SESSION

COMPUTATIONAL SCIENCE, OPTIMIZATION METHODS, AND PERFORMANCE ISSUES

Chair(s)

TBA

Optimization of PID Controller Parameters for Automated Ground Vehicle Control on Dynamic Terrain

J. Mange[1], **S. Pace**[1], **and A. Dunn**[1]
[1]Department of Defense, TARDEC, Computational Methods & System Behavior

Abstract— *Within the ground vehicle domain, there are a variety of contexts in which automated control schemes are necessary, from control of robotic and other semi-autonomous platforms to the automated performance of specified vehicle maneuvers within computer simulations. For basic automated control of speed and steering, particularly within the modeling and simulation arena, proportional-integral-derivative (PID) controllers are commonly used, as these controllers have been extensively studied and are well-understood. However, the performance of these types of controllers is often highly sensitive to the three parameters necessary for their use. In this paper, we explore the use of optimization algorithms for optimal parameter selection for a PID ground vehicle speed controller. In particular, we examine three optimization algorithms – a evolutionary optimization, simulated annealing, and Nelder-Mead simplex optimization – and compare the results of these algorithms on parameter selection for the vehicle controller in a simulation environment.*

Keywords: PID Controller Tuning, Non-linear Optimization, Evolutionary Algorithms, Simulated Annealing, Nelder-Mead Simplex

1. Introduction

Proportional-integral-derivative (PID) controllers are commonly used for automated vehicle control within modeling and simulation. A variety of applications call for this type of automated control. For example, many vehicle mobility tests require a vehicle to follow a specified path at a specified speed through a terrain course, and thus a software controller is needed to keep the vehicle within the required bounds for the test. PID controllers have proved well-suited to this type of task, as they are often able to maintain low-error control without any information about the underlying dynamics of the vehicle, terrain, or test scenario.

Three weight parameters are used for PID controllers, one each for the proportional, integral, and derivative components. The performance of PID controllers in various contexts has proved to be very sensitive to these three parameters. Ground vehicle controllers for modeling and simulation are commonly tuned by subject matter expects manually, based on prior experience and trial-and-error. There exists well known strategies for heuristic-based tuning of controllers, such as [1], as well as a wealth of research into automated parameter selection or optimization for specific applications. [2], [3], [4]. Within this paper, we explore the use of optimization algorithms for optimal selection of these parameters without the need for subject expertise. We make the following contributions:

- a definition of the optimization problem for parameter selection for a PID ground vehicle speed controller.
- a definition of a multiple-component Monte Carlo simulation function over which to perform the optimization.
- a description of the applicability of three optimization algorithms to the problem (Genetic Algorithm, Simulated Annealing, and Nelder-Mead simplex).
- a presentation and discussion of the results of these three optimization algorithms on the problem
- a discussion of the use of these results and the applicability of the approach in other contexts

2. PID Controller

A PID controller is a control loop feedback mechanism that uses a combination of proportional, integral, and derivative terms to control the state of the system. Within this mechanism, the outputs is fed back into the control scheme to correct behavior of the system, and this cyclic cause-and-effect chain of events forms the feedback control loop (see figure 1). A PID controller calculates an error value as the difference between a measured process variable and a desired setpoint. The controller then tries to minimize the error by adjusting the process through the use of a manipulated variable.

The PID control algorithm involves three separate constant terms: proportional, integral, and derivative. These terms can be interpreted in terms of time: proportional depends on the present error, integral on the accumulation of past error, and derivative is a prediction of future error, based on current rate of change [5]. By tuning the weight parameters associated with these three terms, the controller can provide specific control action for unique process requirements. The response of controller can be described in terms of the responsiveness to an error, the degree to which the controller over- or undershoots the desired setpoint, as well as the degree of system oscillation. Defining $u(t)$ as the controller output, the PID algorithm is:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt}e(t) \qquad (1)$$

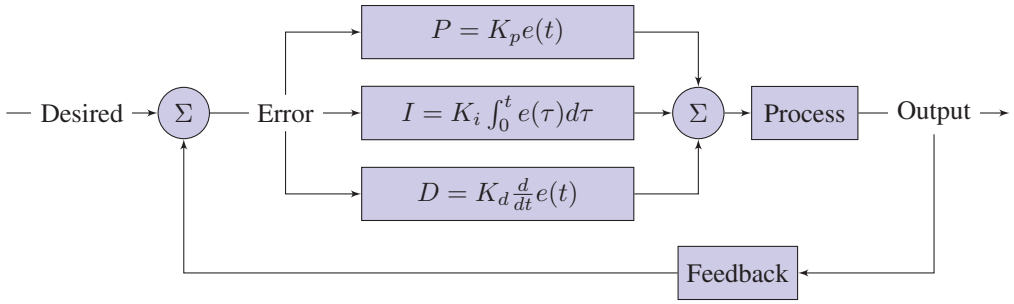


Fig. 1: Notional PID Controller

2.1 Proportional Term

The proportional term is a measure of the current error value. The proportional gain is the ratio of the system response to the current error value. Increasing the proportional gain will increase the speed at which the system responds to the error; increasing the proportional gain may cause instability in the system.

$$P = K_p e(t) \tag{2}$$

K_p is proportional gain, a tuning parameter

e is error

t is instantaneous time

2.2 Integral Term

The integral term is a measure of the error over time. The integral gain is the ratio of the system response to the error values that have accumulated over time. The integral term will increase over time unless the error value is zero.

$$I = K_i \int_0^t e(\tau)d\tau \tag{3}$$

K_i is integral gain, a tuning parameter

e is error

t is instantaneous time

τ is variable of integration

2.3 Derivative Term

The derivative term is a measure of the error rate of change. The derivative gain is the ratio of the system response to the rate of change for the error value. The derivative term is sensitive to rapid changes in the system.

$$D = K_d \frac{d}{dt} e(t) \tag{4}$$

K_d is derivative gain, a tuning parameter

e is error

t is instantaneous time

2.4 Parameter Tuning

Tuning is a process in which the proportional, integral, and derivative gains are set for optimal response of the controller to a unique process requirement. The fundamental difficulty in tuning PID control is that it is a feedback system, with constant parameters and no direct knowledge of the process. For most applications, overall stability (having bounded oscillation) and overall performance (having a bounded error term) are the basic goals for the controller.

3. Autonomous Ground Vehicles

For ground vehicles with automatic transmissions, there are three primary driver inputs: throttle, braking, and steering. For vehicles with manual transmissions, an additional gear input is also necessary. In a modeling and simulation context, since throttle and braking are both related to speed control, a single controller is often used for speed, with the application of throttle and brakes performed in response to the controller output at each time step. Thus for automated control of a simulated vehicle, two controllers are generally used – a speed controller and a steering controller. For a specific vehicle course, then, a set of maneuvers and associated desired speeds is specified, and the two controllers attempt to follow these prescribed steps as closely as possible.

Each of the components (proportional, integral, and derivative) has a rough conceptual equivalent within the simulation for both types of controllers. For the steering controller, the proportional component compensates for direct errors in the vehicle's position, the integral component compensates for any lateral movement of the vehicle (for instance, from a terrain with a side slope), and the derivative component compensates for errors in the vehicle's direction. For the speed controller, the proportional component compensates for errors in the overall speed of the vehicle, the integral component compensates for over- or under-acceleration effects, and the derivative component compensates for errors of acceleration itself.

The selection of the weight parameters for PID controllers within a vehicle simulation is often heavily dependent on the context in which the vehicle will operate within the simulation, in particular the type of terrain and terrain

features present. For example, since the integral component for both steering and speed controllers compensates mainly for specific types of scenarios, the weight parameter for these components is often set proportionally lower than the other two PID parameters. Indeed, sometimes a simpler PD controller (equivalent to setting the integral weight parameter to zero in a PID controller) can be used outside of those specific classes of scenarios for which an integral component is important (see, for instance, [6]).

The basic operation of the PID controllers is the same for both automated speed and steering control within a vehicle simulation. However, since the steering control generally requires more initial input from a simulation user (e.g. starting and ending locations, type(s) of maneuvers along the path), we use a speed controller for the optimization formulation and results in this paper for simplicity of presentation. The same concepts and optimization algorithm results apply to an automated steering controller, although not necessarily the same optimized weight parameters.

4. Optimization Problem Formulation

In order to formulate the problem of selecting automated vehicle speed PID controller weight parameter values as a general optimization problem, we must define optimization variables, an evaluation (or "objective") function to be minimized or maximized, and a domain for the variables of that function.

The optimization variables are straightforward – they are simply the three PID parameters, K_p, K_i, and K_d. The domain for these variables will be $[0, 1]$. Although this is a fairly standard domain for these weight parameters, some technical considerations are involved with this choice. In particular, in our context, since the vehicle simulation itself is time-stepped, the rate of change of the speed of the vehicle is limited, as it would be in a real life scenario by both the vehicle performance and the reaction time of the driver. In order to control this rate of change, we also control the maximum effect of the weighted PID controller value at each timestep, which is facilitated in part by selecting the domain of $[0, 1]$ for the weight parameters.

The evaluation function consists of a measure of the performance of the PID controller with the specified parameters. A full vehicle simulation is performed over a specified time length, and the value of the mean-squared-error is taken as the single evaluation measure of the controller performance. The reasons for these selections and the details of the simulation itself are presented in the following sections.

4.1 Vehicle Simulation Details

As with any simulation-based optimization problem, the vehicle simulation in our context is designed to be both:

- complex enough to model the important features of the problem and create an accurate relationship between parameter input and controller performance output
- simple enough to run quickly for use in an optimization context that requires hundreds or thousands of simulation runs

Our simulation is Monte Carlo based, with randomized terrain for each simulation run. The two main simulation components, aside from the positional and controller elements themselves, are this randomized terrain and a simplified vehicle powertrain model.

The randomized terrain consists of a series of flat, uphill, and downhill segments, roughly modeling on the types and grades of terrains commonly seen in the research context of the authors (US Army ground vehicles). The slope of the terrain at a specific timestep contributes either positively or negatively to the acceleration of the vehicle according to the physics governing the simulation.

In any modern ground vehicle, the relationship between the throttle input from the driver (or speed controller) and the torque output of the powertrain is a complicated one. Gearing, engine characteristics, tire / terrain interaction effects, and many other factors also play significant roles. In order to simplify modeling this relationship, a tractive force vs. speed curve is often used to characterize the average performance of a vehicle's powertrain for simulation purposes. This is the approach we chose in our simulation, again with data roughly representative of the vehicle research context of the authors. Therefore, based on the vehicle's speed at the previous timestep, a powertrain output is calculated, which is then used in conjunction with the terrain profile and controller input to calculate the vehicle's speed at the current timestep.

This simulation meets both of the goals stated above; it is complex enough to fairly accurately model the real-life scenario on which it is based and to produce a meaningful relationship between the speed controller parameters and the speed performance through the simulation, yet simple enough to be used in our optimization context for a variety of optimization algorithms.

4.2 Evaluation Function

The simulation described in the previous section is the heart of the evaluation function for our optimization. However, the questions remain of how exactly to use this simulation and what output metric(s) to consider.

Since our vehicle simulation is Monte Carlo based, the common approach is to run the simulation several times with the same parameters in order to average the output to compensate for the effects of randomization. Based on experimentation, we found that the simulation took approximately 80 runs to converge to a stable output value with this approach. Therefore, for a single objective function evaluation, we run the simulation 100 times and average the results for the function value.

Now we must determine precisely what value the evaluation function should calculate. Several metrics for measuring

PID controller performance have been used [7], with mean-squared error being the most common, as well as various other error measures including integrated absolute error, integrated squared error, and integrated time-weighted squared error (see [7] for some discussion of the advantages and disadvantages of several). For our purposes, mean-squared error was the natural choice for a single evaluation optimization function. However, the same formulation presented in this paper could be used with any of the other mentioned error measures, or indeed any measure of controller performance that is applicable to the context of the user. Furthermore, a multi-objective optimization formulation along the same lines is perfectly feasible. We chose the single objective mean-squared error measure both because of its general purpose usefulness for evaluating controller performance of this kind, as well as for ease of presenting results.

5. Optimization Algorithms

We compare three commonly used nonlinear optimization algorithms for the variables and evaluation function defined above: an evolutionary (genetic) algorithm, the simulated annealing algorithm, and the Nelder-Mead simplex optimization algorithm. A full description of the intricacies of these algorithms is outside the scope of this paper; however, the following sections contain a brief overview of each.

5.1 Genetic Algorithm

Genetic algorithms are algorithms that roughly correspond to the biological process of natural selection. They are part of the broader category of Evolutionary Algorithms, which are population-based optimization algorithms inspired by biological evolution. Although there are many variations, Genetic algorithms tend to involve these three main components:

- Mutation – a subset of the current population is altered probabilistically to produce a new set of individuals
- Crossover – a subset of the current population is "bred", with aspects of two (or more) parents being combined to produce a new child individual
- Selection – based on the original population, new mutated individuals, and new individuals produced by crossover, a subset of this new population is chosen to continue to the next generation, based on the evaluation function values for each individual

Genetic Algorithms, and Evolutionary Algorithms more generally, are some of the most popular optimization techniques in use today. They have proven successful in a wide variety of contexts, and are often good choices for nonlinear optimization problems that can prove difficult for other classes of optimization algorithms [8].

5.2 Simulated Annealing

Simulated Annealing is an optimization approach modeled after the annealing process in materials science, which involves heating a material and then controlling the rate at which it cools in order to produce specific effects. The algorithm starts from (generally) a random point in the optimization search space, and then iteratively inspects other points within the search space and probabilistically determines whether to move to the new point based on its evaluation function value and the state of the system. The effect of the algorithm is generally to examine a widely spaced set of points in the early stages of the algorithm, and converge to examining smaller areas near good solutions as the algorithm progresses. Because of its relative simplicity and lack of a tracked population, Simulated Annealing has also proven to be a popular and effective optimization algorithm in a variety of contexts [9].

5.3 Nelder-Mead Simplex Optimization

The Simplex algorithm is one of the most common optimization algorithms for linear programming problems. The Nelder-Mead method is an extension of the standard Simplex algorithm for nonlinear optimization. The algorithm involves constructing a simplex (a multi-dimensional polytope) with points within the optimization search space. The points of this simplex are then iteratively replaced with new points based on the evaluation function values for the simplex points and new test points. Many variations and improvements for this algorithm have been implemented over the years since its introduction, and it is often considered only applicable to certain types of problems [10]. As will become apparent in the Results section, the problem of choosing optimal PID vehicle controller parameters is well-suited to the Nelder-Mead algorithm.

6. Results

A single simulation produces a speed over time trace, similar to Figure 2, which shows a vehicle PID speed controller attempting to maintain a speed of 30 mph. The difference between the speed at each timestep and the reference speed of 30 mph is that timestep's error, and as previously discussed, the mean squared error is the overall measure of control performance used. The task of each optimization algorithm, then, is to minimize this metric.

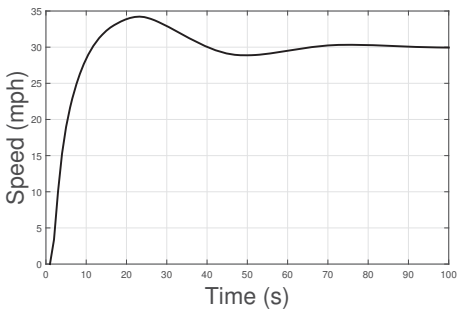


Fig. 2: Single simulation

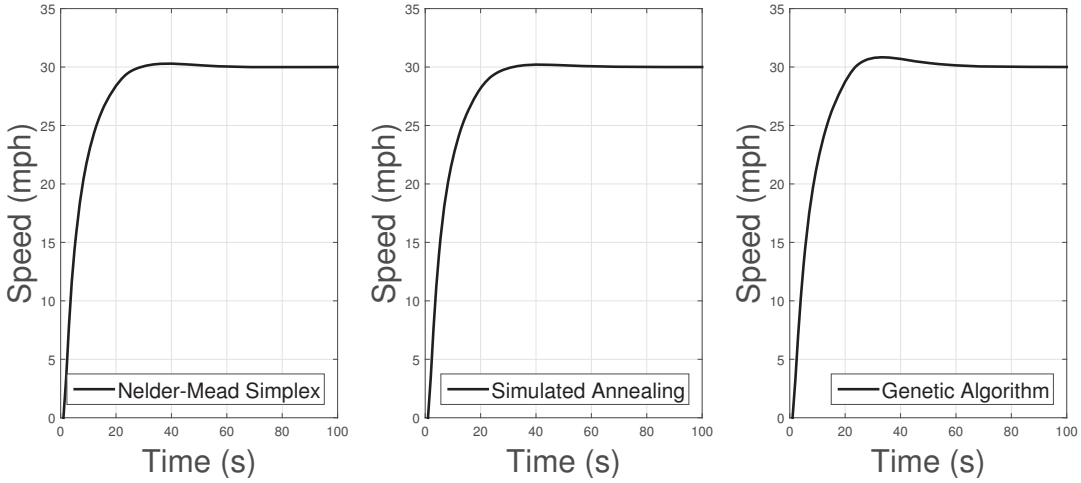


Fig. 3: Comparison after 100 evaluations

Because different optimization algorithms operate in sometimes very different ways, the most common method of directly comparing their performance is by measuring objective function evaluations. For instance, we can compare the results of the best found parameters after n objective function evaluations for two or more different optimization algorithms. Figure 3 shows such a comparison of the three optimization algorithms under consideration, showing a simulated speed trace over the same terrain using the best parameters obtained by each after 100 objective function evaluations. As can be seen, even by this relatively small number of evaluations, the three algorithms have begun to converge to very similar parameter values, thus producing very similar simulation results.

A more complete comparison of the performance of the three algorithms involves tracking the evaluation function value for the best-found parameters every time a objective function evaluation occurs, to see how each algorithm evolves over time. Table 1 shows values for the mean squared error at different number of objective function evaluations for the three algorithms, and figure 4 shows a graph of this comparison for the first 100 evaluations.

	25	**50**	**75**	**100**
Nelder-Mead Simplex	28.6	27.9	27.9	27.9
Simulated Annealing	30.1	30.1	30.0	29.7
Genetic Algorithm	66.9	39.5	36.2	35.3

Table 1: Mean squared error at different numbers of objective function evaluations

As can be seen, the Nelder-Mead Simplex Optimization algorithm performed the best in this time range, as it found the parameters that produced the lowest average mean squared error for the vehicle simulations. Simulated Annealing performed second best, with the Genetic Algorithm exhibiting the worst performance. After 100 evaluations, the lowest average mean squared error for the Simulated Annealing algorithm was 6.4% higher than for the Nelder-Mean algorithm. The lowest for the Genetic Algorithm was 26.9% higher.

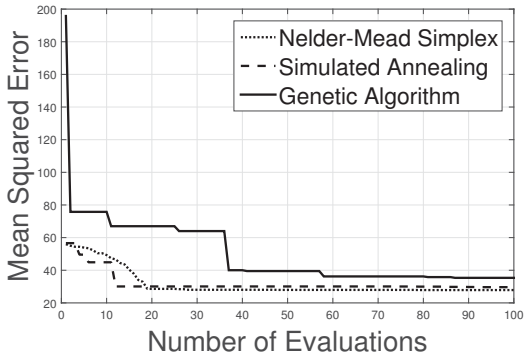


Fig. 4: Error for best known values at each evaluation

The shape of the lines in this graph also fit with the characteristics of the algorithms – since the Genetic Algorithm and Simulated Annealing both involve much more randomness, much larger "jumps" are seen when the best known evaluation value abruptly decreases, whereas the Nelder-Mead best known values decrease at a steadier rate as they converge toward a solution.

Although these differences are somewhat enlightening, it is important to note that 100 objective function evaluations is relatively small for most problems, and for this problem takes under a minute to run on a single core of a modern computer. By 3000 objective function evaluations, all three algorithms had converged to the same parameter values.

A complication of this comparison is that all three algorithms involve a number of algorithm-specific parameters

that can significantly affect their own performance. Attempting to find ideal parameters for the optimization algorithms themselves does not fit with the aims of this paper, so in making this comparison, we attempted to use common general-purpose parameters from the literature. Thus, it is quite likely that some or all of the algorithms could perform better with algorithm-specific parameter tweaking.

7. Conclusions

The goal of this investigation was, first, to formulate the selection of parameters for an automated ground vehicle PID speed controller as an optimization problem, and second, to compare a Genetic Algorithm, the Simulated Annealing algorithm, and the Nelder-Mead Simplex algorithm for optimization of these parameters.

The formulation was demonstrably successful, as all three algorithms were able to optimize these parameters based on the criterion specified. In terms of algorithm performance, the Nelder-Mead Simplex optimization algorithm performed the best, followed by the Simulated Annealing algorithm, followed by the Genetic Algorithm. However, all three algorithms converged to the same parameter values given a reasonably small amount of time. From this, we conclude that the formulation itself, including the details of the vehicle simulation, are of far more practical importance than the selection of optimization algorithm for parameter selection, unless a serious time or resource restriction exists.

The basic ideas of this optimization problem formulation, as well as the concept and implementation of the Monte Carlo vehicle simulation, are general enough to be applicable in a variety of other contexts. In particular, however, the results in this paper have convincingly demonstrated that an optimization approach is a valid alternative to relying on domain-specific expertise for the selection of PID controller parameters for an automated ground vehicle.

References

[1] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *trans. ASME*, vol. 64, no. 11, 1942.

[2] C. Xiangguang, "Research of the application of genetic algorithm (ga) to parameters optimization of pid controller [j]," *Computer Simulation*, vol. 2, p. 011, 2001.

[3] M. Zhuang and D. Atherton, "Automatic tuning of optimum pid controllers," in *Control Theory and Applications, IEE Proceedings D*, vol. 140, no. 3. IET, 1993, pp. 216–224.

[4] M. I. Solihin, L. F. Tack, and M. L. Kean, "Tuning of pid controller using particle swarm optimization (pso)," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 1, no. 4, pp. 458–461, 2011.

[5] M. Araki, "Pid control," *Control systems, robotics and automation*, vol. 2, pp. 1–23, 2002.

[6] Y.-P. Kuo and T. Li, "Ga-based fuzzy pi/pd controller for automotive active suspension system," *Industrial Electronics, IEEE Transactions on*, vol. 46, no. 6, pp. 1051–1056, 1999.

[7] Z.-L. Gaing, "A particle swarm optimization approach for optimum design of pid controller in avr system," *Energy Conversion, IEEE Transactions on*, vol. 19, no. 2, pp. 384–391, 2004.

[8] M. Raghuwanshi and O. Kakde, "Survey on multiobjective evolutionary and real coded genetic algorithms," in *Proceedings of the 8th Asia Pacific symposium on intelligent and evolutionary systems*, 2004, pp. 150–161.

[9] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the operational research society*, vol. 57, no. 10, pp. 1143–1160, 2006.

[10] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder–mead simplex method in low dimensions," *SIAM Journal on optimization*, vol. 9, no. 1, pp. 112–147, 1998.

Queuing Network Approximation Technique for Evaluating Performance of Computer Systems with Finite Input Source

Mayuko Hirose, Madoka Shiratori, Matrazali Noorafiza,
Ryo Tsuboi, Itaru Koike, Toshiyuki Kinoshita

School of Computer Science, Tokyo University of Technology
1404-1 Katakura, Hachioji Tokyo, 192-0982, Japan

Abstract *Queuing network techniques are effective for evaluating the performance of computer systems. We discuss a queuing network technique for computer systems in finite input source. The finite number of terminals exist in the network and a job in the network moves to the server that includes CPU, I/O equipment and memory after think-time at the terminal. When the job arrives at the server, it acquires a part of memory and executes CPU and I/O processing in the server. After the job completes CPU and I/O processing, it releases the memory and goes back to its original terminal. However, because the memory resource can be considered as a secondary resource for the CPU and I/O equipment, the queuing network model has no product form solution and cannot be calculated the exact solutions.*

We proposed here an approximation queuing network technique for calculating the performance measures of computer systems with finite input source on which multiple types of jobs exist. This technique involves dividing the queuing network into two levels; one is "inner level" in which a job executes CPU and I/O processing, and the other is "outer level" that includes terminals and communication lines. By dividing the network into two levels, we can prevent the number of states of the network from increasing and approximately calculate the performance measures of the network. We evaluated the proposed approximation technique by using numerical experiments and clarified the characteristics of the system response time and the mean number of jobs in the inner level.

Keywords *performance evaluation, queuing network, central server model, finite input source*

1. Introduction

Queuing network techniques are effective for evaluating the performance of computer systems. In computer systems, two or more jobs are generally executed at the same time, which causes delays due to conflicts in accessing hardware or software resources such as the CPU, I/O equipment, or data files. We can evaluate how this delay affects the computer system performance by using a queuing network technique. Some queuing networks have an explicit exact solution, which is called a product form solution [1]. With this solution, we can easily calculate the performance measures of computer systems, for example the busy ratio of hardware and the job response time, and so on. However, when the exclusion controls are active or when a memory resource exists, the queuing network does not have the product form solution. To calculate an exact solution of a queuing network that does not have the product form solution, we have to construct a Markov chain that describes the stochastic characteristics of the queuing network and numerically solve its equilibrium equations. When the number of jobs or the amount of hardware in the network increases, the number of states of the queuing network drastically increases. Since the number of unknown quantities in the equilibrium equations is equal to the number of states of the queuing network, the number of unknown quantities in the equilibrium equations also drastically increases. Therefore, we cannot perform calculating the exact solution of the queuing network numerically.

Here, we discuss the queuing network technique for computer systems with finite input source (Figure 1). The finite number of terminals exists in the system and a job is dedicated to its own terminal. After a think-time at the terminal, the job moves to the server and acquires a part of the memory and executes CPU and I/O processing. When the job completes CPU and I/O processing at the server, it releases the memory and goes back to its original terminal. Thus, this model is constructed with two levels, one is outside of the server (terminals and communication lines) and the other is inside of the server (CPU, I/O equipment, and memory). We call them "outer level" and "inner level" respectively.

Since a job executes CPU and I/O processing occupying the memory, the memory can be considered as a secondary resource for the CPU and I/O equipment in inner level. Generally when a queuing network includes a secondary resource, it does not have product form solutions and an approximation technique is required to analyze the network.

We have proposed here an approximation technique for calculating the performance measures of computer systems with

finite input source. We previously reported the results for computer systems with memory resource in open input source, in which jobs arrive from and depart to outside of the system. In this paper, we consider the finite input source model, in which each job is dedicated to its own terminal, and after think-time the job moves to the server that includes CPU, I/O equipment and memory. After the job acquires a part of memory, it executes CPU and I/O processing occupying the memory in the server. When the job completes CPU and I/O processing, it releases the memory and goes back to its original terminal. In order to prevent the number of states of the Markov chain from increasing, we divide the model into two levels, one is outer level that includes the terminals and communication lines, and the other is inner level that includes CPU, I/O equipment and memory resources. Similarly in [7][10], two different types of jobs exist in the network. Both job behavior in the inner and the outer level differs for each job class. When there is a single job class, both the inner and outer level has a product form solution. However, when there exist multiple job classes, inner level does not have a product form solution. Therefore, an approximation is needed to analyze the inner level.

Dividing the model into two levels is one of two-layer queuing network techniques [3][4]. Our proposed technique is also a two-layer technique for computer systems with finite input source.

In our previous study [5], we reported an approximation technique for evaluating performance of computer systems with file resources. Meanwhile, heterogeneous parallel computer systems with distributed memory is researched in [8], and the Markov chain involving two dimensional state transition similar to our proposed model was discussed in [9].

2. Model Description

The CPU and I/O model in the inner level is equivalent to the ordinary central server model with multiple job types (each of which is called a job class). In this model, R job classes exist, and each of them is numbered r = 1, 2, … , R by affixing r. The inner level consists of a single CPU node and multiple I/O nodes. We denote M as the number of I/O nodes. The I/O nodes are numbered m = 1, 2, … , M by affixing m, and the CPU node is numbered m = 0 by also affixing m. The service rate of job class r at the CPU node is $\mu_0{}^r$, and the service rate of job class r at an I/O node m is $\mu_m{}^r$. The service time at each node is a mutually independent random variable subject to common exponential distributions. Jobs are scheduled on a first come first served (FCFS) principle at all nodes. At the end of CPU processing, a job probabilistically selects an I/O node and moves to it, or completes CPU and I/O processing and goes back to the terminal. The selection probability of I/O node m of job class r is $p_m^r\ (m=1,2,\ldots,M; r=1,2,\ldots,R)$ and the completion probability of job class r is p_0^r. Therefore, $\sum_{m=0}^{M} p_m^r = 1\ (r=1,2\ldots R)$.

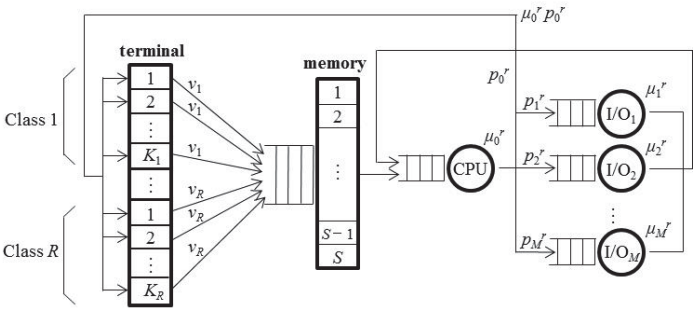


Fig. 1 Central server model in finite input source

Memory resources are added to this central server model (Figure 1). We denote S as the number of memory resources.

In outer level, a job stays at the terminal for short while. The staying time is called "think-time" of a job. The think-time is mutually independent random variable subject to common exponential distribution with parameter v_r of job class r (v_r is job departure rate from the terminal). After the think-time, the job moves to the inner level, and requests and acquires a part of the memory resources before entering the central server model. If all the parts of the memory are occupied, the job joins the system waiting queue and waits for a part of the memory to be released by another job. When the job completes CPU and I/O processing, it releases the memory and leaves the inner model and goes back to its own terminal. Since the job has to acquire a part of memory before entering the central server model, the number of jobs occupying a memory is always equal to the number of jobs in the central server model. Therefore, at most S jobs can execute CPU and I/O processing at the same time. That is, the maximum job multiplicity in the central server model is S. When the number of jobs of job class r in the central server model is denoted by n_r, $\sum_{r=1}^{R} n_r \le S$.

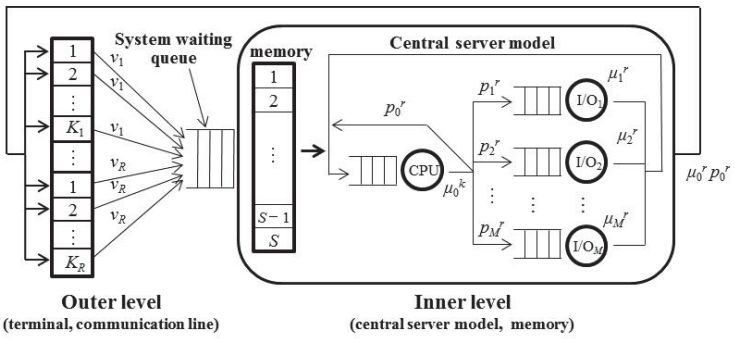


Fig. 2 Concept of approximation

By replacing "CPU → outer level transition" with "CPU → CPU transition," the central server model is modified to a closed model in which the number of jobs is constant (Figure 2). In this model, when "CPU → CPU transition" occurs, the job terminates and a new job is born. Therefore, the mean job response time is the mean time between two successive "CPU → CPU transitions." This mean job response time can be considered as a job lifetime.

3. Approximation Model

To obtain the exact solution of the central server model with finite input source, we have to describe the entire model with a single Markov chain for each job class. However, this causes the number of states of the Markov chain to drastically increase when the number of jobs and the number of nodes in the network increase. By dividing the network into two levels, and describing each level with two Markov chains, we can prevent the number of states of the model from increasing (Figure 2). We set the following notations.

t_r : mean think-time of jobs in job class r
v_r : departure rate from the terminal of job class r
τ_{rm} : total mean service time at node-m in a job lifetime of job class r
n_{rm} : number of jobs in job class r at node-m (r =1, 2, ... , R; m =0, 1, ... , M)
K_r : number of jobs in job class r (= number of terminals of job class r)
$\boldsymbol{n} = (n_1, n_2, \dots, n_R)$: vector of number of jobs (n_r =0, 1, 2, ..., K_r)
$\boldsymbol{n}^* = (n_{10}, n_{11}, \dots, n_{1M}, n_{20}, n_{21}, \dots, n_{2M}, \dots, n_{R0}, n_{R1}, \dots, n_{RM})$: state vector of the central server model
$\boldsymbol{F}(\boldsymbol{n}) = \{\boldsymbol{n}^* \mid \sum_{m=0}^{M} n_{rm} = n_r, n_{rm} \geq 0 \ (m = 0, 1, \dots, M)\}$ $(n_1+n_2+\dots+n_R \leqq S)$: set of all feasible states of the central server model when the number of jobs of job class r is n_r
$P_s(\boldsymbol{n}^*)$: steady-state probability of state $\boldsymbol{n}^*$
$T_{\boldsymbol{n}}^{\,r}$: mean job response time of the central server model when the vector of number of jobs is $\boldsymbol{n}$
$\mu_{\boldsymbol{n}}^{\,r}$: service rate from the central server model of job class r
T^r : system response time of job class r

Since the central server model in inner level is equivalent to the ordinary central server model with multiple job classes, it has the product form solution. Then the steady-state probability $P_s(\boldsymbol{n}^*)$ is represented by the following formula [1][2].

$$P_s(\boldsymbol{n}^*) = \frac{\prod_{r=1}^{R}\prod_{m=0}^{M} \tau_{rm}^{\,n_{rm}}}{\varphi(n_1, n_2, \cdots, n_R, M)}$$

where $\varphi(n_1, n_2, \cdots, n_R, M) = \sum_{n \in F(n)} \prod_{r=1}^{R}\prod_{m=0}^{M} \tau_{rm}^{\,n_{rm}}$ is the normalizing constant of steady-state probabilities when the number of jobs of job class r in the central server model is n_r (=0, 1, 2, … , K_r; r =1, 2, ... , R). From these steady-state probabilities, we can calculate the mean job response time $T_{\boldsymbol{n}}^{\,r}$ of job class r as follows when the number of jobs is n_r.

$$T_n^{\,r} = \frac{n_r \cdot \varphi(n_1, \cdots, n_r, \cdots, n_R, M)}{\varphi(n_1, \cdots, n_r - 1, \cdots, n_R, M)}$$

The memory resource in our model can be considered as an M/M/S queuing model with S servers. In an ordinary M/M/S queuing model, the service rate at a server is constant, regardless of the number of guests in the service. In the memory resource of our model, however, the service rate changes depending on the number of occupied memories. The mean job response time $T_{\boldsymbol{n}}^{\,r}$ of job class r (=1, 2, ... , R) when the vector of number of jobs is $\boldsymbol{n} = (n_1, n_2, \dots, n_r)$ is equal to the

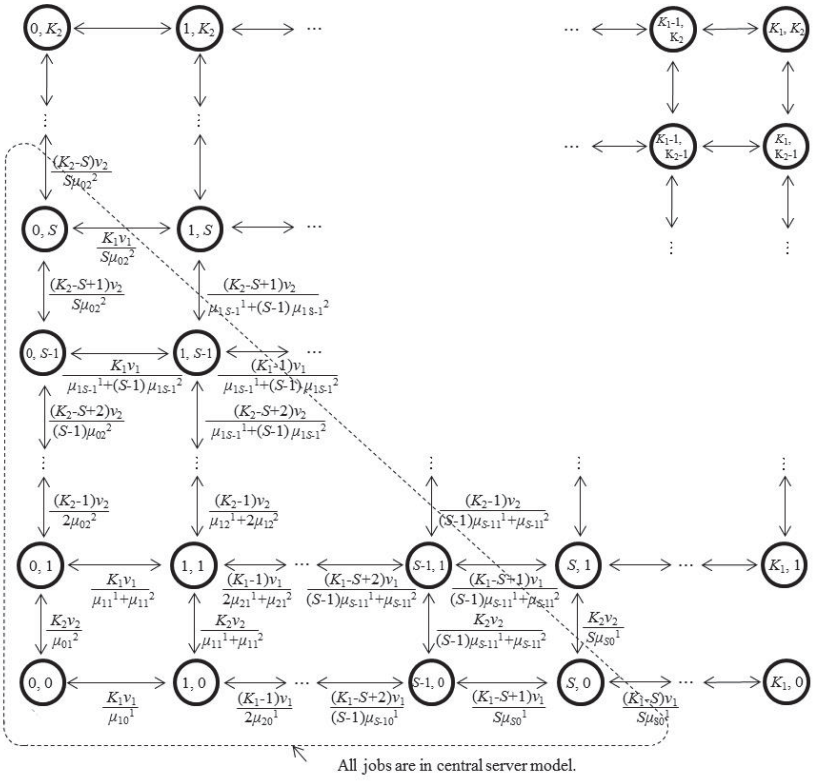


Fig. 3 State Transition diagram (two job classes)

mean time while the memory is occupied. Since the service rate of job class r from the central server model $\mu_{\boldsymbol{n}}^{r}$ is denoted as $\mu_{\boldsymbol{n}}^{r}=\frac{1}{T_n^r}$, $\mu_{\boldsymbol{n}}^{r}$ depends on the number of jobs in the central server model n_r. The state transition of the M/M/S queuing model with two job classes is shown in Figure 3, where the service rates from the central server model change depending on the number of jobs in the central server model. This is a two dimensional birth-death process. The equilibrium equations with the steady-state probability $Q_S(\boldsymbol{n})=Q_S(n_1, n_2)$, when the number of maximum number of jobs in the central server model is S and the number of jobs in the central server model is $\boldsymbol{n}=(n_1, n_2)$, are as follows (similar to the case with higher dimensions).

(1) $n_1=0, n_2=0$

$$(K_1v_1+K_2v_2)\cdot Q_S(0,0)=\mu_{10}{}^{1}\cdot Q_S(1,0)+\mu_{01}{}^{2}\cdot Q_S(0,1)$$

(2) $n_1=1, 2, \dots, S-1, n_2=0$

$$\{(K_1-n_1)v_1+K_2v_2+n_1\mu^1_{n_1 0}\}\cdot Q_S(n_1,0)$$
$$=(K_1-n_1+1)v_1\cdot Q_S(n_1-1,0)+(n_1+1)\mu^1_{n_1+10}\cdot Q_S(n_1+1,0)$$
$$+\mu^2_{n_1 1}\cdot Q_S(n_1,1)$$

(3) $n_1=S, S+1, \dots, K_1-1, n_2=0$

$$\{(K_1-n_1)v_1+K_2v_2+S\mu^1_{S0}\}\cdot Q_S(n_1,0)$$
$$=(K_1-n_1+1)v_1\cdot Q_S(n_1-1,0)+S\mu^1_{S0}\cdot Q_S(n_1+1,0)$$
$$+\mu^2_{n_1 1}\cdot Q_S(n_1,1)$$

(4) $n_1=0, n_2=1, 2, \dots, S-1$

$$\{K_1v_1+(K_2-n_2)v_2+n_2\mu^2_{0n_2}\}\cdot Q_S(0,n_2)$$
$$=(K_2-n_2+1)v_2\cdot Q_S(0,n_2-1)+\mu^1_{1n_2}\cdot Q_S(1,n_2)$$
$$+(n_2+1)\mu^2_{0n_2+1}\cdot Q_S(0,n_2+1)$$

(5) $n_1=0, n_2=S, S+1, \dots, K_2-1$

$$\{K_1v_1+(K_2-n_2)v_2+S\mu^2_{0S}\}\cdot Q_S(0,n_2)$$
$$=(K_2-n_2+1)v_2\cdot Q_S(0,n_2-1)+\mu^1_{1n_2}\cdot Q_S(1,n_2)$$
$$+S\mu^2_{0S}\cdot Q_S(0,n_2+1)$$

(6) $n_1+n_2\leqq S-1, n_1=1, 2, \dots, S-2, n_2=1, 2, \dots, S-2$

$$\{(K_1-n_1)v_1+(K_2-n_2)v_2+n_1\mu^1_{n_1n_2}+n_2\mu^2_{n_1n_2}\}\cdot Q_S(n_1,n_2)$$
$$=(K_1-n_1+1)v_1\cdot Q_S(n_1-1,n_2)+(K_2-n_2+1)v_2\cdot Q_S(n_1,n_2-1)$$
$$+(n_1+1)\mu^1_{n_1+1n_2}\cdot Q_S(n_1+1,n_2)+(n_2+1)\mu^2_{n_1n_2+1}\cdot Q_S(n_1,n_2+1)$$

(7) $n_1+n_2=S, n_1=1, 2, \dots, S-1, n_2=1, 2, \dots, S-1$

$$\{(K_1-n_1)v_1+(K_2-n_2)v_2+n_1\mu^1_{n_1n_2}+n_2\mu^2_{n_1n_2}\}\cdot Q_S(n_1,n_2)$$
$$=(K_1-n_1+1)v_1\cdot Q_S(n_1-1,n_2)+(K_2-n_2+1)v_2\cdot Q_S(n_1,n_2-1)$$
$$+n_1\mu^1_{n_1n_2}\cdot Q_S(n_1+1,n_2)+n_2\mu^2_{n_1n_2}\cdot Q_S(n_1,n_2+1)$$

(8) $n_1+n_2>S, n_1=1, 2, \dots, K_1, n_2=1, 2, \dots, K_2$

When the lattice point (m_1, m_2) such as $m_1+m_2=S$ is on the shortest route from (0, 0) to (n_1, n_2), and $Q_S^{m_1m_2}(n_1, n_2)$ is the steady-state probability along with the route.

$$\{(K_1-n_1)v_1+(K_2-n_2)v_2+m_1\mu^1_{m_1m_2}+m_2\mu^2_{m_1m_2}\}\cdot Q_S^{m_1m_2}(n_1,n_2)$$
$$=(K_1-n_1+1)v_1\cdot Q_S^{m_1m_2}(n_1-1,n_2)$$
$$+(K_2-n_2+1)v_2\cdot Q_S^{m_1m_2}(n_1,n_2-1)$$
$$+m_1\mu^1_{m_1m_2}\cdot Q_S^{m_1m_2}(n_1+1,n_2)+m_2\mu^2_{m_1m_2}\cdot Q_S^{m_1m_2}(n_1,n_2+1)$$

(a) $n_1+n_2>S, n_1=1, 2, \dots, S, n_2=S-n_1+1, S-n_1+2, \dots, S$

$$\Rightarrow Q_S(n_1,n_2)=\sum_{m_1=S-n_2}^{n_1}Q_S^{m_1m_2}(n_1,n_2)$$

(b) $n_1+n_2>S, n_1=S+1, S+2, \dots, K_1, n_2=1, 2, \dots, S$

$$\Rightarrow Q_S(n_1,n_2)=\sum_{m_1=S-n_2}^{S}Q_S^{m_1m_2}(n_1,n_2)$$

(c) $n_1+n_2>S, n_1=1, 2, \dots, S, n_2=S+1, S+2, \dots, K_2$

$$\Rightarrow Q_S(n_1,n_2)=\sum_{m_1=0}^{n_1}Q_S^{m_1m_2}(n_1,n_2)$$

(d) $n_1+n_2>S, n_1=S+1, S+2, \dots, K_1, n_2=S+1, S+2, \dots, K_2$

$$\Rightarrow Q_S(n_1,n_2)=\sum_{m_1=0}^{S}Q_S^{m_1m_2}(n_1,n_2)$$

For the state (n_1, n_2) of the Markov chain, when $n_1+n_2\leqq S$, all jobs are in the central server model and executing CPU and I/O processing, and when $n_1+n_2>S$, n_1+n_2-S jobs are in the system waiting queue and waiting for a part of the memory resources to be released. The transition diagram of the two dimensional birth-death process is shown in Figure 3. However, the equilibrium equation does not have the product form solution. Therefore, some approximation is required to solve it.

When the model has a single job class, it can be described with a one dimensional birth-death process. Its transition diagram is shown in Figure 4, and the equilibrium equation is as follows:

$$K_1v_1\cdot Q_S(0)=\mu_1{}^1\cdot Q_S(1)$$
$$\{(K_1-n_1)v_1+n_1\mu^1_{n_1}\}\cdot Q_S(n_1)=(K_1-n_1+1)v_1\cdot Q_S(n_1-1)$$
$$+(n_1+1)\mu^1_{n_1+1}\cdot Q_S(n_1+1)\quad(n_1=1, 2, \dots, S-1)$$
$$\{(K_1-n_1)v_1+S\mu^1_S\}\cdot Q_S(n_1)=(K_1-n_1+1)v_1\cdot Q_S(n_1-1)$$
$$+S\mu^1_S\cdot Q_S(n_1+1)\qquad(n_1=S, S+1, \dots, K_1-1)$$
$$S\mu^1_S\cdot Q_S(K_1)=v_1\cdot Q_S(K_1-1)$$

Solutions for the equilibrium equation are in the following product form.

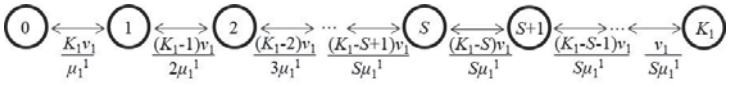


Fig. 4 State transition diagram (single job class)

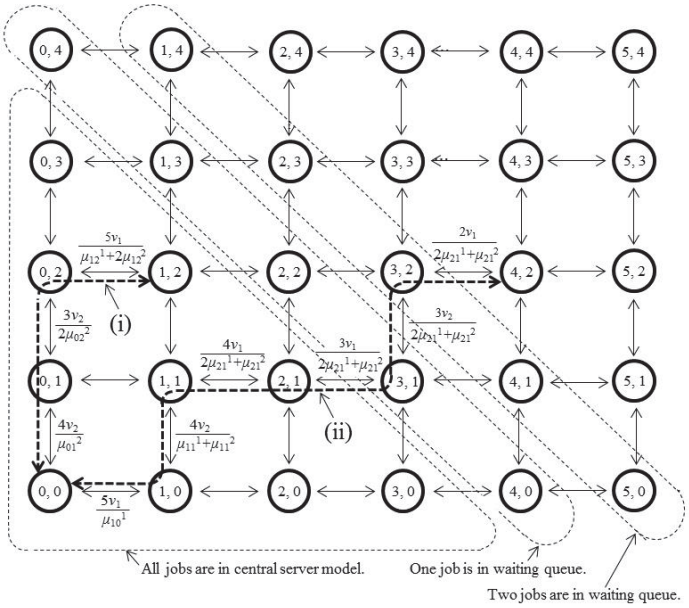


Fig. 5 Calculating state probability for two job classes

$$Q_S(n_1)=\begin{cases} Q_S(0)\cdot\prod_{i=1}^{n_1}\dfrac{(K_1-i+1)\nu_1}{i\cdot\mu_i^1} & (n_1=1,2,\ldots,S-1) \\ Q_S(0)\cdot\prod_{i=1}^{S-1}\dfrac{(K_1-i+1)\nu_1}{i\cdot\mu_i^1}\cdot\prod_{i=S}^{n_1}\dfrac{(K_1-i+1)\nu_1}{S\cdot\mu_S^1} & (n_1=S,S+1,\ldots,K_1) \end{cases}$$

In this formula, for the state transition at i =1, 2, ..., $S-1$, multiply by factor $\dfrac{(K_1-i+1)\nu_1}{i\cdot\mu_i^1}$ while for the state transition at i = S, S+1, ..., K_1 multiply by factor $\dfrac{(K_1-i+1)\nu_1}{S\cdot\mu_S^1}$. For two dimension case, we consider a route from lattice point (0, 0) to (n_1, n_2) shown in Figure 5, and for the horizontal state transition at the lattice point (i_1, i_2) such as $i_1+i_2 \leqq S$ on the route, multiply by factor $\dfrac{(K_1-i_1+1)\nu_1}{i_1\cdot\mu_{i_1i_2}^1+i_2\cdot\mu_{i_1i_2}^2}$, and multiply by factor $\dfrac{(K_2-i_2+1)\nu_2}{i_1\cdot\mu_{i_1i_2}^1+i_2\cdot\mu_{i_1i_2}^2}$ for the vertical state transition. When the lattice point (i_1, i_2) such as $i_1+i_2>$S, for the state transition outside of the lattice point (m_1, m_2) such as $m_1+m_2=S$ on the route (between (m_1, m_2) and (i_1, i_2)), multiply by factor $\dfrac{(K_1-m_1+1)\nu_1}{m_1\cdot\mu_{m_1m_2}^1+m_2\cdot\mu_{m_1m_2}^2}$ or $\dfrac{(K_2-m_2+1)\nu_2}{m_1\cdot\mu_{m_1m_2}^1+m_2\cdot\mu_{m_1m_2}^2}$. Thus, the coefficient of $Q_S(n_1, n_2)$ related to $Q_S(0, 0)$ is represented as the summation of the multiplication based on all the routes from (0, 0) to (n_1, n_2). For example, for the route from (0, 0) to (1, 2) when S=3, and K_1=5, K_2=4, which is the case of $n_1+n_2 \leqq S$, the multiplication along the route of broken line (i) in Figure 5 is $Q_S(0,0)\cdot\dfrac{4\nu_2}{\mu_{01}^2}\cdot\dfrac{3\nu_2}{2\mu_{02}^2}\cdot\dfrac{5\nu_1}{\mu_{12}^1+2\mu_{12}^2}$. For the route from (0, 0) to (4,2), which is the case of $n_1+n_2>$S, the multiplication along the route (ii) is $Q_S(0,0)\cdot\dfrac{5\nu_1}{\mu_{01}^2}\cdot\dfrac{4\nu_2}{\mu_{11}^1+\mu_{11}^2}\cdot\dfrac{4\nu_1}{2\mu_{21}^1+\mu_{21}^2}\cdot\dfrac{3\nu_1}{2\mu_{21}^1+\mu_{21}^2}\times\dfrac{3\nu_2}{2\mu_{21}^1+\mu_{21}^2}\cdot\dfrac{2\nu_1}{2\mu_{21}^1+\mu_{21}^2}$. Since there are multiple routes from (0, 0) to (n_1, n_2), the coefficient of $Q_S(n_1, n_2)$ related to $Q_S(0, 0)$ is approximately represented as the total of the multiplication based on all routes. Similarly to the case above, we can approximately calculate the state probability of a queuing network with multiple job classes when R >2.

4. Numerical Experiments

We evaluated the proposed approximation technique through numerical experiments. We used the following parameters.

1. Number of terminals: K_1 = 3, 4, …, 20; K_2 = 3
2. Number of memory resources: S = 3
3. Think-time: (t_1, t_2) = (20, 10), (5, 2.5),
   where t_r is the think-time of job class r (r = 1, 2).
4. Number of I/O nodes: M = 2
5. Total service time at each node
   τ_{10}=1.0, τ_{11}=τ_{12}=1.0
   τ_{20}=1.0, τ_{21}=τ_{22}=0.5,
   where τ_{rm} is the total service time of job class r at node m.

Figures 6 ~ 9 show the mean system response times and mean number of jobs in inner level of job classes 1 and 2 respectively, when K_2 is fixed at 3, and K_1 changes from 3 to 20. The mean system response time is the mean time from job arrival to departure from the inner level (that is the mean time from departure from the terminal to coming back to the terminal). Similarly to the case of a single job class, the mean system response time for both job class increases monotonically in a convex curve. As the number of terminals K_1 of job class 1 is increased and K_2 is fixed (the only traffic of job class 1 is increased), not only the mean response time of job class 1 but also job class 2 increases, because the entire central server model is more crowded by increasing the traffic of the job class 1. We can see that the mean response time of job class 1 and

job class 2 is nearly increasing in the range of heavier traffic. This reason can be presumed that the behavior of the mean system response time in the heavier traffic range is approximately linear to the number of jobs.

5. *Conclusion*

We proposed an approximation technique for evaluating the performance of computer systems in finite input source using a queuing network and analyzed its performance measures through numerical experiments. The concept of the approximation is based on separately analyzing the inner level (CPU, I/O equipment, and memory) and outer level (terminals and communication lines). The numerical experiments clarified the characteristics of the system response time.

In the future we plan to examine the accuracy of the proposed approximation technique by comparing it with exact solutions or simulation results.

REFERENCES

[1] F. Baskett, K. M. Chandy, R. R. Muntz and F. G. Palacious, "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," J. ACM, Vol.22, No.2, pp.248--260, April 1975.

[2] H. Kobayashi, "Modeling and Analysis," Addison-Wesley Publishing Company, Inc. 1978.

[3] T. Kurasugi and I. Kino, "Approximation Method for Two-layer Queueing Models," Performance Evaluation 36--37, pp.55--70, 1999.

[4] J. A. Rolia and K. C. Sevcik, "The Method of Layers," IEEE Trans. on Software Engineering, Vol.21, No.8, pp.689--700, Aug. 1995.

[5] T. Kinoshita and Y. Takahashi, "A Queuing Network Modeling and Performance Evaluation Method for Computer Systems with Resource Requirement," IEICE D-I, Vol. J 82-D-I, No.6, pp.701--710, Jun. 1999.

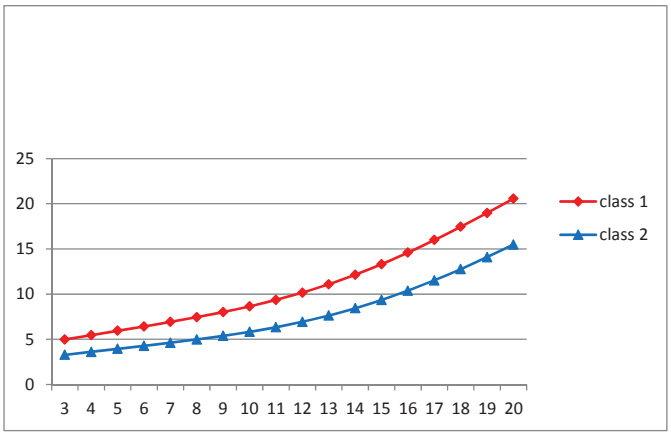


Figure 6. System response time ((t_1, t_2)=(20, 10))

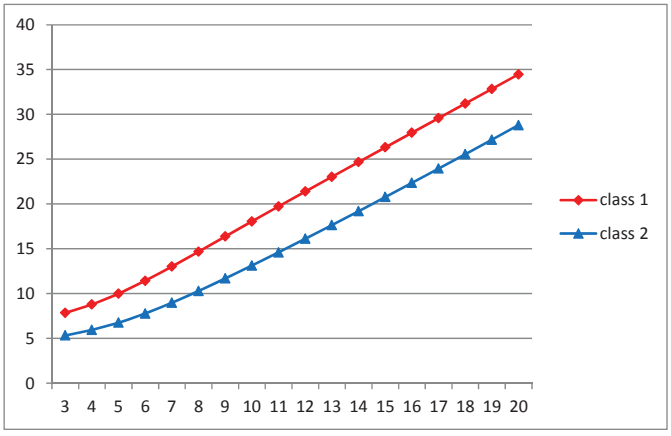


Figure 8. System response time ((t_1, t_2)=(5.0, 2.5))

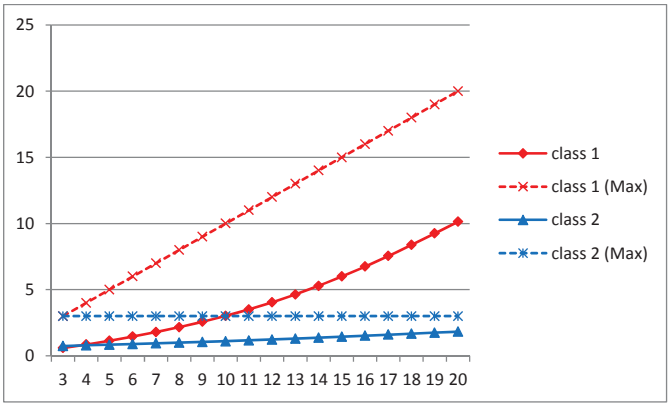


Figure 7. Mean number of jobs in inner level ((t_1, t_2)=(20, 10))

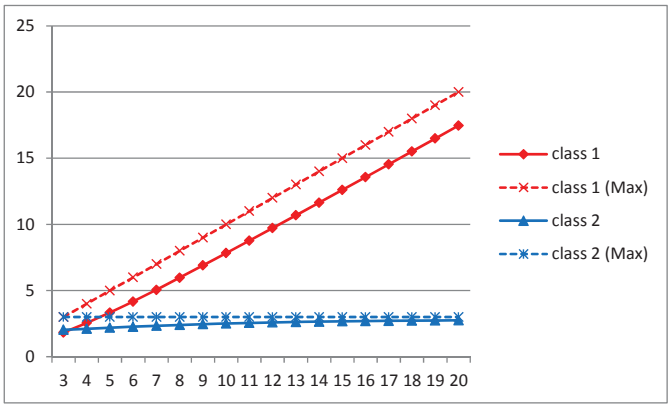


Figure 9. Mean number of jobs in inner level ((t_1, t_2)=(5, 2.5))

[6] T. Kinoshita and X. Gao, “Queuing Network Approximation Technique for Evaluating Performance of Computer Systems with Memory Resources,” PDPTA2010, pp.640--, 646, July 2010

[7] A. Razali, T. Kinoshita and A. Tanabe, “Queuing Network Approximation Technique for Evaluating Performance of Computer Systems with Multiple Memory Resource Requirements,” PDPTA2012, pp.758--763, July 2012

[8] O. E. Oguike, M. N. Agu and S. C. Echezona, ”Modeling Variation of Waiting Time of Distributed Memory Heterogeneous Parallel Computer System Using Recursive Models,” International Journal of Soft Computing and Engineering, vol. 2, Issue 6, Jan 2013

[9] A. Gandhi, S. Doroudi, M. Harchol-Balter and A. Scheller-Wolf, “Exact Analysis of the M/M/k/setup Class of Markov Chains via Recursive Renewal Reward,” SIGMETRICS’13, pp.153--166, June 2013

[10] M. Takaya, M. Ogiwara, N. Matrazali, C. Itaba, I. Koike, T. Kinoshita, “Queuing Network Approximation Technique for Evaluating Performance of Computer Systems with Memory Resource used by Multiple job types,” CSC2014, pp.41--46, July 2014

Shuffled Frog Leaping Algorithm for 0/1 Knapsack Problem on the GPU

Pranav Bhandari, Rahul Chandrashekhar, and Peter Yoon
Department of Computer Science
Trinity College
Hartford, CT 06106 USA

Abstract— *This paper presents an accelerated implementation of the discrete shuffled frog leaping algorithm (DSFLA) to solve 0/1 Knapsack Problem, an important class of combinatorial optimization problems. The DSFLA is known for its numerical stability, but, as the number of objects in the dataset increases, the number of possible solutions also increases exponentially, making this approach computationally impractical. To that end, the proposed parallel algorithm exploits a massively parallel architecture of modern graphics processing units (GPUs) to make the DSFLA computationally competitive in real situations. The experimental results show that it is very effective in solving 0/1 knapsack problems of various sizes, rendering an optimal solution. When implemented on a multi-GPU system, our parallel algorithm exhibits a high-level of scalability on large datasets.*

1. Introduction

Given n items, each having weight w_i and value v_i, we consider the problem of collecting the items in the knapsack in such a way that the total value of the knapsack is maximized while the capacity of the knapsack, W, is not exceeded, that is,

$$\text{maximize} \sum_{i=1}^{n} v_i \delta_i \tag{1}$$

$$\text{subject to} \sum_{i=1}^{n} w_i \delta_i \leq W \tag{2}$$

where

$$\delta_i \in \{0, 1\}, \quad i = 1, \ldots, n \tag{3}$$

This problem is known as *0/1 knapsack problem*, an important class of combinatorial problems.

There are several applications of the 0/1 knapsack problem, including industrial applications such as cargo loading, project selection, and budget control. For example, a shipping company would want to maximize the volume of shipments that they are sending in each shipment. Choosing the right combination of items to send will help the company to ship more efficiently. Another application is in the real estate industry. The 0/1 knapsack problem can be defined in which the items are real estate properties and the knapsack capacity is the available investment. The aim is to buy properties such that the profit is maximized based on the data from the previous period.

Several methods have been proposed over the years in order to solve the 0/1 knapsack problem. When the brute force method is used, since there are n items to choose from, there will be 2^n number of possible combinations. Each combination is then checked to see which one has the highest value and is below the weight limit of the knapsack. This simple approach might be useful for a small data, but, as the data size increase, it becomes impractical because of the computational time it would require. Other approaches like dynamic programming and approximation algorithm have also been investigated [3], [4], [5], [6].

Recently, it has been shown that the shuffled frog leaping algorithm (SFLA), a memetic meta-heuristic algorithm, can be used to efficiently solve the 0/1 knapsack problem [2]. The algorithm creates a population of solutions which are known as "frogs." It uses particle swarm optimization in order to perform the local search and then improve the population to bring it closer to the optimal solution.

However, the SFLA in itself cannot be used to solve the 0/1 knapsack problem. In order to satisfy the condition (3) of the 0/1 knapsack problem, the SFLA has to be modified slightly. For the 0/1 knapsack problem, each solution is a series of bits of length n where each bit represents an item in the list and the value of the bit shows if the item is included in the knapsack. If the ith bit is 1, then the item i is included; otherwise it is excluded. This algorithm is called the discrete shuffled frog leaping algorithm (DSFLA).

Although the DSFLA can lead to an optimal solution, the algorithm might take too long when the problem set increases. As the number of objects in the list increases, the number of frogs that should be processed also increases, making the procedure computationally impractical.

In this paper, we present an efficient implementation of the DSFLA on multi-GPU systems in order to accelerate the overall computations. As the algorithm is embarrassingly parallel, there is a potential to achieve good speed-ups. Using the GPU, threads can be created which can each create a frog at the same time. The division of memeplex is also implemented in parallel. Each memeplex is assigned a thread where it undergoes local search. This is where the most

computation time is saved as the local search is the most computationally expensive part of the algorithm. After each memeplex goes through the local search, the memeplexes are combined to determine the final population at the end. The population converges towards the optimal solutions and the best frog of the final population is the best solution.

The remainder of this paper is organized as follows: Section 2 briefly describes the DSFLA, Section 3 presents a detailed parallel implementation, Section 4 presents the experimental results followed by discussion and future directions in Section 5.

2. Shuffled Frog Leaping Algorithm

The SFLA is a popular choice of solving the 0/1 knapsack problem for its fast convergence and stability. It has been used to solve many nonlinear optimization problems. A detailed description of the algorithm is given in [2].

In this algorithm based on the divide and conquer paradigm, we construct a "population" of "frogs," each of which represents a solution to the problem. These frogs are then partitioned into families of frogs, known as *memeplexes*. Each memeplex acts as an independent family of frogs and performs a *local search*. Within each memeplex, the frogs can be influenced by the state of another frog and reorganize themselves by going through a somewhat evolutionary process to determine the best solution.

After performing a number of evolutions, all the frogs are regrouped, shuffled and divided into families again. This causes an exchange of information and can have a greater influence on other memeplexes. As the SFLA is a meta heuristic algorithm, it is not guaranteed that the evolutions will always converge to a unique solution, but it will converge to an optimal solution. Local searches continue until a predetermined convergence criteria is met.

To begin a search process, an initial population of frogs, P, is first created randomly. Here, the number of frogs is predetermined. The frogs are then sorted in a descending order on the basis of their fitness and divided into m memeplexes each containing n frogs. The frogs are assigned to each memeplex, so that the first frog goes to the first memeplex, the second frog goes to the second memeplex, and so on. We continue to assign each frog in cyclic fashion, that is, the $(m+1)^{st}$ frog ends up in the first memeplex.

Next, we find the best frog, X_b, and worst frog, X_w in each memeplex based on their fitness. At this point, the global best frog, X_g, is also determined. As a part of every evolution, the frog with the worst fitness in each memeplex undergoes an improvement step given by

$$D_i = \texttt{rand} \cdot (X_b - X_w), \tag{4}$$

where D_i is the change in the position of the ith frog and `rand` represents a random number from $(0, 1]$. The new position of the frog is then determined by

$$X_w^{(new)} = X_w + D_i \tag{5}$$

$$-D_{max} \leq D_i \leq D_{max} \tag{6}$$

Here, D_{max} represents the maximum change that is allowed to a frog's position. If this step yields a better solution, it replaces the worst frog X_w. If not, just replace X_b by X_g in (4). If it still does not improve the solution, a randomly generated frog is used to replace the worst frog. The whole process continues until the convergence criteria is satisfied.

2.1 Discrete Shuffled Frog Leaping Algorithm

Unlike other knapsack problems, the 0/1 knapsack problem does not allow a fraction of an item to be included. It is either included or excluded in entirety. Thus, we use a slightly modified version of the SFLA, called the Discrete Shuffled Frog Leaping Algorithm (DSFLA). With this algorithm the worst frog in each memeplex is modified by

$$D_i = \begin{cases} \min\{\texttt{rand} \cdot (X_b - X_w), D_{max}\} \text{ for a positive step} \\ \\ \max\{\texttt{rand} \cdot (X_b - X_w), -D_{max}\} \text{ for a negative step} \end{cases} \tag{7}$$

The new position of the frog is determined in the same way described in (5). The DSFLA is very similar to the SFLA in the sense that the basic approach to perform the local search remains the same. If there are no improvements possible for the worst frog, the best frog is replaced by the global best frog. If that fails as well, we use a randomly generated frog. The other important parameters such as the population P, the number of memeplexes m, the number of frogs per memeplex n, and the number of shuffling iterations remain the same. The DSFLA is summarized in the following:

Algorithm 1 Algorithm for DSFLA

1: Generate random population of *P* solutions (or frogs)
2: **for** each individual $i \in P$ **do**
3: Calculate fitness(i)
4: Sort the population P in descending order of their fitness
5: Divide P into m memeplexes
6: **for** each memeplex **do**
7: Determine the best frog, X_b, and the worst frog, X_w.
8: Improve the position of X_w using (5) and (7).
9: Repeat the process for p times,
10: where p is a predefined number of iterations.
11: Combine the evolved memeplexes
12: Sort the population P according to their fitness
13: **if** terminated **then return** the best solution

2.2 Modifications for the 0/1 Knapsack Problem

It has been empirically observed that some modifications must be made to the DSFLA in order to make the local search process more suitable for the 0/1 knapsack problem. The local search procedure accounts for high convergence speed, and the purpose of using the DSFLA is to take advantage of this property. But at the same time, we also risk of a possibility of the local search oscillating between a range of optimum solutions. Thus, we must maintain the balance between the convergent and divergent properties of the approach.

2.2.1 Discretization

A critical part of the DSFLA is where we need to determine which frog is the worst frog, the one above or below the weight limit. Thus, we first check the magnitude of the frogs at two extremes. The one which is the farthest is the one that needs improvement. Therefore, the points farther away from the weight limit converge first.

Based on this, we can make two types of improvements: a positive improvement and a negative improvement. The positive improvement is done when the frog below the weight limit is selected. To improve this frog, a 0 bit is selected at random and changed into a 1. Therefore, the weight and the value of the item added will be a part of the solution. On the other hand, the negative improvement is possible when when the frog above the weight limit is selected. To improve this frog, a 1 bit is selected at random and changed into a 0. Therefore, the weight and the value of the item is removed from the solution as illustrated in the following:

$$t = X_w + D \tag{8}$$

$$X_w^{(new)} = \begin{cases} 0 & t \leq 0 \\ \texttt{round}(t) & 0 < t < 1 \\ 1 & t \geq 1 \end{cases} \tag{9}$$

This method ensures that we get only integral values and makes the convergence faster.

2.2.2 Stopping criteria

As the DSFLA is a meta heuristic algorithm, the choice of termination condition is critical. But as the algorithm is highly sensitive to the underlying datasets, we cannot always be sure as to when to terminate. To that end, we use two approaches.

The first approach is to terminate when we reach the maximum number of iterations. The maximum number of iterations is one of the parameters that has to be passed to the algorithm. Once the local search runs for the maximum number of times the algorithm is terminated and the memeplexes are then combined to determine the solution.

Another strategy is to maintain a threshold and terminate the algorithm when a suitable level of accuracy has been obtained. In other words, the program terminates when a desired accuracy has been reached or if there is no further improvement. We can conclude that there is no further improvement when every frog in the memeplex becomes equivalent to the best frog.

3. GPU Implementation

The main purpose of our GPU implementation is to exploit the embarrassingly parallel nature of the algorithm. Recently, GPUs have been widely used for graphics processing on multiple platforms. It has been also found that they can be used for massively parallel numerical, analytical and scientific problems. By using the numerous GPU cores, the overall processing can be accelerated significantly. Our implementation was done using CUDA, a computational platform specifically designed for general-purpose computing with Nvidia GPUs.

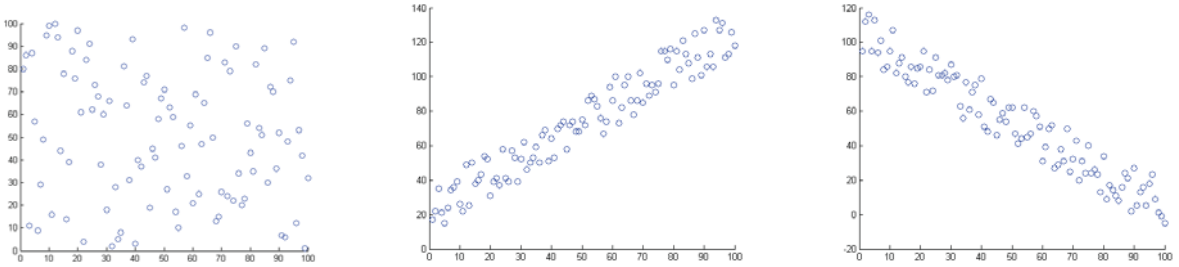
3.1 CUDA

The availability of massive number of cores in the GPU makes CUDA a powerful programming environment to develop parallel algorithms. The CPU and the main memory are known as *host* while the GPUs are referred to as *device*. A *kernel* is a routine that executes on the device. The kernels designated as `__global__` can only be invoked by the host while those designated as `__device__` can only be called by a kernel. To successfully exploit parallelism, CUDA employs units called *blocks* and *threads*. A GPU device assigns a copy of kernel code to every block which is further broken down into threads, which work simultaneously.

Unlike CPU cores, GPU devices do not share their memory. CUDA's unified memory model provides a framework where the CPU and the GPU can share their memory in one unit. We make an extensive use of the unified memory model in order to carry out our algorithm in parallel. While this is just from the programmer's perspective, CUDA manages the memory and transfers of data in the background. We use a hybrid computational model in order to carry out our implementation. This model employs both CPUs and GPUs where the CPUs coordinate the communication between the GPUs and the GPUs perform the actual computations. Because the limited GPU memory for the program, we fixed the size of the GPU workspace in order to avoid running out of GPU memory irrespective of the input size. As the size of the input is independent of the GPU memory, we must account for dynamic partitioning of the data and tasks. The following sections further explain the hybrid model used in our implementation.

3.2 Creating Frogs

The creation of the frogs is the first step of the algorithm where each frog is a structure with attributes: weight,



(a) No correlation (b) Positive correlation (c) Negative correlation

Fig. 1: Random datasets representing Weight vs Value

value and a bitstring. The bitstring is of the length n, the number of objects in context. Each bit is either 0 or 1, indicating whether the respective object is included or excluded from the solution. Similarly, every frog represents a unique solution to the given data set. The algorithm, being embarrassingly parallel, gives us the freedom to implement this step on the device as well. Since the creation of every frog is independent from the other frogs, a GPU thread is assigned to carry out this task. Each thread first spawns a random potential solution which is a series of bits of 0's and 1's of length n. Then the thread calculates the weight and the value of the frog using the bits. An algorithm outlining the parallel creation of frogs is as follows :

Algorithm 2 Algorithm for creating frogs

1: **Input** : The population of frogs P of size m, set of objects S of size n
2: **for** each frog P_i in the population **in parallel do**
3: Spawn a new GPU thread t_j
4: Assign t_j to a frog P_i
5: Generate a random bitstring b_i in t_j
6: Calculate the weight and value of P_i based on b_i
7: **return** P

3.3 Construction of Memeplexes

After each thread completes creating and evaluating the frog, these frogs are sorted in the descending order of their value. As the sorting process involves the interaction of multiple frogs, this process cannot be carried out in parallel. Since the aim of the implementation is optimization, using conventional sorting algorithms such as bubble sort or selection sort do not provide as much of a speed up. To harness the computing power of GPUs, we use Nvidia's Thrust library [7] to perform the sorting. Thrust is a C++ template library for CUDA which is based on the Standard Template Library. Its sorting procedure implemented on CUDA gives a speed-up of as much as 10x running on the GPU as compared to the `sort()` function of the C standard library. We next divide the population into memeplexes. This division is also performed in parallel by assigning one thread to one memeplex. For every iteration, a thread will select a frog from the sorted population and place it in the respective memeplex. This is carried out simultaneously by all the threads in the block making this process parallelizable as well.

Algorithm 3 Algorithm for constructing memeplexes

1: **Input** : The population of frogs P of size m
2: Spawn a GPU thread to sort the population P with the population as the parameter
3: Sort the population P in descending order of their fitness using the Thrust library
4: **for** each frog P_i in the population **in parallel do**
5: Spawn a new GPU thread t_j
6: Assign t_j to a frog P_i
7: Determine the index *tid* of t_j using the formula: `threadIdx.x + blockIdx.x * blockDim.x`
8: Assign P_i to the memeplex number (tid mod m)
9: **return** set of memeplexes M

3.4 Local Search

Each memeplex is assigned to a thread where the local search takes place. To maintain the uniqueness of every thread, we associate an ID with each memeplex. This helps to relate a thread to its respective block on the device and can be used as an index for the thread. Note that this is the most computationally expensive part of the algorithm. As mentioned earlier, since the evolution of every memeplex is independent from the other in a single iteration of the local search, we employ a single GPU thread to perform a local search on every memeplex. This continues until the termination criteria is met. After the local search is completed, the frogs in the memeplex are sorted in the descending order again. This is also performed through Nvidia's Thrust Library.

After the local search is completed, the memeplexes are combined back together and sorted in descending order

again. After this is done, the frog with the highest value under the knapsack limit is the best solution. All the frogs in the memeplex converge to a central value with each iteration as each frog is improved in each step. A detailed algorithm is given in Algorithm 4.

Algorithm 4 Algorithm for parallel local search

1: **Input** : Set of memeplexes M, set of objects, Global best frog X_g
2: **for** each memeplex M_i **in parallel do**
3: Spawn a new GPU thread t_j
4: Assign t_j to a memeplex M_i
5: Determine the best frog, X_b, and the worst frog, X_w for M_i
6: Improve the position of X_w using (5) and (7).
7: Calculate number of frogs within optimum threshold
8: **if** accuracy has improved **then** repeat local search
9: Combine the evolved memeplexes
10: Sort the population P according to their fitness
11: **if** terminated **then return** the best solution

3.5 Synchronization

As we are using a hybrid approach, synchronization becomes a key factor for our implementation to function smoothly on a CPU-GPU hybrid model. To exploit the massively parallel nature of GPUs, we employ thousands of threads to carry out the tasks. Threads are the smallest unit of execution path on the GPU. Even though they seem to be working simultaneously, there is a fraction of a delay between the scheduling of threads. Owing to this, the threads need to be synchronized at multiple points in the algorithm. This is achieved through the `__syncthreads()` call. The threads belonging to a specific block are all scheduled under one streaming microprocessor. Hence the synchronization is computationally inexpensive on the GPU.

4. Experimental Results

We tested our implementation of the DSFLA on a system which comprises a dual Intel Xeon E5-2620 CPUs with 64 GB main memory, and four NVIDIA Tesla K20c GPU, each with 5 GB global memory.

4.1 Datasets

We tested our implementation using two types of datasets: random and real estate data.

4.1.1 Random Data

This dataset contains randomly generated data without correlation, with positive correlation and with negative correlation as seen in Figure 1. To ensure that a solution exists to the input data, we use a data generator which prepares a data set on the basis of several parameters. The parameters that we use are the number of objects and the capacity of the knapsack. To be able to test our implementation successfully, it is essential to know that a solution exists to the input data hence the first step is to generate solutions to the given input set. To achieve this, the generator determines the average weight of the objects and distributes the weights across the objects so that the total weight is the weight limit. This guarantees the existence of a few solutions in the generated frogs and we can be sure that the algorithm will eventually converge to some solution. The other possible solutions are generated keeping the value to weight ratio of the known solutions much higher than the value to weight ratio of the newly generated ones. This accounts for easier verification of the solution set and fast convergence.

4.1.2 Real Estate Data

We also experimented our GPU-based DSFLA with real-world data in the form of real estate information. In this application, we consider a real estate property to be an item. The weight of the item here is the price of the property and the value is the net increase in the price of the property from last year. Thus, here what the DSFLA will do is try to maximize the price increase. For the price increase we use percent increase or exact monetary values. Instead of data of particular properties, average data can be used when looking to choose between different areas to invest in.

There are various ways this approach can be used to get information in the real estate area. Instead of the price increase, we can use other information such as the value of the frog to extract additional facts. If the value is set to decrease in price, it will show the properties that lost the most value.

Figure 1 shows the the real estate data from the state of Connecticut we used for the experiment [8]. The datasets provide information such as the number of properties sold in the particular county, the median price at which it was sold and the change in the price of the property from the last quarter. We collected the increase in price and average cost of a property of various counties and ran the DSFLA on this data. It must be noted that since this is an application of the algorithm, the solution set is purely computational and not very realistic. In order to achieve pragmatic results, pre-processing of the datasets was required. We also observed that purchasing more properties in the same county can be more productive and hence alter the number of frogs for a property accordingly.

For the pre-processing we first calculated the expected number of houses that will be sold in a county in the next quarter depending on the data of the previous quarter. After the prediction, we assigned each county with *market influence constant*, k, which estimates the percentage of market that you can influence. It basically is an estimation of how many houses one will realistically be able to buy considering all of them are on sale. It is very unlikely that a

Table 1: Real Estate Data from Berkshire Hathaway with Market Influence Constant $k = 0.5$

County	Median price (in USD)	Change in price (in %)	Profit (in USD)	Sales	Change in sales (in %)	Expected sales	# frogs
Berlin	265,000	7.1	18,815	51	6.3	54	27
Bloomfield	156,000	-16.6	-25,896	53	12.8	59	29
Farmington	397,000	10.3	40891	42	9.95	46	31
Southington	274,500	-0.1	-274.5	127	8.5	137	68
New Britain	140,000	12	16800	73	-21.5	57	28
Manchester	162,500	-2.7	-4387.5	143	-10.1	128	64
West Hartford	322,000	5.6	18032	137	-8.23	125	57
Windsor	204,500	-9.1	-18609.5	84	18.3	99	49
Enfield	168,000	8.4	14112	129	31.6	169	84
Hartland	270,000	0.8	2160	55	-20.3	43	21
Marlborough	275,000	13.6	37400	19	-9.5	17	8
New Britain	140,000	12	16800	73	-21.5	57	28
Plainville	175,000	-15.5	-27125	45	-15.1	38	19
Rocky Hill	304,300	10.6	32255.8	32	-8.6	29	14
Suffield	330,000	15.2	50160	36	37.5	50	26

person or company will purchase all the properties within a certain county. So depending on the estimation the value of k can be set, k times the number of estimated houses sold gives the number of frogs for that particular county. For this purpose we have set k to 0.5.

The weight, in our case, is the cost of the property and the maximum weight is the amount of money that is available for investment. The profit made by the property in the last quarter is considered as the value. We assume that the rate of change of the price will be the same and consider the potential profit that could be made.

4.2 Discussions and Conclusion

The results obtained indicated good speedups. For lower dimensional problems, it is expected that the serial algorithm will run faster than the parallel one, but as we increase the dimension of the problem the speedup becomes more and more evident. The multi-CPU vs multi-GPU comparison also shows a favorable speedup in the case of the latter.

Figure 2 shows the Accuracy Result and the Timing Result of the algorithm on randomly generated data for different dimensions of the memeplexes on a single GPU. It shows how the dimension of the memeplex also slightly affects the performance of the algorithm. The Accuracy Result shows how having fewer memeplexes with more frogs is favorable for the accuracy as compared to more memeplexes with fewer frogs. The Timing Result also favors the former when it comes to faster running time.

For most cases, the data must be exchanged between the CPU and the GPU and for smaller dimensional problems, this is what slightly sets back the GPU based implementation as compared to the serial version. For smaller data sets, the time taken for this data transfer is very small and almost

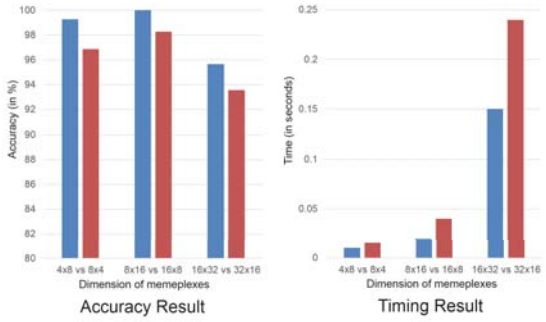


Fig. 2: Accuracy and Timing Result for randomly generated data without correlation

negligible. And as the problem size grows, the time taken for exchange of data gets magnified hence giving us an exponential increase in the time taken. The number of items that we have in the problem affects the time taken by the algorithm. When we increase the number of items, there is an exponential increase in the amount of time taken.

This can be seen clearly in Table 2 where the increase in the step size of the objects exhibits an exponential increase in the running time of the implementation. The data used is positively correlated. As seen in the case of the multi-CPU system, the running time is faster for smaller cases but the multi-GPU system dominates when it comes to large test cases. The speed-up is at least 7x for all sizes of the problem. The accuracy achieved was 82.7% on average.

Through multiple test runs across various implementations, there are some patterns that were observed which affect the performance of the algorithm. It is observed that increasing the number of frogs in each memeplex increases

Table 2: Running time (in seconds) for positively correlated data with speed-up of a multi-GPU system (4 Nvidia K20c) over multi-CPU system (16 cores)

N	Multi-CPU	GPU	Multi-GPU	Speed-Up
500	30.174	6.45	4.13	7
1000	77.53	9.88	7.32	10
2000	178.32	29.13	21.33	8
2500	197.17	37.71	27.88	7

the number of steps taken to converge to the result in most cases. This is expected because adding more frogs also accounts for more frogs to be improved through the local search. But since the algorithm is a meta-heuristic algorithm and the correct solution is not guaranteed, increasing the number of frogs improves the accuracy of the solution set generated. As the number of frogs increases, there are more frogs that converge near the correct solution giving us a more accurate result.

The tests performed on correlated data have shown better accuracy than that in a general case with randomly generated data. The correlated data follows a trend and since the frogs are sorted before dividing them up into memeplexes, they are evenly spread out across the memeplexes. Owing to this, the number of changes required in the bitstring of the frog to make it converge are less. And since the number of bit flips is lower, the number of steps required also decreases. This factor accounts for quicker convergence with fewer steps.

The number of iterations also plays an important role in the results while considering larger test cases. For smaller cases, the program terminates on converging to the most optimal solution. But as we increase the number of frogs, it becomes increasingly difficult to keep all the frogs within the optimal threshold. It was observed in a few cases that some frogs, after reaching the optimal solution tend to diverge from it over the course of multiple iterations. No specific pattern has been observed with respect to this anomaly due to which the accuracy has suffered.

The real estate data required a significant amount of polishing before it could have been used in order to obtain practical results. The results that we obtained were mathematically correct but were not feasible at all which made us do the pre-processing in the data after which the results have been satisfactory.

For our application of the DSFLA on real estate data, we used \$2 million as the maximum "weight", W. As the data set was relatively small, the results could be verified for plausibility. After performing multiple test runs on the real estate data, the results proved to be consistent with what would be a good investment in terms of maximizing profit. On an average, the set of estates that would be ideal for investing within the allotted budget were: Suffield, Farmington, Marlborough, Rocky Hill, Berlin and West Hartford. The results from this application are consistent with those achieved for the datasets with randomly generated data as presented in Table 2. The average speedup was nearly 8x.

Through the analysis performed and results collected, it has become clear that there exists a trade-off in the implementation when it comes to accuracy and the size of the problem. As the problem size increases so does the number of frogs. An increase in the number of frogs exhibits a better accuracy up until a certain point but for data sets of size $> 3,000$ and above, the accuracy suffers as the problem size increases. This leads us to a trade-off between the accuracy and the size of the problem.

It must be noted that these observations are based on multiple test runs and the average of the cases has been considered. The DSFLA, being a meta-heuristic algorithm does not guarantee convergence to a unique solution and so the solution set that is generated might be different for the same input set run multiple times. However, the overall performance of our GPU implementation of the DSFLA has been proven effective in solving the 0/1 knapsack problem.

Acknowledgments

The authors would like to thank CUDA Teaching Center Program, Nvidia Research, and Interdisciplinary Science Program, Trinity College, for supporting this research.

References

[1] T. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithm, 3rd Edition*, The MIT Press, 2009.

[2] M.M. Eusuff, K. Lansey, F. Pasha, *Shuffled frog leaping algorithm: a memetic meta-heuristic for discrete optimization*, Engineering Optimization, 38:129-154, 2006.

[3] Robert M. Nauss, *The 0/1 knapsack problem with multiple choice constraints*, European Journal of Operational Research, 2(2):125-131, 1978.

[4] Sartaj Sahni, *Approximate algorithms for the 0/1 knapsack problem*, Journal of the ACM, 22(1):115-124, 1975.

[5] Silvano Martello, *New trends in exact algorithms for the 0/1 knapsack problem*, European Journal of Operational Research, 123(2):325-332, 2000.

[6] David Pisinger, *Linear time algorithms for knapsack problems with bounded weights*, Journal of Algorithms, 33(1):1-14, 1999.

[7] Thrust User's Guide, Nvidia Corporation, 2012.

[8] www.bhhsneproperties.com.

Analysis of a Genetic Algorithm-based Approach in the Optimization of the SourceAFIS's Matching Algorithm

A. G. A. Silva[1], I. A. Almeida Junior[1], Rodrigo L. Parente[1], L. V. Batista[1], João J. B. Primo[2], Adriano S. Marinho[2], Pedro Alves[2]
[1]Informatics Center, Federal University of Paraíba, João Pessoa, Paraíba, Brazil
[2]Research Department, VSoft Tecnologia LTDA, João Pessoa, Paraíba, Brazil

Abstract— *Fingerprints are an important biometric trait, being the most widely deployed biometric characteristic. Minutiae-based methods are the most popular technique to compare fingerprints, but as the pairing is rarely perfect the score between two fingerprints may also depend on another factor. Usually, a constant weight is computed either empirically or statically and assigned for each factor. Optimize these weights can be a hard task for a large N-dimensional attributes space. Genetic Algorithms (GA) is an optimization approach based on Darwin's theory of evolution of species and it has been proved to be quite successful in finding good solutions to such complex problems. In this work, we analyze an GA-based optimization approach for SourceAFIS's fingerprint authentication algorithm. The DB2 fingerprint database from FVC 2006 project was used to validate our method. As best result, an EER of 0.379% was achieved.*

Keywords: fingerprint, matching, scoring, genetic algorithm, optimization

1. Introduction

Reliable identification systems have become a key issue for applications that authenticate users. Traditional methods to establish user's identity include mechanisms based upon knowledge (e.g., passwords) or tokens (e.g., identification cards). However, such mechanisms may be lost, stolen or even manipulated to spoof the system. In such a context, biometrics rises as an alternative. [16].

Biometrics (biometric recognition) refers to the use of distinctive anatomical and behavioral characteristics such as fingerprints, iris, face and voice for automatically recognizing a person. Biometrics provides better security, higher efficiency, and, in many instances, increased user convenience. Because biometric identifiers cannot be easily misplaced, forged, or shared, they are considered more reliable for person recognition than a traditional token or knowledge-based methods. Thus, biometric recognition systems are increasingly being deployed in a large number of government and civilian applications [13].

Because fingerprints are unique to individuals, invariant to age, and convenient in practice [12], they are the most widely deployed biometric characteristics. Fingerprint are used in every forensics and law enforcement agency worldwide routinely [13].

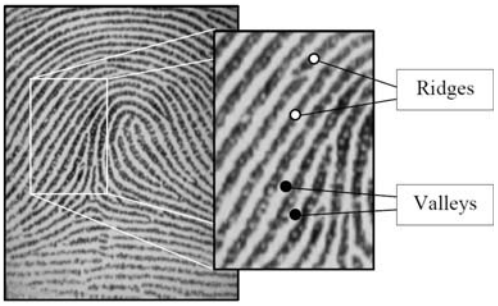


Fig. 1: Ridges and Valleys

A fingerprint image consists of interleaving ridges and valleys. Usually, interleaving ridges are dark lines and valleys are the white lines between ridges (see Figure 1). Ridge terminations and bifurcations are kinds of minutiae which are characteristic features of fingerprints [19]. Usually, minutiae have localization and direction as attributes. Most Automated Fingerprint Identification Systems (AFIS) are based on minutiae.

Fingerprint images usually need to be segmented in background and foreground - where the foreground is the Region of Interest (ROI). The segmentation removes uninterested regions before some other steps such as enhancement and minutiae detection. Hence, the image processing will consume less CPU time and avoid undesired errors as detection of spurious minutiae in low-quality image regions.

A matching algorithm is used to compare two given fingerprints. In case of identification, the algorithm returns a similarity level between fingerprints. In authentication, on the other hand, an answer indicating if two fingerprints belong or not to the same person is returned. Usually, a threshold value previously defined is used to answer it.

The most popular and widely used technique to compare fingerprints is based on minutiae. Minutiae are extracted from the two fingerprints and stored as sets of points in the two-dimensional plane. Minutiae-based matching essentially consists of finding the alignment between fingerprint's minutiae sets that results in the maximum number of minutiae

pairings. [13].

The pairing of minutiae in most cases generates a score value. Because the pairing is rarely perfect, this score may also depend on other factors like minutiae type, minutiae position, minutiae location and minutiae direction error. Usually, a constant weight which is computed either empirically or statically is assigned for each factor. Optimize these weights can be a hard task for a large N-dimensional attributes space.

Genetic Algorithms (GA) are an optimization approach based on the principle of natural selection of Charles Darwin. These algorithms input is an N-dimensional vector that will be optimized according to a fitness function. GAs proved to be quite successful in finding good solutions to such complex problems as the traveling salesman, the knapsack problem, large scheduling problems and others [4].

In this work, our goal is to analyze the optimization gain performed by a GA approach in the fingerprint matching algorithm SourceAFIS. The database DB2 of FVC 2006 project [5] were used to validate our study.

2. Related Works

In this section, we list some related works that may be useful to the reader. First, is appropriate to cite the work of John Holland [9] who was the first scientist to describe evolutionary algorithms, during the 60s. Such work provides a good background about Holland's goal in understanding the life adaptation as like it occurs in nature and the ways of developing systems based on these principles.

In order to a good theoretical foundation on evolutionary algorithms Back et al. [1] provides an overview of the three main branches of evolutionary algorithms (EA): evolution strategies, evolutionary programming, and genetic algorithms. In their work, certain characteristic components of EAs are considered: the representation scheme of object variables, mutation, recombination, and selection operators.

Considering the importance of parameter optimization on biometric systems, the work by Goranin et al. [7] analyzes GA application in that context. According to this paper, the use of evolutionary algorithms may ensure a qualitative increase of biometric system parameters, such as speed, error rate, and flexibility.

With regard to fingerprint matching optimization, Jiang and Yau [10] use the local and global structures of minutiae in their approach. The local structure of a minutia describes a rotation and translation invariant feature of the minutia in its neighborhood while the global structure tries to determine the uniqueness of a fingerprint.

The matching algorithm proposed by [11] uses triangular matching to deal with the deformations of fingerprints. A Dynamic Time Warping (DWT) is used to validate the final results of fingerprint matching. Without DWT, the results are not acceptable though.

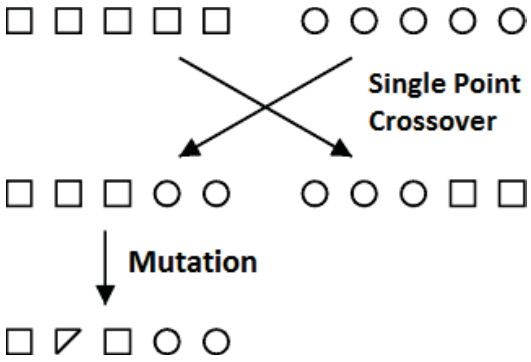


Fig. 2: Example of Crossover and Mutation operators. Where the circles represents one type of genes and the squares another one. Note that after mutation operation a gene mutates.

The closest work in comparison with ours is the work of Scheidat et al. [17]. Such work suggests another good solution for parameter optimization in the biometrics area. They planned their application in some way that it can be used without great effort by other biometric systems, even firstly developed for fingerprint recognition. All databases from FVC projects of the years 2000, 2002 and 2004 were used to generate the results. In the best case, a relative improvement of 40% in EER was obtained for database 1 of FVC 2000.

3. Genetic Algorithms

Genetic algorithms were proposed by [8] as a tool to find solutions to optimization problems in poorly understood large spaces. They are based on the genetic processes of biological organisms, especially on the principle of natural selection by Charles Darwin [3]. Although, this slogan seems to be slightly tautological in the natural environment, where fitness is defined as the ability to survive, it makes good sense in the world of optimization problems where fitness of a string is given as the value of the function to be optimized at the argument encoded by the string.

Typically, a genetic algorithm works on a population of individuals. Each individual is represented by one chromosome formed by a set of genes representing the parameters to be optimized. Some operations are realized in order to produce new generations of individuals based on their capability to generate good results: crossover, selection and mutation.

The crossover is the key operator to generate new individuals in the population. Inspired by the example of nature, crossover is intended to join the genetic material of chromosomes with a high fitness in order to produce even better individuals.

The selection operator is intended to implement the idea of "survival of the fittest". It basically determines which of the

chromosomes in the current population is allowed to inherit their genetic material to the next generation.

The mutation operator should allow the GA to find solutions which contain genes values that are non-existent in the initial population. The parameter governing this operator is called mutation probability. Whereas the selection operator reduces the diversity in the population, the mutation operator increases it again. The higher the mutation probability, the smaller is the danger of premature convergence. A high mutation probability, however, transforms a GA into a pure random search algorithm, which is of course not the intention of this.

Let *P* be a random population of *N* chromossomes (x_1, x_2, ..., x_n) and *f(x)* a fitness function. The following pseudocode describes the steps of genetic algorithms.

1) Create a random population *P* of *N* chromosomes (candidate solutions for the problem).
2) Evaluate *f(x)* of each chromossome *x* in the population.
3) Generate a new population by repeating the following steps until the new population reaches population N:
   a) Select two parent chromosomes from the population, giving preference to highly fit chromosomes (high *f(x)* values). Automatically copy the fittest chromosome to the next generation.
   b) With a given crossover probability, crossover the parent chromosomes to form two new offspring. If no crossover was performed, offspring is an exact copy of parents.
   c) With a given mutation probability, randomly swap two genes in the offspring.
   d) Copy the new offspring into a new population.
4) Copy the newly generated population over the previous (existing) population.
5) If the loop termination condition is satisfied, then stop and return the best solution in current population. Otherwise, go to Step 2.

4. Proposed Technique

In this paper, we analyze the improvement in parameters optimization of SourceAFIS's fingerprint matching using genetic algorithms. The DB2 database of FVC2006 [2] project from Biolab is used to test and validate the proposed method. It has 1680 samples of 140 persons with 12 samples/person. The experiments were carried out on a computer with an I7 Intel Quad-Core processor(3.2 GHz) and 8GB of RAM.

4.1 Matching Algorithm

The matching algorithm used in this work is based on the algorithm proposed by Robert Vazan in the open source project SourceAFIS [18].

The matching algorithm of SourceAFIS also uses a minutiae-based method. However, the comparison is not directly based on coordinates, types, and direction. It is based on a set of vectors generated by the connection between minutiae and their neighbors. It ignores position differences and possible rotations caused by failures on capture.

Fig. 3: Example of k segments created for a minutia and its neighboors. The enumerated minutiae are divided by types, where blues are endings and reds are bifurcations.

Let (M_1, M_2, M_3, ..., M_n) be the set of *n* minutiae of template A. For each minutia, the algorithm will search the *k* nearest neighbors and make vectors to each of them, where k is a parameter of the system (see Figure 3). These vectors will, in general, represent a unique relation between two minutiae and a more reliable information than a simple minutiae location. It may decrease False Positive Rates.

Apart from the length and angle information of each vector, the following properties are also computed to enlarge the relationship between the vectors and its minutiae:

- *Reference Angle*: stores the angle difference between reference minutia, from where the vectors depart, and the vector angle.
- *NeighborAngle*: similarly to Reference Angle, this attribute stores the difference between the neighbor minutiae direction, to where the vector arrive, and the opposite of the vector angle.

These relations between minutae directions and vector angle are important to improve the reliability of matching. This algorithm doesn't store any information about the location of the minutiae. It happens because the algorithm ignores the position of founded pattern.

After all vectors constructions, all information about created segments are stored in a matrix $M[n, k]$. On the second step of the algorithm, each generated segment of a template B will search a correspondent segment on template A matrix. The comparison between two vectors is made by computing the subtraction of vectors length, References Angle, and Neighbors Angle. The segments are matched if these values are lower than a defined angle and size thresholds.

When a correspondent vector of template A is found on template B, the algorithm realizes that the minutiae from where the vectors depart, on each template, are corresponding (see Figure 4). Based on this premise, it's possible

Fig. 4: Example of corresponding minutiae and it's neighbors segments from different fingerprints.

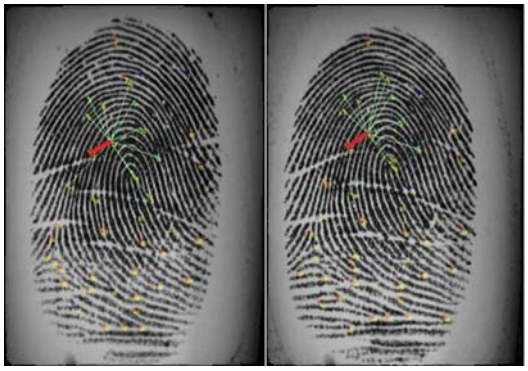


Fig. 5: Example of corresponding segment on a probably region intersection.

the minutiae in neighborhood are also able to find their corresponding because this region is candidate to be an intersection region - Figure 5 shows an example of region intersection. Thus, the paired minutiae of template B searches on its neighbors segments if any of them has correspondence with the neighbor segments of template A's paired minutiae.

When paired neighbors are found, the cycle of searching for neighbors restarts. Each newly paired minutiae searches for corresponding vectors on its neighbors. This recursive cycle continues until the minutiae graph reaches the intersection borders between the two samples, where there will be no corresponding minutiae in both templates. The graph tends to be significantly smaller in fingerprint pairs of different individuals than in pairs of the same individual.

After all steps, a list of paired minutiae between templates A and B is obtained. Figure 6 shows the best graph generated by matching between two templates. So many information can be inferred from this list, such as percentage of paired minutiae about the total of minutiae of each template; amount of equal types, since the algorithm does not consider the minutia types for graph construction; sum of the distance errors between all vectors, and many others. Such information may be used to establish a matching score for each primer pair.

4.2 Genetic Algorithm

The Optimera library [6] developed by Chas Egan was chosen as framework to apply the Genetic Algorithm in this work. It's written in C# and supports multithread as well.

In the Fingerprint Matching algorithm used in this work, graphs initialized by different minutiae can produce distinct scores of matching. In order to let Genetic Algorithm decide the weight of used parameters, all possible matchings scores - computed by all possible graphs - were computed. The purpose of the genetic algorithm, in our work, is to optimize the weight of these used parameters.

The parameters used on each graph evaluation were:

- *Pair Count*: Represents the quantity of matched minutiae found.
- *Correct Type*: Quantity of matched minutiae with correct type (Ending or Bifurcation).
- *Supported Count*: Quantity of segments that found the same minutia as part of matched segment.
- *Edge Count Factor*: Quantity of matched segments.
- *Distance Accuracy*: This variable represents the sum of distance errors between all matched vectors. This value contributes negativaly.
- *Angle Accuracy Factor*: Represents the sum of angle errors between all matched vectors. This value contributes negatively.

5. Results and Discussions

To evaluate the matching algorithm, the DB2 database of FVC2006 Project was used. The FVC protocol [2] was adopted to compare SourceAFIS and the proposed method. Hence, both methods were compared through EER performance although FAR1000, FAR and FRR perfomances are also compared.

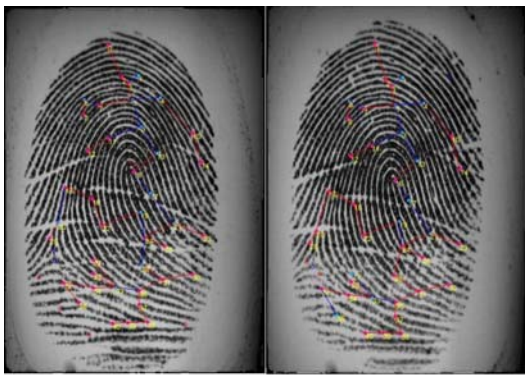


Fig. 6: Example of completed matching graph of corresponding segments.

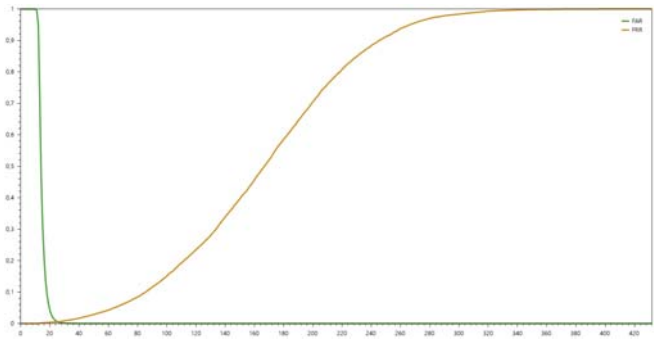


Fig. 7: FAR and FRR curves of SourceAFIS method.

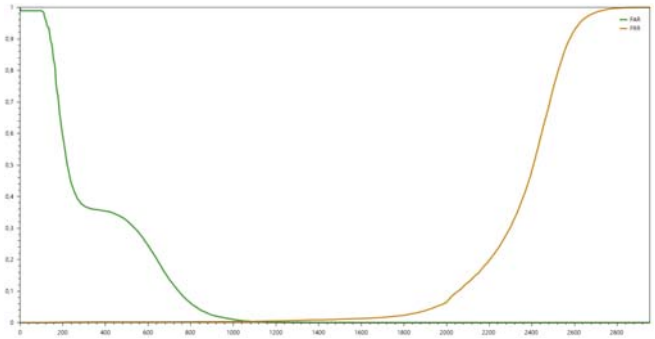


Fig. 8: FAR and FRR curves of SourceAFIS method using genetic algorithms.

Figures 7 and 8 show FAR and FRR curves of SourceAFIS method and the proposed one. A significant improvement can be observed in the ROC curve, shown in Figure 9.

Table 1 compares both performance rates. A relative performance improvement of at least 32% was achieved in all performance rates.

The best results were achieved with the following parameters to genetic algorithm: Population Size p = 120, CrossOver Rate c=0.8 and Mutation Rate m= 0.05. The genetic algorithm usually took 2000 generations to converge to same results.

Table 1: Comparison of performance rates between SourceAFIS and the proposed method.

	SourceAFIS	*Proposed Method*	*Improvement*
EER	**0.5592%**	**0.3798%**	**32.08%**
FAR1000	**1.0173%**	**0.6602%**	**35.10%**
FAR	**0.5447%**	**0.3699%**	**32.09%**
FRR	**0.5736%**	**0.3896%**	**32.07%**

6. Conclusion

Improvement of EER performance plays an important role in automatic fingerprint authentication. In this paper, a genetic algorithm based approach was used to optimize

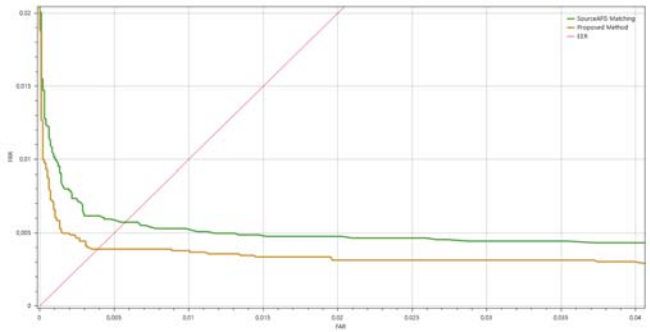


Fig. 9: Comparison of ROC curves. The green line represents SourceAFIS curve.

parameters of fingerprint matching algorithms. It has shown to be effective decreasing error rates of the basic algorithm, SourceAFIS, with a low cost process. An improvement of at least 32% was achieved in most used fingerprint authentication performance rates. The future research activity will be devoted to further improve the SourceAFIS algorithm, decreasing the EER performance and submit our algorithm to validation in FVC project.

References

[1] Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1):1–23, 1993.

[2] Raffaele Cappelli, Matteo Ferrara, Annalisa Franco, and Davide Maltoni. Fingerprint verification competition 2006. *Biometric Technology Today*, 15(7-8):7-9, 2007.

[3] Charles Darwin and Gillian Beer. *The origin of species*. Oxford University Press Oxford, 1951.

[4] Herbert Dawid. Genetic algorithms. In *Adaptive Learning by Genetic Algorithms*, volume 441 of *Lecture Notes in Economics and Mathematical Systems*, pages 37–60. Springer Berlin Heidelberg, 1996.

[5] Bernadette Dorizzi, Raffaele Cappelli, Matteo Ferrara, Dario Maio, Davide Maltoni, Nesma Houmani, Sonia Garcia-Salicetti, and Aurélien Mayoue. Fingerprint and on-line signature verification competitions at ICB 2009. In *Advances in Biometrics, Third International Conference, ICB 2009, Alghero, Italy, June 2-5, 2009. Proceedings*, pages 725–732, 2009.

[6] Chas Egan. Optimera - a multithreaded genetic algorithm library in $C\#$, April 2013.

[7] N Goranin and A Cenys. Evolutionary algorithms application analysis in biometric systems. *Journal of Engineering Science and Technology Review*, 3(1):70–79, 2010.

[8] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975. second edition, 1992.

[9] John Holland. Outline for a logical theory of adaptive systems. *Journal of the Association of Computing Machinery*, 3, 1962.

[10] Xudong Jiang and Wei-Yun Yau. Fingerprint minutiae matching based on the local and global structures. In *Pattern recognition, 2000. Proceedings. 15th international conference on*, volume 2, pages 1038–1041. IEEE, 2000.

[11] Zsolt Miklos Kovacs-Vajna. A fingerprint verification system based on triangular matching and dynamic time warping. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1266–1276, 2000.

[12] Eryun Liu, Heng Zhao, Fangfei Guo, Jimin Liang, and Jie Tian. Fingerprint segmentation based on an adaboost classifier. *Frontiers of Computer Science in China*, 5(2):148–157, 2011.

[13] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar. *Handbook of Fingerprint Recognition*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

[14] Melanie Mitchell. Genetic algorithms: An overview. *Complexity*, 1(1):31–39, 1995.

[15] Fernández J.A. Prieto and Velasco J. R. Pérez. Adaptive genetic algorithm control parameter optimization to verify the network protocol performance. In *Proceedings of IPMU*, volume 08, pages 785–791, 2008.

[16] Arun A. Ross, Karthik Nandakumar, and Anil K. Jain. *Handbook of Multibiometrics (International Series on Biometrics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[17] Tobias Scheidat, Andreas Engel, and Claus Vielhauer. Parameter optimization for biometric fingerprint recognition using genetic algorithms. In *Proceedings of the 8th Workshop on Multimedia and Security*, pages 130–134. ACM, 2006.

[18] Robert Važan. Source AFIS project @ONLINE, June 2009.

[19] En Zhu, Jianping Yin, Chunfeng Hu, and Guomin Zhang. A systematic method for fingerprint ridge orientation estimation and image segmentation. *Pattern Recogn.*, 39(8):1452–1472, August 2006.

TOPSIS-Based Multi-Objective ABC Algorithm to Attune Plan of PV/Wind Generation System

H. Shayeghi[1], Y. Hashemi[1], H. A. Shayanfar[*2], S. Kardar[3]
[1]Technical Engineering Department University of Mohaghegh Ardabili, Ardabil, Iran
[2]College of Electrical Engineering Center of Excellence for Power System Automation and Operation, Iran University of Science and Technology, Tehran, Iran
[3]College of Environmental Engineering, Science and Research Branch, Islamic Azad University Tehran, Iran

Abstract – *A hybrid wind/photovoltaic system based on storage device is designed to supply demand of a building in this paper. The aim of this plant design is minimization of the annualized cost of the hybrid renewable generation system and environmental cost. The design problem is formulated as a multi-objective optimization issue and solved by a Multi-objective Artificial Bee Colony (MABC) technique. Solar radiation, wind speed and load data are assumed which are entirely deterministic. The used prices for all devices are practical and assumed that all components are commercially available. The considered test system is in the northwest region of Iran. The presented method intends to offer the optimal number of hybrid renewable system that the economic and environmental profits achieved during the operational lifetime period of the system are maximized. A decision-making methodology based on Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is applied for finding the best compromise solution from the set of Pareto-optimal solutions obtained by MABC algorithm.*

Keywords: Multi-objective optimization, Decision making, Photovoltaic, Wind turbine.

1 Introduction

Wind and solar power can be considered as viable options for future electricity generation. Besides being emission-free, the energy coming from the wind and the sunrays is available at no cost. In addition, they offer a solution for power supply to remote areas that are not accessible by the Utility Company, and to developing countries that are poor in fossil-based resources. Some literatures have proposed methods to require the optimal combination of hybrid renewable energies as configured in Fig. 1 [1]. Homer software has been applied to construct the PV/micro hydro electric hybrid system in north of Africa [2]. Several possible combinations of PV/wind generation capacities were established. The total annual cost for each combination with the lowest cost is selected to present the optimal mixture. The additional cost of this approach because of inappropriate combination is the important drawback related to it. Homer software utilized to minimize the cost of PV/wind/micro hydro electric/diesel hybrid system in Malaysia [3]. Use of diesel generator has environmental problems. To obtain the optimal size of PV/wind integrated hybrid energy system, a evolutionary algorithm has been applied in [4]. A technique to design hybrid system has been presented in [5] by Koutroulis et al. A method to size a stand-alone hybrid wind/PV/diesel energy system has been offered in [6]. The proposed method use a deterministic algorithm to specify the optimal number and type of units while handle total cost minimization and energy availability. Heuristic method based on the evolutionary algorithm has been applied by Ekren et al. [7] for optimizing size of a PV/wind integrated hybrid energy system with battery storage. The proposed methodology uses a stochastic gradient search for the global optimization. In the study, the objective function is the minimization of the hybrid energy system total cost.

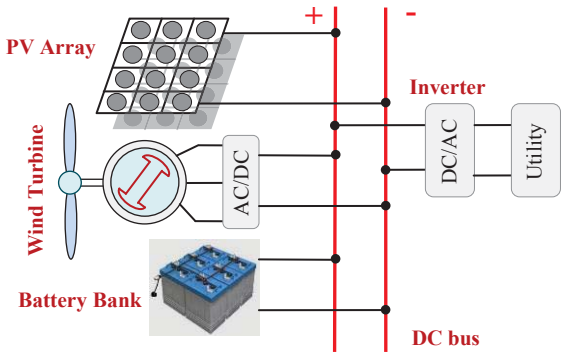


Fig. 1: Configuration of the PV/Wind generation system

In this paper, the proposed optimization procedure is based on a dynamic evaluation of the wind and solar energetic potential corresponding to a long term reference. This dynamic evaluation of the energetic potential of the site permits the introduction of new constraints making the optimization procedure more flexible like the maximum acceptable time of energy unavailability and the minimum power level authorized regarding the power demand. Consequently, this approach

* Corresponding author, E-mail address: hashayanfar@yahoo.com

results in a more realistic optimization of the size of hybrid wind/PV power generation system. A methodology for the design optimization of PV/wind hybrid system is proposed to supply the demand of a building. Multi-objective approach is employed to find optimal size of the hybrid PV/wind system by considering two objectives function including economic, and environmental profit obtained by the hybrid system. Multi-objective optimization design problrm is solved by a multi-objective artificial bee colony (MABC) algorithm. A decision-making procedure based on TOPSIS method is applied to rank the Pareto-optimal solutions from the best to worst and to determine the best optimal solution.

2 MABC Concept

Multi-objective minimization: A multi-objective problem (MOP) involves the simultaneous satisfaction of two or more objective functions. Furthermore, in such problems, the objectives to be optimized are usually in conflict, which means that they do not have a single best solution, but a set of solutions [8]. To find this set of solutions, it is used the Pareto optimality theory [9]. The general multi-objective minimization problem, without constraints, can be stated as (1):

$$Minimize\ f(x)=(f_1(x), f_2(x),\cdots, f_m(x)) \quad (1)$$

Subject to $x\in\Omega$

Where, $x\in\Omega$ is a feasible solution vector, Ω is the feasible region of the problem, m is the number of objectives and $f_i(x)$ is the ith objective function of the problem.

In this case, the purpose is to optimize m objective functions simultaneously, with the goal to find a good trade-off of solutions that represent the better compromise between the objectives. So, given $f(x)=(f_1(x), f_2(x),\cdots, f_m(x))$ and $f(y)=(f_1(y), f_2(y),\cdots, f_m(y))$, $f(x)$ dominates $f(y)$, denoted by $f(x)\prec f(y)$, if and only if (minimization):

$$\begin{aligned}&\forall i\in\{1,2,\ldots,m\}: f_i(x)\le f_i(y), \text{ and}\\ &\exists i\in\{1,2,\ldots,m\}: f_i(x)< f_i(y)\end{aligned} \quad (2)$$

$f(x)$ is non-dominated if there is no $f(y)$ that dominates $f(x)$. Also if there is no solution y that dominates x, x is called Pareto-optimal and $f(x)$ is a non-dominated objective vector. The set of all Pareto-optimal solutions is called Pareto optimal set, denoted by P^*, and the set of all non-dominated objective vector is called Pareto Front, denoted by PF^*.

Artificial Bee Colony Optimization Concept: In the ABC algorithm, each employed bee, at each iteration of the algorithm determines a new neighboring food source of its currently associated food source by Eq. (3), and computes the nectar amount of this new food source [10]:

$$v_{ij}=z_{ij}+\theta_{ij}(z_{ij}-z_{kj}) \quad (3)$$

Where, $k\in\{1,2,\ldots,BN\}$ and $j\in\{1,2,\ldots,D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. θ_{ij} is a random number between [−1, 1]. It controls the production of a neighbor food source position around z_{ij} and the modification represents the comparison of the neighbor food positions visually by the bee. Equation (3) shows that as the difference between the parameters of the z_{ij} and z_{kj} decreases, the perturbation on the position z_{ij} decreases, too. If the nectar amount of this new food source is higher than that of its currently associated food source, then this employed bee moves to this new food source, otherwise it continues with the old one. After all employed bees complete the search process; they share the information about their food sources with onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount by Eq. (4). This method, known as roulette wheel selection method, provides better candidates to have a greater chance of being selected:

$$p_i = fit_i \Big/ \sum_{n=1}^{SN} fit_i \quad (4)$$

Where, fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees. The flowchart of considered algorithm has been given in Fig. 2.

3 TOPSIS Method

When solutions based on the estimated Pareto-optimal set are found, it is required to choose one of them for implementation [11]. The theory of TOPSIS is used for finding the best compromise solution in this paper. The normalized decision matrix can be written as [12, 13]:

$$NDM_{ij} = DM_{ij} \Big/ \sum_{p=1}^{n} DM_{pj} \quad (5)$$

Where, $DM=\{DM_{ij}\,|\,i=1,2,\ldots,n; j=1,2,\ldots,m\}$ (n, m are the number of Pareto-optimal solutions and number of objectives respectively) is the $n\times m$ decision matrix and DM_{ij} is the performance rating of alternative X_j (Pareto-optimal solution) with respect to attribute AT_i. The amount of decision information can be measured by the entropy value as:

$$EV_j = \frac{-1}{\ln n}\sum_{i=1}^{n} NDM_{ij}\ln(NDM_{ij}) \quad (6)$$

The degree of divergence (D_j) of the average intrinsic information contained by each attribute AT_j (j=1, 2, ..., m) can be calculated as:

$$D_j = 1 - EV_j \quad (7)$$

and objective weighted normalized value OW_{ij} is calculated as:

$$OW_{ij} = w_i \times NDM_{ij} \quad (8)$$

To produce an overall performance index for each alternative the positive ideal solution (AT^+) and the negative ideal solution (AT^-) are used, which are defined, respectively, by:

$$AT^+ = (\max(OW_{i1}) \ \max(OW_{i2}) \ \dots \ \max(OW_{im})) = (OW_1^+, OW_2^+, \dots, OW_m^+) \quad (9)$$
$$AT^- = (\min(OW_{i1}) \ \min(OW_{i2}) \ \dots \ \min(OW_{im})) = (OW_1^-, OW_2^-, \dots, OW_m^-)$$

Separation (distance) between alternatives can be measured by the *n*-dimensional Euclidean distance. The separation of each alternative from the ideal solution is given as:

$$D_j^+ = \left\{ \sum_{i=1}^{m} \left(OW_{ji} - OW_i^+ \right)^2 \right\}^{1/2}, \quad j = 1, 2, \dots, n \quad (10)$$

Similary, the separation from the negative ideal solution is given as:

$$D_j^- = \left\{ \sum_{i=1}^{m} \left(OW_{ji} - OW_i^- \right)^2 \right\}^{1/2}, \quad j = 1, 2, \dots, n \quad (11)$$

The relative closeness to the ideal solution of alternative X_j with respect to AT^+ is defined as:

$$RC_j = \frac{D_j^-}{D_j^+ + D_j^-}, \quad j = 1, 2, \dots, n \quad (12)$$

Since $D_j^- \geq 0$ and $D_j^+ \geq 0$, then, clearly, $RC_j \in [0,1]$.

Choose an alternative with maximum RC_j, in descending order. It is clear that an alternatives X_j is closer to AT^+ than to AT^- as RC_j approaches 1.

4 Model of PV and Wind turbine Generation System

Model of PV generation system:

The output voltage and current obtained by PV module are described in the following equation [14, 15]:

$$I_{mpp} = I_{SC} \{ \frac{G_T}{G_{ref}} - \lambda_1 [\exp(\frac{V_{\max}}{\lambda_2 V_{OC}}) - 1] \} + \sigma_{I,SC} (T_c - T_{c,ref}) \quad (13)$$

$$V_{mpp} = V_{\max} + \sigma_{V,OC} (T_c - T_{c,ref}) \quad (14)$$

$$\begin{cases} \lambda_1 = (1 - \frac{I_{\max}}{I_{SC}}) \exp(-\frac{V_{\max}}{\lambda_2 V_{OC}}), \lambda_2 = (\frac{V_{\max}}{V_{OC}} - 1)[\ln(1 - \frac{I_{\max}}{I_{SC}})]^{-1} \\ T_c = T_a + \frac{NOCT - 20}{800} G_T \end{cases}$$

Thus, PV panel power at the maximum power point can be written as:

$$P_{mpp} = V_{mpp} I_{mpp} \quad (15)$$

The total number of PV module is defined as:

$$N_{PV} = N_{PV}^s N_{PV}^p \quad (16)$$

The number of series module N_{PV}^s is specified by the chosen DC bus voltage V_{bus}:

$$N_{PV}^s = V_{bus} / V_{PV}^{nom} \quad (17)$$

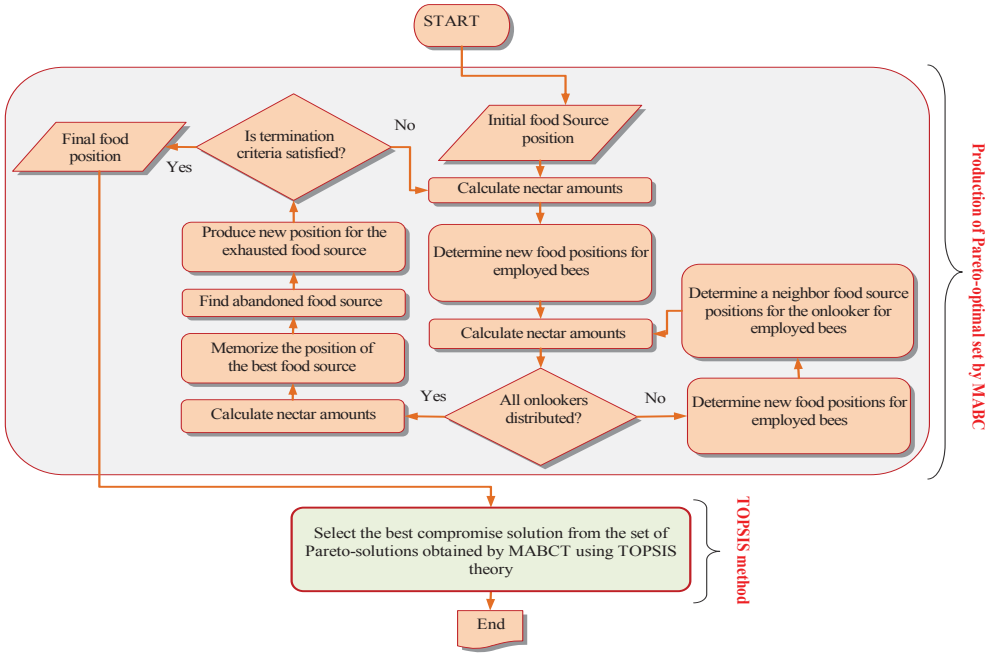


Fig. 2: Flowchart of the TOPSIS-based MABCT

Model of wind turbine generator: According to Fig. 3, the output power of wind turbine generator is a function of wind speed. The mathematical model of wind turbine generator can be written as the following equation [16]:

$$P_{WTG}=\begin{cases}\dfrac{P_R v^3}{v_r^3-v_{ci}^3}-\dfrac{P_R v_{ci}^3}{v_r^3-v_{ci}^3} & , \quad v_{ci}<v<v_r \\ P_R & , \quad v_r<v<v_{co} \\ 0 & , \quad otherwise\end{cases} \tag{18}$$

To convert the wind speed at reference height H_r into wind speed at hub height H_b, the following function is used:

$$v=v_{Hr}\left(\frac{H_b}{H_r}\right)^{\kappa} \tag{19}$$

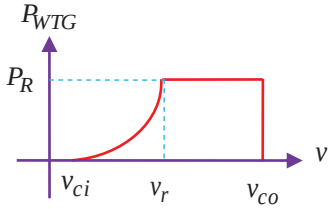


Fig. 3: Power-speed curve of wind turbine generator

Model of battery bank:

The state of charge of a battery bank can be written as [6]:

$$SOC(t+\Delta t)=SOC(t)+\eta_{bat}(P_B(t)/V_{bus})\Delta t \tag{20}$$

Maximum charging rate ($SOC_{\max}$) is equal to the total nominal capacity of the battery bank ($C_n(Ah)$). The state of charge of battery bank should be confined to minimum and maximum values of SOC as follows [17]:

$$SOC_{\min}=(1-DOD)SOC_{\max}<SOC<SOC_{\max} \tag{21}$$

The number of series batteries depends on the DC bus voltage and the nominal voltage of each battery is given by:

$$N_{battery}^{s}=\frac{V_{bus}}{V_{bus}^{nom}} \tag{22}$$

5 Problem Formulation

The annualized cost of each component i is described as [18]:

$$AP_i=N_i\{[CC_i+RC_i\,k_i(ir,L_i,y_i)]\times CRF(ir,R)+O\&MC_i\} \tag{23}$$

Where, N_i is number of the ith component, CC is capital cost (U$/unit), RC is cost of each replacement ($/unit), O&MC is annual operation and maintenance cost (U$/unit-year) of the component, R is lifetime of the project, and ir is the real interest rate which depends on nominal interest rate ($ir_{nominal}$) and annual inflation rate (f_r) and is defined as follows [19]:

$$ir=\frac{ir_{nominal}-f_r}{1+f_r} \tag{24}$$

CRF and K are capital recovery factor and single payment present worth, respectively, and are formulated as:

$$CRF=\frac{ir\times(1+ir)^R}{(1+ir)^R-1} \tag{25}$$

$$K_i(ir,L_i,y_i)=\sum_{n=1}^{y_i}\frac{1}{(1+ir)^{n\times L_i}} \tag{26}$$

L and y are lifetime and number of replacements of the component during lifetime of the project, respectively. Finally, the cost function of planning process is formulated as follows:

$$F_1=\sum_i AP_i \tag{27}$$

The second objective function is the maximization of total environmental profit function, f_2 (kg CO_2). This function defines the total CO_2 emissions, which is avoided due to the use of hybrid renewable energy system and given by:

$$F_2=E_{conv}-E_{RES}-E_{inst} \tag{28}$$

The CO_2 emission created by the conventional energy system is given as:

$$E_{conv}=E_{tot}\cdot f_{conv} \tag{29}$$

f_{conv} (kg CO_2 per kWh) is a parameter that describes the CO_2 emission created by the conventional energy source system.

The CO_2 emission created by renewable energies is verified by:

$$E_{RES}=E_{total}\cdot f_{RES} \tag{30}$$

f_{RES} (kg CO_2 per kWh) is a parameter that describes the CO_2 emissions created by a hybrid renewable energy system per energy unit. This parameter is considered as 0.098 kg/kWh.

Also, the CO_2 emission created during the production and installation of renewable energy system is defined by:

$$E_{inst}=P_{RES}\cdot f_{pro} \tag{31}$$

f_{pro} (kg CO_2 per kW) is a parameter that describes the CO_2 emission created during the production and installation of renewable energy system and P_{RES} is the installed power of renewable energy system.

In general and after aggregating the objectives and constraints, the problem can be mathematically formulated as follows:

$$Min\,F(x,u,p)=Min\{F_1,F_2\} \tag{32}$$

6 System Design

The method presented in this paper has been applied for the optimal design of PV/wind generation system for power supply in a building located in Iran, with geographical coordinates of latitude: 38.24 and longitude: 48.29. The collection of 12 months of wind speed, solar irradiation, and ambient temperature data recorded for every day in 2013 have been used and plotted in Figs. 4-6 [20]. The wind speed has been extracted at 10 m height which is considered as the reference height for the site. The daily distribution of the consumer power requirement during 2013 for building is

shown in Fig. 7. The specifications and related costs of each component type are listed in Table 2.

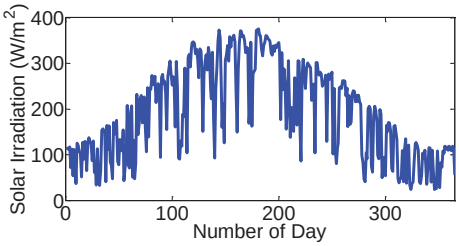


Fig. 4: Hourly mean values of solar radiation during a period of one year

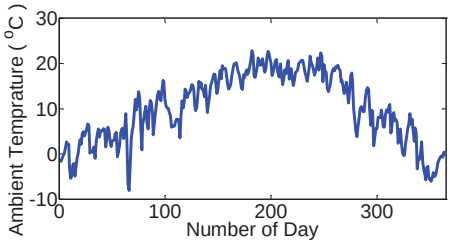


Fig. 5: Hourly mean values of ambient temperature during a period of one year

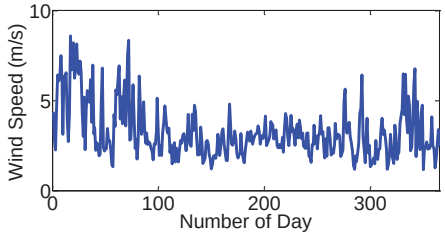


Fig. 6: Hourly mean values of wind speed during a period of one year

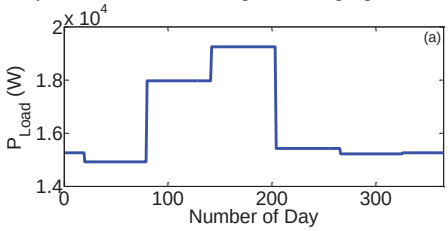


Fig. 7: Hourly demanded power per year

6.1. Hybrid system sizing results

The MABC algorithm is applied for the simultaneous optimization of the two-objective functions, describing the economic and environmental benefits of the PV/wind hybrid system. The Pareto-front curves calculated by the application of the MABC is shown in Fig. 8.

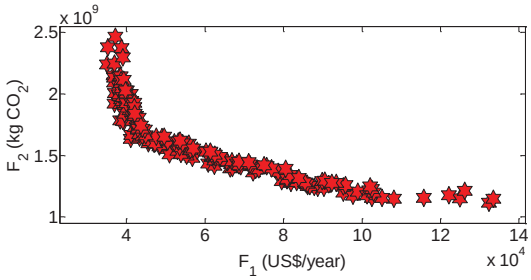
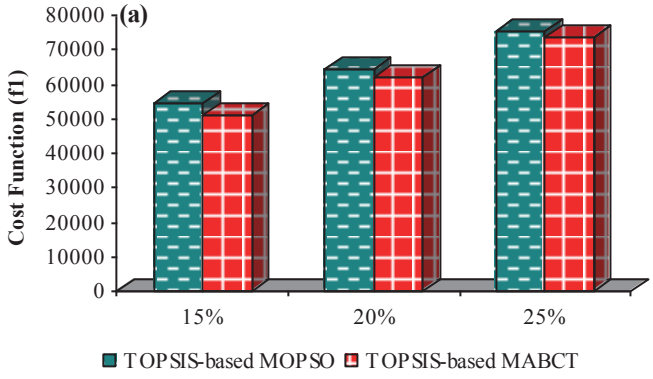


Fig. 8: Pareto-optimal archive with MABCT in two-dimensional objective space

The optimal sizing results according to the objective functions of the environmental and financial benefits in different cases have been tabulated in Table 1. A decision-making technique based on TOPSIS theory is used to find the best compromise solution from the set of Pareto-solutions obtained by MABC technique. The obtained results of TOPSIS approach are shown in Fig. 9. It can be seen that, the best result has been achieved by employing MABC algorithm between two different solution algorithms.

Table 1: Value of functions obtained by Pareto-archive in different cases

	f_1	f_2
C_1: min f_1	34986	2.2405×10^9
C_2	38351	1.9169×10^9
C_3	46982	1.5939×10^9
C_4	75699	1.4128×10^9
C_5	83868	1.3156×10^9
C_6	91583	1.2673×10^9
C_7: min f_2	132360	1.1147×10^9



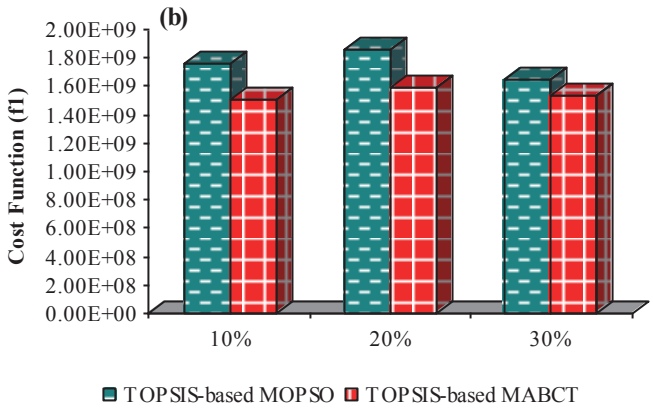


Fig. 9: Value of objective functions by applying TOPSIS with different solution methods in three inflation rates, 15%, 20% and 25%, (a) F_1 (b) F_2

The optimal sizing results of the hybrid systems in different cases, C_1 to C_7, and different solution methods are presented in Tables 2 and 3, where the different types of devices and their numbers are listed.

Table 2: Optimal configuration of wind/PV system in different cases

	Number of panels	Number of batteries	Number of wind turbines	Number of charge controllers	Number of inverters
C_1: min f_1	34	11	9	14	6
C_2	27	17	9	12	5
C_3	25	29	7	10	4
C_4	24	60	5	9	3
C_5	18	68	7	9	3
C_6	17	76	6	8	3
C_7: min f_2	16	117	5	7	3

Table 3: Optimal configuration of wind/PV system by applying TOPSIS technique for different solution methods

Number of devices	Panel	Battery	Wind turbine	Charge controller	Inverter
MOPSO	28	43	7	11	5
MABCT	23	36	7	10	4

The results of a simulation conducted on a period of one year are performed for case 1 or C_1 as the sample. The power supply from the renewable sources P_{RE}, power produced by the hybrid system P_p, input/output battery bank system P_b, and variations of SOC have been reported in Figs. 10.

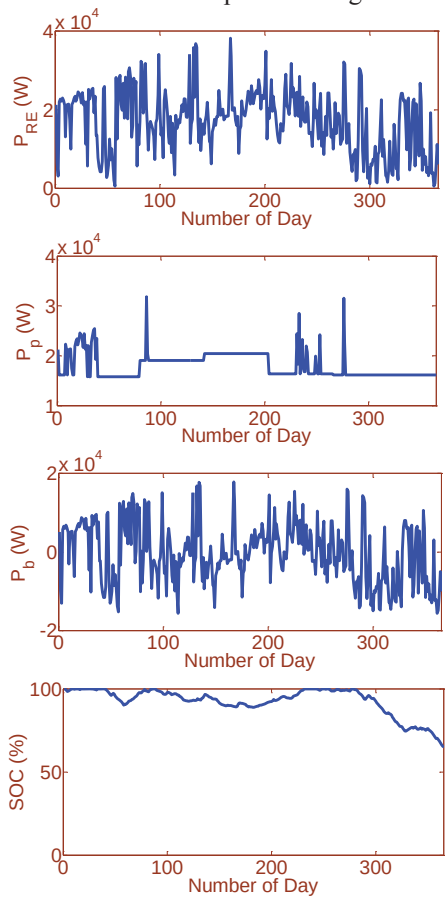


Fig. 10: Curve of power supply from the renewable sources P_{RE}, power produced by the hybrid system P_p, input/output battery bank system P_b, and variations of SOC

According to Fig. 10, it can be verified that the power produced by the hybrid system covers completely the considered loads. It can be observed that, when the renewable source power is greater than the power demand, the surplus power is stored in the battery bank. When the renewable source power is smaller than the power demand, the insufficient power is supplied by the battery bank. When the power generated by the renewable sources is greater than the demand power and the battery bank is completely charged, the surplus of energy can be used for water pumping or other actions. According to the SOC curve, it can be observed that the charge state of the battery bank can never exceed the permissible maximum value SOC_{max} and can never be below the permissible minimum value SOC_{min}.

7 Conclusions

In this paper, an optimization model is presented for designing a hybrid wind/PV system for a building implemented in the north part of Iran. The optimization solution is provided by MABC algorithm. The main purpose of combination PV and wind unit is to reach a reliable applying with minimum initial and operation cost. The developed methodology is based on the use of long-term data of wind speed, solar irradiation, and ambient temperature of the considered site. Based on two objectives of economic and environmental indices, a multi-objective optimization process is systematized and MABC technique is applied to solve the problem. TOPSIS method is used to determine the solution considering all the relevant attributes from the set of Pareto-solutions. A set of tradeoff solutions is obtained using the multi-index met-heuristic method, which suggests many design alternatives to the decision-maker. The results show that the optimized configuration produces has high efficiency. Implementation of this energy system will supply the demand of the areas while having no emissions and reducing the environmental pollutions.

8 References

[1] L. Wang and C. Singh, "Multicriteria design of hybrid power generation systems based on a modified particle swarm optimization algorithm," *IEEE Trans. Energy Convers*, vol. 24, pp. 163-172, 2009.

[2] G. Bekele and B. Palm, "Wind energy potential assessment at four typical locations in Ethiopia," *Appl Energ*, vol. 86, pp. 388-396, 2009.

[3] R. Luna-Rubio, *et al.*, "Optimal sizing of renewable hybrids energy systems: A review of methodologies," *Sol Energy*, vol. 86, pp. 1077-1088, 2012.

[4] E. Koutroulis, *et al.*, "Methodology for optimal sizing of stand-alone photovoltaic/wind-generator systems using genetic algorithms," *Sol Energy*, vol. 80, pp. 1072-1088, 2006.

[5] H. Yang, *et al.*, "Optimal sizing method for stand-alone hybrid solar–wind system with LPSP technology by using genetic algorithm," *Sol Energy*, vol. 82, pp. 354-367, 2008.

[6] R. Belfkira, *et al.*, "Optimal sizing study of hybrid wind/PV/diesel power generation unit," *Sol Energy*, vol. 85, pp. 100-110, 2011.

[7] J. L. Bernal-Agustín, *et al.*, "Design of isolated hybrid systems minimizing costs and pollutant emissions," *Renew Energ*, vol. 31, pp. 2227-2244, 2006.

[8] F. Gunes and U. Ozkaya, "Multiobjective FET modeling using particle swarm optimization based on scattering parameters with Pareto-optimal analysis," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 20, pp. 353-365, 2012.
[9] C. A. C. Coello, *et al.*, *Evolutionary algorithms for solving multi-objective problems*. New York: Springer, 2007.
[10] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of global optimization,* vol. 39, pp. 459-471, 2007.
[11] L. Xuebin, "Study of multi-objective optimization and multi-attribute decision-making for dynamic economic emission dispatch," *Electric Power Components and Systems,* vol. 37, pp. 1133-1148, 2009.
[12] P. Schubert and W. Dettling, "Extended web assessment method (EWAM) evaluation of e-commerce applications from the customer's viewpoint," in *35th Annual Hawaii International Conference on System Sciences (HICSS)*, 2002.
[13] S. Ramesh, *et al.*, "Application of modified NSGA-II algorithm to multi-objective reactive power planning," *Applied Soft Computing,* vol. 12, pp. 741-753, 2012.
[14] M. Alsayed, *et al.*, "Design of hybrid power generation systems based on multi criteria decision analysis," *Sol Energy,* vol. 105, pp. 548-560, 2014.
[15] A. Kornelakis and E. Koutroulis, "Methodology for the design optimisation and the economic analysis of grid-connected photovoltaic systems," *IET Renew Power Gen*, vol. 3, pp. 476-492, 2009.
[16] A. Chauhan and R. Saini, "A review on integrated renewable energy system based power generation for stand-alone applications: configurations, storage options, sizing methodologies and control," *Renew Sust Energ Rev*, vol. 38, pp. 99-120, 2014.
[17] M. Bortolini, *et al.*, "Technical and economic design of photovoltaic and battery energy storage system," *Energ Convers Manage,* vol. 86, pp. 81-92, 2014.
[18] A. Maheri, "Multi-objective design optimisation of standalone hybrid wind-PV-diesel systems under uncertainties," *Renew Energ,* vol. 66, pp. 650-661, 2014.
[19] A. Kashefi Kaviani, *et al.*, "Optimal design of a reliable hydrogen-based stand-alone wind/PV generating system, considering component outages," *Renew Energ*, vol. 34, pp. 2380-2390, 2009.
[20] S. N. Singh and J. Ostergaard, "Use of demand response in electricity markets: An overview and key issues," in *Energy Market (EEM), 7th International Conference on the European*, 2010, pp. 1-6.

Biographies

Hossein Shayeghi received the B.S. and M.S.E. degrees in Electrical and Control Engineering in 1996 and 1998, respectively. He received his Ph.D. degree in Electrical Engineering from Iran University of Science and Technology, Tehran, Iran in 2006. Currently, he is a full Professor in Technical Engineering Department of University of Mohaghegh Ardabili, Ardabil, Iran. His research interests are in the application of robust control, artificial intelligence and heuristic optimization methods to power system control design, operation and planning and power system restructuring. He has authored and co-authored of six books in Electrical Engineering area all in Farsi, one book and two book chapters in international publishers and more than 300 papers in international journals and conference proceedings. Also, he collaborates with several international journals as reviewer boards and works as editorial committee of eight international journals. He has served on several other committees and panels in governmental, industrial, and technical conferences. He was selected as distinguished researcher of the University of Mohaghegh Ardabili several times. In 2007, 2010, 2011 and 2013 he was also elected as distinguished researcher in engineering field in Ardabil province of Iran. Also, he is a member of Iranian Association of Electrical and Electronic Engineers (IAEEE) and IEEE.

Yashar Hashemi received the B.Sc. and M.S.E. degrees in Electrical Engineering in 2009 and 2011 respectively. Currently, he is Ph.D. student in Electrical Engineering Department, University of Mohaghegh Ardabili, Ardabil, Iran. His research interests include Power Systems Analysis, Wide Area Measurement and Control, Planning and Control of Renewable Energies, Dynamic Stability, Operation and Planning, Power System Restructuring and FACTS Devices Applications in Power System.

Heidarali Shayanfar received the B.S. and M.S.E. degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph. D. degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, he is a Full Professor in Electrical Engineering Department of Iran University of Science and Technology, Tehran, Iran. His research interests are in the Application of Artificial Intelligence to Power System Control Design, Dynamic Load Modeling, Power System Observability Studies, Voltage Collapse, Congestion Management in a Restructured Power System, Reliability Improvement in Distribution Systems, Smart Grids and Reactive Pricing in Deregulated Power Systems. He has published more than 497 technical papers in the International Journals and Conferences proceedings. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.

Saeed Kardar received the B.S. and M.S.E. degrees in Civil Engineering in 1998 and 2001, respectively. He received his Ph. D. degree in Environmental Engineering from Science and Research Branch, Islamic Azad University, Iran, in 2007. Currently, he is an Assistant Professor in Environmental Engineering College of Science and Research Branch, Islamic Azad University, Tehran, Iran.

A Complete Solution to the Set Covering Problem

Qi Yang, Adam Nofsinger, Jamie McPeek, Joel Phinney, Ryan Knuesel
Department of Computer Science and Software Engineering
University of Wisconsin - Platteville
Platteville, WI 53818, USA

Abstract: The set-covering problem is a classical problem in computational complexity theory. It has been proved to be NP hard and different heuristic algorithms have been presented to solve the problem. We have developed a new algorithm and optimized the input process. We will provide analysis and show that the complexity of our algorithm is better than that of earlier solutions in most cases. Our experiments show that our new solution performs significantly better than earlier solutions and provides a complete and practical solution to the set-covering problem.

Keywords: NP-Hard; Greedy algorithm; Set covering problem; Linked List; Binary Search Trees

1 INTRODUCTION

The set-covering problem is a classical problem in computational complexity theory [3], [4], [5] and also one of Karp's 21 NP-complete problems shown to be NP-complete [1]. Given N sets, let X be the union of all the sets. An element is covered by a set if the element is in the set. A cover of X is a group of sets from the N sets such that every element of X is covered by at least one set in the group. The set-covering problem is to find a cover of X of the minimum size. A proof that the problem is NP hard can be found in [4].

In [6], the set-covering problem is found to be equivalent to the problem of identifying redundant search engines on the Web, and finding an effective and efficient practical algorithm to the problem becomes a key issue in building a very large-scale Web meta-search engine. Another application of the set-covering problem is the job-scheduling problem: There are a list of jobs and a list of workers. For each job, multiple workers have the needed skills and can to do the job, and each worker has the needed skills to do different jobs. The problem is to find the minimum number workers needed to perform all the jobs. More applications of the set-covering problem can be found in [2].

Since the problem is NP hard, only approximation solutions are practical. In [3], a greedy algorithm is presented with a time complexity of O (M * N * min (M, N)), where N is the number of sets and M is the number of all elements of the union X. In [6], an algorithm called Check-And-Remove (CAR) is proposed and its time complexity is O (N * M). Experimental results show that the Greedy algorithm is not efficient and runs very slow. In some cases, it takes thousands of seconds while the CAR algorithm runs in 20 seconds or less. On the other hand, the CAR algorithm is not as effective as the Greedy algorithm and produces larger cover sets than the Greedy algorithm in quite some cases.

We have designed a different greedy algorithm List and Remove (LAR) [7], which optimizes the major phase of the Greedy algorithm based on a different data structure and also adapts the advantage of the CAR algorithm. Our experimental results show that our LAR algorithm is both efficient and effective. We have also studied the input phase and proposed a hybrid approach to improve the input process [8].

In this paper, we provide further analysis and show that the time complexity of our LAR algorithm is O ($N^2 + \sum |S_i|$, i = 1 to N), where N is the number of sets. This paper includes all the results to provide a complete efficient and effective solution to the set-covering problem.

2 MATRIX AND BST

The set-covering problem can best represented by a matrix. The matrix in Table I represents the case with five sets and six elements. Each row of the matrix represents a set and each column represents an element from the union of all the sets. A 1 in a cell of the matrix indicates the element of the column is contained in the set of the row, and a zero indicates otherwise. In this case, S1 = {b, e}, S2 = {d, f}, S3 = {a, b}, S4 = {a, b, c, d} and S5 = {d, e, f}.

TABLE I. MATRIX REPRESENTATION

	a	b	c	d	e	F
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1

TABLE II. MATRIX WITH CURRENT COVER

	a	b	c	D	e	F
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S1, S2}	0	1	0	1	1	1

We can add one more row at the bottom of the matrix for the cover to be generated to indicate which elements are covered by the current cover. In Table II, the added row shows that the covered elements are {b, d, e, f} when the current cover has two sets, S1 and S2. It is clear that the number of rows of the matrix (without the top row and the bottom row for the current cover) is N, the number of sets, and the number of columns of the matrix (without the left column for the sets) is M, the number of elements in the union of all the N sets. We assume that the value of N is known and the number of elements of each set is also known, but the value of M is unknown until all sets have been read in, since an element could be covered by multiple sets.

The implementation in [6] uses a Binary Search Tree (BST) to input data and sort all the elements at the same time. Each node of the tree stores one element with a bitmap to indicate which sets cover the element. We can see that the bitmap represents the corresponding column of the matrix. After the tree is built, it is converted to an array of bitmap and both Greedy and CAR algorithms work with the array. The value of M, the total number of elements of the union of all sets, is known at this stage.

3 Algorithm Greedy and Algorithm CAR

The two algorithms are presented in the following, where ResultCover is the cover to be generated and Covered is the set of elements that are covered by ResultCover. The value of M is known when the algorithms run.

3.1 Algorithm Greedy

Algorithm Greedy

1. Set both ResultCover and Covered to the empty set
2. While the size of Covered is less than M
   2.1 Select a set S that is not in ResultCover and contains the most elements not in Covered
   2.2 Add S to ResultCover
   2.3 Add the elements of S to Covered

Table III shows how algorithm Greedy works on the previous example. At beginning, both ResultCover and Covered are empty. Then, the best set S4 is added to ResultCover, since it covers four elements not in Covered, and each other set covers at most three elements. After S4 is added to ResultCover, Covered = {a, b, c, d}. In the next iteration of step 2.1, S5 is found to be the best set after all remaining sets are examined, since it contains two elements (e and f) not in Covered, and each other set contains at most one. So, S5 is added to ResultCover in the second iteration. After adding S5, Covered contains six elements, the same as M, and the algorithm terminates with ResultCover {S4, S5}.

The Greedy algorithm tries to find the best set (the one with the most uncovered elements) to add to the result cover. The algorithm should produce a better result (a smaller cover) but run slower, since it spends a lot of time in step 2.1 to find the best set: Each remaining set needs to be examined against Covered to calculate the number of elements not covered.

TABLE III. THE GREEDY ALGORITHM

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {}	0	0	0	0	0	0

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S4}	1	1	1	1	0	0

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S4, S5}	1	1	1	1	1	1

3.2 Algorithm CAR

Algorithm CAR (Check and Remove)

1. Set both ResultCover and Covered to the empty set
2. For each set S
   2.1 Determine if S has an element that is not in Covered
   2.2 Add S to ResultCover if S has such an element
   2.3 Add all elements of S to Covered
   3.4 Exit the for loop if Covered has M elements
3. For each set S in ResultCover
   3.1 Determine if S has an element that is not covered by any other set of ResultCover
   3.2 Remove S from ResultCover if S has no such elements

Table IV shows how algorithm CAR works on the same example. At beginning, both ResultCover and Covered are empty. Then, the first set S1 is added to ResultCover, since it contains two elements not in Covered. After S1 is added, Covered

= {b, e}. In the second iteration of step 2, S2 is added to ResultCover, since it contains element d, which is not in Covered, and Covered = {b, d, e, f}. In the next iteration, the third set S3 is added to ResultCover, since S3 contains one element (a) not in Covered, and Covered = {a, b, d, e, f}. The fourth set S4 is added in iteration 4, and Covered has six elements. Since ResultCover has M elements after S4 is added, step 2 is terminated with ResultCover {S1, S2, S3, S4}. In step 3, set S3 is found that all its elements are covered by other sets in ResultCover and hence removed from ResultCover. Then the algorithm terminates with ResultCover {S1, S2, S4}.

We can see that algorithm CAR takes the opposite approach compared to the Greedy algorithm: add any set to ResultCover as long as it has at least one uncovered element. The algorithm should run faster, but the produced cover may not be as good as that from the Greedy algorithm.

TABLE IV. THE CAR ALGORITHM

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {}	0	0	0	0	0	0

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S1}	0	1	0	0	1	0

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S1, S2}	0	1	0	1	1	1

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S1, S2, S3}	1	1	0	1	1	1

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S1, S2, S4}	1	1	1	1	1	1

	a	b	c	d	e	f
S1	0	1	0	0	1	0
S2	0	0	0	1	0	1
S3	1	1	0	0	0	0
S4	1	1	1	1	0	0
S5	0	0	0	1	1	1
ResultCover {S1, S2, S4}	1	1	1	1	1	1

Notice that set S3 is added to ResultCover in step 2 and then removed in step 3. That is why the algorithm has a remove phase. Similar situation could happen to the Greedy algorithm (not on this example), but the Greedy algorithm does not have such a remove phase.

3.3 Performance Comparison

The Greedy algorithm spends a lot of time in step 2.1 to find the best set to be added to ResultCover, while the CAR algorithm selects a set as long as it contains one uncovered element. It is expected that the Greedy algorithm would run much slower than the CAR algorithm, but would produce a better ResultCover with a smaller size. Experimental results in [6] have verified that. Some results are shown in the following two tables. In all cases, the CAR algorithm runs much faster than the Greedy algorithm. The running time does not include the time for building the BST tree, since it's the same for the two algorithms. The cover size from the CAR algorithm is larger than that from the Greedy algorithm, except when the cover size becomes very large, in which cases the remove phase (Step 3) of the CAR algorithm produces better results.

TABLE V. THE RUNNING TIMES OF THE TWO ALGORITHMS

Greedy	0.63	53.9	300	1220	2130	3457	5056
CAR	0.01	0.31	1.63	6.36	11.15	16.92	20.70

TABLE VI. THE COVER SIZES OF THE TWO ALGORITHMS

Greedy	10	87	191	424	625	849	984
CAR	16	120	235	467	648	824	975

4 ALGORITHM LAR

The CAR algorithm picks any set to add to ResultCover and hence runs fast but produces larger cover sets in most cases. It can also be seen that the result cover from the CAR algorithm depends on the order of the input sets. For the example in Table IV, the algorithm would produce the same result cover {S5, S4} as the Greedy algorithm if the input sets

were ordered as {S5, S4, S3, S2, S1}. But it's unlikely that such a good order can be found easily in general.

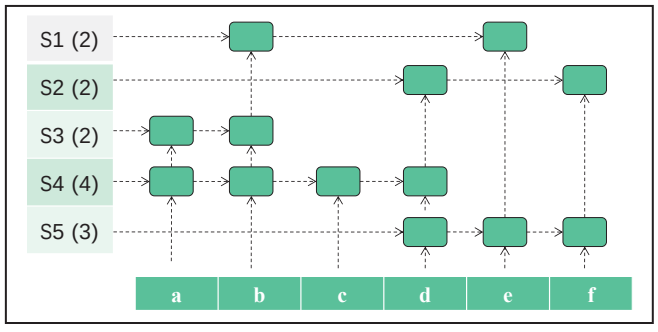
The Greedy algorithm spends a lot of time in finding the best set to add to ResultCover in order to produce a good cover set and hence runs very slow. To find the best set to add to ResultCover, the elements of each remaining set need to be examined against the elements in Covered to calculate the count of uncovered elements. This process has to be repeated each time a set is added to ResultCover, and it is clear that a lot of redundant calculation has occurred. The performance would be improved significantly if the process to find the best set could be optimized.

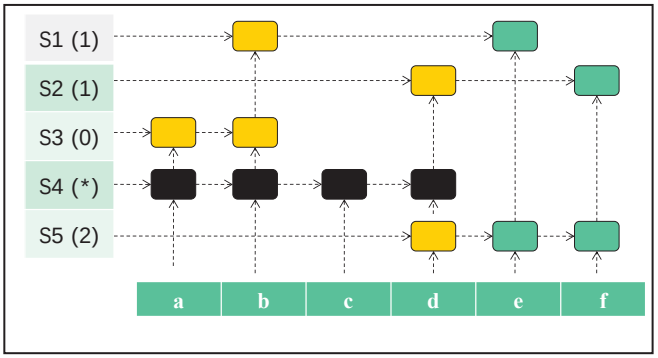
One way to optimize the process is to keep a count (UncoveredCount) for each set as the number of elements uncovered by the ResultCover. Then the best set to be added can be found easily. This requires an efficient way to modify UncoveredCount after a set is added to ResultCover. We have found that using double linked list for the matrix can achieve the goal with minimum cost.

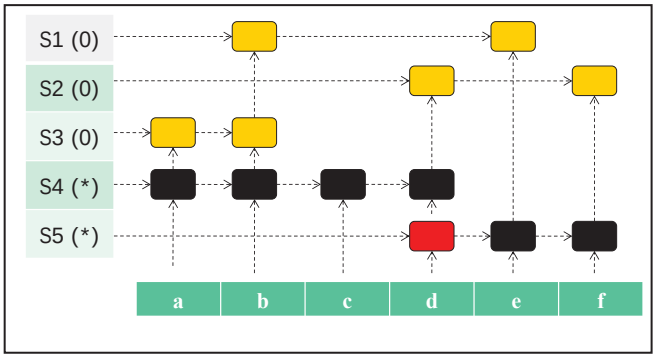
Table VII shows this approach on the same example as before. A cell with value 1 in the matrix becomes a node, all nodes corresponding to the cells of one row of the matrix are linked for one set and all nodes corresponding to the cells of one column of the matrix are also linked for one element.

At the beginning, ResultCover is empty, and the initial value for UncoveredCount is 2, 2, 2, 4 and 3. S4 is the first set to be added to ResultCover. The row list for S4 is traversed and all the nodes are marked as covered (black). When a node is marked as covered, the column list for the corresponding element is traversed, each node on the column list is marked as covered through another set (yellow) and UncoveredCount of the corresponding set is decremented by one. The second set to add is S5. When the row list for S5 is traversed, the first node for element d is yellow indicating it's covered already, so the column list for element d will not be traversed and the node is marked red indicating it's visited twice. Then the nodes for elements e and f are visited, marked black, and the corresponding column lists are traversed. After that the algorithm terminates with ResultCover {S4, S5}.

TABLE VII. LINKED LISTS FOR MATRIX







We also incorporated the remove phase from the CAR algorithm and call it List and Remove (LAR).

Algorithm LAR (List and Remove)

1. Set both ResultCover and Covered to the empty set
2. For each set, assign the number of elements of the set to UncoveredCount
3. While there is a set with UncoveredCount > 0
   - 3.1 Select a set with the largest UncoveredCount among all sets with UncoveredCount > 0
   - 3.2 Add the set to ResultCover
   - 3.3 Update the value of UncoveredCount for each affected set
4. For each set in ResultCover
   - 4.1 Determine if it has an element that is not covered by any other sets in ResultCover
   - 4.2 Remove the set from ResultCover if it does not have such an element

In [6], it was claimed that the time complexity of algorithm LAR is O (M * N), the same as that of algorithm CAR. After a careful analysis, we have found that the time complexity of algorithm LAR is actually less than that of algorithm CAR.

For algorithm LAR, step 3.1 takes time O (N^2), since all sets need to be examined to determine the set to be added in each iteration and step 3 could run N iterations. To update the UncoveredCount for the affected sets in each iteration one row list is traversed and one or more column lists will be traversed. But overall for all iterations, each row list will be traversed at most once when the set is added to ResultCover. Similarly each column list will be traversed exactly once overall for all iterations when the first time a set containing the

corresponding element is added to ResultCover and all nodes on the list will be marked as black or yellow; when another set containing the same element is added to ResultCover, the same column list won't be traversed again, since the corresponding node is yellow and will be changed to red. Thus each node will be visited at most twice and the overall time needed for step 3.3 to update the UncoveredCount is O ($\sum |S_i|$, i = 1 to N), where N is the number of all sets.

For step 4, we can traverse each column list first to create a SetCoverCount for each element that indicates the number of sets in ResultCover containing the element. Then for each set in ResultCover, the corresponding row list is traversed. If the SetCoverCount for the each element corresponding to the nodes of the row list is greater than one, then this set is redundant and removed from ResultCover with all the corresponding SetCoverCount decremented by one. So for step 4, each node will be visited at most twice and the overall time needed for step 4 is also O ($\sum |S_i|$, i = 1 to N).

Thus the time complexity for algorithm LAR is O ($N^2 + \sum |S_i|$, i = 1 to N). This is clearly better than O (M * N), the time complexity of algorithm CAR, in most cases, because M > N and M * N > $\sum |S_i|$, where N is the number of sets and M is the number of all elements of the union S_i, i = 1… N.

Our experimental results are shown in Table VIII and Table IX (see [7] for more details). The running time does not include the time for building the BST tree, but it does include the time to convert the BST tree to the data structure for the matrix, which is an array of bitmaps for the Greedy and CAR algorithms and linked lists for our LAR algorithm. It takes longer time to convert the tree to linked lists than to the array, that's why the LAR algorithm runs slower than the CAR algorithm when the total running time is very short (less than one second). But in most cases, the LAR algorithm runs faster than the CAR algorithm. This is the advantage of combining linked list and updating the uncovered count for each set. Furthermore, the cover size from algorithm LAR is smaller than that from algorithm CAR in all cases, and even smaller than the Greedy algorithm in some cases, since it adapts the remove phase from the CAR algorithm.

TABLE VIII. THE RUNNING TIMES OF THE THREE ALGORITHMS

Greedy	0.63	53.9	300	1220	2130	3457	5056
CAR	0.01	0.31	1.63	6.36	11.15	16.92	20.70
LAR	0.21	0.35	0.51	0.86	1.11	1.40	1.66

TABLE IX. THE COVER SIZES OF THE THREE ALGORITHMS

Greedy	10	87	191	424	625	849	984
CAR	16	120	235	467	648	824	975
LAR	10	87	191	422	607	815	971

5 IMPROVING THE INPUT OPERATION

While testing these algorithms, some anomalous tests ran slower than expected. Timing different phases of the programs revealed that the data-input phase was quite slow compared to the rest. As mentioned before, a standard binary search tree (BST) is used in the input phase to sort all elements, and the average time for inserting one element should be O (log(M)), where M is the number of elements of the union of all N sets. However, when the elements are in sorted order before input, the time for inserting one element becomes O (M).

To investigate this issue, we have implemented two self-balancing trees: the Red-Black and the AVL trees. The Red-Black tree used in our experiments assumes null nodes are black, and the AVL tree is a custom iterative implementation.

Further analysis of the input phase revealed that a tree could be eliminated with some changes to the double-linked-list matrix. By sorting the elements of each set as they're read in, the elements could be inserted into the matrix without any trees. This was then implemented, using an efficient heap sort algorithm to sort the elements.

The results of the different approaches are shown in Table X (see [8] for more details). In most cases when the elements are not sorted before input, the AVL tree and the Red-Black tree take about the same amount of time, which is slightly more than that of the BST tree, and the No-Tree approach takes much longer time. But in the anomalous cases where the elements are almost sorted before input, the No-Tree approach performs the best, followed by the AVL tree, and the BST tree and the Red-Black tree takes significantly more time.

TABLE X. TIMES FOR DIFFERENT INPUT APPROACHES

BST	0.87	1.88	1.97	2.03	2.28	2.07	157.31
AVL	0.99	1.92	2.19	2.37	2.51	2.62	28.71
Red-Black	0.83	1.97	2.19	2.25	2.53	2.52	187.14
No-Tree	1.95	6.37	8.36	10.23	11.31	15.66	5.58

We propose the following hybrid approach for the input phase.

Hybrid Input Approach

If it is known that the elements are mostly sorted
    Use heap sorting to sort the elements of each set and insert them into the matrix without any tree
Else if it is known the elements are in random order
    Use the BST tree to input and sort the elements
Else
    Use the AVL tree to input and sort the elements

6 DATA GENERATION

The implementation reported in [6] takes a special approach to generate data. The minimum cover size is determined

beforehand and is passed to the data generator program, which first generates that many non-overlapping sets, then generates other sets by randomly selecting elements from the union of those non-overlapping sets. After generating all the sets, it shuffles them and outputs data to data files. The advantage of the approach is that the minimum cover size is known, but the produced data sets may not represent general cases.

We take a different approach to generate the test data. A range for the set size is decided before hand, and the size of each set is determined randomly according to the uniform or normal distribution. Similarly, a range for the elements is given, and the elements of each set are generated according to the uniform or normal distribution. The minimum cover size is unknown, but by changing the ranges and the choice of distributions, more general data sets can be generated. All of our experiments use test data sets from the new approach.

For the data sets generated in our experiments, we do not know the actual cover size, and it's not practical to find the actual size. We have run both algorithm CAR and algorithm LAR on the data sets generated by the approach from [6]. Recall that in this approach the actual cover size (CoverSize) is decided first, then CoverSize non-disjoint sets are generated, and other sets are generated by selecting elements from the union of the CoverSize sets. The total number of sets is fixed at 1000. The cover sizes from algorithm LAR are the same as the actual sizes in all cases. This is because the greedy algorithm is always trying to find the best sets to add to the result cover and will find the non-disjoint sets. For algorithm CAR, the cover size is the same as the actual cover size when the cover size is 200 and above; but when the actual cover size is smaller, the cover size is much larger than the actual size; this is because many other sets are selected before any of the non-disjoint sets get a chance to be selected.

TABLE XI. COVER SIZES WHEN THE MINIMUM COVER IS KNOWN

Actual	50	70	90	110	200	500	900
LAR	50	70	90	110	200	500	900
CAR	291	391	496	528	200	500	900

7 SUMMARY

Our LAR algorithm employs the double linked list presentation of a matrix and adapts the remove phase of the CAR algorithm. With the major step of the Greedy algorithm optimized, the LAR algorithm has a better time complexity than the CAR algorithm and produces better results than the Greedy algorithm. With the hybrid input approach, the LAR algorithm provides a complete solution to the set-covering problem. Experimental results show that the LAR algorithm is both efficient and effective.

REFERENCES

[1] Richard M. Karp, "Reducibility Among Combinatorial Problems". In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85–103, 1972.

[2] E. Balas and M.W. Padberg, "Set Partitioning: A Survey", SIAM Review, 18, 710-760, 1976.

[3] Feige, Uriel, "A threshold of ln n for approximating set cover", Journal of the ACM (ACM) 45 (4): 634–652, 1998.

[4] T. H. Cormen, C.E. Leiserson, R. L. Rivest. "Introduction to Algorithms". The MIT Press, 1991.

[5] Alon, Noga; Moshkovitz, Dana; Safra, Shmuel, "Algorithmic construction of sets for k-restrictions", *ACM Trans. Algorithms* (ACM) **2** (2): 153–177, 2006.

[6] R. Desai1, Q. Yang, Z. Wu, W. Meng, C. Yu. "Identifying Redundant Search Engines in a Very Large Scale Metasearch Engine Context". ACM WIDM'06 (8th ACM International Workshop on Web Information and Data Management, November 10, 2006, Arlington, Virginia, USA, pp. 51-58.

[7] Q. Yang, J. MacPeek, A. Nofsinger, "Efficient and Effective Practical Algorithms for the Set-Covering Problem," The 2008 International Conference on Scientific Computing (CSC'08), The 2008 World Congress in Computer Science, Computer Engineering and Applied Computing (WORLDCOMP'08), Las Vegas, July 14-17, 2008.

[8] J. Phinney, R. Knuesel, Q. Yang, "Improving the Input Operation for the Set-Covering Problem," The Midwest Instructional and Computing Symposium, April 16-17, 2010, University of Wisconsin-Eau Claire.

A fast algorithm for coordinate rotation without using transcendental functions

Jorge Resa[1], **Domingo Cortes**[1], **and David Navarro**[1]
[1]Instituto Politecnico Nacional
Superior School of Electrical and Mechanical Engineering
Av. Santa Ana No. 1000 Mexico City, Mexico.
Email: jorgeresa@yahoo.com,
domingo.cortes@gmail.com,
david.navarro.d@gmail.com

Abstract— *Coordinates rotation is widely used in science and engineering. It has applications on astronomy, image processing, robotics, power electronics, etc. This paper presents an efficient algorithm to calculate a rotation transform using digital devices. The method is based on the rational trigonometry. Unlike conventional trigonometry which is based on the concepts of angle and distance, the rational trigonometry is based on the concepts of spread and quadrature. In is also presented an analysis of the number of operation that can be saved using a standard math library in a typical situation.*

Keywords: Coordinate transformation, Rotation matrix, Spread, Rational trigonometry

1. Introduction

Coordinate transform is a key concept in mathematics and a very useful tool in engineering. Particularly, coordinate rotation transform is widely employed in image processing, robotics and power electronics among others industrial and scientific applications [1], [2]. Typical application in these areas could require hundreds of rotation transforms per second. The most standard method to carry out a rotation transform is to multiply the original coordinates times a special matrix called rotation matrix. The elements of such matrix are trigonometric functions of the rotation angle.

Numerical methods are necessary to calculate trigonometric functions using digital devices. In general, first power terms of the power series is employed. For example to calculate $\sin(\alpha)$ the first five terms of its Taylor series can be used if $\alpha \in [-\pi, \pi]$. However, if the angle is not in this interval the error increase rapidly. Hence it is necessary to preprocess the angle before using the Taylor series. Furthermore, calculation of the five terms of Taylor series, depending on the variable type (float, frac, etc.), requires additional calculations to obtain the coefficients for each term.

The fact that evaluation of several trigonometric functions is necessary to calculate a single rotation matrix together with the high number of rotation transforms per second that are necessary to perform in a typical application make the use of high power computing devices mandatory for such applications. In power electronics for example, the need of many rotation transform per second prevent the use of low cost microcontrollers for common industrial applications like inverters, active filters and motor drives. This circumstance could be changed with a more efficient methods to perform a rotation transform. The search for efficient methods to perform coordinates rotation has a long history [3] and still continues. The cordic method for example has attracted many effort in the past decades [3], [4], [5], [6].

Recently, some mathematicians have questioned the need of real numbers in math [7], [8]. They say that all math ideas could be expressed with rationals and some irrational numbers. Seeking to solve trigonometrical problems without using real numbers the concept of spread has been introduced [7], [8]. Spread substitutes the angle notion and hence eliminates the necessity of using transcendental functions as $\sin$ and $\cos$. These ideas has risen a debate about whether the real numbers are necessary or not. No matter how this debate ends, in this paper it is shown that the spread concept has practical implications. The spread concept is extended to perform rotation transforms more efficiently than the standard method. As a consequence, the time for these transformations are significantly reduced. Moreover, the proposed method is easier to program and requires less memory.

The paper is organized as follows. In Section 2 the standard method to perform a rotation transform is revisited. To keep the figures simple the two dimensional case is used. Nevertheless the tree dimensional case is very similar. In Section 3 the spread concept is explained. In this Section is also shown how the spread concept allow to solve trigonometric problems without transcendental functions and why it is necessary to extend the concept to make it useful for coordinates rotation. In Section 4 the spread concept is extended to allow rotation transforms. In Section 5 an analysis is carried out to precise the performance improvement that could be expected with the proposed method in comparison with the standard method. Finally some conclusions are

given.

2. The 2d rotation matrix

Let $a = (x, y)$ a point in the coordinate system XY (see Figure 1). Let consider another coordinate system $X'Y'$ with the same origin but rotated an angle α. The coordinates rotation problem is to find the coodinates of $a = (x'y')$ in the new coordinate system.

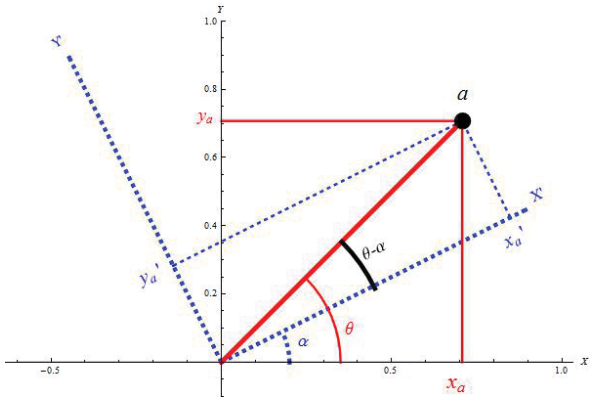


Fig. 1: Coordinates rotation

The usual method to solve this problem is as follows. From Figure 1 note that

$$x = r\cos(\theta) \tag{1a}$$
$$y = r\sin(\theta) \tag{1b}$$

where

$$r = \sqrt{x^2 + y^2}, \quad \tan^{-1}(\theta) = \frac{y}{x} \tag{2}$$

From Figure 1 can also be observed that

$$x' = r\cos(\theta - \alpha) \tag{3a}$$
$$y' = r\sin(\theta - \alpha) \tag{3b}$$

Using trigonometric identities for $\cos(\theta - \alpha)$ and $\sin(\theta - \alpha)$ results

$$x' = r\left(\cos(\theta)\cos(\alpha) + \sin(\theta)\sin(\alpha)\right) \tag{4a}$$
$$y' = r\left(\sin(\theta)\cos(\alpha) - \cos(\theta)\sin(\alpha)\right) \tag{4b}$$

using (1) in (4) yields

$$x' = x\cos(\alpha) + y\sin(\alpha) \tag{5a}$$
$$y' = -x\sin(\alpha) + y\cos(\alpha) \tag{5b}$$

which can be written as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{6}$$

That is, coordinates (x', y') can be obtained multiplying coordinates (x, y) by a Matrix. Such matrix is called the rotation matrix.

In some applications, such as the control of electrical machines, the rotation transform must be performed hundreds of times per second. A cos and a sin functions has to be calculated at every time that a rotation transform is performed. Calculation of trigonometric function takes many clock cycles. Hence, a device with high computing power is usually necessary for these applications.

Aimed to reduce the computing power neccesary for some power electronics and electrical machines control applications, in what follows a more efficient method to perform a rotation transform is proposed.

3. Spread and Quadratures

3.1 The spread concept

Consider the right triangle of Figure 2. The "spread" S_1 is defined as

$$S_1 = \frac{y^2}{h^2} \tag{7}$$

where $h^2 = (x^2 + y^2)$. That is squared oposite leg over squared hypotenuse.

Note that the spread S_2 is given by

$$S_2 = \frac{x^2}{h^2} \tag{8}$$

An easy to obtain property but an important one is

$$S_2 = 1 - S_1 \tag{9}$$

It is important to point out that

$$S_1 = \sin^2(\theta) \tag{10}$$

and

$$S_2 = \cos^2(\theta) \tag{11}$$

Note that S_1 and S_2 are always positive and sin and cos are squared in (10-11). As it is explained below, this fact prevent the use of spread for coordinates transformation. That is why it is necesary to introduce the extension presented in Section 4

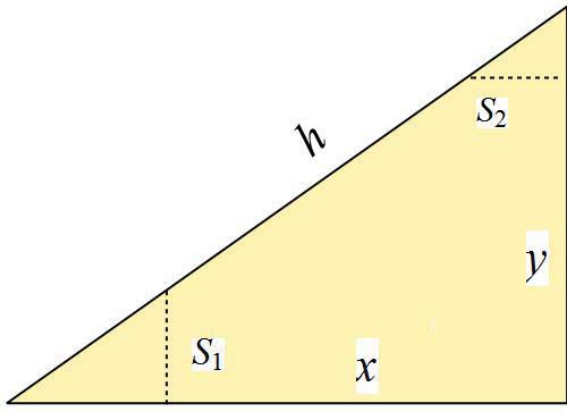


Fig. 2: The spread definition

It can be observed that for a right triangle if a two legs or a leg and spread is known all other data (legs or spread) can be calculated. For example, given x and h (see Figure 2) then S_2 can be calculated from (8). Having S_2, S_1 can be calculated from (9). Finally y can be calculated from (7). Hence, it can be said that it is possible to solve trigonometric problems using the spread concept. Furthermore, trascendental functions are not needed and all the numbers are rational. That is the cause for this trigonometry is called rational.

3.2 The quadrature concept

It is possible to extend the spread concept for not right triangles. Consider three points on a line as it is shown in Fig. 3. Let the distances

$$q_1 = p_2 - p_1, \quad q_2 = p_3 - p_2, \quad q_3 = p_3 - p_1 \tag{12}$$

The cuadrature is defined as the squared distance, that is

$$Q_1 = q_1^2, \quad Q_2 = q_2^2, \quad Q_3 = q_3^2, \tag{13}$$

Since the distance q_1, q_2 and q_3 there is a relation between quadratures. From Fig. 3, it can be observed that

$$Q_3 = Q_1 + Q_2 + 2Q \tag{14}$$

and

$$Q = \sqrt{Q_1}\sqrt{Q_2} \tag{15}$$

Sustituting (15) in (14) yields the so called quadrature equation

$$(Q_3 - Q_1 - Q_2)^2 = 4Q_1Q_2 \tag{16}$$

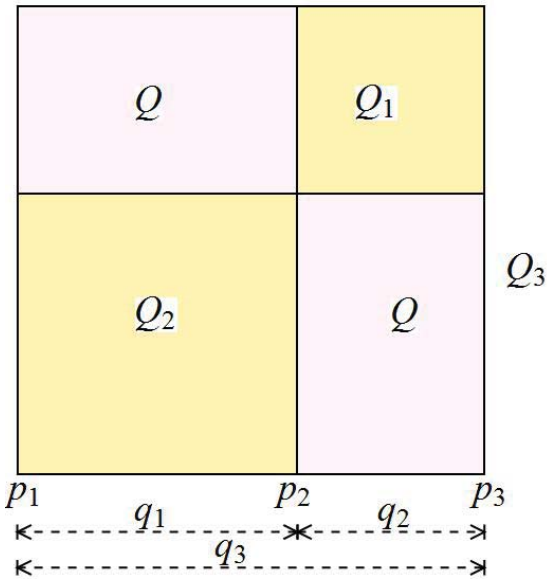


Fig. 3: Quadrature definition.

Now it will be shown that from (16) and the spread concept it is possible to solve trigonometric problems for not right triangles.

First note that any non right triangles can be splitted in two right triangles (see Fig. 4). Using the Pithagoras theorem and the spread, quadratures Q_a, Q_b and Q_h can be expressed as

$$Q_a = Q_1 - Q_h \tag{17a}$$
$$Q_b = Q_3 - Q_h \tag{17b}$$
$$Q_h = S_3 Q_1 \tag{17c}$$

As Q_a, Q_b and Q_2 are the quadratures fo three distances in a line, then (16) can be used, yielding

$$(Q_2 + Q_b - Q_a)^2 = 4Q_2Q_b \tag{18}$$

Substituting (17) in (18) results

$$(Q_1 + Q_2 - Q_3)^2 = 4Q_1Q_2(1 + S_3) \tag{19}$$

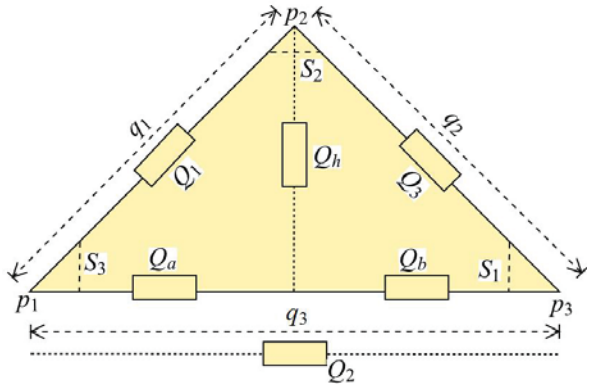


Fig. 4: Quadratures of a triangle

Using (19) it is possible to calculate the spread of two non parallels lines if a point of each line is known. This is a consequence that the intersection point and a point of each line form a triangle (not necessarily a right triangle). The algebra to find the spread of two lines that intersect on the origin is easy and yields

$$S = \frac{(x_2y_1 - x_1y_2)^2}{(x_1^2 + y_1^2)(x_2^2 + y_2^2)} \tag{20}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of a point in line 1 and line 2 respectively.

3.3 The problem of rotation transform using spread and quadratures

Although expressions (7-9) and (20) allow us to solve trigonometric problems, in its current form they are not useful to perform rotation transforms because two restrictions

- Spread is only defined in the interval $[0 - 1]$ (that is $0 - 90^o$ degrees).

- Spread between two lines does not distinguis which line has more spread (or angle) with respect to X-axis.

These restrictions cause that the four cases depicted in Fig. 5 are indistiguisable in terms of spread because all of them yields the same value. In the next Section the spread concept is equiped with other consideration to distinguish the four cases of Figure 5.

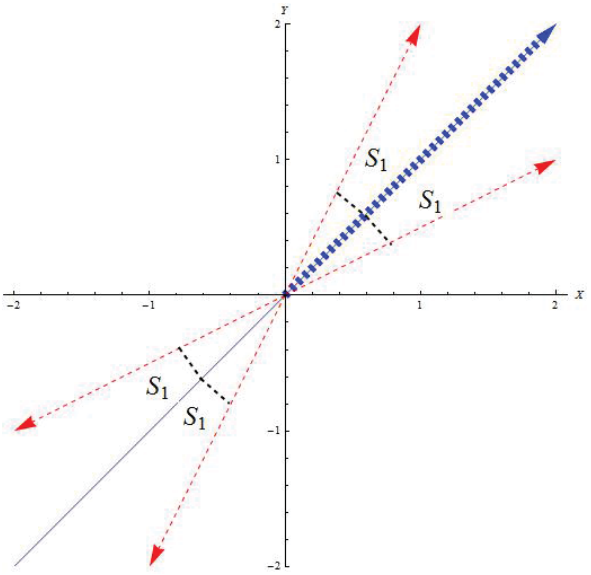


Fig. 5: Four indistinguishable cases

4. Extending the spread concept to allow rotation transforms

In order to distinguish each case of Figure 5 let first distinguish if the vector is in the shaded or non-shaded area of Figure 6.

Consider a point (x_1, y_1) on the positive side of X' axis (see Figure 6). Note that any point (x, y) on the X' axis satifies

$$y = \frac{y_1}{x_1} x \tag{21}$$

or

$$xy_1 - x_1 y = 0 \tag{22}$$

From 22, any point on the shaded area satisfies

$$y_1 x - x_1 y < 0 \tag{23}$$

on the other hand any point on the non-shaded area acomplish

$$y_1 x - x_1 y > 0 \tag{24}$$

From (10) and (23-24) it can be obtained the following equivalence

$$\sin(\theta) = -\operatorname{sign}(y_1 x - x_1 y)\sqrt{(S)} \tag{25}$$

Defining $v_1 = -\operatorname{sign}(y_1 x - x_1 y)$, (25) becomes

$$\sin(\theta) = v_1\sqrt{(S)} \tag{26}$$

Consider now the Figure 7 and a point on the Y' axis. Such a point can be obtained from the point (x_1, y_1) as follows

$$\begin{bmatrix} x_{1\perp} \\ y_{1\perp} \end{bmatrix} = \begin{bmatrix} 0 & (-1) \\ (1) & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{27}$$

Having the point $(x_{1\perp}, y_{1\perp})$, it can be seen that any point on the Y' axis satisfies

$$y = \frac{y_{1\perp}}{x_{1\perp}} x \tag{28}$$

or

$$xy_{1\perp} - x_{1\perp} y = 0 \tag{29}$$

From Figure 7 any point (x, y) on the shaded area satisfies

$$y_{1\perp} x - x_{1\perp} y < 0 \tag{30}$$

and the points on the non-shaded area acomplish

$$y_{1\perp} x - x_{1\perp} y > 0 \tag{31}$$

From (11) and (30-31) it can be obtained the equivalence

$$\cos(\theta) = \operatorname{sign}(y_{1\perp} x - x_{1\perp} y)\sqrt{(1 - S)} \tag{32}$$

defining $v_2 = \operatorname{sign}(y_{1\perp} x - x_{1\perp} y)$, (32) becomes

$$\cos(\theta) = v_2\sqrt{(1 - S)} \tag{33}$$

Substituting (26) and (33) in (6) results in

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} v_2\sqrt{1 - S}\sqrt{x^2 + y^2} \\ -v_1\sqrt{S}\sqrt{x^2 + y^2} \end{bmatrix} \tag{34}$$

where S is given by (20)

Note that the rotation matrix given by (34) only requires aritmetic operation, two sign and two square root extraction. It does not require any preprocessing, hence is easier to program and requires less memory.

5. A typical application

To compare the proposed method and the standar procedure a typical engineering problem is considered. Suppose there are two vectors $v_1 = (x_1, y_1)$ and $v_2 = (x_2, y_2)$. Consider the problem of decomposing v_2 in two components, one parallel and one orthogonal to v_1 (see Figure 8). Such problem arise in many power electronics applications. The usuall way to proceed is first to calculate the angle θ between the two vectors and then to find the proyection of v_2 on v_1 and its orthogonal vector. In the first step the calculus of a $\tan^{-1}$ function is necesary or a $\sin$ and a $\cos$. The second step requires calculation of at least one $\sin$ and one $\cos$. Because digital systems use power series to evaluate trigonometric functions, a significant amount of time is required to solve this problem. On the other hand only

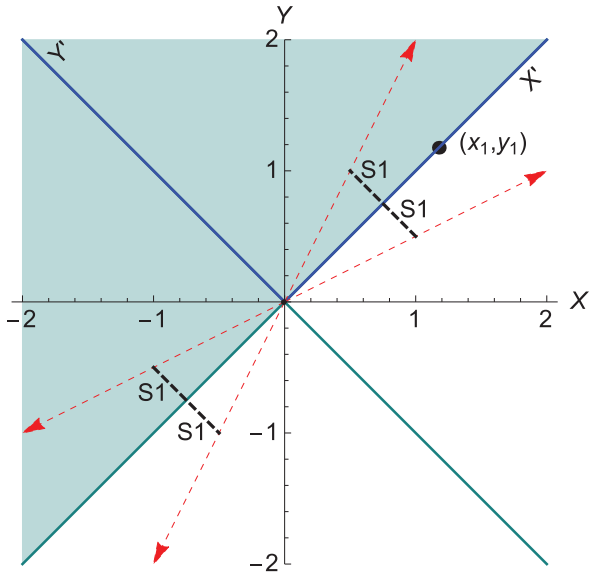


Fig. 6: Establishing the equivalence of sin and S

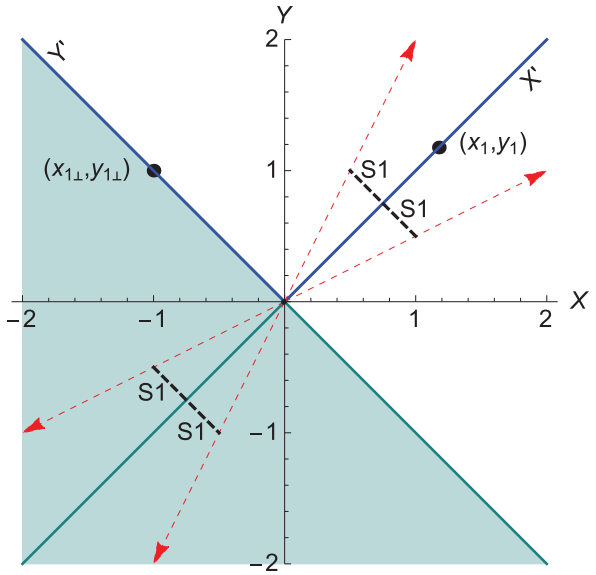


Fig. 7: Establishing the equvalence of cos and S

19 aritmetic operation and two sign operations are needed using the spread concept.

For comparison the two methods were programmed in a Freescale board FRDM-K64F @60MHz using the standard C math library. The test carried out with and without floating point unit (FPU) [9]. Without the FPU the performance of the proposed method was 2.68 times faster than the standar method. When the FPU was used the proposed method is 2.28 faster.

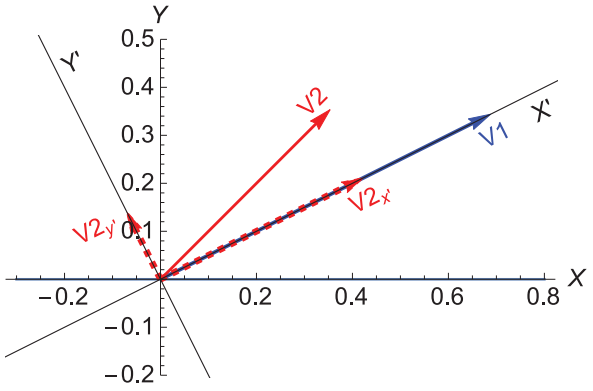


Fig. 8: Typical application of coordinates rotatio

6. Conclusions

Use of spread instead of angles to solve trigonometrical problems has the advantage of only requiring aritmetic oprations. However the spread only works for a quadrant in the plane. In this paper the spread concept has been extended to work in the entire plane. As a result a new rotation matrix that not use sin and cos functions was obtained. The proposed method is faster, easy to program and requires less memory.

References

[1] G. E. Owen, *Fundamentals of Scientific Mathematics*. Dover Publications, Inc., 2003.

[2] W. A. Meyer, *Geometry and Its Applications*. Elsiever Academic Press, 2006, iSBN 0080478034, 9780080478036.

[3] P. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50 years of cordic: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems I*, vol. 56, no. 9, p. 14, September 2009.

[4] T.-B. Juang, S.-F. Hsiao, and M.-Y. Tsai, "Para-cordic: parallel cordic rotation algorithm," *IEEE Transactions on Circuits and Systems I*, vol. 51, no. 8, p. 10, August 2004.

[5] J. E. Volder, "The cordic trigonometric computing technique," *IRE Transactions on Electronic Computers*, vol. EC-8, September 1959.

[6] Y. Hu, "The quantization effects of the cordic algorithm," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, p. 11, April 1992.

[7] N. J. Wildberger, *Divine Proportions: Rational Trigonometry to Universal Geometry*. Wild Egg Books, 2005, iSBN: 0-9757492-0-X.

[8] ——, "Affine and projective universal geometry," *arXiv:math/0612499*, p. 22, December 2006.

[9] *IEEE Standard for Floating-Point Arithmetic*, IEEE Std., 2008.

SESSION

MODELING AND SIMULATION FRAMEWORKS

Chair(s)

Dr. Douglas D. Hodson
Prof. Michael R. Grimaila

The Unified Behavior Framework for the Simulation of Autonomous Agents

Daniel Roberson*, Douglas Hodson†, Gilbert Peterson‡, and Brian Woolley§
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson Air Force Base, Ohio 45433
Email: {daniel.roberson, douglas.hodson, gilbert.peterson, brian.woolley}@afit.edu
Phone: *443-504-9177, †937-255-3636 x4719 , ‡937-255-3636 x4281 , §937-255-3636 x4618

***Abstract*—Since the 1980s, researchers have designed a variety of robot control architectures intending to imbue robots with some degree of autonomy. A recently developed architecture, the Unified Behavior Framework (UBF), implements a variation of the three-layer architecture with a reactive controller to rapidly make behavior decisions. Additionally, the UBF utilizes software design patterns that promote the reuse of code and free designers to dynamically switch between behavior paradigms. This paper explores the application of the UBF to the simulation domain. By employing software engineering principles to implement the UBF architecture within an open-source simulation framework, we have extended the versatility of both. The consolidation of these frameworks assists the designer in efficiently constructing simulations of one or more autonomous agents that exhibit similar behaviors. A typical air-to-air engagement scenario between six UBF agents controlling both friendly and enemy aircraft demonstrates the utility of the UBF architecture as a flexible mechanism for reusing behavior code and rapidly creating autonomous agents in simulation.**

I. Introduction

The pursuit of autonomous agents is one of the main thrusts of the artificial intelligence research community. This has manifested in the robotics community, where development has progressed towards the creation of robots that can autonomously pursue goals in the real world. Building robots to explore autonomy is practical, but it requires investment of time and resources beyond the design and development of the software. On the other hand, simulation is an effective and inexpensive way of exploring autonomy that does not require the hardware, integration effort, and risk of damage inherent in designing, constructing, and testing robots. Not only that, but robots can be simulated in a variety of environments that push the limits of their autonomous capability. The ability to stress and analyze a robot might otherwise be impractical in a real-world context. So, it seems that simulation is a good option for researching and testing applications of autonomous robots. However, there is a plethora of robot control architectures available, and simulating each of them individually would require a huge code base. With the application of software engineering principles, it is possible to reduce this coding requirement. Doing so grants the designer access to a wide range of autonomous architectures within a single, flexible framework.

The Unified Behavior Framework (UBF) applies such software engineering principles by implementing well-established design patterns and an extensible behavior paradigm. Because of it's flexibility, UBF can be used to explore multiple robot control architectures simultaneously. Currently, UBF has been implemented mainly on robot platforms [1]. However, due to it's adaptability, it is ripe for implementation on other AI platforms. In this paper, we discuss a basic implementation and demonstration of UBF within a simulation environment. The OpenEaagles (Open Extensible Architecture for the Analysis and Generation of Linked Simulations) is an open-source framework that simplifies the creation of simulation applications. Again, by utilizing software engineering principles, OpenEaagles lends itself to the rapid creation of applications, and therefore is a copacetic simulation framework in which to implement the UBF. Additionally, a simple example of an air engagement scenario was developed in order to demonstrate the utility of the implementation. This scenario, known generically as the sweep mission, verifies the ability for rapid scenario development with UBF-based agents, and it demonstrates its application in a military context.

In this paper, we will first discuss some relevant background concerning behavior trees, previous applications of UBF, the OpenEaagles framework, and a breakdown of a sweep mission scenario used to test UBF's implementation in OpenEaagles. We then delve into this implementation, examining the UBF structure within OpenEaagles, and the specific implementation of the sweep mission within the our UBF implementation. We will discuss the results of our implementation of the sweep mission, and end with a look at future work and a conclusion.

II. Background

A. Behavior trees

While the robotics community has progressed from Sense-Plan-Act (SPA) architectures, through subsumption, to three layer architectures for controlling their robotic agents, the commercial gaming industry has faced similar problems when trying to create realistic non-player characters (NPCs). Like robots, these NPCs are expected to be autonomous, acting with realistic, human-like intelligence within the game environment. As Isla states, "a 'common sense' AI is a long-standing goal for much of the research AI community." In pursuit of

this goal, Isla introduced an AI concept, colloquially referred to as "behavior trees," which was first implemented in the popular console game Halo 2. More technically, behavior trees are hierarchical finite state machines (HFSMs) implemented as directed acyclic graphs (DAGs) [8], [9].

In the same way that recent robot architectures focus on individual tasks, or behaviors, an agent's behavior tree executes relatively short behavior scripts directly onto the NPC, so that it exhibits the specified behavior. These scripts are built into a tree structure that is traversed depth-first node-by-node. The tree is queried, or "ticked" at a certain frequency, and behaviors are executed (or not) based on the structure of the tree and the types of nodes that are being ticked. In order to facilitate decision-making, the tree contains multiple types of nodes. As Marzinotto defines them, these node types are either specified as internal or external (leaf) nodes. The internal node types are selector, sequence, parallel, and decorator, while the external/leaf nodes are either actions or conditions. In addition, after being ticked all nodes will either return a success, failure, or running condition, indicating whether the behavior was successful, or if it is still running [9]. Figure 1 provides a simple example of a behavior tree that utilizes at least one of each type of node and implements autonomous vehicle-driving behavior.

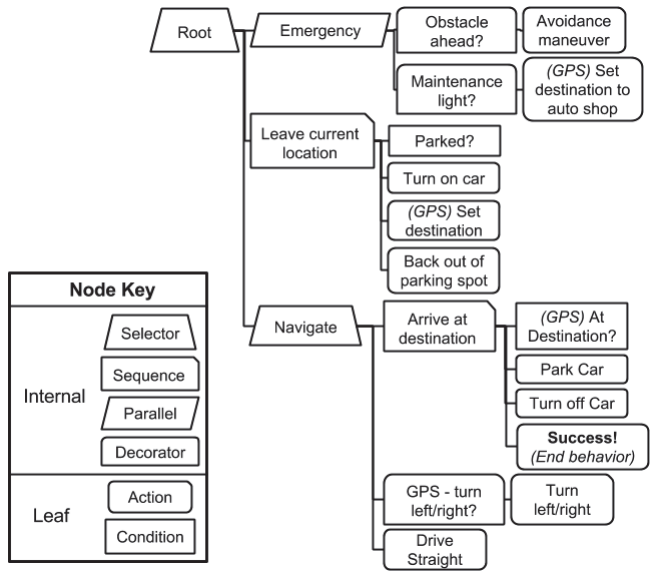


Fig. 1. An example behavior tree implementing autonomous driving behavior (with the aid of a GPS). Note that the nodes in the tree will be "ticked" from top to bottom, implying that behaviors higher in the tree have higher priority.

At the leaf level, action nodes are the only nodes that actually implement control steps upon the agent. When an action node is ticked, it will execute the control step and return running until the control step is complete. Once completed, success or failure return values indicate whether the control step achieved the desired state.

Condition nodes, like action nodes, evaluate the agent's state and return either success or failure, however, they cannot exercise control over the agent, and therefore cannot return running.

Internally, selector, sequence, parallel, and decorator nodes represent different elements of the agent's decision-making process. Selector nodes select one child by ticking each child in order until one of the children returns running or success, which the selector node also returns. If all children return failure, the selector node fails.

Sequence nodes execute each child in order, by ticking each until one of the children returns running or failure. If none of the sequence node's children fails, it will return success, otherwise, it will return running or failure based on the running/failed child's return condition.

Parallel nodes tick all children regardless of return condition, ticking each child node in sequence. The parallel node maintains a count of the return values of every child. If either the success or return value counts are greater than established thresholds, the parallel node will return the respective success or failure condition. If neither threshold is met, the parallel node will return running.

Finally, decorator nodes have internal variables and conditions that are evaluated when ticked, and are only allowed one child node. If the conditions based on the internal variables of the decorator node are met, the child node is also ticked. The return value of a decorator node is based on a function as applied to the node's internal variables.

Due to their ease of understanding and the ability to quickly construct large trees, behavior trees are extremely effective for building AI agents in commercial games. As Marzinotto demonstrated, with slight modifications, behavior trees can also be effectively applied to robot control architectures [9]. There are a few limitations when it comes to robot control, however. First is the necessity for the behavior tree action nodes to have direct control over the robot's actuators. This is less of a problem, as the running return value of nodes accounts for the time it takes for a node to complete the relevant behavior. However, in addition to requiring direct control over the actuators, the entire behavior tree also needs access to the current world state. In commercial games, these are not issues, as the NPCs can be given complete and 100% accurate information about the virtual world at any time, with no sensors or world model-building required. In the robot control domain, however, the state of the robot must be gathered from the sensors and built into some sort of world model, which is sometimes inaccurate, due to the world changing. Marzinotto admits that "a large number of checks has to be performed over the state spaces of the Actions in the [behavior] tree," acknowledging this shortfall of behavior trees for robot control. In his case, Marzinotto works around this problem of behavior trees by being willing to accept a delayed state update rather than interrupt the ticking over the behavior tree [9]. Also, behavior trees lack the flexibility of behavior-switching and goal-setting provided by sequencer and deliberator (respectively) of the three layer architecture.

B. Unified Behavior Framework

In response to the issues of behavior trees for robot control, the Unified Behavior Framework (UBF) decouples the behavior tree from the state and actions. By reintroducing the controller, the UBF enforces a tight coupling between sensors and actuators, ensuring the rapid response times of reactive control architectures. UBF also utilizes the strategy and the composite design patterns to guarantee design flexibility and versatility over multiple behavior paradigms [10]. In this way, UBF reduces latency in the autonomous robots, while offering implementation flexibility by applying software engineering principles. Additionally, the modular design of UBF speeds up the development and testing phases of software design and promotes the reuse of code [1].

The UBF was initially implemented on robot platforms, as a way to accomplish real-time, reactive robot control [10], [1]. In robot control implementations, a driving factor is the speed with which the robot reacts to the changing environment. Again, the current methodology for ensuring quick response time in reactive control is to tightly couple sensors to actuators through the use of a controller. Figure 2 contains a UML diagram of the Unified Behavior Framework.

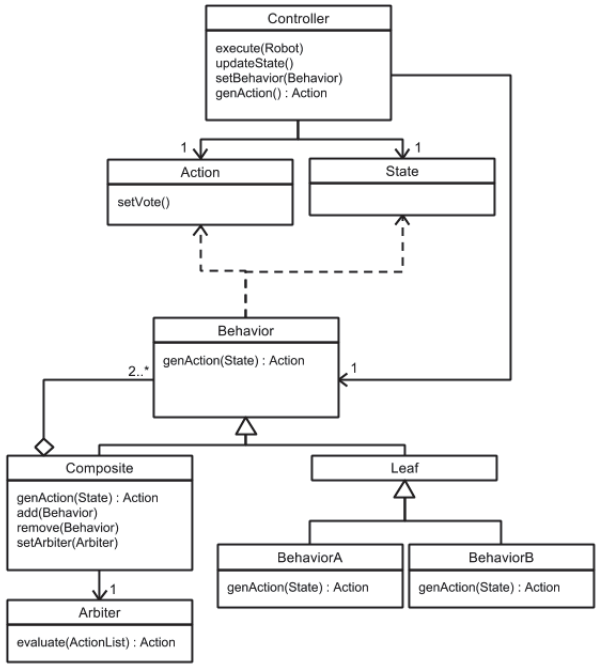


Fig. 2. A UML diagram of the Unified Behavior Framework (UBF) [10].

1) Behavior: The initial success of the subsumption architecture came from viewing the functional units of the robot control architecture as individual robot tasks or behaviors, instead of chronological steps in the robot's decision making process. The UBF utilizes this concept, viewing the smallest units of the architecture as individual behaviors. And, taking a page from the commercial game industry, these behaviors are developed individually and added to a tree structure. However, similar to the three layer architecture, behaviors are not given access to the robot's sensors or actuators; instead, the sensing and actuation is left to the controller, as discussed next. As expected, these behaviors are the central part of the agent's "intelligence," and they define individual tasks that the robot intends to perform. In practice, UBF behaviors interpret the perceived state of the robot (as represented by the UBF State class). Then, based on the task being performed, the behavior may test certain conditions or otherwise evaluate the state passed to it. After interpreting the state, the behavior recommends a specific action to take. During each traversal of the UBF tree, every behavior recommends and returns an action for the robot to take.

2) Controller, State & Action: As with three layer architectures, the controller is the direct interface between the UBF and the sensors and actuators of the robot. As the layer closest to the hardware, the controller has two primary responsibilities. First, the controller develops the "world model," or the state, by interpreting the incoming sensor data. Then the controller actuates the robot's motors and controls based on the characteristics of the action output by the UBF behavior tree.

As is the case with any robot control architecture, some representation of the real world, or the world model, is present in UBF. This is referred to as the state. Through the updateState() method, the controller interprets the sensor data for the robot. Because of the possible inaccuracies and failures of sensors in robot control applications, the state is described more accurately as the "perceived state," as the actual world state cannot be known, but can only be interpreted based on input from the sensors.

A quick philosophical aside: although we might imply that these robots are somehow inferior due to their limitations in perceiving the world state correctly, we must humble ourselves; we humans are also limited to the inputs from our "sensors" - our eyes, ears, mouth, skin, etc. So, in the same way, our understanding of the world's state may also be flawed, despite our inherent trust in our perspective.

As described in the previous section, each behavior in the UBF tree recommends an action for the robot to take. This action is a representation of what a behavior is recommending that the robot do, it does not actually control the motors on the robot, keeping in line with three layer architectures. By this method, the UBF behavior tree remains decoupled from the specifics of the robot, enhancing the flexibility of the framework for use in different applications. Actions might represent small adjustments to the robots actuators, but are typically more abstract representations, such as vectors indicating a desired direction and magnitude for the robot to go. As such, the action can be tailored to the desired effect on the robot, but the details of the actuation of controls is left to the controller, and is therefore not dealt with inside the UBF behavior tree.

Because the controller is the only direct link to the sensors and actuators, other elements of the UBF behavior tree are interchangeable between different robots by making adjustments

to the controller. In the same way, differing UBF behavior trees and architectures can be swapped in and out on the same robot by retaining the controller. Due to this structuring, the behavior packages can even be swapped in and out at runtime [10].

3) Arbiter: Because each behavior recommends an action, multiple actions are being passed up the UBF behavior tree as return values from behaviors' children. Therefore, a method of choosing the "correct" action from child behaviors the UBF behavior tree is required. This is the reason for the UBF Arbiter class. The arbiter is contained within UBF behaviors that are internal nodes in the UBF behavior tree. These internal behaviors have one or more children that will be recommending actions for the robot to perform. The arbiter acts as another decision-maker, determining which of its children is the appropriate action to pass further up the tree to its parent, until the desired action is returned from the UBF behavior tree's root node (which also contains an arbiter). In this way, the root node of the tree will use its arbiter to recommend a single action, based on the returned actions of the entire tree.

Arbiters can have differing schemes for determining the most important action. Simple arbiters, such as a winner-takes-all (WTA) arbiter, might just choose the highest-voted action from that behavior's children nodes. A more complex arbiter might "fuse" multiple returned actions into one, where the components of the composite action are weighted by each individual action's vote. This is known as a "fusion" arbiter.

In a general sense, fusion arbiters combine one or more actions returned by its children in the UBF behavior tree. By "fusing," a single action will be created and returned by the fusion arbiter which has elements from multiple of the child behaviors' recommended actions. Typically, some set of the highest-voted actions returned to the fusion arbiter are selected, and those actions combinable attributes are all added to a single action which is then returned by the arbiter. There are varying ways that this can be achieved. One method would be to select the highest-voted actions, and combine their non-conflicting attributes. Or, to achieve "fairness" between the highest-voted actions, their attributes might be weighted relative to their respective votes before being "fused" into the arbiter's returned action. In this way, a fusion arbiter is really a larger category of arbiters with infinite possibilities of how to combine the actions returned by the UBF tree.

The variety and customization available for arbitration implementations allows for great flexibility, whereby the entire behavior of a robot can be modified by using a different arbitration scheme, even if the rest of the UBF behavior tree remains unchanged.

C. OpenEaagles Simulation Framework

UBF has been implemented as a robot control architecture, but is clearly ripe for implementation in simulation. To maintain the flexibility and versatility that UBF provides, a simulation framework that was also developed using these principles is necessary. The Open Extensible Architecture for the Analysis and Generation of Linked Simulations (OpenEaagles) is such a framework. OpenEaagles is open-source, meaning that the code base is readily accessible. With the express purpose of "[aiding] the design of robust, scalable, virtual, constructive, stand-alone, and distributed simulation applications," OpenEaagles is a worthwhile tool in which to add UBF capability [11].

OpenEaagles is an open-source simulation framework that defines the design pattern shown in Figure 3 for constructing a wide variety of simulation applications. The framework itself is written in C++ and leverages modern object-oriented software design principles while incorporating fundamental real-time system design techniques to build time sensitive, low latency, fast response time applications, if needed. By providing abstract representations of many different system components (that the object-oriented design philosophy promotes), multiple levels of fidelity can be easily intermixed and selected for optimal runtime performance. Abstract representations of systems allow a developer to tune the application to run efficiently so, for example, interaction latency deadlines for human-in-the-loop simulations can be met. On the flip side, constructive-only simulation applications that do not need to meet time-critical deadlines can use models with even higher levels of fidelity.

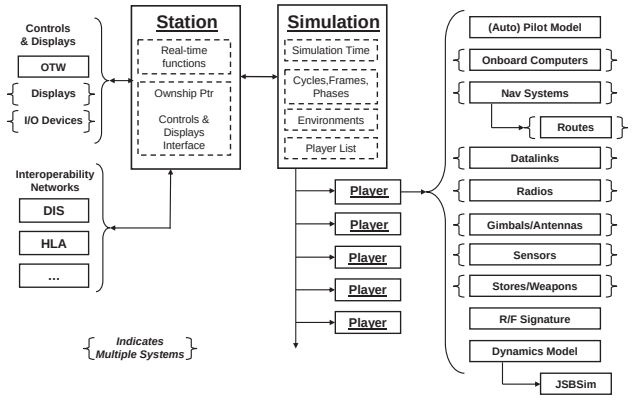


Fig. 3. A graphical depiction of the structure of the OpenEaagles simulation framework.

The framework embraces the Model-View-Controller (MVC) software design pattern by partitioning functional components into packages. As shown, the Station class serves as a view-controller or central connection point that associates simulation of systems (M) with specific views (V) which include graphics, I/O and networks in the case of a distributed simulation.

As a simulation framework, OpenEaagles is not an application itself applications which are stand-alone executable software programs designed to support specific simulation experiments are built leveraging the framework.

Currently, OpenEaagles has a sophisticated autopilot system, but that is the extent of built-in mechanisms for player or entity autonomy. Other than that, no AI exists in the framework for making simulation entities autonomous. Due to UBF's abstract design structure, it was implemented within OpenEaagles as a set of cooperating classes to define agents which can be attached to Players (i.e., entities) to provide more intelligent features than currently available. Within this structure, UBF agents have access to Player state (world model) and all Player systems which are attached as components such as antennas, sensors, weapons, etc. The Players themselves also include a sophisticated autopilot system which can be used to augment and provide low level control functionality.

III. Methods & results

To demonstrate and test our implementation of UBF within OpenEaagles, we defined a sweep mission scenario. Due to differences between a robot platform and a simulation environment, appropriate adjustments had to be made before implementing UBF. After a discussion concerning revision made to UBF, we revisit the sweep mission to discuss details of bringing it to life. Finally, we will discuss the specific UBF behaviors built and utilized by our agents to successfully navigate this defined scenario.

A. Scenario Description

A simple military mission known as a "sweep" was defined to demonstrate and test UBF-based agents. In this mission, a flight of friendly aircraft navigate towards enemy-controlled or contested airspace. The friendly aircraft search for and engage any enemy aircraft encountered, leaving the area upon destruction of the enemies or an emergency condition being met. The mission is split into four phases: Ingress, Beyond Visual Range (BVR) Engagement, Within Visual Range (WVR) Engagement, and Egress. Figure 4 and Figure 5 depict graphically the progression of the typical sweep mission.

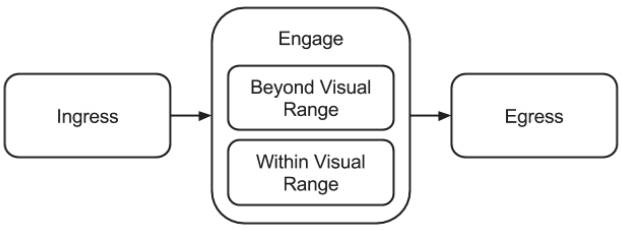


Fig. 4. A graphical depiction of the sweep mission phases.

1) Ingress Phase: The ingress phase of the sweep mission consists of navigating along a set of waypoints to the designated mission area. The flight of friendly aircraft follows the flight lead in formation towards the mission area, watching and evaluating their radar for potential enemy target aircraft. Upon acquiring a target, the friendly aircraft proceed to the engagement phase of the mission.

2) Engagement Phase: Engagement is the mission phase upon which the friendly aircraft launch missiles and fire guns against the enemy targets in attempt to shoot down those aircraft. Engagement is broken into two sub-phases based on the distance to the target.

a) Beyond Visual Range: The beyond visual range (BVR) phase of engagement consists of any combat that occurs when an enemy target is not visible to the friendly pilot through the windscreen, only on radar and signal warning systems. If targets are not detected until they are visible to the friendly pilots, it is possible to skip the BVR phase of engagement. While BVR, the friendly aircraft will confirm that the radar targets are indeed enemy aircraft, and then will engage the target(s) with long-range missiles. If the targets are not destroyed while BVR, and they become visible to the pilots, the within visual range engagement phase is entered.

b) Within Visual Range: Within Visual Range (WVR) combat occurs when enemy aircraft are close enough that the friendly pilots can see them from the cockpit, and not exclusively on radar or signal warning systems. Within visual range combat tends to involve more complicated aircraft maneuvering in order to achieve an advantageous position relative to the enemy aircraft. When an advantageous position is attained, the friendly aircraft may engage the enemy with short range missiles or guns.

3) Egress Phase: The egress phase of the mission occurs after the desired mission objective is completed; namely, if the mission area is clear of enemies. Egress may also be necessary if other emergency conditions are met. If friendly aircraft are low on fuel, or if multiple flight members have been shot down by enemy aircraft, it might be necessary to exit the mission area as quickly as possible. During egress, remaining friendly aircraft proceed to the home airfield, again, sometimes by way of navigation waypoints exiting the mission area, or possibly by the most direct route to base.

B. The UBF Implementation in OpenEaagles

Using the basic structure of UBF as described in section II-B, the architecture was built on top of the object system defined by OpenEaagles. Then, abstract classes defined by the architecture were extended to provide specific functionality (i.e., behaviors, arbiters) relevant to the sweep mission being implemented. Some changes were made to the original UBF structure to tailor it to the OpenEaagles simulation environment, which are described in detail in the following sections. Figure 6 contains a UML diagram of the UBF including the changes that were made in the OpenEaagles implementation.

1) Agent: As discussed in section II-B, UBF within a robot control application provides flexibility between platforms by utilizing different controllers that interface to hardware. Within a simulation environment, hardware is simulated, and can be accessed through the OpenEaagles object system. Therefore, a controller isn't implemented in the same way as it would be on a robot. Also, within OpenEaagles, player entities are built using the composite design pattern; each entity is a composite of many individual components, which each are composites

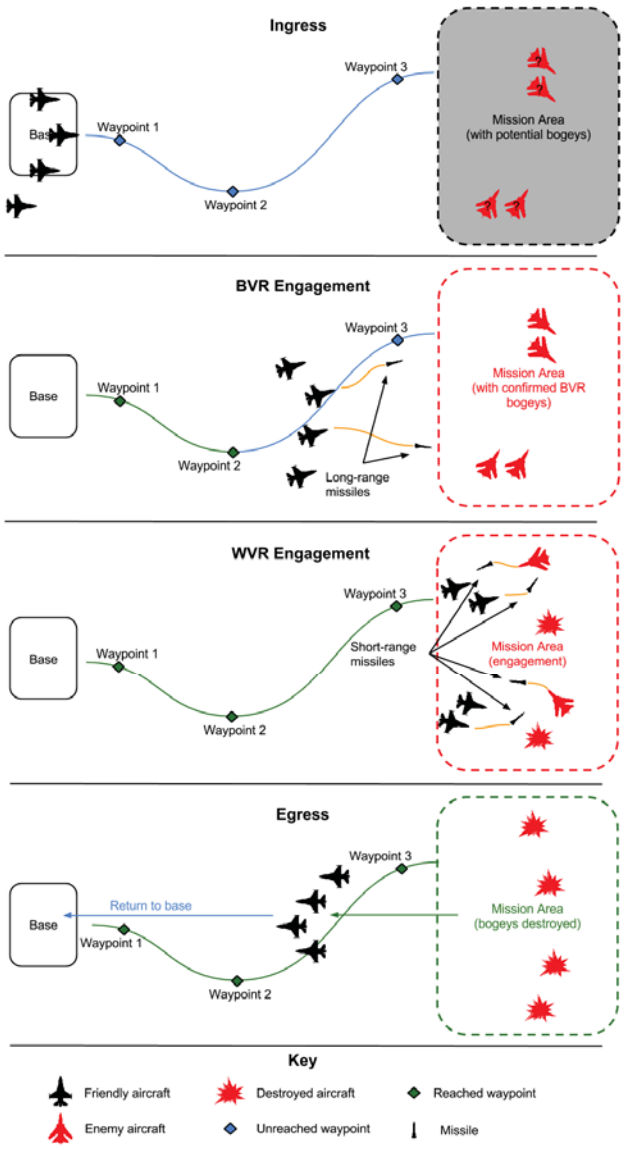


Fig. 5. A birds-eye-view depiction of the sweep mission.

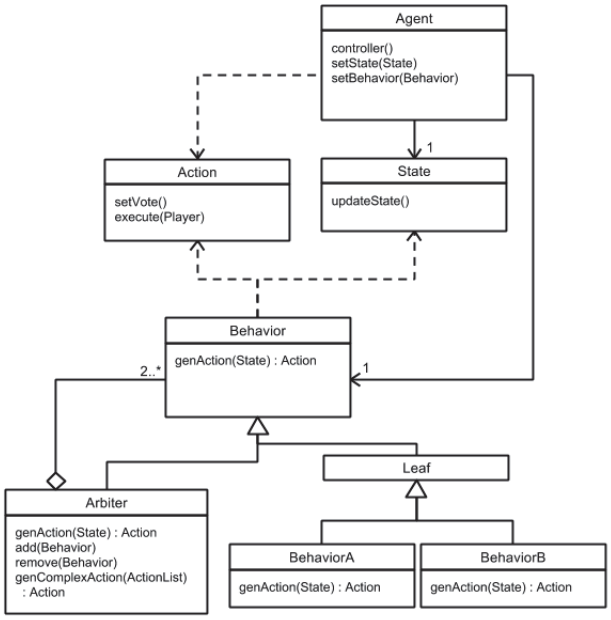


Fig. 6. A UML diagram of the OpenEaagles implementation of the Unified Behavior Framework (UBF).

of their own components. To maintain consistency with this design pattern, UBF needed an overarching component object that contains the whole of the UBF structure. The most effective method was to create an Agent class that contains multiple elements of the UBF, namely, the controller, the root behavior, and the state. This UBF agent can be added - as a component - to an (intended) autonomous player entity in order to add UBF functionality. Through the periodic time phase updates of the Simulation, the agent trickles down requests for updates to the state, and requests for execution of the actions on the autonomous player entity.

2) Controller: Since direct hardware access is not necessary when using a software framework like OpenEaagles, the controller was implemented somewhat differently. Not only does the controller no longer directly update the state of UBF, it is also implemented as a method in the Agent class, rather than its own class separate from actions. This structure retains the decoupling between the UBF behavior tree and the controller, and it enables actions/controllers to be tailored to a specific (simulated) platform. This is more appropriate for simulation: unlike robot architectures, the variety of platforms available in simulation means that different platforms will not only have different control mechanisms, but the actions that can logically be performed between them might be drastically different. For instance, increasing altitude on an aircraft is a logical action for that aircraft, but trying to use that Action on a ground vehicle does not make sense. In this case, it is more appropriate to have different versions of the action class in addition to differing controllers.

By implementing the controller as an action method, the UBF agent's "perceived" state is also no longer tied to the controller, but is separated into its own state class, which will be discussed in detail in the next section.

3) State: As an abstraction, state actually contains no data other than that specific to the OpenEaagles object system. Within individual implementations, state can be populated with world model information that is important to a specific agent. The controller previously contained the updateState() method, as it alone had access to the robot hardware, specifically, the sensors needed to evaluate the state. The OpenEaagles framework, allows for much wider access to the simulation environment details that might be important to a UBF agent. Therefore, updating the state in practice might occur differently than on a robot. An update can occur by evaluating the simulated sensors' input data, emulating the operation of a robot control application. However, software-simulated entities generally have privileged access to true (though simulated)

world state details. In the interest of simplifying a scenario, state was granted this privileged access to the actual simulation state. On the other hand, there is flexibility to implement a more true-to-life state update, one that emulates a robot's state update process, if desired. By separating state into it's own class, rather than relegating it to the controller, the state update can be implementation-defined, adding to the flexibility of UBF in the OpenEaagles simulation environment.

It should also be noted that in OpenEaagles (as in most real-time simulation frameworks), simulation occurs via discrete time steps. Therefore, the State class contains the updateState() method that is tied to the simulation's time-step process, received as requests from the Agent class' controller() method, by which the state is updated as the simulation progresses.

4) Action: As aforementioned, the OpenEaagles implementation of the Action class includes a execute() method that interprets the details of the action and then executes it by "actuating" the relevant controls within the simulated player entity. As is the case with the state, the action/execute() combination allows for more flexibility in the implementation of UBF to specific platforms.

5) Behavior: Behaviors are the smallest functional unit of the UBF, in accordance with the original principles of the subsumption architecture. Behaviors comprise individual tasks that a player entity might perform, which might be as simple as flying straight, or as complicated as following an enemy aircraft. As the UBF's design originally intended, UBF behaviors in OpenEaagles accept the state of the UBF agent's player entity and return an action via the genAction() method. Each internal behavior node also passes the state down the tree to its children, so that every behavior in the tree will receive an updated state every time the UBF tree is polled for an action. Based on the specific behavior involved, each behavior returns a recommended action. Associated with each action is a vote, which indicates the priority of that action as determined by the behavior. A higher value vote indicates a higher priority action. As the returned actions are passed up the tree, arbiters must decide which of the actions (or which combination) will be returned further up the UBF behavior tree.

6) Arbiter: Unlike the original UBF design, arbiters are not a component of internal behavior nodes in the OpenEaagles UBF implementation. Instead, the Arbiter class is subclassed off of the Behavior class, so that the arbiters *are* the internal behavior nodes, though a more specific version of a behavior. In a nutshell, this implementation combines "arbiter" functionality with the composite behavior. This facilitates the selection of actions as behaviors return actions up the UBF tree. Each arbiter, as described, has some decision scheme that selects or constructs the action that is returned up the UBF behavior tree. In the OpenEaagles implementation, the Arbiter class includes a genComplexAction() method, which is the method for returning an action based on the recommendations of its children.

C. Scenario Implementation

1) Reducing the complexity: Complexity is a very relevant issue when trying to build a well-software-engineered product. While any project will become more complex as it grows, the intent is generally to reduce complexity and maintain simplicity. In this case, reducing the complexity of the scenario was necessary to obtain an effective demonstration.

To reduce the complexity, some sacrifices were made with regard to the pilot mental model fidelity. Where pilots might fly specific maneuvers in order to pursue an enemy aircraft, the UBF agent essentially turns on the autopilot and sets it to follow the enemy aircraft. In the same way, the pilots defensive maneuvering is limited to a break maneuver, whereas a human pilot likely has a large repertoire of defensive maneuvers at his/her disposal to defend against an incoming missile or a pursuing enemy. These sacrifices were necessary to successfully implement the desired scenario, but with more work and study on a human pilots decision making, a much more accurate representation of the pilots mind could be obtained with the UBF tree.

In addition to the mental model fidelity, complexity was also reduced with regards to the maneuverability of the aircraft. The OpenEaagles simulation framework includes a very detailed aerodynamics model called JSBSim. In order to create a more manageable implementation, however, this UBF agent utilizes a more simplistic aerodynamics model called the LaeroModel. While the LaeroModel prevents hands-on-stick-and-throttle (HOTAS) control of the aircraft allowing for detailed maneuvers and upside down flight, the simplicity of the model interface greatly reduces the UBF action code required. This was a necessary and acceptable sacrifice in order to implement the sweep mission scenario. As with any simulation, detail is a function of the defined experiment.

2) Scenario Arbiters, State, and Action classes:

a) Arbiters: As mentioned in section II-B3, there are a variety of arbitration schemes available to facilitate decision making in the UBF behavior tree. In our scenario, two separate Arbiters were designed. Unfortunately, due to time constraints, only one was tested and verified with the sweep mission scenario.

b) Winner-takes-all Arbiter: The winner-takes-all (WTA) arbiter simply selects the action with the highest vote. This is the simpler of the two arbiters implemented, not requiring any special manipulation of returned actions. Because of its simplicity, the WTA was the arbiter used in the scenario implementation. This allowed for straightforward construction of the UBF behavior tree and unambiguous confirmation that behaviors were responding as expected.

c) Fusion Arbiter: In addition to the WTA arbiter, a simple fusion arbiter was implemented, but again, it was not tested or verified. Our arbiter takes an extremely basic approach, simply averaging the altitude, velocity, and heading components of each action, and launching a missile if there is a highly-voted action recommending weapons-release.

d) PlaneState: For the OpenEaagles implementation, the PlaneState class was subclassed off of the generic State class. This class contains useful information to an aircraft such as

heading, altitude, velocity, missiles onboard, etc. During the updateState() routine, the PlaneState class polls the simulation to ascertain and populate the PlaneState object. Each of the UBF agents has a state created when the agent is initialized, and the state is not destroyed, rather it is changed as it is updated with the simulation time steps.

e) PlaneAction: The PlaneAction class is the subclass of Action that implements actions for the aircraft agent. Some leniency had to be taken with this class in order to simplify the flying of the aircraft. While specific controls such as the throttle or control column could be actuated to direct the aircraft to the desired vector or location, this is clearly an extremely complicated way of flying the aircraft. Essentially, a UBF agent would require the flying skill of an experienced pilot in order to even perform basic flight maneuvers. As implemented, however, actions are able to use more effective, if less realistic control methods, without requiring an experienced pilot's flying ability. OpenEaagles provides a simplistic aerodynamic model that will "fly" the plane absent of any actual inputs to the simulated controls; only a basic understanding of some elements of flight (altitude, velocity, and heading) is required. In this manner, the PlaneActions controller() method stores the details of the intended action, and modifies the heading, velocity, altitude, launches missiles, engages the autopilot, etc., to allow the aircraft to act according to the agents desire.

Again, the inherent flexibility of this implementation method allows for a future implementation to utilize a more accurate emulation of the original UBF's design, if desired.

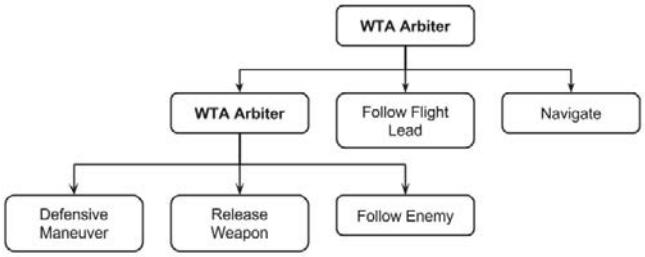


Fig. 7. The UBF behavior tree for the sweep mission scenario.

3) UBF behaviors: In designing the scenario, multiple behaviors were created so that the pilot agents could seek out their sweep mission goal of destroying the enemy aircraft. These behaviors are implemented for the agent's navigation, missile evasion, pursuit of the enemy, and weapons release. They are discussed individually in the sections below. Figure 7 shows the UBF tree structure for the sweep mission scenario.

a) Navigate: In order to successfully complete the mission, the UBF agent needs a way of navigating along a mission path towards the intended mission area. In a real sweep mission, the intended waypoints would be known and planned ahead of mission execution, and the pilot would follow those waypoints until the engagement phase. In this way, the mission waypoints were programmed into the navigation computer of the aircraft before the mission started. The UBF agent turns on the autopilot, instructing it to follow those waypoints, in order to execute the navigation required for the mission.

b) Follow the Lead: Following is a behavior that is necessary for formation flight. While having all of the friendly UBF agents navigating to the same waypoints might imitate this behavior, it does not truly replicate how a pilot would behave, keeping track of the lead aircraft and following his movements. That being said, however, all of the friendly UBF agents in our scenario were given knowledge of the waypoints within their navigation computers. This allowed for an agent to take over as the flight lead if the current one was shot down.

Because a particular formation is specified, the wall formation (shown in figure 8), the UBF agents can use their flight ranking to determine their physical position in relation to the flight lead. In this way, the UBF agent can tell its flight ranking based on the players predefined name assigned when constructing the simulation. To make things simpler, a naming convention of "*(flight name)(rank)*" was used to identify which flight the agent is a part of, and their intended rank in the flight. As rank could change if flight leaders were shot down, there was a mechanism built into PlaneStates updateState() method that determines the actual current ranking, rather than just the original predefined ranking.

To actually follow the flight leader in proper formation, the autopilot was again utilized for the convenient functionality provided within OpenEaagles. The autopilot has a following mode built in, which allows the user to define who to follow, and the position relative to the leader. For instance, in our scenario, "eagle2" followed 6000 feet left, 1000 feet behind, and 500 feet below eagle1.

Utilizing this autopilot functionality, along with the naming convention that defines a flight and its members, the follow behavior was implemented that allows the "eagle" flight (and the enemy "bogey" flight) to fly in wall formation during any non-engagement portions of the scenario.

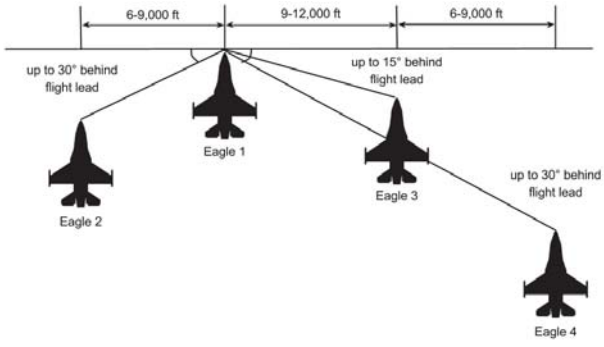


Fig. 8. A graphical depiction of the "wall" flight formation.

c) Pursue an Enemy: Pursuing the enemy is a behavior that is necessary for eventually attacking the enemy, which ultimately is the purpose of the sweep mission. To implement this behavior, again, the autopilot following functionality was utilized. This, while not an accurate representation of how a pilot might maneuver to engage an enemy, does provide a

simple, convenient way to implement the pursuing behavior. This is certainly an area for future improvement, whereas a complex model that is more representative of an actual pilot could be implemented.

In this case, the UBF agent first detects the enemy using its onboard radar systems. After detecting the enemy, the agent is given special access to simulation information about the enemy player in order to provide the data necessary for the autopilot to enter following mode against that enemy.

d) Release Weapon: As it is the ultimate mission of the sweep mission, the UBF agent requires a behavior that decides when to release a weapon against an enemy target. A pilot would normally have some idea of how probable a kill is based on the location of the enemy aircraft in relation to his own aircraft. The term for the region with the highest probability of a kill is a weapons employment zone, or WEZ.

The UBF agent evaluates whether an enemy target is visible (on radar), and then whether that target is within the agents WEZ. If so, the behavior recommends the release of a weapon, which is performed through the stores management system of the UBF agents player.

e) Break (Defensive Maneuver): Finally, a maneuver that attempts to evade incoming missiles is necessary. This behavior detects a missile based on its radar track. As with the pursuit of an enemy aircraft, this behavior could be modified to be more accurate to a true pilots behavior. In the meantime, the detection of the missile is performed within the simulation, which of course has omniscience about whether the radar track is a missile or not.

Once detected, the incoming missile also needs to be determined to be coming at the UBF agent interested in it. As before, the simulation is polled to determine the missiles target. If the target is the current UBF agents player entity, the UBF agent knows that the missile is pursuing it, and can then initiate defensive maneuvering.

In order to be simple, the current defensive maneuver implementation has two phases. The first phase occurs if the missile is detected outside of a two nautical mile radius of the UBF agent. When the missile is far away, as determined by this arbitrary boundary, the UBF agent maneuvers his plane towards the incoming missile, and increases altitude. This is designed as a preparation phase for when the missile is danger close, within the two nautical mile radius. Upon the missile breaching two nautical miles, the UBF agent then performs a break maneuver, or a hard, diving turn (to either side, depending on the angle of the incoming missile).

IV. Assessment

Through the development and implementation of the Unified Behavior Framework within the OpenEaagles simulation framework, we have demonstrated the potential for creating simulated autonomous agents in a military simulation context. Some specific issues that arose during the process were the granularity of behaviors, and the contrast between UBF and behavior trees. In this section, we will briefly discuss these issues as they relate to our implementation.

A. Granularity of behaviors

A difficult design decision presented itself when building the UBF tree of behaviors for our scenario. Behaviors can be as "simple" as performing a basic stick or throttle control change, but they can also be very complex, attempting to attain a specific heading, altitude and velocity by a long series of control input changes. When designing behaviors, it is necessary to make some decisions about how complex, or "granular," the individual UBF behaviors will be. The granularity of the behaviors will also have a direct effect on the size of the UBF behavior tree, and it can affect the arbitration scheme drastically. As the behaviors become simpler and smaller, the UBF tree will grow, and vice versa. WTA arbiters are useful for "large grain" behaviors and small trees, while a fusion arbiter becomes much more interesting as the UBF tree grows and includes "small grain" behaviors that can be fused in interesting ways.

In this case, the design decision was made to allow for very complex, "large grain" behaviors. In this way, the scenario behavior tree remained fairly small in size. This decision was due to the exponential jump in complexity of breaking some tasks down into multiple behaviors. In addition, behaviors that utilized the autopilot navigation and follow modes would have been much more complex if not using the autopilot, and instead building multiple less-complex, autopilot-lacking behaviors. Instead of having a large UBF sub-tree dedicated to navigating to the next waypoint, the UBF agent only required one behavior that turned on the autopilot when navigation was the desired behavior. Though it results in a much more complex exhibited behavior, by choosing this level of granularity, the behavior was actually much simpler to implement.

B. UBF versus Behavior Trees

A question that arose when implementing our sweep mission scenario using UBF was, would the sweep mission be easier to implement with Behavior Trees? The answer, of course, is complicated. When thinking about the sweep mission scenario, the behaviors desired from the pilot agent are well-understood and well-defined. This lends itself to behavior trees, with behaviors that are expected and scripted, rather than unexpected, or "emergent" behaviors. Clearly, the benefits of UBF are lost on such a simple and well-defined scenario. On the other hand, the design elements of UBF lend it to future experimentation within the simulation environment. With the UBF framework in place, the opportunity to simulate pilot agents that exhibit unpredictable behavior is now ripe for exploration. Instead of defining a scenario based on detailed pilot procedures, agents can be designed to behave like we would expect a pilot to in various situations, and then put those agents through their paces to understand how an agent might behave under unpredictable circumstances. While behavior trees would presumably produce consistent behavior, the UBF agents would allow for emergence of behaviors that give deeper insight into agent design.

V. Future Work

One of the major benefits of the Unified Behavior Framework's arbiter scheme, is the opportunity for emergent behavior. Emergent behavior is somewhat of a misnomer; in truth, the actions are emergent. Specific behaviors in the UBF tree are deterministic when considered on their own. When utilizing an arbitration scheme that allows for actions to combine multiple behaviors returned actions, such as fusion, those deterministic responses can now become unpredictable, or emergent. While this may produce odd and possibly detrimental behavior, it also provides for complex combinations of actions that may have been unexpected. By introducing this element of unpredictability and randomness, the capability of the UBF agent grows beyond that of the scripted nature of behavior trees.

A fusion arbiter was developed as part of this effort, but it was not utilized as part of the scenario. Along with increasing the fidelity of the pilot mental model, the fusion arbiter is certainly ready for future work.

VI. Conclusion

Three layer architectures demonstrate the usefulness of separating complex planning algorithms from the reactive control mechanisms needed for rapid action in dynamic environments. In our scenario, these higher-level planning activities were not necessary, as our agents were seeking a very specific goal: destroy any enemies encountered. On the other hand, a real-world pilot would likely come across situations that required a change of goal; an emergency condition or a change of waypoints. While our agents' single-mindedness did not affect the results of the simulation, it demonstrates a lack of capability that could be remedied with the addition of a sequencer to the OpenEaagles UBF agent. In later implementations, adding a sequencer would be an effective way to define planning abilities, so that agents could switch between UBF behavior trees if a goal change was necessary during the middle of a mission.

Acknowledgment

The authors would like to acknowledge the support of David P. Gehl of L-3 Communications for his support in understanding the design and organization of the OpenEaagles framework.

References

[1] B. G. Woolley, G. L. Peterson, and J. T. Kresge, "Real-time behavior-based robot control," *Autonomous Robots*, vol. 30, no. 3, pp. 233–242, 2011.

[2] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, ser. Bradford Books. MIT Press, 1986. [Online]. Available: http://books.google.com/books?id=7KkUAT_q_sQC

[3] N. J. Nilsson, "Shakey the robot," DTIC Document, Tech. Rep., 1984.

[4] E. Gat *et al.*, "On three-layer architectures," 1998.

[5] R. A. Brooks, "A robust layered control system for a mobile robot," *Robotics and Automation, IEEE Journal of*, vol. 2, no. 1, pp. 14–23, 1986.

[6] R. C. Arkin, "Survivable robotic systems: Reactive and homeostatic control," in *Robotics and remote systems for hazardous environments*. Prentice-Hall, Inc., 1993, pp. 135–154.

[7] R. Peter Bonasso, R. James Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack, "Experiences with an architecture for intelligent, reactive agents," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 237–256, 1997.

[8] D. Isla, "Gdc 2005 proceeding: Handling complexity in the halo 2 ai," *Retrieved October*, vol. 21, p. 2009, 2005.

[9] A. Marzinotto, M. Colledanchise, C. Smith, and P. Ogren, "Towards a unified behavior trees framework for robot control," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014.

[10] B. G. Woolley and G. L. Peterson, "Unified behavior framework for reactive robot control," *Journal of Intelligent and Robotic Systems*, vol. 55, no. 2-3, pp. 155–176, 2009.

[11] D. D. Hodson, D. P. Gehl, and R. O. Baldwin, "Building distributed simulations utilizing the eaagles framework," in *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*, vol. 2006, no. 1. NTSA, 2006.

[12] M. Cutumisu and D. Szafron, "An architecture for game behavior ai: Behavior multi-queues." in *AIIDE*, 2009.

[13] A. September, "Ieee standard glossary of software engineering terminology," *Office*, vol. 121990, no. 1, p. 1, 1990.

[14] R. E. Johnson and B. Foote, "Designing reusable classes," *Journal of object-oriented programming*, vol. 1, no. 2, pp. 22–35, 1988.

An Efficient Testing Process for a Quantum Key Distribution System Modeling Framework

Jennifer A. Holes[1], Logan O. Mailloux[1], Michael R. Grimaila[1], Douglas D. Hodson[1]
[1]Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio 45433, United States
[1]{Jennifer.Holes.ctr, Logan.Mailloux, Michael.Grimaila, Douglas.Hodson}@afit.edu

Abstract*—Quantum Key Distribution (QKD) is an innovative technology which exploits the laws of quantum mechanics to generate and distribute unconditionally secure shared keying material for use in cryptographic applications. However, QKD is a nascent technology where real-world systems are constructed from non-ideal components, which can significantly impact system security and performance. In this paper, we discuss a testing process to support a QKD modeling framework built to enable the efficient study of real-world QKD systems. Specifically, we designed a testing process that leveraged existing open source tools to efficiently perform verification testing of QKD optical and electro-optical components. This process provides an efficient means to verify the functionality of an entire library of modeled optical components to expected analytical results based on commercial specifications. Three example test cases of components commonly used in QKD systems are provided to demonstrate the utility of the test process. This research describes a generalizable efficient way to verify and develop modeled behaviors in support of model and simulation frameworks.

***Keywords*—Quantum Key Distribution; Model and Simulation; Verification**

I. Introduction

Quantum Key Distribution (QKD) is an innovative technology which exploits the laws of quantum mechanics to generate and distribute unconditionally secure cryptographic keying material between two parties. QKD has emerged as an important development in the cryptographic security solution space with practical systems being developed by several prominent research groups across Asia, Europe, and North America [1], [2]. Moreover, initial commercial offerings are now available from several venders [3], [4], [5], [6], [7]. However, QKD is a nascent technology with many questions about its claim of "unconditionally secure" key distribution. These apprehensions are seemingly justified as real-world QKD systems are constructed from non-ideal components, which can significantly impact system security and performance. While the Industry Specification Group (ISG) of the European Telecommunications Standards Institute (ETSI) has conducted initial work on developing standards, there has been no globally accepted security certification developed for QKD systems [8], [9]. This work supports an ongoing research effort to model and study the impact of these non-idealities and practical engineering limitations [10], [11].

In this paper, a testing process is described which is used to verify a library of optical and electro-optical models defined within our QKD modeling and simulation framework. Testing is accomplished by verifying correct optical pulse transformations through comparisons of the model results to the expected analytical results. Component verification was conducted for each of the modeled components as described in[11]. The testing process verifies the described primary behavior(s) of each modeled component to commercial specifications. The testing process is more assessable and understandable to a larger group of developers because we leverage Python [12] as opposed to defining tests in C++ to exercise models developed for use within the OMNeT++ simulation framework [13]. We examine three test cases to demonstrate the testing capability, which can be used to verify current, future, and notional optical, electro-optical, and opto-mechanical components required for QKD systems.

The remainder of the paper is organized as follows. In Section II, a brief introduction to QKD is provided along with a description of the QKD simulation framework [11]. In Section III, a thorough description and depiction of the testing methodology is shown. In Section IV, three example test cases are presented to illustrate the utility of the described testing process. Finally, Section V provides conclusions and comments on the application of the rapid testing process towards the design and development of new capabilities within the QKD simulation framework.

II. Background

The genesis of QKD can be traced back to Wiesner, who introduced the idea of quantum conjugate coding in the late 1960s [14]. He proposed two quantum security applications: quantum money and quantum information multiplexing. In 1984, Bennett and

Brassard extended these ideas and designed the first QKD protocol, known as BB84 [15]. The BB84 protocol describes a means for two parties to generate an unconditionally secure shared cryptographic key. The BB84 is depicted in Figure 1, where the sender "Alice" prepares quantum bits (qubits) in one of four polarization states, for example: $|\leftrightarrow\rangle$, $|\updownarrow\rangle$, $|\nearrow\rangle$, or $|\searrow\rangle$. These qubits are encoded according to a randomly selected basis ($\oplus$ for "rectilinear" or $\otimes$ for "diagonal") and bit value (0 or 1). Alice then sends the encoded qubits over a quantum channel, typically an otherwise unused optical fiber or direct-line-of-sight configuration, to the receiver "Bob." A classical channel is also used to facilitate the QKD protocol between Alice and Bob, where Alice generally controls the entire process.

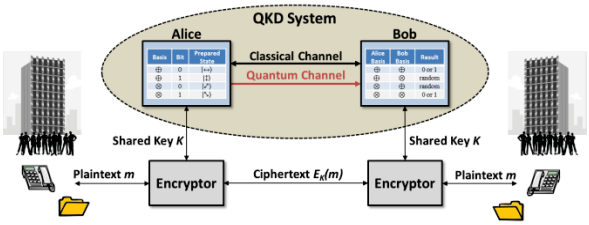


Figure 1. A BB84 polarization-based prepare and measure Quantum Key Distribution (QKD) system.

Bob measures each qubit using a randomly selected basis ($\oplus$ or $\otimes$). If Bob measures the qubit in the matching basis, the encoded message is obtained with a high degree of accuracy. Conversely, if he measures the qubits in the incorrect basis, a random result occurs and the encoded information is destroyed. This unique phenomenon is inherent to quantum communications, where measuring an encoded qubit disturbs its quantum state [16]. The process of preparing, sending, and measuring qubits is known as "quantum exchange" and results in raw keys at both Alice and Bob. After quantum exchange, the QKD system employs classical information theory techniques (error correction, entropy estimation, and privacy amplification) to generate an error free secure shared key [17]. For an accessible description of these processes please see [10], and for comprehensive treatments please see [16], [17].

The QKD-generated shared secret key can be used to enhance the security of conventional symmetric encryption algorithms such as DES, 3DES, or AES through frequent re-keying [18]. Alternatively, the QKD-generated key is often discussed in conjunction with the One-Time Pad (OTP) encryption algorithm to provide an unbreakable cryptographic solution [19], [20]. Despite its great appeal, the OTP is not commonly implemented because of its strict keying requirements – the key must be: 1. as long as the message to be encrypted, 2. truly random, and 3. never re-used [20]. QKD's great appeal is generally found in its ability to meet these keying requirements by generating unlimited amounts of unconditionally secure random key; thus making previously unrealistic OTP secure communications possible.

However, the BB84 protocol (and other QKD protocols) assumes several idealities, which are not valid when building real-world QKD systems [21], [22], [23]. For example: (1) Alice emits perfect single photons; (2) the optical link between Alice and Bob is noisy but lossless; (3) Bob has single photon detectors with perfect efficiency; and (4) the basis alignment between Alice and Bob is perfect. The impact of these non-idealities on system security and performance is not well understood; thus, there exists a need to study these behaviors. The QKD modeling framework, known as "qkdX" and described in [11], is a response to these needs. The qkdX has the ability to analyze the impact of these non-idealities and study the security and performance of QKD systems. It was built in OMNeT++, which provides an infrastructure to develop hardware-focused QKD systems, execute simulations, and analyze results. The qkdX is designed to efficiently model QKD systems with the desired accuracy to answer specific research questions.

As part of the QKD modeling framework, a library of reusable optical component models exists. Each component is modeled with a number of configurable parameters derived from desired operational behaviors and device specification sheets (examples provided in Section IV with complete details provided in the Appendix of [11]). The described testing process supports the rapid verification testing of these component models through analytical comparisons as described in the next section.

III. Testing Methodology

Figure 2 provides a depiction of the proposed testing process; its focus is to verify the primary behaviors of each modeled component to its commercial specification. This is accomplished by ensuring each modeled component correctly performs optical pulse transformations verified against expected analytical "truth data" based on theory combined with commercial specifications and measured data. Verification tests were conducted for each modeled component in the qkdX library (i.e., laser, polarization modulator, electronic variable optical attenuator, fixed optical attenuator, fiber channel, beamsplitter, polarizing beamsplitter, half wave plate, bandpass filter, circulator, in-line polarizer, isolator, optical switch 1x2, polarization controller, and wave division multiplexer).

The testing process accounts for each component in a systematic way, which allows users to quickly and repeatedly test component behaviors over their

operational ranges; it is intended to provide a flexible capability for verification of optical components, where new or modified components can be easily added and tested. The testing process is easily adaptable and allows the user to customize the testing parameters, including optical pulses and operational performance parameters for each component.

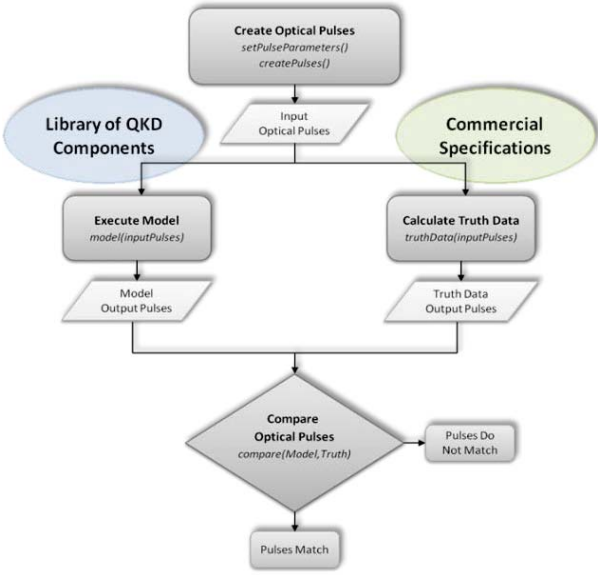


Figure 2. Test Process

The testing process is implemented as a set of Python files and leverages the Simplified Wrapper and Interface Generator (SWIG) to 'wrap' C++-based models for instantiation in Python. The main implementation file *genericdefinitionstemplate.py* contains generic definitions used to test each optical component associated with the qkdX. Specifically, the definitions *setPulseParameters()*, *createPulses()*, and *compareParameters(Truth,Model)* are used to setup and control program flow. Each optical component also has a unique test file containing a device-specific set of definitions. While each component file is distinctive, they are structured in a similar manner. The component specific definitions include *model(inputPulses)*, and *truthData(inputPulses)*. Additionally, definitions common to more than one component were coded in a generic component file in order to reduce redundancy.

Referring to the testing flow chart presented in Figure 1, an input list of optical pulses is first created for input into the modeled component and the matching truth data calculation. Specifically, *setPulseParameters()* defines the optical pulse parameters and the corresponding bounds, while *createPulses()* creates and outputs a linked list of optical pulses. The definition *setPulseParameters()* sets the amplitude, orientation, ellipticity, global phase, central frequency, and duration of each modeled pulse. Each of these parameters has an associated maximum, minimum, number of steps, and step size. Note that the wavelength, peak power, MPN, and pulse energy are derivations of the above parameters. A detailed description of the optical pulse model is available in [24]. The *createPulses()* definition creates optical pulses set to the prescribed parameters, which are appended to a list of "inputPulses".

Next, the inputPulses list is presented to the component model and the truth data calculation. The component model, *model(inputPulses),* executes the modeled optical component. For the truth data, *truthData(inputPulses)* calculates the parameters of interest. The output is separate linked lists of optical pulses. The component model definition *model(inputPulses)* creates the desired modeled optical component, where the test parameters of interest varies over the prescribed operational range(s). The model transforms the input optical pulses through the defined behaviors (i.e., the SWIG provided models) and appended to a model output list. The number and type of output pulses (i.e., output, isolated, nonreflected, reflected) created depends on the component.

The corresponding expected analytical results (i.e., the truth data) is calculated by the definition *truthData(inputPulses)* according to the specified component parameters of interest, the operational range, associated theory, and commercial specifications. The results (i.e., the expected optical pulses) are captured in a "Truth" list, which is then compared to the modeled optical pulses. A comparison of the optical pulse transformations is made by *compareParameters(Model,Truth)*, which compares each optical pulse in the output lists to determine if the pulses match or do not match. Comparisons are made between the model and truth data of each pulse's amplitude, orientation, ellipticity, global phase, central frequency, and duration over the specified operational ranges within a specified epsilon around the expected truth data. This comparison is performed for each transformed pulse in the list, where each device will either pass the suite of tests if the pulses match or fail if the pulses do not match. Additionally, this definition produces an error message to indicate which component and the location of the error when not equal.

The testing process is repeated for each component in the optical component library to accomplish verification of the optical component in the qkdX. This process supports regression testing of current components and facilities verification testing for modified and future components. The testing process also allows for customization of a wide swath of testing parameters inherent to the optical pulse (currently six) and components of interest (ranging from 11 to 21).

IV. Testing Examples

Three examples are provided to demonstrate the applicability of the testing process. In each example, the primary behaviors and parameters of interest were verified. In the first example, an Electrical Variable Optical Attenuator (EVOA) was studied as a simple model with one input, one configuration parameter of interest, and one output. Second, a Polarizing BeamSplitter (PBS) provides a more complex study with one input, multiple parameters of interest, and two outputs. Third, a Single Photon Detector (SPD) is presented as an electro-optical component with an optical pulse input, multiple configuration parameters, and an electrical output.

1.) Electrical Variable Optical Attenuator (EVOA)

The EVOA is an electrically-controlled optical device with one electrical port and two bidirectional optical ports as shown in Figure 4. Note, directional arrows are shown to illustrate the intended direction of travel for the reader. The device's modeled behavior is based on [25, 26, 27, 28]. The EVOA is configured to attenuate optical pulses as they traverse through the device (from the designated input port to the complementary output port). In the qkdX, the EVOA is used to apply a precisely controlled attenuation to optical pulses to meet the system's desired Mean Photon Number (MPN). In QKD systems, the pulse's strength is commonly referred to as by its MPN, which indicates a probabilistic likelihood (i.e., a Poisson Distribution) of a photon existing in each pulse.

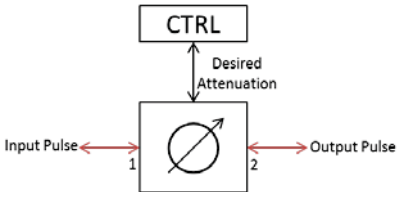


Figure 3. Example Electrical Variable Optical Attenuator (EVOA).

The primary behavior of interest in the EVOA is the attenuation of optical pulses through the component as described in Eq. (1) with the amplitude of the output pulses based upon the device's current attenuation, insertion loss, and the amplitude of the input pulse. The amplitude of the input pulse is based on the system's configuration, while the device's insertion loss is generally fixed and relatively small in comparison to the desired variable attenuation (e.g., 0.5 db compared to 10 dB). The variable attenuation can be adjusted at any time, regardless of the model's state.

$$\text{Amplitude}_{\text{Output}} = \text{Amplitude}_{\text{Input}} * \sqrt{10^{\frac{-(\text{InsertionLoss}+\text{VariableAttenutation})}{10}}} \quad (1)$$

In this example, 1,000 optical pulses were input into the EVOA, each with a MPN of ~8.6, as the EVOA's attenuation increased from 0.0 to 30.0db in steps of 0.1 dB. Simultaneously, a copy of the 1,000 optical pulses was transformed by the truth data calculation using Equation (1). Figure 4 graphs the attenuation of the EVOA versus the MPN of the output optical pulses with results from both the model and the truth data shown. The model and truth data results match precisely, with the MPN of the output optical pulses decreasing from ~8.6 to ~0.009 due to a decrease in the amplitude. Of note, we've also indicated the location of common QKD MPNs of 0.5 and 0.1 at 13 dB and 19 dB, respectively. This result is consistent with the EVOA desired functionality in the qkdX.

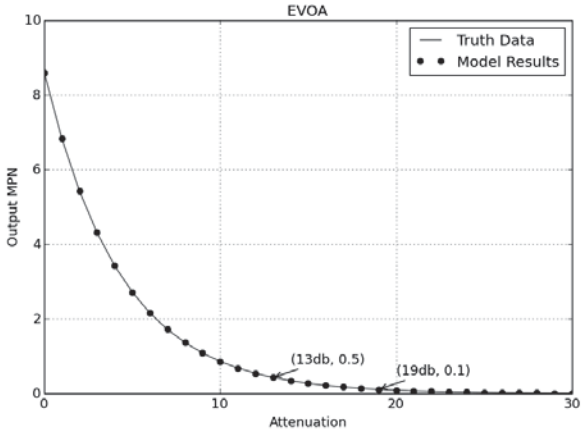


Figure 4. Example Model and Analytical Results.

2.) Polarizing Beamsplitter (PBS)

The PBS is a passive, four port bidirectional device designed to split a beam of light into two orthogonally polarized outputs or to combine two input streams of light into one output stream. Since each optical port of the PBS is a bidirectional device, polarization beam splitting (or combining) may occur at each port effectively creating a four port input/output device. In the qkdX framework, the PBS is often used to split input light into two polarization dependent outputs (i.e., the transmitted polarization is 0 and the reflected polarization is $\pi/2$).

Figure 5 depicts the modeled PBS with directional arrows to illustrate the intended direction of travel for the reader with the fourth optical port being unused.

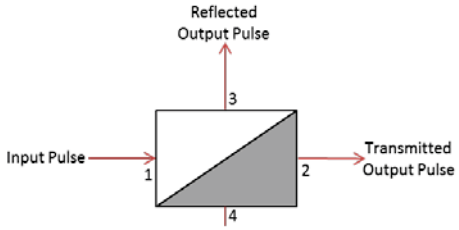


Figure 5. Example Polarizing Beamsplitter (PBS).

The modeled PBS has six performance parameters of interest: insertion loss, excess loss, polarization dependent loss in the x component, polarization dependent loss in the y component, the offset angle, and the extinction ratio. The output pulses are calculated based upon the input optical pulse, polarization dependent loss in the x and y components, excess loss, device offset angle, and extinction ratio as seen in Eqs. (2)-(6). Note that α is the orientation of the input optical pulse and γ is the component offset angle. The modeled behaviors are based on [29, 30, 31, 32, 33].

$$\mathrm{Amp}_{\mathrm{Ouput}} = \sqrt{\left(\mathrm{Amp}_{\mathrm{X}_{\mathrm{Output}}}\right)^2 + \left(\mathrm{Amp}_{\mathrm{Y}_{\mathrm{Output}}}\right)^2} \quad (2)$$

$$\mathrm{Amp}_{\mathrm{X}_{\mathrm{Output}}} = \mathrm{Amp}_{\mathrm{Input}} * \sin(\alpha+\gamma) * \sqrt{10^{\frac{-\mathrm{ExcessLoss}}{10}}} * \sqrt{10^{\frac{-\mathrm{PolarDependentLosses_X}}{10}}} * \sqrt{10^{\frac{-\mathrm{InsertionLoss}}{10}}} \quad (3)$$

$$\mathrm{Amp}_{\mathrm{Y}_{\mathrm{Output}}} = \mathrm{ExtinctionRatio} \quad (4)$$

$$\mathrm{Orientation}_{\mathrm{Output}} = \mathrm{atan2}(\mathrm{Amp}_{\mathrm{Y}}, \mathrm{Amp}_{\mathrm{X}}) \quad (5)$$

$$\mathrm{GlobalPhase}_{\mathrm{Reflected}} = \mathrm{GlobalPhase}_{\mathrm{Input}} + \gamma + \frac{\pi}{2} \quad (6)$$

The function of the PBS is to separate light that enter the component into orthogonally polarized light (i.e., a polarization of 0 or $\pi/2$) . Therefore, the primary behaviors of the component impact the MPN and orientation of the transmitted and reflected output pulses. In this example, five optical pulses are shown input to the PBS with an MPN set to 0.1 and an orientation increasing from 0 to $\pi/2$. Note that insertion loss, excess loss, and polarization dependent losses in the x and y components are set to 0, since loss was already demonstrated in the first example. Simultaneously, these five optical pulses were transformed by the truth data calculations using Equations (2)-(6). As shown in Table 1, the model and truth data results match. For both the modeled and truth data results, the orientation of the transmitted pulses is 0 and the output MPN can be seen decreasing from 0.1 to 0.0. For the corresponding reflected pulses, the orientation is $\pi/2$ and the MPN increases from 0.0 to 0.1. This result is consistent with the current understanding of the PBS and its function based on the above equations.

Table 1. Example Model and Analytical Results.

Input Pulses			**Model Output Pulses**		**Truth Data Output Pulses**	
Index	MPN	Orientation	Transmitted MPN (Orientation = 0)	Reflected MPN (Orientation = $\pi/2$)	Transmitted MPN (Orientation = 0)	Reflected MPN (Orientation = $\pi/2$)
1	0.1	0	0.1	0.0	0.1	0.0
2	0.1	$\pi/8$	0.085	0.015	0.085	0.015
3	0.1	$\pi/4$	0.05	0.05	0.05	0.05
4	0.1	$3\pi/8$	0.015	0.085	0.015	0.085
5	0.1	$\pi/2$	0.0	0.1	0.0	0.1

3.) Single Photon Detector (SPD)

The SPD is an optical-electrical component with one bidirectional optical port, one electrical input configured to "gate" the device when a qubit is expected to arrive, and one electrical output port configured to generate an electrical "click" signal when a photon is detected as shown in Figure 6. Example Single Photon Detector (SPD).. In the qkdX framework, the SPD is modeled as an Indium Gallium Arsenide Avalanche PhotoDiode (APD) with additional control logic (not shown) operating at a wavelength of 1550 *nm* [34]. The APD is configured to detect qubits encoded in one of four polarization states: $|\leftrightarrow\rangle$, $|\updownarrow\rangle$, $|\nearrow\rangle$, or $|\nwarrow\rangle$ in response to the PBS's previously described.

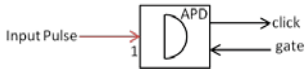


Figure 6. Example Single Photon Detector (SPD).

The detection of single photons, is a complex phenomenon with a number of configuration parameters. Currently, the qkdX employs two SPD models; the simpler version is discussed here. There are three performance parameters for consideration in the simple SPD model: detector efficiency, probability of a dark count, and probability of an afterpulse. APDs have relatively low detection efficiencies of 10-20% at 1550 nm, while dark counts and after pulses represent to erroneous noise in the device.

For the simple SPD model, detection of single photons occurs as follows (for further details see [11]): the device assumed gated during the expected arrive time for each optical pulse. During each gating period, the likelihood of a dark count (error due to stray light) needs to be considered. If a pulse arrives, the number of arriving photons is probabilistically determined according to the MPN of the pulse and the Poisson distribution. If a photon arrives, the detector efficiency needs to be taken into consideration to determine if the photon was successfully detected. For each successful detection, afterpulsing (error due to spontaneous emission within the APD) must also be considered during the next gating period. This behavior is described in Eq. (7)

$$\mathrm{Click} = \mathrm{P}(\text{Pulse with } 1 \geq \text{photons}) * \eta_{\mathrm{Detector}} + \mathrm{P}(\mathrm{DarkCount}) + \mathrm{P}(\mathrm{Afterpulse} \mid \mathrm{Detection}) \quad (7)$$

where a click can be caused by one photon arriving, multiple photons arriving, a dark count, or an afterpulse. For the reader's interest, additional parameters considered in the more complex APD include, bias voltage, temperature, recovery time, jitter time, and quench time. For a detailed description of SPD behaviors and technologies, please see [35, 36].

The desired functionality of the SPD is to detect quantum-level changes in energy; therefore, the primary behavior is whether or not a "click" has occurred. The testing parameters of interest are the MPN, detection efficiency, dark count rate, and afterpulse rate. In this example, four separate test cases were conducted with 1000 pulses each with results presented in Figure 7: SPD Example Model Results.:

1. Ideal test case - the detector efficiency is set to 100%, and the dark count and afterpulsing rates are set to 0. However, since the pulse's MPN is 0.1, only ~9.5% of the pulses contain one or more photons. Therefore, we would expect to detect ~95 of the optical pulses. In this test case, we detected 95 optical pulses.
2. Test case 1 - the detector efficiency is set to 10%, and the dark count and afterpulsing rates are set to 0. We would expect to detect ~9% of the optical pulses. In this test case, we detected 10 optical pulses.
3. Test case 2 - the detector efficiency is set to 10%, the dark count rate is increased to 5e-6, and the afterpulse rate is set to 0. We expect to detect ~9% of the optical pulses, roughly the same as in test case 2. In this test case, we detected 11 optical pulses.
4. Test case 3 - the detector efficiency is set to 10%, the dark count rate is increased to 5e-6, and the afterpulse rate is set to 0.008. We expect to detect ~17% of the optical pulses, a little more than in test case 3. In this test case, we detected 13 optical pulses.

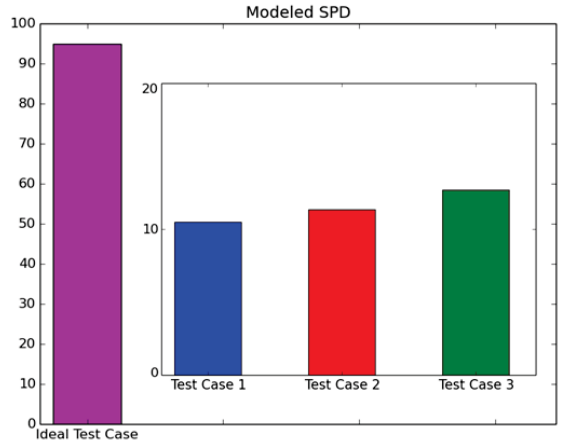


Figure 7: SPD Example Model Results. An inlay is used to appropriately scale the results.

V. Conclusion

QKD is an emerging technology with the ability to generate and distribute unconditionally secure shared cryptographic key. However, real-world QKD systems are built from non-ideal components, which impact system security and performance. A simulation framework provides an efficiency way to study these systems; however, the model(s) must be properly verified, which can be a time consuming effort. In this paper, we presented a rapid testing process built to support the modeling and simulation of QKD systems. Specifically, we discussed and demonstrated the verification of a library of QKD optical components. The testing process offers an efficient means to perform analytical verification of modeled behaviors to commercial specifications.

Additionally, this process lends itself well to the design, implementation, and test of modified or new optical component models. Future work includes verification of the optical component models in a Fock state and the continuation of verification for extended and new optical component models associated with a QKD system.

Acknowledgments

This work was supported by the Laboratory for Telecommunication Sciences [grant number 5743400-304-6448].

Disclaimer

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the U.S. Government.

Bibliography

[1] L. Oesterling, D. Hayford and G. Friend, "Comparison of commercial and next generation quantum key distribution: Technologies for secure communication of information," in *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, 2012.

[2] J. D. Morris, M. R. Grimaila, D. D. Hodson and D. Jacques, "A Survey of Quantum Key Distribution (QKD) Technologies," in *Emerging Trends in ICT Security*, Elsevier, 2013, pp. 141-152.

[3] ID Quantique, "ID Quantique Main Page," 2013. [Online]. Available: http://www.idquantique.com/. [Accessed 1 Nov 2013].

[4] SeQureNet, [Online]. Available: www.sequrenet.com. [Accessed 30 September 2014].

[5] Quintessence Labs, [Online]. Available: www.quintessencelabs.com. [Accessed 30 September 2014].

[6] MagiQ Technologies, [Online]. Available: www.magiqtech.com. [Accessed 30 September 2014].

[7] QuantumCTek, [Online]. Available: http://www.quantum-info.com/en.php. [Accessed 19 March 2015].

[8] T. Länger and G. Lenhar, "Standardization of quantum key distribution and the ETSI standardization initiative ISG-QKD," *New Journal of Physics,* vol. 11, no. 5, p. 055051, 2009.

[9] European Telecommunications Standards Institute, "Quantum Key Distribution," [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/quantum-key-distribution. [Accessed 13 April 2015].

[10] L. O. Mailloux, M. R. Grimaila, D. D. Hodson, G. Baumgartner and C. McLaughlin, "Performance evaluations of quantum key distribution system architectures," *IEEE Security and Privacy,* vol. 13, no. 1, pp. 30-40, 2015.

[11] L. O. Mailloux, J. D. Morris, M. R. Grimaila, D. D. Hodson, D. R. Jacques, J. M. Colombi, C. V. Mclaughlin and J. Holes, "A Modeling Framework for Studying Quantum Key Distribution System Implementation Nonidealities," *Access, IEEE,* vol. 3, pp. 110-130, 2015.

[12] Python Software Foundation, "Python Language Reference," [Online]. Available: https://www.python.org/. [Accessed 11 04 2015].

[13] OMNeT++, "Home page," 03 Mar 2014. [Online]. Available: http://www.omnetpp.org/. [Accessed 08 11 2013].

[14] S. Wiesner, "Conjugate Coding," *ACM SIGACT News,* vol. 15, no. 1.

[15] C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Key Tossing," *Proc. IEEE Int. Conf. Comput., Syst. Signal Process,* pp. 475-480, 1984.

[16] N. Gisin, G. Ribordy, W. Tittel and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics,* vol. 74, no. 1, pp. 145-195, 2002.

[17] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus and M. Peev, "The security of practical quantum key distribution," *Reviews of Modern Physics,* vol. 81, no. 3, pp. 1301-1350, 2009.

[18] ID Quantique, "Cerberis Quantum key Distribution (QKD) Server," 08 Mar 2014. [Online]. Available: http://www.idquantique.com/network-encryption/products/cerberis-quantum-key-distribution.html.

[19] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *American Institute of Electrical Engineers, Transactions of the,* vol. 45, pp. 295-301, 1926.

[20] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal,* vol. 28, pp. 656-715, 1949.

[21] R. Renner, N. Gisin and B. Kraus, An information-theoretic security proof for QKD protocols, arXiv:quant-ph/0502064, 2005.

[22] D. Gottesman, H.-K. Lo, N. Lutkenhaus and J. Preskill, "Security of quantum key distribution with imperfect devices," in *In Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, 2004.

[23] V. Scarani and C. Kurtsiefer, "The black paper of quantum cryptography: real implementation problems," *arXiv:0906.4547v2,* 2009.

[24] L. O. Mailloux, M. R. Grimaila, D. D. Hodson and C. McLaughlin, "Modeling Continuous Time Optical Pulses in a Quantum Key Distribution Discrete Event Simulation," in *International Conference on Security and Management SAM'14*, 2014.

[25] OPLINK, "Electronically Variable Optical Attenuators," 2014. [Online]. Available: http://www.oplink.com/pdf/EVOA-S0012.pdf .

[26] Lightwaves 2020, "Liquid crystal based variable optical attenuation for open-loop architecture," 2014. [Online]. Available: http://www.amstechnologies.com/fileadmin/amsmedia/downloads/1073_VOA-LowTDL.pdf .

[27] OZ Optics, "Fixed attenuators and attenuating fiber patchcord," [Online]. Available: http://www.ozoptics.com/ALLNEW_PDF/DTS0030.pdf. [Accessed 13 06 2014].

[28] OZ Optics, "Electronically controlled variable fiber optic attenuator," 2014. [Online]. Available: http://www.ozoptics.com/ALLNEW_PDF/DTS0010.pdf.

[29] Thor Labs, "Fiber-Based Polarization Beam Combiners / Splitters, 1 SM and 2 PM Ports," 2014. [Online]. Available: http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=6673.

[30] Thor Labs, "NIR bandpas & laser line Filters 700 - 1650 nm center wavelength," 2014. [Online]. Available: http://www.newport.com/Tunable-Bandpass-Fiber-Optic-Filter/835502/1033/info.aspx#tab_orderinfo.

[31] OZ Optics, "Beam splitters/combiners," 2014. [Online]. Available: http://www.ozoptics.com/ALLNEW_PDF/DTS0095.pdf.

[32] Thor Labs, "Variable Polarization Beamsplitter Kit," 2014. [Online]. Available: http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=316.

[33] DPM Photonics, "Specifications," 2014. [Online]. Available: http://www.dpmphotonics.com/product_detail.php?id=170 .

[34] Thor Labs, "InGaAs Avalanche Photodetectors," 2014. [Online]. Available: http://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=4047.

[35] R. H. Hadfield, "Single-photon detectors for optical quantum information applications," *Nature photonics* , vol. 3, no. 12, pp. 696-705, 2009.

[36] M. D. Eisaman, J. Fan, A. Migdall and S. V. Polyakov, "Invited review article: Single-photon sources and detectors," *Review of Scientific Instruments* , vol. 82, no. 7, p. 071101, 2011 .

[37] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus and M. Peev, "The security of practical quantum key distribution," *Reviews of modern physics,* vol. 81, no. 3, p. 1301, 2009.

[38] [Online]. Available: http://www.swig.org/. [Accessed 11 April 2015].

Single and Multi-Station Virtual Simulation Design Patterns

Douglas D. Hodson[1], **David P. Gehl**[2]
[1]Air Force Institute of Technology, WPAFB, OH, USA
[2]L-3 Communications, Link Simulation & Training, Dayton, OH, USA
Email: douglas.hodson@afit.edu, david.gehl@l-3com.com

Abstract— *In a virtual simulation, people and real system hardware interact with a simulated system. Introducing these real-world elements into the simulation environment imposes timing constraints which, from a software standpoint, places the design into the class of a real-time system. Considering these requirements, we present several software design patterns appropriate for the development of single- and multi-station real-time virtual simulations. A variant of the model-view-controller architectural pattern is introduced followed by the development of a supporting component pattern that facilitates the development of single-station hierarchical simulation models, graphical displays, and network input/output (I/O) that need to meet real-time constraints. These patterns are extended to system designs that include multiple PCs with several graphics cards to support the development of multi-station virtual simulations (i.e., simulators that include multiple people or operators).*

Keywords: Distributed Simulation Real-Time

1. Introduction

Virtual simulations involve people or real system hardware interacting with a simulated system. In either case, the software system (the simulation) interfaces and interacts with driving functions (input signals) [1] generated by a person or hardware component and responds by producing outputs. For a typical flight simulator, interaction includes input from stick and throttle devices and output in the form of graphical displays. Software systems designed to meet latency requirements due to these real-world interactions fall into the class of real-time systems. The organization of real-time systems and the quantitative methods used to evaluate particular designs can be found here [2], [3].

In software engineering, a *design pattern* is a general reusable solution to a commonly occurring problem in software design [4]. It is a description or template for how to solve a problem in many different situations. We are particularly interested in tailoring the *Model-View-Controller* (MVC) and the *Component* design patterns to the domain of virtual simulation. The MVC pattern provides a high-level architectural structure of an application and classifies objects according to the roles they play. The *Component* pattern is used as a basis to implement those specific objects.

We incorporate accepted real-time software organization paradigms into these patterns so that quantitative methods can be used to estimate the performance of virtual simulation applications. Incorporated paradigms include the separation of software code into foreground and background tasks while the scheduling of individual jobs (i.e., software code) mimics a fixed cyclic scheduler. The patterns also incorporate hierarchical modeling concepts to define modeled systems.

For each pattern, we assume an implementation that leverages modern object-oriented software techniques. This provides the flexibility to conveniently utilize the concepts of "selective abstraction" and "focused fidelity" to prune object trees, thereby improving system performance.

2. Model-View-Controller (MVC)

In the MVC pattern, there are three types of objects: model objects, view objects, and controller objects. Figure 1 shows the roles these objects play in the application and their lines of communication. When designing an application, choosing or creating custom classes for objects that fall into one of these three groups is a major step since it determines boundaries and communication with other types of objects occurs across those boundaries [5].

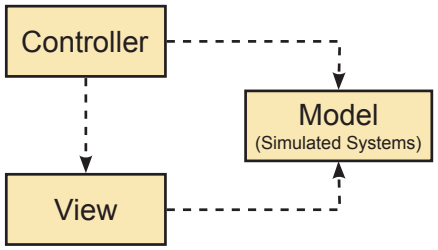


Fig. 1: Model-View-Controller Pattern

For a particular application domain, Model objects represent special knowledge and expertise; they hold an application's data and define the logic that manipulates that data. A well-designed MVC application has all its important data encapsulated in model objects and, ideally, a model object has no explicit connection to the user interface [5].

For a virtual simulation application, the model object is the simulation itself. It contains all simulation state data, behaviors in the form of hierarchical system models, and manages time advancement. The model object as defined in the MVC pattern should not be confused with simulated system models.

A view object knows how to display or present data to an external viewer. The view is not responsible for storing the data it is displaying and comes in many different varieties. For a virtual simulation, the view includes the drawing of graphical displays such as GUI interfaces and interactive operator displays. In the case of distributed virtual simulations (DVS) or Live-Virtual-Constructive (LVC) simulations, a view is responsible for sharing simulation state data across a network to other interconnected simulation applications.

The controller object acts as the intermediary between the application's view objects and its model objects. Ideally, it is the sole source of user inputs and connects the simulation to its graphical displays. Practically, one often merges the roles played by an object. For example, an object that fulfills the roles of both controller and view is called a "view-controller" [5].

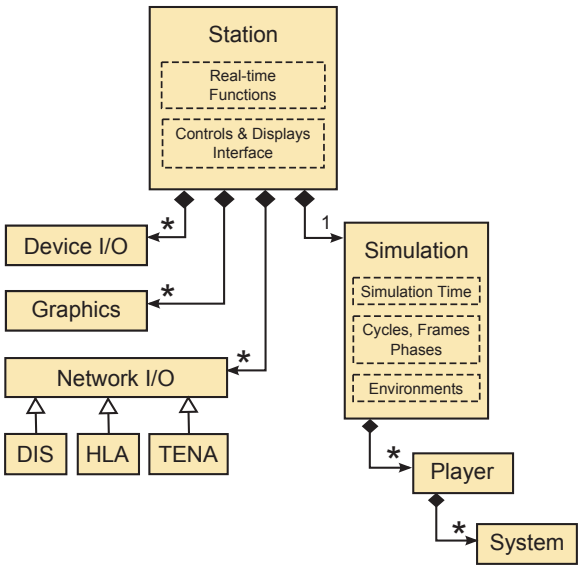


Fig. 2: Simulation Design Pattern

A view-controller is view layer centric. It "owns" the views, while still managing the interface and communications with the model. Combining roles like this is common [5] and is reflected in our tailored MVC design pattern for virtual simulations as shown in Figure 2.

The simulation pattern in Figure 2 consists of a top-level "Station" object containing one simulation object (i.e., the model in the MVC pattern) and multiple view-controllers. We call this top-level object a Station to reflect its close association between the management of I/O functions and visual displays which is what real operators interact with. The Station object also manages high-level functions for creating the threads associated with each of the view-controllers, as needed.

The simulation object consists of a list of players which are assembled as a set of hierarchical-based system models consisting of systems with sub-systems. The simulation object manages simulation time and provides features needed to implement a fixed cyclic scheduler. The view-controllers consist of handlers that read and/or write to I/O devices, interactive graphical displays and interoperability network interfaces. The network interoperability interface for sharing simulation state data supports concrete implementations of a variety of standards such as the Distributed Interactive Simulation (DIS) [6], the High-Level Architecture (HLA) [7], and the Test and Training Enabling Architecture (TENA) [8] specifications.

2.1 Multi-Threading

Ideally, the execution of a simulation application based upon the MVC simulation pattern would consist of a loop that would sequentially read inputs, execute the system models, and generate outputs (i.e., update graphics and process network activities) once per frame. This is an acceptable execution strategy for constructive simulations, where the requirement to execute in real-time (i.e., in sync with wallclock time) is often relaxed since everything is simulated by models. But a virtual simulation that performs all of these tasks in real-time, however, is limited by processing power, So this approach becomes problematic as frame rates increase, thereby reducing the amount of time to complete all tasks.

To resolve this fundamental problem, the processing time associated with input devices, graphic display(s) and interoperability network management functions (i.e., the view-controllers) can be partitioned into separate periodic tasks, each executed asynchronously with respect to each other, at particular frequencies. For example, the update rate associated with graphical displays might be much less than the rate at which the simulation advances time. Furthermore, the division of software code into foreground and background tasks reduces the workload associated with processing time-critical tasks. The challenge is to organize software code to promote this separation of work. This is one of the central goals of our tailored *Component* design pattern.

3. Component Pattern

The simulation pattern, as shown in Figure 2, is the first step towards separating a virtual simulation application into high-level objects that can be executed independently. Further improvement can be made by partitioning the real-time and non-real-time jobs[1] defined by those independent objects (i.e., the simulation and view-controller objects) into foreground and background tasks. We introduce a *Component* design pattern which facilitates this separation while simultaneously supporting hierarchical modeling.

[1]For an object-oriented system, a "job" is a code fragment to be executed (e.g. a function or method call).

As most systems selected for simulation-based analysis are complex [9], and because managing the complexity of models is a challenging task, large systems are seldom modeled in a monolithic fashion. In fact, they are usually divided into smaller, interacting subsystems. The subsystems themselves are further divided into smaller sub-subsystems until a manageable level of complexity is reached. In other words, the *system under study* can be viewed as a "system of systems". This subdivision of a system results in a hierarchical organization of the *system under study* itself.

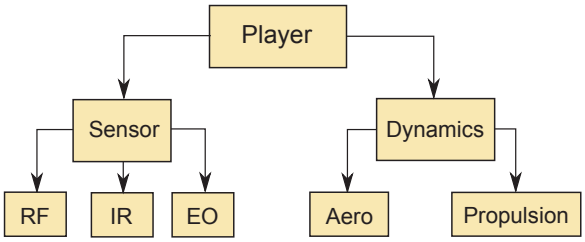


Fig. 3: Hierarchical Player Model

An example hierarchy is shown in Figure 3, where the top level model is a "player" or "entity" within the simulation. The player is composed of both a dynamics and a sensor model. The sensor model is a composite of several sensors, namely, radio frequency (RF), infrared (IR), and electro-optical (EO) models. Dynamics is composed of an aerodynamics and propulsion model.

Hierarchical models from a software engineering point of view are software "components." Conceptually, a component is an entity, a real-world object, that is viewed as a "black box." Its interface, behavior, and functionality are visible but its implementation is not [10]. Components naturally map to object-oriented implementation paradigms supported by languages such as C++.

Gamma [4] contains a catalog of commonly used design patterns in software development and provides solutions developed and evolved over time. *Structural* design patterns provide classes and objects that form larger structures. Of particular interest for hierarchical modeling is the *composite* pattern in Figure 4 which implements hierarchical models in object-oriented programming languages.

The composite pattern uses a tree structure where components can have *children*, i.e., subsystems and sub-subsystems. The *Component* class declares the interface for objects in the composition and implements default behaviors for all the classes. The *Leaf* class has no children while the *Composite* class defines behavior for components that have children. The *operation* method is a placeholder for the functionality of the model. Using this structure, modeled systems can be divided into sub-systems and defined as *Components* through inheritance.

When implementing a composite pattern there are trade-offs related to software design safety and transparency.

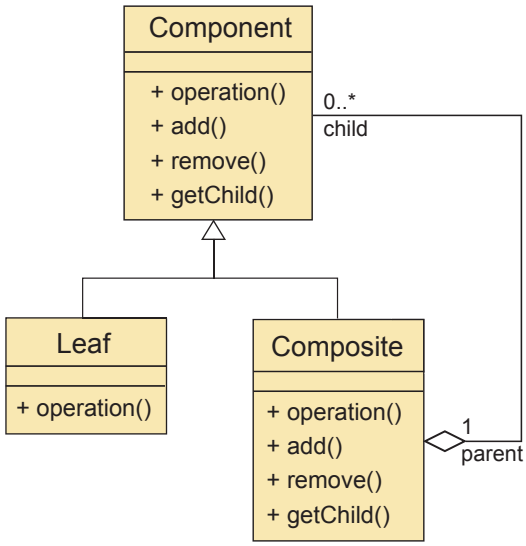


Fig. 4: Structural Composite Pattern

Gamma provides an extensive discussion that considers several implementation approaches. For example, the component class declares the *add* and *remove* methods to provide a transparent interface for all components, but these do not make sense for a leaf. We consider these trade-offs as we adapt this pattern to the domain of system modeling and real-time processing.

The hierarchical-based approach addresses model complexity, but does not address the temporal performance of code execution, specifically, the reliable completion of jobs at or before their deadline. We recommend partitioning of code into real-time foreground and non-real-time background tasks.

Given hierarchy models with the structural composite patterns shown in Figure 4, software partitioning can be incorporated by replacing the single *operation* method by two methods, *updateTC*, and *updateData* as shown in Figure 5. The *updateTC* method (where TC means time critical) is a placeholder to implement a real-time task which includes calculations associated with updating model state space. Less time-critical jobs, such as saving or logging data to a hard drive is placed within the updateData method.

We add and explicitly pass the simulation step-size (sometimes referred to as *delta-time*) as a parameter. Step-size is used by mathematical calculations associated with system models. Since *updateTC* automatically calls all of its children's *updateTC* methods, executing a complete hierarchical model (implemented as a component tree) occurs with a single method call to the root component.

Our component design pattern considers all components to be composites. In other words, when modeling systems, sub-system, and sub-sub systems, there are no leaves, as each

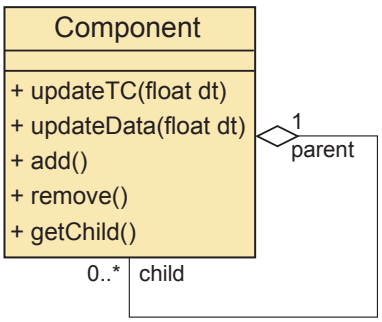


Fig. 5: Component With Partitioned Code

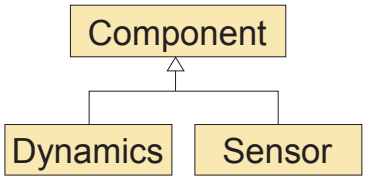


Fig. 6: Example Component Models

model is an abstraction at some level.

Consider, for example, the player model in Figure 3. To implement this system, several models are created by subclassing from the *Component* class as shown in Figure 6. Component models whose functionality is described by a set of differential equations might include a numerical solver in the *updateTC* method. Other background, less time critical jobs, such as saving vehicle position data at each simulation step for analysis, is in the *updateData* method. After each component model is built, the complete flight control system is assembled into a component tree that is the complete modeled system. Subsequent execution or simulation of the modeled system occurs by calling the updateTC method of the root component.

3.1 Scheduling Code Execution

Designing a software system to meet temporal requirements is a scheduling problem. More formally, to meet a program's temporal requirements in real-time systems, a strategy is needed for ordering the use of system resources [3]. This strategy results in a schedule for executing jobs. We are particularly interested in how to schedule jobs to maintain a consistent simulation state space.

To accomplish this, we design a cyclic scheduler which specifies when jobs are executed. The schedule is static, which may not be optimal, but is highly predictable and simple to implement. A cyclic scheduler makes decisions periodically. The time interval between scheduling decision points are called frames (F). Scheduling decisions are made at the beginning of every frame and there is no preemption within a frame.

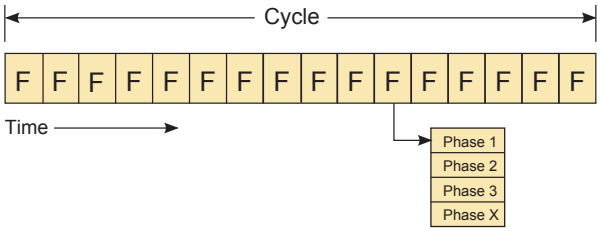


Fig. 7: Cyclic Scheduler

A notional structure for a scheduler is shown in Figure 7. Frames are grouped into a "cycle," and subdivided into an arbitrary number of *phases*. Frames are divided into phases to resolve data and control dependencies among jobs and specify an execution order.

Adding features to support static scheduling in our *Component* class is as simple as adding attributes, specifically, cycle, frame and phase attributes in the form of class variables as shown in Figure 8. Subclasses of *Component* can be built that not only partition model code (i.e., jobs) for execution in the foreground and background, but explicitly define which frames and phases jobs should be processed. Providing direct access to scheduling attributes allows the developer to design a model or set of models that balances execution load across frames.

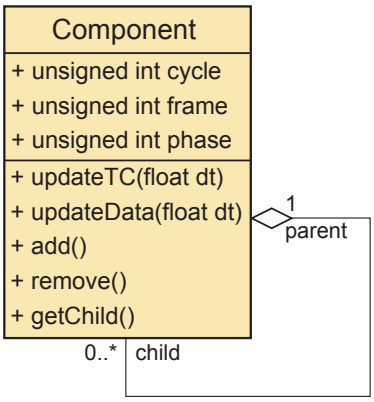


Fig. 8: Component with Scheduling Support

Consider a system model derived from the Component class with the *updateTC* method coded below:

```
updateTC(dt) {
  switch (phase) {
    case 0:
      // update position
      break;
    case 1:
      // process radar interactions
      break;
  }
}
```

The phase attribute is used to impose an execution order within each frame for modeling systems. Conditional code before the switch statement can be inserted to limit processing to selected frames within a cycle. A very common technique conditionally selects a single frame, all even or odd frames, or all frames within a cycle for execution. The parameter "dt" (delta time) is the simulation time advance step-size and is passed to the *updateTC* method and made available for system model calculations.

3.2 Modeling a Player

Consider a player or entity defined by an object tree specified by the set of Components instances $\{C_1, C_2, C_3, ..., C_k\}$. The cyclic scheduler for the object tree has p phases, and f frames per cycle. The maximum execution time for the task defined by this single hierarchical system model can be determined by computing the execution time in each frame,

$$e_f = \sum_{comp}^{c} \sum_{phase}^{p} e_{f,c,p}, \tag{1}$$

where comp is the set of components, followed by selecting the maximum frame execution time in the cycle,

$$e_{Player} = \max_{1 \le f \le cycle} \{e_f\}. \tag{2}$$

For a virtual simulation, this is the execution time of a single instance of a player managed by the simulation object shown in Figure 2. Specific application of rate monotonic quantitative methods [11] for a single PC architecture can be found here [12].

3.3 Graphics and Input/Output

To support unique features of view-controller objects, specialized *Components* can be created with additional methods. For example, just as the single *operation* method in *Component* was replaced with *updateTC* and *updateData* to partition jobs, additional methods can be added to support the execution of specific jobs unique to a particular view-controller. Effectively, each new method defines an independent execution path through a hierarchical system model or object tree.

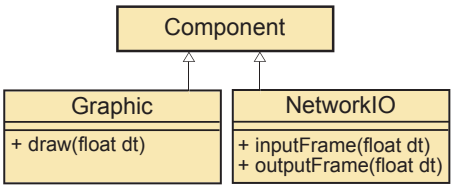


Fig. 9: Graphic and Network Classes

As shown in Figure 9, specialized *Component* classes to support graphics and interoperability networks are defined. Analogous to the *updateTC* method provided by *Component*, the *Graphic* class provides a *draw* method for specifying graphic operations. In a similar vein, the *NetworkIO* class provides two methods for receiving and transmitting state data across a network, *inputFrame* and *outputFrame*. The *NetworkIO* class also serves as an abstract interface to support a wide range of interoperability protocols providing a clear separation between models and specific interoperability implementations.

Since the *Graphic* and *NetworkIO* classes are specialized *Components*, they can use *updateTC* for real-time model execution and *updateData* for background processing. For example, *Graphic*-based components can use *updateTC*, graphic operations in draw, and non-real-time background processing in *updateData*. Strictly speaking, this violates the spirit of the MVC pattern as the model would be closely coupled with the view and controller, but is acceptable to meet temporal constraints.

4. System Abstraction

Implementing hierarchical, component-based models using the *Component* design pattern efficiently implements "selective abstraction." Selective abstraction [13] reduces the complexity of models by identifying and discarding details of the model which have minimal impact on the overall results. This allows the developer to *prune* the object tree at selected points to reduce the level of complexity to improve runtime performance.

Another approach starts with highly abstract system representations and adds fidelity as needed. We introduce the term "focused fidelity" to capture this concept. Focused fidelity provides the appropriate level of detail (resolution) to the *system under study* to provide the required accuracy while eliminating undesirable system inputs. This is important because complex models that are not directly under study can affect independent variables, which are inputs into the *system under study*, and can therefore confound the study results. Additionally, it is inefficient and often counter-productive to develop more complex models than needed for the simulation.

Our *Component* class provides the means to implement *selective abstraction* and *focused fidelity* concepts. Applying them reduces simulation development time and cost, while simultaneously improving runtime performance and validity of simulation results.

5. Multi-Station Simulator Patterns

The simulation design pattern presented in Figure 2 is most frequently used to develop single-station simulators (i.e., simulators that contain a single person with a limited set of graphical displays). These simulators consist of a single model (i.e., the simulation) and multiple view-controllers executing on the same PC. As organized, this approach takes

full advantage of multi-core, multi-CPU PC-based architectures. If multiple single-station simulators are assembled to create a DVS or an LVC, each application creates their own simulation object and shares state data through a common interoperability interface.

We now consider multi-station simulators which often consist of many more graphical displays than a single PC or a single graphics card can drive. This is a common situation for simulators that include multiple people each with their own set of interactive graphical displays. This case is different than the organization of a standard DVS or LVC because there is a single simulation object.

The design considered consists of a single simulation "executive" with scalable well-defined interfaces to support the data distribution and change request requirements from multiple PC's that are responsible for providing their own set of interactive graphical displays. To support this requirement, we extend the simulation pattern through the introduction of a *Interface* class which provides another level of indirection between station graphics (i.e., view-controller(s)) and the simulation object (i.e., model).

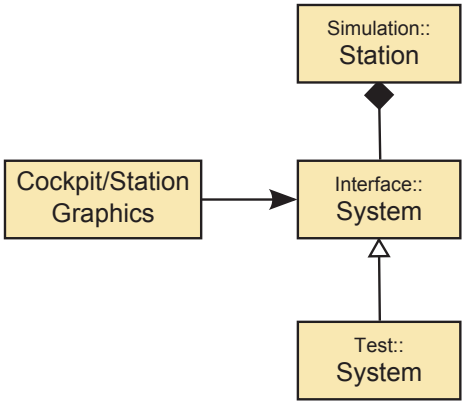


Fig. 10: Single PC Test Appartus

As an example, Figure 10 shows an apparatus that can be used to fully test graphical displays independent of the simulation. As shown, a graphical display is associated with an interface object (Interface::System) which is responsible for setting and getting data associated with a particular modeled system. A specialized version of this interface (i.e., Test::System) serves as a surrogate simulation for testing purposes.

Figures 11, 12 and 13, graphically depicts the design patterns associated with the logical development of a multiple PC, multi-station simulator. Figure 11 provides a different application of the *Interface* class on a single PC by associating the specialized version of the interface (Local::System) with a one of the simulated Player system models. Figure 12 effectively implements the test apparatus shown in Figure 10 across multiple PCs. To support this, a server

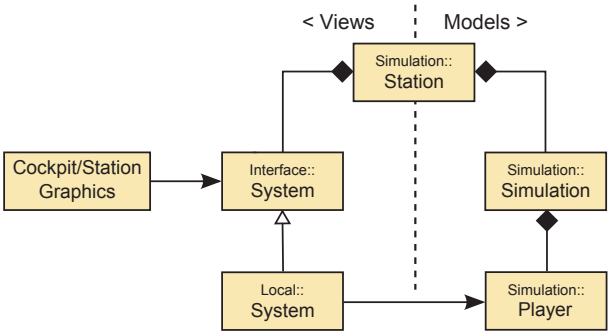


Fig. 11: Single PC with Simulation Object

(Server::System) interacts through a network infrastructure with a client (Client::System) to provide data distribution and change request services. Finally, Figure 13 shows the implementation of a full system design that includes multiple computers – a single "simulation executive" connected to multiple PCs each drawing interactive graphical displays.

6. Final Thoughts

The patterns presented have not been developed in isolation. In fact, they have been carefully crafted and used for many years by simulation engineers. They are heavily used by the open-source OPENEAAGLES [14] framework. Embracing these design patterns promotes good software designs that consider real-time requirements and leverage multiple PCs to develop multi-station simulators.

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government.

References

[1] F. E. Cellier and E. Kofman, *Continuous System Simulation*. New York, NY: Springer, 2006.
[2] J. W. S. Liu, *Real-Time Systems*. Upper Saddle River, NJ: Prentice Hall, 2000.
[3] P. A. Laplante, *Real-Time Systems Design, 3rd Edition*. Piscataway, NJ: IEEE Press/Wiley Interscience, 2004.
[4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Design*. Upper Saddle River, NJ: Addison-Wesley, 1995.
[5] Apple, "Cocoa Fundamentals Guide," 2007.
[6] *IEEE Std for Distributed Interactive Simulation, Standard 1278*, 1998.
[7] *IEEE Std for High Level Architecture, Standard 1516*, 2000.
[8] *DoD Foundation Initiative 2010: The Test and Training Enabling Architecture – Architecture Reference Document*, 2002, `https://www.tena-sda.org`.
[9] D. M. Rao, "A study of dynamic component substitutions," Ph.D. Dissertation, University of Cincinnati, 2003.
[10] D. M. Rao, D. D. Hodson, M. S. Jr, C. B. Johnson, P. Kidambi, and S. Narayanan, *Design & Implementation of Virtual and Constructive Simulations Using OpenEaagles*. Linus Publications, 2009.
[11] C. L. Lui and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 46–61, 1973.

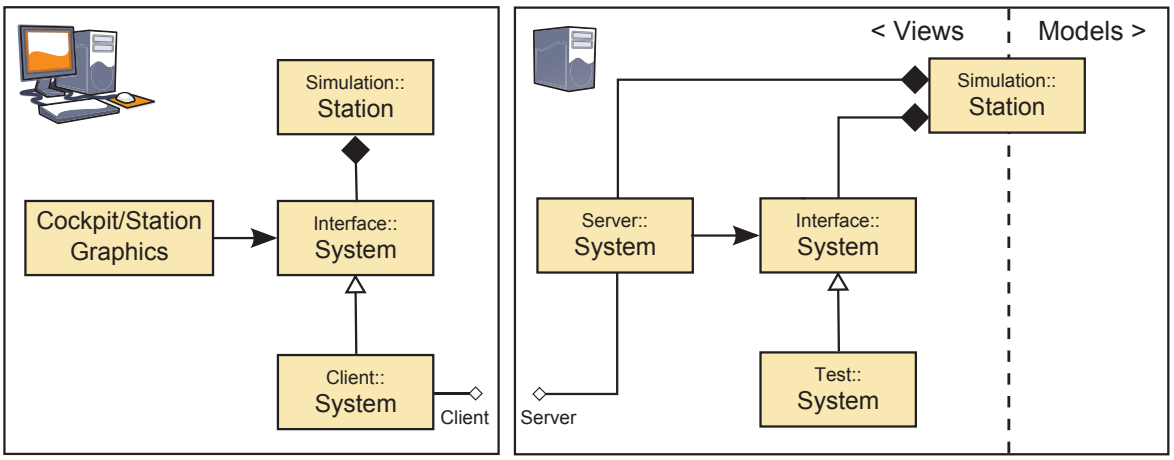


Fig. 12: Multiple PCs Test Apparatus

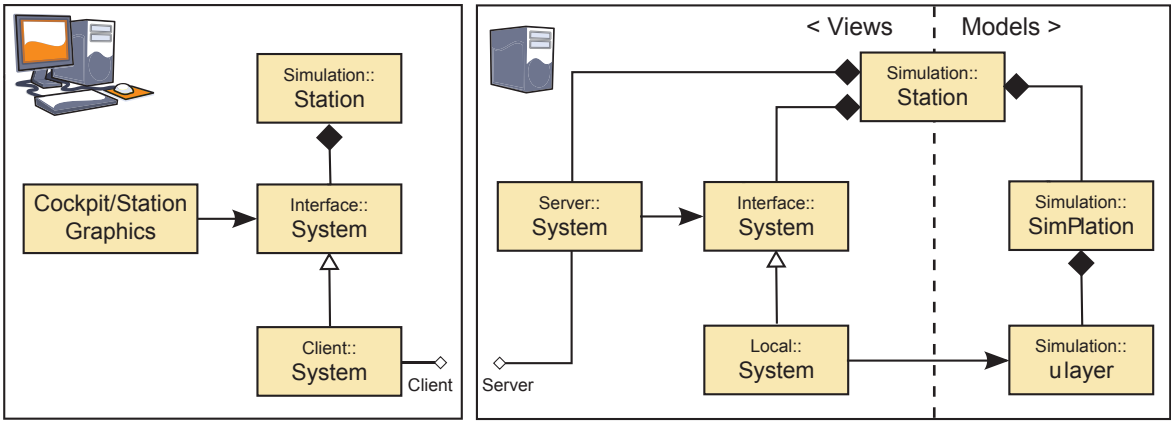


Fig. 13: Multiple PCs with Simulation Executive

[12] D. Hodson, R. Baldwin, D. Gehl, J. Weber, and S. Narayanan, "Real-time design patterns in virtual simulations," *International Journal of Modelling and Simulation, In Press*, 2009.

[13] A. F. Sisti and S. D. Farr, "Model abstraction techniques: An intuitive overview," *Aerospace and Electronics Conference, NAECON, IEEE National*, 1998.

[14] OpenEaagles, "The OpenEaagles simulation framework." June 2009, `http://www.openeaagles.org`.

[15] R. M. Fujimoto, *Parallel and Distributed Simulation Systems*. New York, NY: Wiley-Interscience, 2000.

[16] S. Ghosh and T. S. Lee, *Modeling and Asychronous Distributed Simulation – Analyzing Complex Systems*. IEEE Press, 2000.

[17] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis, Third Edition*. New York, NY: McGraw Hill, 2000.

[18] S. Narayanan, N. L. Schneider, C. Patel, T. M. Carrico, J. DiPasquale, and N. Reddy, "An object-based architecture for developing interactive simulations using Java," *Simulation*, vol. 69, no. 3, pp. 153–171, 1997.

Advanced Framework for Simulation, Integration and Modeling (AFSIM)

(Case Number: 88ABW-2015-2258)

Peter D Clive[*], Jeffrey A Johnson[*], Michael J Moss[*], James M Zeh[†], Brian M Birkmire[†], and Douglas D Hodson[‡]
[*]The Boeing Company
St. Louis, MO
[†]Aerospace Systems Directorate
Air Force Research Laboratory
Wright Patterson Air Force Base, OH
[‡]Computer Science and Engineering Department
Air Force Institute of Technology, USA

***Abstract*— The Advanced Framework for Simulation, Integration and Modeling (AFSIM) is an engagement and mission level simulation environment written in C++ originally developed by Boeing and now managed by the Air Force Research Laboratory (AFRL). AFSIM was developed to address analysis capability shortcomings in existing legacy simulation environments as well as to provide an environment built with more modern programming paradigms in mind. AFSIM can simulate missions from subsurface to space and across multiple levels of model fidelity. The AFSIM environment consists of three pieces of software: the framework itself which provides the backbone for defining platforms and interactions, an integrated development environment (IDE) for scenario creation and scripting, and a visualization tool called VESPA. AFSIM also provides a flexible and easy to use agent modeling architecture which utilizes behavior trees and hierarchical tasking called the Reactive Integrated Planning aRchitecture (RIPR). AFSIM is currently ITAR restricted and AFRL only distributes AFSIM within the DoD community. However, work is under way to modify the base architecture facilitating the maintenance of AFSIM versions across multiple levels of releasability.**

***Index Terms*— Simulation Framework, Mission Level Model, Artificial Intelligence Framework, Agent Framework**

I. Introduction

AFSIM is a government-approved C++ simulation framework for use in constructing engagement and mission-level analytic simulations for the Operations Analysis community, as well as virtual experimentation. The primary goal of AFSIM applications is the assessment of new system concepts and designs with advanced capabilities not easily assessed within traditional engagement and mission level simulations. Development activities include modeling weapon kinematics, sensor systems, electronic warfare systems, communication networks, advanced tracking, correlation, and fusion algorithms, and automated tactics and battle management software.

In this section, the reasons for the development and history of AFSIM are presented. The next section provides an overview of the AFSIM architecture, integrated development environment, visualization tools and AFSIM's agent modeling architecture. The following section highlights the current/planned effort to create a Component Based Architecture for AFSIM which will allow multiple levels of releasability. The last section provides a conclusion on AFSIM and its current state.

A. Background

AFSIM is based on The Boeing Company's Analytic Framework for Network-Enabled Systems (AFNES). Under contract, Boeing delivered AFNES to the Air Force (specifically AFRL/RQQD) with unlimited rights, including source code, in February 2013. AFRL/RQQD rebranded AFNES as AFSIM and has begun to distribute AFSIM within the Air Force and DoD, including DoD contractors.

The Boeing Company developed and funded the AFNES simulation framework through internal research and development (IR&D) funding from 2003-2014. Beginning in 2005, Boeing began developing a customized AFNES capability to simulate threat Integrated Air Defense Systems (IADS) to assess advanced air vehicle concepts performing Precision Engagement missions. The requirements of this new IADS simulation capability included being able to match results with the Air Force-approved mission level model. The reason for developing an AFNES alternative to the Air Force IADS modeling capability relates to the limitations associated with the Air Force mission level model. Examples of areas in which the Air Force mission level model is lacking include: expansion of representations of Electronic Warfare (EW) techniques; the integration of independent tracking and correlation systems; utilization of vendor-supplied auto-routers and mission optimization capabilities; net-centric communications systems; the contribution of Space assets; and integration of special, existing models, such as AGI's System Tool Kit (STK).

The AFNES IADS capability became operational in 2008, and is currently being utilized by multiple Boeing development programs, as well as government contracted programs, to assess the ability of advanced air vehicle design concepts to

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

penetrate advanced Air Defense networks and conduct precision engagement missions. In 2010, the AFRL/RQQD Aerospace Vehicles Technology Assessment & Simulation (AVTAS) Lab (formerly AFRL/RBCD) commissioned a trade study of M&S Frameworks for the purpose of assessing potential alternatives to replace or augment their current constructive simulation environment. The result of the AFRL trade study was the selection of AFNES as the best M&S framework to meet their air vehicle mission effectiveness analysis requirements.

II. AFSIM Software Suite

The AFSIM software suite consists of three distinct pieces or applications. The first piece is the framework itself which provides the underlying architecture and services allowing the creation of simulation applications. The second piece is the integrated development environment (IDE) which facilitates the creation of scenarios. Lastly the Visualization Environment for Scenario, Preparation and Analysis (VESPA) application allows for post-processing and visualization of scenario executions. This section provides detail on all three.

A. Functional Architecture

AFSIM is an object-oriented, C++ simulation environment that facilitates the prototyping of customized engagement and mission level warfare simulations. AFSIM includes a set of software libraries, shown as a functional architecture in Figure 1, containing routines commonly used to create analytic applications. The AFSIM infrastructure includes routines for the top-level control and management of the simulation; management of time and events within the simulation; management of terrain databases; general purpose math and coordinate transformation utilities; and support of standard simulation interfaces, such as those supporting the Distributed Interactive Simulation (DIS) protocol. The AFSIM component software routines support the definition of entities (platforms) to populate scenarios. These software routines contain models for a variety of user-defined movers, sensors, weapons, processors for defining system behavior and information flow, communications and track management.

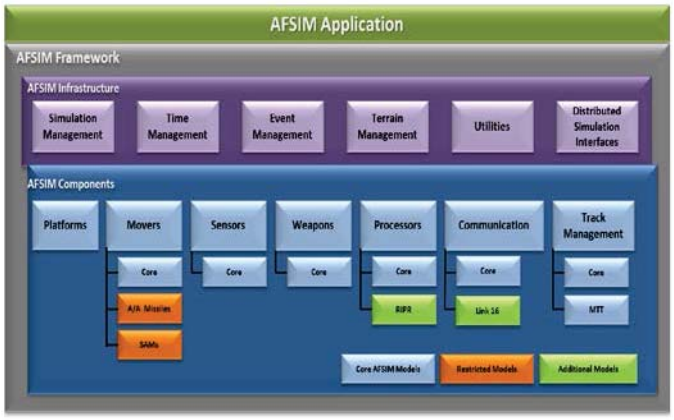


Fig. 1. The AFSIM functional architecture.

The top-level characteristics and capabilities of the AFSIM framework include:

- A class hierarchy of simulation objects, including data driven platforms, movers, sensors, communications networks, processors, weapons, and simulation observers.
- Simulation and Event classes to control time and/or event processing for AFSIM-based models, and the logging of entity data.
- Standard math libraries for coordinate systems (WGS-84, Spherical, ENU, NED), random number generation, DIS communication, High-Level Architecture (HLA) publish and subscribe, and generalized software routines, such as container classes for storing objects and data.
- A common geo-spatial environment and terrain representation, importing standard formats such as National Geospatial-Intelligence Agency (NGA) Digital Terrain Elevation Data (DTED), ESRI, GeoTiff and VMAP database formats.
- A general-purpose scripting language to provide access to framework objects using text input files (i.e., scripts) rather than through the Application Programming Interface (API).
- Communications network modeling, including basic radio transceivers and advanced communications algorithms, including addressable nodes, routers, multi-access protocols, contention and queuing.
- Electronic warfare modeling, including noise and deceptive jamming techniques, as well as the ability to jam and degrade any type of electro-magnetic receiver, including communications systems.
- Modeling of information flow and tasking between player and system elements to define candidate Network Centric Operation (NCO) concepts.
- The ability to run any AFSIM application in both constructive (batch processing) and virtual (real-time) modes.
- User interface elements for integrated scenario generation and post-processor visualization software.

In addition to the AFSIM core, several capabilities are available. Additional capabilities include: multitarget tracking algorithms; Link-16 modeling of both the physical and message layers; and Reactive Integrated Planning aRchitecture (RIPR) intelligent agent algorithms for implementing complex object behaviors. RIPR utilizes a Boeing-developed Quantum Tasker concept for commander subordinate interaction and task de-confliction. Section 3 provides additional details of the RIPR model. Restricted capabilities include missile flyout models.

The baseline AFSIM constructive application is called the Simulation of Autonomously Generated Entities (SAGE), which was one of the first constructive applications developed using the AFSIM framework. SAGE is a simple application that reads in a user-defined input file, executes the simulation, and outputs any user-defined data files. The original purpose

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

for SAGE was to simulate background air, road or maritime traffic. Although SAGE retains the capability to generate background traffic, the user can exercise all of the resident AFSIM capabilities.

B. AFSIM IDE

AFSIM permits the user to create subsystem definitions in separate files and to include those definitions in a hierarchal manner to define representations. This enables subsystem configuration control and reuse. This flexibility leads to large numbers of subsystem definition files when creating scenarios with a wide variety of different complex systems. The VESPA application facilitates the creation of the scenario initial conditions files. It does not, however, address the problems associated with defining and integrating system and subsystem models or defining system-level relationships such as command chains and peers using ASCII data files. Any input file errors are not discovered until an AFSIM application is executed.

In early 2011, Boeing initiated the development of the AFSIM Integrated Development Environment (IDE) to support the analyst in defining and integrating system and subsystem models. The AFSIM IDE patterns itself on IDEs created for use with software development. With software IDEs, a single application is used to edit files, compile, link, and run the software executable, and view output results or error messages. Likewise, the AFSIM IDE permits the analyst to edit input files, execute the AFSIM-based application, and visualize the output results and any error messages. This iterative process allows the analyst to receive immediate feedback as system and subsystem models are defined and scenarios are created.

Current capabilities of the IDE support input file creation including support for syntax highlighting, auto-completion, context-sensitive command documentation and a variety of scenario browsers. Syntax highlighting makes reading and understanding the content easier for the analyst. Unknown keywords or commands are underlined in red for easy discovery. Examples of unknown keywords or commands include misspelling of keywords or using keywords out of scope. The auto-completion feature provides a list of suggestions for the analyst to choose from, based on the context. The analyst can select one of the suggestions, and the command will be completed without having to manually type the command. Context-sensitive command documentation allows the analyst to bring up documentation associated with a command to illustrate the scope and use of the command. Other IDE capabilities are available to assist the analyst in defining system and subsystem models and scenarios.

The IDE can execute any AFSIM-based application using the input files defined by the analyst. Any screen output from the application is displayed in an IDE output window along with any error messages. Current capabilities of the IDE to view simulation results include the ability to run the VESPA application from the IDE using the AFSIM replay file created during the simulation run.

C. Visual Environment for Scenario Preparation and Analysis (VESPA)

To support the analyst, Boeing developed tools to facilitate scenario generation and post-process data analysis and visualization. Specifically, the Visual Environment for Scenario Preparation and Analysis (VESPA) software application was developed to support the creation of scenario initial condition files compatible with any AFSIM-based application. In addition, VESPA can be used to visualize object positional time histories and other event information generated as output from any AFSIM-based application. This allows the analyst to quickly understand and analyze the output from the simulation. Since VESPA is a "DIS-listener" visualization tool, it may also be used to display real-time entity interactions from any real-time simulation that publishes DIS data.

VESPA includes a graphical user interface (GUI) that includes a drawing area with a geospatial map and a data input area, as shown in Figure 2.

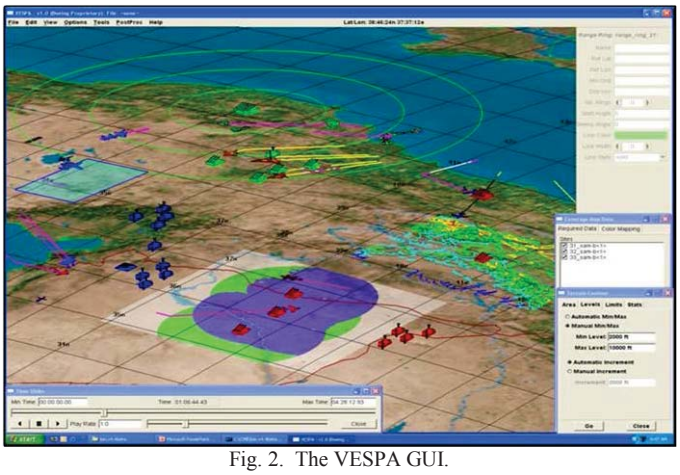


Fig. 2. The VESPA GUI.

Using VESPA, the analyst can place icons representing objects at specific latitude and longitude locations on a geospatial map. Initial conditions can then be assigned for each selected object. For example, the initial conditions of an aircraft could be its speed, heading and altitude. Visual features associated with objects, called attachments, can also be created. Examples include routes, range rings and zones.

VESPA can be used to display object positional histories and events using an AFSIM replay file generated during an AFSIM simulation run. The AFSIM replay file is a binary file containing the DIS output from the AFSIM simulation. In addition, plots can be generated for selected events that occurred during the simulation.

III. REACTIVE INTEGRATED PLANNING ARCHITECTURE (RIPR)

RIPR is the framework included with AFSIM that enables behavior modeling. RIPR is agent based, meaning that each agent acts according to its own knowledge; however, it is common for agents to cooperate and communicate with each other. RIPR is best thought of as a collection of utilities and algorithms that are used to construct intelligent agents. Most

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

modern RIPR agents, however, do contain a Perception Processor and a Quantum Tasker Processor. The agent senses the world by querying the platform and its subsystems, for information. The agent builds knowledge internally, makes decisions, and then takes action by controlling its platform accordingly. Most platform queries and control actions take place inside of the AFSIM scripting language. The knowledge-building and decision-making actions that RIPR performs are aided by various artificial intelligence technologies described in this section.

A. Cognitive Model

A RIPR agent maintains its own perception of threats, assets, and peers. This represents an agent's limited brain and the information can be delayed or erroneous. To represent players of varying skill, each agent has its own tunable cognitive model. For example, an "expert" pilot agent can maintain knowledge of 16 threats that he updates (looks at radar) every 5 seconds. Much of the cognitive model's ability is contained within the Perception Processor.

B. Quantum Tasker

The RIPR Quantum Tasker is used for commander subordinate interaction and task de-confliction. The Quantum Tasker comprises task generator(s), task-asset pair evaluator(s), an allocation algorithm, and various strategy settings (such as how to handle rejected task assignments). Each component (generator, evaluator, allocator) can be selected from pre-defined options, or custom created in script. The RIPR Quantum Tasker tasking system is also compatible with platforms using the older task manager (WSF_TASK_MANAGER and WSF_TASK_PROCESSOR). It can send and/or receive tasks to/from other RIPR agents and other task manager platforms. Figure 3 illustrates the various pieces of the Quantum Tasker and their connections.

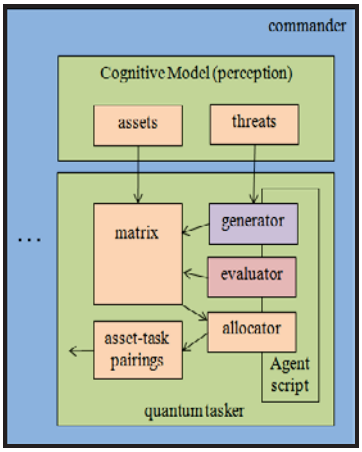


Fig. 3. Quantum tasker mode of operation.

The Quantum Tasker's method of operation:

- Acquire perception of assets from cognitive model for matrix columns.
- Acquire perception of threats from cognitive model
- Generator generates tasks for matrix rows.
- Strategy dictates how previously assigned tasks, rejected tasks, or new tasks are handled.
- Evaluator calculates values for possible asset-task pairs for matrix body.
- The allocator runs on the task-asset matrix to find appropriate task allocation, e.g. greedy, optimal, etc.
- Tasks are assigned over comm, handshaking performed for acceptance/rejection.

C. Behavior Tree

RIPR agents typically make use of a RIPR behavior tree to define their behavior. A behavior is a compact modular piece of script that performs some unique action. Behaviors should be parameterized and reusable. A behavior tree allows connection of behaviors in interesting ways so they perform in certain orders or subsets. The whole tree aggregates the behaviors to model an agent's behavior. Figure 4 provides an example of a RIPR behavior tree.

RIPR behavior trees provide five different intermediate connector-node types:

- Selector - chooses and performs first child behavior to pass its precondition check.
- Sequence - performs all child behaviors in sequence until one fails its precondition check.
- Parallel - performs all child behaviors whose precondition check passes.
- Weight Random - makes a weighted random selection from its child behaviors.
- Priority Selector - selects the child behavior who returns the largest precondition value.

Behavior trees provide for maximum utility for developing and editing agents. A properly constructed behavior tree allows a user to find relevant script fast, and swap in other behaviors at appropriate places. For example: try separating out behaviors for choosing desired heading, altitude, and speed from the behavior that actually performs the flight task. When you develop a new flying behavior, e.g. one that used a new route finder, you can swap that for the old one while keeping the logic in place for calculating desired direction.

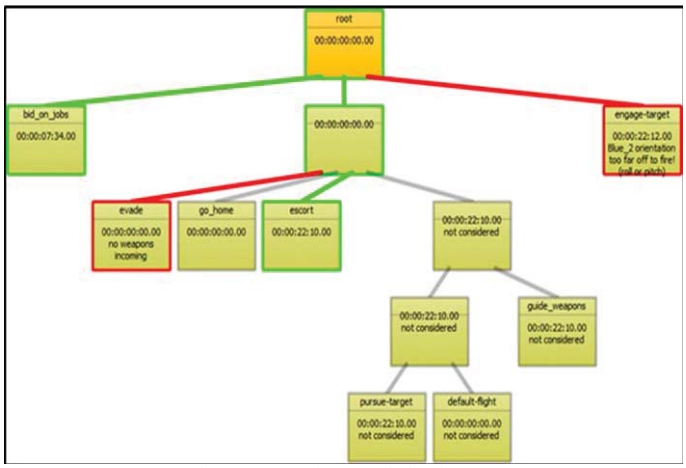


Fig. 4. Example RIPR behavior tree.

D. Cluster Manager

Some RIPR agents take advantage of the Cluster Manager to perform clustering on threat or asset perception in order to think of these larger sets as smaller groups. For example, it is common for a commander to group incoming threats into two clusters so it can send each of its two squadrons after separate

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

groups. The Cluster Manager can cluster based on desired similarity thresholds or based on the desired number of clusters. Similarity measurements can be based on ground distance, 3D distance, or 3D distance and speed. The Cluster Manager can use one of three clustering algorithms:

- Hierarchical Tree Max - default, guaranteed to be optimal, no cluster member dissimilar to any other member past the threshold (this method provides for tighter "classic" groups of members)
- Hierarchical Tree Min - guaranteed to be optimal, no cluster member dissimilar to at least one other member past the threshold (this method allows for long "stringy" chains of members)
- K-Means - not guaranteed to be optimal, fastest, clusters are centered on K different mean points.

E. Example Agent Interaction

Below is an example sequence of interactions within the RIPR architecture for a group of agents:

1. A commander agent obtains threats from his cognitive model (Perception Processor).
2. Commander's Quantum Tasker generator clusters threats into groups and creates a task for each group.
3. Commander's Quantum Tasker evaluator scores his squadrons (assets) against each group.
4. Commander's Quantum Tasker allocator finds optimal task assignment.
5. Commander assigns task(s) to subordinate flight leads over comm.
6. Flight lead uses asset and threat perception from cognitive model while interpreting task.
7. Flight lead agent's Quantum Tasker generates, evaluates, allocates, and assigns tasks to pilot agents.
8. Pilot agent uses peer and threat perception from cognitive model.
9. Pilot agent's behavior tree checks for evade, disengage, bingo conditions.
10. Pilot agent's behavior tree flies to intercept and eventually engages threat from task.
11. Pilot agent uses route finder to fly around SAM zones during ingress towards target.

IV. Future Work

The current state of the AFSIM framework only allows distribution to DoD agencies and DoD contractor's due to International Traffic in Arms Regulations (ITAR) restrictions. It is the desire of the AFRL to allow wider dissemination of the framework in order to provide more modeling and simulation collaboration opportunities. However, the current architecture of AFSIM does not easily lend itself to maintaining multiple versions across multiple release restrictions, which is why an architecture rework is underway to create a Component Based Architecture.

A. Component Based Architecture

Figure 5 details the current base level architecture of AFSIM. Since the base components of AFSIM are directly named in code this makes it difficult to add or remove base component types. Also it is currently difficult to extend other non-platform components.

Fig. 5. Existing AFSIM base level architecture.

In order to better facilitate the ability to add and remove base components work is underway to create a Component Based Architecture, which relies on an underlying generic component class where all components can be derived from. This architecture allows access via naming for components that already exist and will ease the addition and removal of certain component types. This solution maximizes commonality with the original architecture while at the same time providing a means to maintain a release version with no weapons or electronic warfare capabilities included as well as an ITAR release, which would include those components. The new architecture is shown in Figure 6.

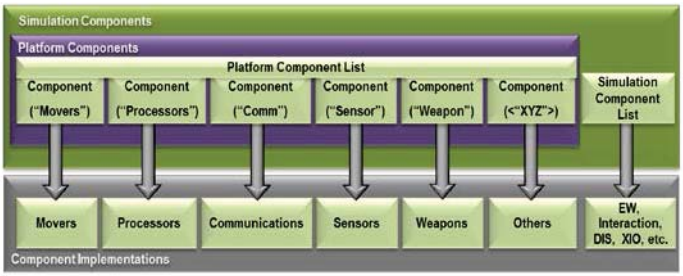


Fig. 6. New AFSIM Component Based Architecture.

V. Conclusion

In this paper we have provided a high level overview of the AFSIM simulation environment. AFSIM has been under development by Boeing under IR&D funds for more than 10 years. Under contract, Boeing delivered AFSIM to the Air Force (specifically AFRL/RQQD) with unlimited government rights (including source code) in February 2013. AFRL has now begun to distribute AFSIM within the DoD community. The AFSIM distribution comes with three pieces of software: the framework itself, an IDE and the visualization tool VESPA. Although AFSIM is currently ITAR restricted future work is planned to modify the underlying architecture to facilitate maintaining multiple versions with varying releasability. Under AFRL management AFSIM will continue to grow as a valuable modeling and simulation tool.

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

SESSION

FUZZY LOGIC AND SYSTEMS + DATA SCIENCE + BIG DATA ANALYTICS AND NOVEL APPLICATIONS + SEMANTIC WEB

Chair(s)

TBA

Computing With Fuzzy Rule Continua

Bart Kosko *

Abstract

This paper shows how to extend additive fuzzy rulebased systems to continuum-many rules and still control the problem of exponential rule explosion. Fuzzy systems fire all their rules for each input. The new system fires only a special random sample of rules taken from a continuum of rules. Additive systems add the fired if-then rules to compute an output. Adding gives rise to an inherent probability mixture-density structure if the system computes an output by taking the centroid of the summed then-part sets of the fired rules. Then the system output is a convex combination of the centroids of the fired then-part sets. This probabilistic convex-sum structure extends to the uncountably infinite case of a rule continuum. A new higher-level mixture structure can define wave-like meta-rules on the rule continuum. Statistical hill-climbing algorithms can tune these mixture meta-rules. But the infinite size of the rule continuum requires that some form of Monte Carlo importance sampling compute the system outputs. The meta-rules grow only linearly even though the underlying fuzzy if-then rules can have high-dimensional if-part and then-part sets.

Keywords: Additive fuzzy systems, rule explosion, fuzzy function approximation, mixture densities, rule continua, Monte Carlo importance sampling

1 From Rule Explosion to Rule Continua

Fuzzy systems suffer from exponential rule explosion in high dimensions [4, 10–16, 18]. This holds if at least two fuzzy sets (such as `SMALL` and `LARGE`) cover each input and output axis because fuzzy if-then rules combine such sets into Cartesian products in the input-output product space. The Cartesian products define a graph cover that grows exponentially with the number of input or output dimensions. So the graph cover of a vector-valued fuzzy system $F : \mathbb{R}^n \to \mathbb{R}^p$ tends to require $O(k^{n+p-1})$ rules.

*Bart Kosko is with the Department of Electrical Engineering, Signal and Image Processing Institute University of Southern California, Los Angeles, California 90089, USA (email: kosko@usc.edu)

A linguistic fuzzy rule combines fuzzy-set adjectives into if-then conditional statements. A paragraph of such statements can define a fuzzy system. A fuzzy set $A \subset \mathbb{R}$ maps input values $x \in \mathbb{R}$ to degrees of membership and thus defines a function $a : \mathbb{R} \to [0, 1]$ [23]. The fuzzy set `COOL` of cool air temperatures maps each real temperature value t to a membership degree in the unit interval $[0, 1]$. So all air temperatures are cool to some degree even if most are cool only to zero degree. Temperature acts here as a *lingistic variable* that takes on fuzzy-set adjective values such as `COOL` or `COLD` or `WARM` [24, 25]. But neither the fuzzy sets nor the rules need have any tie to words or natural language. The sets simply quantize an input or output variable or axis [13, 18].

Fuzzy rules define fuzzy patches in the product space. Consider the rules that control an air conditioner. Figure 1 shows that the linguistic control rule "If the air is `COOL` then set the air conditioner's motor speed to `SLOW`" defines a fuzzy subset of the 2-D product space of air temperatures and motor speeds. This rule arises from the Cartesian product of the triangular if-part fuzzy set `COOL` and the trapezoidal then-part fuzzy set `SLOW`. Each pair (t, m) of input tenperature values t and output motor speeds m belongs to the product space to some degree. So the rule `COOL`×`SLOW` looks like a barn or hill of membership values that stands above the 2-D planar space [15]. The rule looks like a patch or blob if one views it from above as in Figure 1. The rule patches need not be connected. They almost always are connected in practice because users almost always use connected fuzzy sets for both the if-part and then-part sets.

Thus a rule patch geometrizes a minimal knowledge unit because it geometrizes an if-then conditional. But the same graph-cover geometry that makes it easy to build simple fuzzy systems from words and conditionals also creates rule explosion in the product space.

A fuzzy system F approximates a function f by covering its graph with rule patches and then averaging patches that overlap. The averaging corresponds to taking the centroid of the fired rule then-parts. Data clusters can estimate the rule patches [10]. Supervised learning can further shape and tune the rules [4, 13, 18].

Additive fuzzy systems can *uniformly* approximate any continuous function on a compact set [11]. A uniform

approximation lets the user pick an error tolerance level $\varepsilon > 0$ in advance and be sure that the error in the approximation is less than ε for *all* input values x: $\|F - f\| < \varepsilon$. But achieving such a uniform approximation may require a prohibitive number of rules. Optimal lone rules cover the extrema of the function f [12]: They "patch the bumps." Supervised learning tends to move rule patches quickly to cover extrema and then move the extra rules in the graph cover to fill in between extrema [4, 13].

Some additive fuzzy systems F can exactly *represent* f in the sense that $F(x) = f(x)$ for all x. The Watkins Representation Theorem states the surprising result that an additive fuzzy system F with just *two* rules can represent any real-valued function f of n real variables if f is bounded [21]. The Watkins result does require that one both know the functional form of f and build it into the structure of the if-part sets of the two rules. This two-rule representation has special force in modern Bayesian statistics because so many common prior and likelihood probabiity densities are bounded [19].

The sets themselves need not be fuzzy at all. Rectangular sets define ordinary binary sets and still lead to uniform approximation for enough rules. So the power of fuzzy systems lies in their ability to approximate functions and not technically in the use of fuzzy sets or their linguistic counterparts. But the fuzzy or nonfuzzy graph cover still achieves such function approximation in high dimensions at the cost of a rulebase curse of dimensionality. Tuning or adapting the fuzzy rules only compounds the complexity cost [18]. Tuning even four independent variables often proves intractable in practice.

So extending a fuzzy system to infinitely many rules does not seem to make sense. Firing or tuning infinitely many rules would appear to describe the ultimate form of rule explosion. But this need not be if we replace the current default of firing all fuzzy rules for each input with firing a carefully chosen random subset of rules.

We show below that working with rule continua lets the user define and tune wave-like meta-rules as a higher-level mixture density defined on a virtual rule continuum. Then statistical algorithms such as the expectation-maximization algorithm can tune the mixture meta-rules with training data. The number of such rules grows only linearly. The new cost becomes the difficulty of computing system outputs $F(x)$ from a rule continuum. Some form of Monte Carlo importance sampling can ameliorate this new burden.

The next section reviews the convex structure of centroidal additive fuzzy systems. This convexity implies an inherent probabilistic structure: Centroidal additive fuzzy systems are generalized mixture densities. The last section shows how to extend these convex fuzzy systems to rule continua.

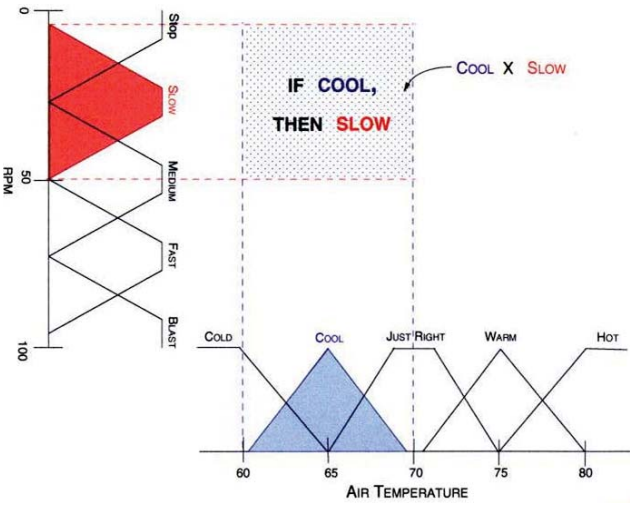


Figure 1: A fuzzy rule as a product-state patch. A Cartesian product combines the if-part fuzzy set COOL with the then-part fuzzy set SLOW to produce the linguistic rule "If the air temperature is COOL then set the air conditioner's motor speed to SLOW." The rule COOL × SLOW defines a patch or fuzzy subset of the input-output product space of temperature values and motor speeds. The figure shows only the base of the rule and not the barn-like set of membership values above it.

2 Additive Fuzzy Systems as Probabilistic Convex Mixtures

Additive fuzzy systems exploit the convex-sum structure that results from additively combining fired if-then rules and computing outputs as centroids [4, 10–16, 18]. They generalize mixture-density models from machine learning and pattern recognition because such mixtures are convex sums that do not depend on an input value.

A fuzzy system is a mapping $F : \mathbb{R}^n \to \mathbb{R}$. It uses a set of fuzzy if-then rules to convert a vector input x to an output $F(x)$. There is no loss of generality if the fuzzy system is scalar and thus if it maps to the real line $\mathbb{R}$. All results still hold with appropriate vector notation for vector-valued fuzzy systems $F : \mathbb{R}^n \to \mathbb{R}^p$.

We first show that *any* centroidal fuzzy system defines a conditional expectation and hence is a probabilistic or statistical system. The fuzzy system need not be additive. A non-additive system could combine rules through a maximum operation or through any other aggregation operation [7,8,22]. Early fuzzy systems combined outputs with a maximum or supremum operation [17].

A centroidal output suffices to produce a conditional expectation. So the conditional-expectation result does not require an independent probabilistic assumption. It follows instead from just the nonnegativity and the integrability of the then-part fuzzy sets B_j that all fuzzy if-then rules use. We first state some notation for fuzzy systems and then state and prove the conditional-

expectation result as Theorem 1.

A centroidal fuzzy system $F : \mathbb{R}^n \to \mathbb{R}$ is a fuzzy system that computes the output $F(x)$ by taking the centroid of a finite number m of combined "fired" then-part sets: $F(x) = \texttt{Centroid}(B(x))$. Later we will drop the finite assumption. The term $B(x)$ stands for the combined fired then-parts. The argument x implies that the vector input x has fired the m rules. The fired combination $B(x)$ has a generalized set function $b : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^+$ that has a finite integral. The j-th rule $R_{A_j \to B_j}$ has the linguistic form "If $X = A_j$ then $Y = B_j$" for if-part fuzzy set $A \subset \mathbb{R}^n$ and scalar then-part fuzzy set $B_j \subset \mathbb{R}$. The unfired then-part set B_j has set function $b_j : \mathbb{R} \to [0, 1]$. But its fired version $B_j(x)$ has a two-place argument and thus corresponds to the set function $b_j(x, y) : \mathbb{R}^n \times \mathbb{R} \to [0, 1]$ for vector input $x \in \mathbb{R}^n$. But we still write the set function in single-argument notation $b_j(x)$ for simplicity. The rule $R_{A_j \to B_j}$ is a fuzzy subset of the input-output product space $\mathbb{R}^n \times \mathbb{R}$ because all input-output pairs (x, y) satisfy the rule to some degree. So the rule corresponds to a two-placed set function $r_{A_j \to B_j} : \mathbb{R}^n \times \mathbb{R} \to [0, 1]$. An input vector x_0 fires the rule by convolving the rule's set function $r_{A_j \to B_j}$ with the input delta spike $\delta(x - x_0)$ [13].

The n-dimensional fuzzy set $A_j \subset \mathbb{R}^n$ corresponds to a joint set membership or multivalued indicator function $a_j : \mathbb{R}^n \to [0, 1]$. Users often assume in practice that the joint membership function factors into a product of scalar membership functions: $a_j(x) = \prod_{k=1}^m a_j^k\left(x^k\right)$ where each factor set $A_j^k \subset \mathbb{R}$ has set function $a_j^k : \mathbb{R} \to [0, 1]$ for row vector $x = \left(x^1, \ldots, x^n\right)$ [13]. Earlier fuzzy systems sometimes formed the joint set function a_j by taking pairwise minima $a_j(x) = \min\left(a_j^1\left(x^1\right), \ldots, a_j^k\left(x^n\right)\right)$ or some other pairwise triangular-norm operation [7, 8]. But the minimum function ignores the information in all scalar inputs except the smallest one when the inputs differ. The standard additive fuzzy systems below always work with the simpler product factorization. The product function preserves the relative values of the scalar inputs. The then-part set function can be a generalized set function. The then-part fuzzy sets B_j need only have positive and integral set functions $b_j : \mathbb{R} \to \mathbb{R}^+$ because of the normalization involved in taking the centroid. They do not need to map to the unit interval [13, 18, 19].

Now suppose the vector input $x = \left(x^1, \ldots, x^n\right)$ activates the scalar fuzzy system $F : \mathbb{R}^n \to \mathbb{R}$ to produce the combined rule firings $B(x)$. Then a centroidal fuzzy system computes the system output $F(x)$ by taking the centroid or center of gravity of $B(x)$: $F(x) = \texttt{Centroid}(B(x))$.

Theorem 1 states that taking the centroid results in a conditional expectation for *any* fuzzy system that combines rules to produce $B(x)$ [13]. Again the fuzzy system need not be additive. It can combine fired then-part sets with a global union (pairwise maximum or other triangular co-norm or aggregation operator [22]) or with any combination operator compatible with the so-called "extension principle" of classical fuzzy set theory [5, 7, 8, 23].

Theorem 1. *Every centroidal fuzzy system is a conditional expectation:*

$$F(x) = E[Y|X = x]. \tag{1}$$

Proof. Assume that the then-part sets B_j are nonnegative and integrable. Assume that the input x leads to nontrivial rule firings and thus leads to a nonzero combination of fired rules $B(x)$: $b(x) > 0$. Then direct expansion gives

$$\begin{aligned}
F(x) &= \texttt{Centroid}(B(x)) && (2)\\
&= \frac{\int_{-\infty}^{\infty} y b(x)\ dy}{\int_{-\infty}^{\infty} b(x)\ dy} && (3)\\
&= \frac{\int_{-\infty}^{\infty} y b(x, y)\ dy}{\int_{-\infty}^{\infty} b(x, y)\ dy} && (4)\\
&= \int_{-\infty}^{\infty} y \left[\frac{b(x, y)}{\int_{-\infty}^{\infty} b(x, y)\ dy}\right] dy && (5)\\
&= \int_{-\infty}^{\infty} y p(y|x)\ dy && (6)\\
&= E[Y|X = x] && (7)
\end{aligned}$$

The expectation structure results because $p(y|x) = \frac{b(x,y)}{\int_{-\infty}^{\infty} b(x,y)\ dy}$ is nonnegative and because $\int_{-\infty}^{\infty} p(y|x)\ dy = 1$ holds from the nonnegativity and integrability of the b function if $b(x, y) > 0$. So $p(y|x)$ is a proper conditional probability density function. Then $E[Y|X = x]$ is a realization of the condition-expectation random variable $E[Y|X]$. □

Additive fuzzy systems add fired then-part sets to compute the combined set $B(x)$. This leads to the central fact of additive systems: Their outputs equal the *convex combination* of the centroids of the fired then-part sets.

We first prove that all additive centroidal fuzzy systems are convex sums of fired then-part centroids. An additive fuzzy system combines the m fired then-part sets $B_j(x)$ by adding them:

$$B(x) = \sum_{j=1}^{m} w_j B_j(x) \tag{8}$$

for positive rule weights $w_j > 0$. The rule weights need not sum to unity. And they can depend on the input x. They drop out of the centroidal output $F(x)$ if they are all equal: $w_1 = \cdots = w_n$. Then the combined set $B(x)$ has a generalized set function $b(y|x)$ for each input x as y ranges over the range space $\mathbb{R}$: $b(y|x) = \sum_{j=1}^m w_j b_j(y|x)$. We here use the conditional notation $b_j(y|x) : \mathbb{R} \times \mathbb{R}^n \to [0, 1]$ for the set function of the fired then-part set $B_j(x)$. So the inputs x parametrize the fired then-part sets.

Each fired then-part set $B_j(x)$ has an area or volume $V_j(x) = \int_{-\infty}^{\infty} b_j(y|x)\ dy$. We again assume that all such integrals are finite and positive. This gives in turn an input-dependent centroid $c_j(x)$ for the fired then-part set

$B_j(x)$:

$$c_j(x) = \frac{\int_{-\infty}^{\infty} y b_j(y|x)\ dy}{\int_{-\infty}^{\infty} b_j(y|x)\ dy} \tag{9}$$

$$= \frac{1}{V_j(x)} \int_{-\infty}^{\infty} y b_j(y|x)\ dy. \tag{10}$$

Then Theorem 2 states that all additive centroidal fuzzy systems equal a convex combination of fired then-part centroids. We omit the proof for reasons of space.

Theorem 2. *Additive centroidal fuzzy systems are convex combinations of fired then-part centroids:*

$$F(x) = \sum_{j=1}^{m} p_j(x)\, c_j(x) \tag{11}$$

where the convex coefficients $p_j(x)$ have the ratio form

$$p_j(x) = \frac{w_j V_j(x)}{\sum_{k=1}^{m} w_k V_k(x)}. \tag{12}$$

Additivity produces convexity in the global conditional probability density function (pdf) $p(y|x)$. This density decomposes into a convex sum of the m local rule-specific conditional pdfs $p_{B_j}(y|x)$:

$$p_{B_j}(y|x) = \frac{b_j(y|x)}{\int_{-\infty}^{\infty} b_j(y|x)\ dy}. \tag{13}$$

The normalizing denominator is just the input-dependent area or volume $V_j(x)$. This gives the important "mixture" of pdfs result:

$$p(y|x) = \sum_{j=1}^{m} p_j(x)\, p_{B_j}(y|x) \tag{14}$$

for the mixture probabilities $p_j(x)$ in Theorem 2.

Next comes the key simplification of firing then-part sets by scaling them [4, 10–14]. We say that an additive fuzzy system $F : \mathbb{R}^n \to \mathbb{R}$ is a *standard additive model* (SAM) if the fired if-part set value $a_j(x)$ multiplicatively scales the then-part B_j: $B_j(x) = a_j(x) B_j$. The multiplicative scaling shrinks the then-part set B_j over the same base. This scaling leaves the relative structure of the then-part set unchanged unlike the still-common min-clip $\min(a_j(x), B_j)$ that discards all then-part set information above the threshold $a_j(x)$.

The SAM structure greatly simplifies the above results for additive fuzzy systems. Now $a_j(x)$ factors out of the key SAM calculations. This leads to an important cancellation that converts the local conditional probability $p_{B_j}(y|x)$ to the *unconditional* probability $p_{B_j}(y)$ so long as $a_j(x) > 0$. Then the SAM volumes or areas V_j and centroids c_j are constant and so the user can pre-compute them. The SAM Theorem is just Theorem 2 with these simplifications [13]. We state it here as Theorem 3.

Theorem 3. SAM Theorem. *Standard additive model centroidal fuzzy systems are convex combinations of fixed then-part centroids:*

$$F(x) = \sum_{j=1}^{m} p_j(x)\, c_j \tag{15}$$

where the convex coefficients $p_j(x)$ have the ratio form

$$p_j(x) = \frac{a_j(x)\, w_j V_j}{\sum_{k=1}^{m} a_k(x)\, w_k V_k}. \tag{16}$$

The SAM structure likewise simplifies the above mixture sum (14) to a probabilistic mixture of *unconditional* pdfs p_{B_j}:

$$p(y|x) = \sum_{j=1}^{m} p_j(x)\, p_{B_j}(y). \tag{17}$$

The next section directly extends this proobabilistic mixture result to the continuous case.

We close this section with a useful corollary of the SAM Theorem. It shows that all higher-order moments of an additive centroidal fuzzy system inherit the same convex structure. We state this result in the simpler SAM form where the then-part volumes and centroids do not depend on the input x.

Theorem 4. *All higher-order moments of SAM systems are convex sums:*

$$F(x) = E[Y|X = x] = \sum_{j=1}^{m} p_j(x)\, c_j \tag{18}$$

$$V[Y|X = x] = \sum_{j=1}^{m} p_j(x)\, \sigma_{B_j}^2 + \sum_{j=1}^{m} p_j(x)\, [c_j - F(x)]^2 \tag{19}$$

$$E\left[(Y - E[Y|X = x])^k | X = x\right] = \sum_{j=1}^{m} p_j(x) \sum_{l=0}^{k} \binom{k}{l} E_{B_j}[(Y - c_j)^l](c_j - F(x))^{k-l} \tag{20}$$

for all positive integers k and where

$$\sigma_{B_j}^2 = \int_{-\infty}^{\infty} (y - c_j)^2 p_{B_j}(y)\ dy. \tag{21}$$

The 2005 paper [16] appears to be the first to plot the conditional variance surface that corresponds to a fuzzy system and thus to its representation as a conditional expectation. The conditional variance $V[Y|X = x]$ gives a direct measure of confidence in the fuzzy "answer" $F(x)$ to an input "question" x relative to the rules in the rule-base. The first convex sum on the righthand side of the second-moment term $V[Y|X = x]$ reflects the inherent

uncertainty in the m then-part sets. This term is positive even if all then-part sets B_j have the same shape because then the convex sum just equals the common unconditional then-part variance $\sigma^2_{B_j} > 0$. Even then different common shapes produce different uncertainly levels as $\sigma^2_{B_j}$ varies. So the shape of then-part sets matters. The second convex sum is an interpolation penalty. The uncertainty in the fuzzy system's output $F(x)$ goes up substantially if the jth rule fires to a high degree ($p_j(x) \approx 1$) when its then-part centroid c_j differs greatly from $F(x)$. This conditional variance result extends directly to conditional covariance matrices for vector outputs.

3 Fuzzy Rule Continua

We now exploit the mixture structure of centroidal additive fuzzy systems to extend them to rule continua.

Mixture models are finite convex combinations of pdfs [6]. A convex combination mixture of m pdfs $f_1, \dots, f_m$ gives a new pdf f with m modes if the m pdfs are unimodal and if they are sufficiently spread out:

$$f(x) = \sum_{j=1}^{m} \pi_j f_j(x). \tag{22}$$

The nonnegative *mixing weights* $\pi_1, \dots, \pi_m$ sum to unity: $\sum_{j=1}^{m} \pi_j = 1$. This convex sum $f(x)$ can model taking random samples from a population made up of m-many subpopulations such as m words or images or other patterns. The estimation task is to find the m mixture weights and the parameters of the mixed pdfs. The most popular mixture by far is the Gaussian mixture where f_j is a scalar or vector Gaussian $\mathcal{N}\left(\mu_j, \sigma_j^2\right)$.

The mixture sum is not arbitrary. It follows from the elementary theorem on total probability. Suppose that m hypothesis sets $H_1, \cdots, H_m$ partition a sample space Ω and that the set $E \subset \Omega$ represents some observed evidence. Then the theorem on total probability states that the unconditional probability of the evidence $P(E)$ equals the convex combination of the prior probabilities $P(H_j)$ and the likelihoods $P(E|H_j)$: $P(E) = \sum_{j=1}^{m} P(H_j) P(E|H_j)$. This corresponds to the above mixture sum if the evidence is the input x and if π_j is the prior probability of the j-th class or mixture element. So $f_j(x) = f(x|j)$ holds if the conditional density $f(x|j)$ is the likelihood that we would observe such an x if it came from the j-th class or subpopulation.

The ubiquitous Expectation-Maximization (EM) algorithm often estimates the mixing weights and the Gaussian means and variances by iteratively maximizing the likelihood function [6]. The class memberships of the m subpopulations correspond to the hidden or latent variables in the EM algorithm. Then carefully injected noise can always speed up convergence of the EM algorithm [1, 2, 20] as it climbs the nearest hill of likelihood.

The SAM and other additive systems generalize mixture models by making the mixture weights π_j depend on the input x: $\pi_j(x) = p_j(x)$. This in turn makes the mixture's means and variances (and other moments) depend on x and thus become conditional moments as in Theorem 4. So mixture models correspond to *fixed*-input centroidal SAM fuzzy systems. So (14) reduces to the defining mixture-density combination for unfired then-part sets if $y = x$:

$$p(y) = \sum_{j=1}^{m} p_j \, p_{B_j}(y). \tag{23}$$

Thus mixture models sample from convex combinations of m suitably normalized then-part fuzzy sets B_j.

We now extend SAM models to systems with infinitely many fuzzy rules. The cardinality of the rulebase can be countably or uncountably infinite. We will work with the latter continuum case. This follows from the direct extension of mixture models to *compounding* models that weight one pdf with another and then integrate out the continuous mixture index [6]. The complexity of the SAM systems will instead require that we impose a higher-level mixture structure on the continuum of rules.

Suppose now that the real parameter θ indexes the continuum-many if-part set functions $a_\theta : \mathbb{R}^n \to [0, 1]$ and the then-part sets B_θ in continuum-many rules of the form "If $X = A_\theta$ then $Y = B_\theta$". Then integration replaces the rulebase sum to give the combined rule firings:

$$b(y|x) = \int_{\theta=-\infty}^{\theta=\infty} w_\theta \, b_\theta(y|x) \; d\theta \tag{24}$$

if the integral exists for appropriate nonnegative rule weights w_θ. Then the discrete mixture result becomes $b(y|x) = \int p_\theta(x) p_{B_\theta}(y|x) \; dx$ in general. Then the proof of Theorem 3 still goes through if (definite) integrals replace the finite sums in the SAM case:

$$F(x) = \int p_\theta(x) \, c_\theta \; d\theta \tag{25}$$

$$p_\theta(x) = \frac{a_\theta(x) \, w_\theta V_\theta}{\int a_\phi(x) \, w_\phi V_\phi \; d\phi}. \tag{26}$$

Consider a Gaussian rule continuum for a scalar parameter θ. The rules have vector-Gaussian if-part set functions a_θ and scalar Gaussian then-part set functions b_θ: $a_\theta(\cdot) = \mathcal{N}(\theta \bullet \mathbb{1}, K_\theta)$ and $b_\theta(\cdot) = \mathcal{N}\left(\theta, \sigma^2\right)$ if $\theta \bullet \mathbb{1}$ denotes the n-vector with all elements equal to θ (θ can also index mean vectors μ_θ). K_θ is an n-by-n covariance matrix. It equals the identity matrix in the simplest or "white" case. Assume unit rule weights $w_\theta = 1$. Put $c_\theta = \theta$ and $V_\theta = 1$ since $b_\theta(\cdot) = \mathcal{N}\left(\theta, \sigma^2\right)$. This gives the output $F(x)$ as a simple unconditional expectation for each x: $F(x) = \int p_\theta(x) \, \theta \; d\theta = E_{p_\theta(x)}[\Theta]$. The question is how to compute the expectation $E_{p_\theta(x)}[\Theta]$.

Computing the convex integral for $F(x)$ is more complicated than in the simpler case of probabilistic compounding. Compounding allows the modeler to pick the weighting pdf p_θ as a normal or gamma or other well-behaved closed-form pdf [6]. But the SAM convex-sum p_θ

involves a highly nonlinear transformation of continuum-many if-part set functions a_θ. This transformation may not be tractable. Integrating it to produce $F(x)$ can only compound the computational intractability.

Monte Carlo simulation offers a practical way to compute the output expectation $E_{p_\theta(x)}[\Theta]$. This technique relies on the weak law of large numbers (WLLN). The WLLN states that the sample mean $\overline{X}_n = \frac{1}{n}\sum_{k=1}^{n} X_k$ of independent and identically distributed finite-variance random variables $X_1, X_2, \ldots$ converges in probability to the population mean $E[X]$: $\lim_{n\to\infty} P\left(\left|\overline{X}_n - E[X]\right| > \epsilon\right) = 0$ for all $\epsilon > 0$. Monte Carlo simulation interprets an ordinary definite integral $\int_a^b g(x)\ dx$ as the expectation of a function g of a random variable X that has a uniform distribution over (a, b) [6]:

$$\int_a^b g(x)\ dx = (b-a)\int_a^b g(x)\,\frac{dx}{b-a} = (b-a)\,E[X] \tag{27}$$

for $X \sim \mathscr{U}(a, b)$. The user need not integrate the integrand $(b-a)\,g(x)$. The user need only compute values $(b-a)\,g(x_k)$ for random uniform draws x_k from (a, b). The random draws can come from any uniform random number generator. Then the WLLN ensures that

$$\frac{1}{n}\sum_{k=1}^{n}(b-a)\,g(x_k) \approx (b-a)\,E[X] = \int_a^b g(x)\ dx \tag{28}$$

for enough random draws x_k. The variance in the WLLN estimate decreases linearly with the number n of draws.

Monte Carlo simulation can estimate the integrals in the continuum-rule SAM system for a given input x. Assume there are n random draws of θ from some finite interval $[c, d]$ and thus $c \le F(x) \le d$. Then

$$F(x) = \frac{\int a_\theta(x)\, w_\theta V_\theta c_\theta\ d\theta}{\int a_\phi(x)\, w_\phi V_\phi\ d\phi} \tag{29}$$

$$\approx \frac{\frac{1}{n}\sum_{k=1}^{n} w_k a_k(x)\, V_k c_k}{\frac{1}{n}\sum_{j=1}^{n} w_j a_j(x)\, V_j} \tag{30}$$

$$= \frac{\sum_{k=1}^{n} w_k a_k(x)\, V_k c_k}{\sum_{j=1}^{n} w_j a_j(x)\, V_j} \tag{31}$$

$$= \sum_{k=1}^{n} p_k(x)\, c_k. \tag{32}$$

The final result has the same convex-sum form as the finite-rule SAM in Theorem 3. But now the sum is over *random* choices of rules instead of over all rules.

The last task is to control and shape the overall distribution of the rule continuum. This allows the fuzzy engineer to define *meta-rules* at a much higher level of abstraction. An engineer can also give the wave-like groupings of rules a linguistic interpretation such as "small negative" or "medium positive" and the like. The engineer should be able to pick an initial set of such meta-rules just as in the case of setting up a finite SAM. Then the engineer should have some practical way to tune these meta-rules with data to give different levels of control or function approximation. This requires a Bayesian-like approach that puts some probabilistic structure on the parameter θ: $\Theta \sim h(\theta)$. Imposing such random structure corresponds to the old fuzzy-engineering task of picking the shapes of the if-part and then-part sets [18].

Mixture densities offer a natural way to define fuzzy meta-rules over the rule continuum. The mixture variable is not x at this level. It is now θ. Suppose the fuzzy engineer wants to impose k-many fuzzy meta-rules. This requires mixing k-many (likely Gaussian) densities f_i:

$$h(\theta) = \sum_{i=1}^{k} \pi_i f_i(\theta). \tag{33}$$

The engineer might center the mixed pdfs f_i closer together in regions of the input space where he desires greater control. An early example of such proximity control was the fuzzy truck-backer-upper [9]. The truck-and-trailer rig backed up to a loading dock in a parking lot. Closer and narrower if-part sets near the loading dock gave finer error control near that equilibrium point. A few wide if-part sets covered much of the remaining parking lot. The engineer can distribute these meta-rule mixture pdfs in the same way: Cluster the mixed terms more closely to achieve finer control or approximation in the state space. The underlying if-then rules can still support a linguistic interpretation.

Standard statistical techniques can then compute fuzzy outputs $F(x)$ and tune the fuzzy meta-rules. Monte Carlo simulation can estimate the output $F(x)$ for a given x. But the sampling now cannot be from a uniform density in general. That would always give the same output on average. The sampling must come instead from the meta-rule mixture density $h(\theta)$ itself to reflect the distribution of the meta-rules. This is just the well-known technique of *importance sampling* from mixtures [3]. Then the E-M algorithm or its variants can tune the mixture parameters based on sampled inputs x.

Suppose there are n random training samples $(x_1, f(x_1)), ..., (x_n, f(x_n))$ from some real function f. Then the total summed squared error E of the training sample is $E = \sum_{k=1}^{n}(f(x_k) - F(x_k))^2$. So minimizing E is the same as maximizing the probability e^{-E}. This exponentiation converts the learning problem into one of maximum likelihood estimation. The E-M algorithm maximizes the likelihood by interatively estimating the latent or "hidden" mixture weights and the means and variances of the mixed Gaussian pdfs [6, 20].

The growth in meta-rules is only *linear* in the number k of mixed densities. That shifts much of the computational burden to the sampling task involved in converting an input x to an output $F(x)$.

4 Conclusions

Centroidal additive fuzzy systems generalize probabilistic mixture densities. But they suffer from exponential rule explosion as they fire all rules for each input. A compromise is to fire only a special random subset of the rules. The rules themselves can come from a virtual rule continuum. Sampling from a higher-order mixture of densities avoids a direct rule explosion because the number of mixed densities grows only linearly. An open research problem is to find effective algorithms that can tune such a higher-level mixture control structure.

References

[1] K. Audhkhasi, O. Osoba, and B. Kosko, "Noise benefits in backpropagation and deep bidirectional pretraining," in *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN-2013)*. IEEE, 2013, pp. 2254–2261.

[2] ——, "Noise benefits in convolutional neural networks," in *Proceedings of the International Conference on Advances in Big Data Analytics (ABDA'14)*, 2014, pp. 73–80.

[3] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. P. Robert, "Adaptive importance sampling in general mixture classes," *Statistics and Computing*, vol. 18, no. 4, pp. 447–459, 2008.

[4] J. A. Dickerson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules," *IEEETransactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 4, pp. 542–560, 1996.

[5] B. R. Gaines, "Foundations of fuzzy reasoning," *International Journal of Man-Machine Studies*, vol. 8, pp. 623–688, 1976.

[6] R. V. Hogg, J. McKean, and A. T. Craig, *Introduction to Mathematical Statistics*. Pearson, 2013.

[7] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Addison-Wesley, 1986.

[8] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Prentice Hall, 1988.

[9] S.-G. Kong and B. Kosko, "Adaptive fuzzy systems for backing up a truck-and-trailer," *IEEE Transactions on Neural Networks*, vol. 3, no. 2, pp. 211–223, 1992.

[10] B. Kosko, *Neural Networks and Fuzzy Systems*. Prentice-Hall, 1991.

[11] ——, "Fuzzy systems as universal approximators," *IEEE Transactions on Computers*, vol. 43, no. 11, pp. 1329–1333, 1994.

[12] ——, "Optimal fuzzy rules cover extrema," *International Journal of Intelligent Systems*, vol. 10, no. 2, pp. 249–255, 1995.

[13] ——, *Fuzzy Engineering*. Prentice-Hall, 1996.

[14] ——, "Global stability of generalized additive fuzzy systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 28, no. 3, pp. 441–452, 1998.

[15] B. Kosko and S. Isaka, "Fuzzy logic," *Scientific American*, vol. 269, no. 1, pp. 62–7, 1993.

[16] I. Lee, B. Kosko, and W. F. Anderson, "Modeling gunshot bruises in soft body armor with an adaptive fuzzy system," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 6, pp. 1374–1390, 2005.

[17] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers*, vol. 26, no. 12, pp. 1182–1191, 1977.

[18] S. Mitaim and B. Kosko, "The shape of fuzzy sets in adaptive function approximation," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 637–656, 2001.

[19] O. Osoba, S. Mitaim, and B. Kosko, "Bayesian inference with adaptive fuzzy priors and likelihoods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 5, pp. 1183–1197, 2011.

[20] ——, "The noisy expectation–maximization algorithm," *Fluctuation and Noise Letters*, vol. 12, no. 3, pp. 1 350 012–1–1 350 012–30, 2013.

[21] F. Watkins, "The representation problem for additive fuzzy systems," in *Proceedings of the International Conference on Fuzzy Systems (IEEE FUZZ-95)*, 1995, pp. 117–122.

[22] R. R. Yager, "On ordered weighted averging aggregation operators in multicriteria decision making," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[23] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[24] ——, "Outline of a new approach to the analysis of complex systems and decision analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 1, pp. 28–44, 1973.

[25] ——, "The concept of a linguistic variable and its application to approximate reasoning," *Information Sciences*, vol. 8, pp. 199–249, 1975.

Designing Bilateralism and Developing Fuzzy Inference System in the Political Domain

Dr. Sameera Alshayji, Nasser Al-Sabah, and Abdulla Al-Sabah
Political and Economic Affairs Department, Amiri Diwan, Seif Palace, Kuwait

Abstract - *The repercussions of and the reactions to Islamic organizations such as ISIS, Houthis, and Al-Qaeda, make it imperative for the leaders to redirect their investment compass in a proper way. This is especially true when considering whether to strengthen bilateral economic relations among nations, as critical decisions are influenced by certain variables that are based on heterogeneous and vague information. A common language is thus needed to describe variables that require human interpretation. Applying a fuzzy ontology method is one of the possible solutions to address the lack of conceptual clarity. Fuzzy logic is based on natural language and is tolerant of imprecise data. Furthermore, a fuzzy Inference System's (FIS) greatest strength lies in its ability to handle imprecise data. This research focuses on developing fuzzy inferences in the political domain, especially highlighting the concept of bilateral meetings as a case study of fuzzy ontology.*

Keywords: Fuzzy logic, FIS, Ontology, Political centers, Bilateralism, Bilateral meetings

1 Introduction

The repercussions of the uprisings and revolts brought about by the Arab Spring undoubtedly call for the heads of state to rethink their investment compasses, especially when it comes to strengthening their bilateral economic relationships. The world's increasing interconnectedness and the recent increase in the number of notable regional and international events pose ever-greater challenges for political decision-making processes. Many times, the response of a decision maker is to question the wishes of the other nations, that his or her country already has some economic bilateral relations with; thus, the answer becomes a silent thought. Such silent thoughts must have scientific logical parameters, A silence thought requires someone to analyze it, as the people from the information systems require clear variables to add into the system, This is especially true, for example, in the "MATLAP" system, which has the ability to convert linguistic variables to fix the number by conducting fuzzy logic methodologies. Alshayji et. al [9] presented the "MATLAP" system; to be more precise, we need to depict these variables in a database rather than in documents.

Overview

Considerable knowledge has been generated, organized, and digitized in various governmental sectors, but the political field still needs to be more organized for decision-makers. Most political terms are language-based and need to be interpreted. For example, existing relationships between countries can be described from a variety of perspectives, such as "strongly positive," "positive," "neutral," "negative," and "strongly negative." A conscientious decision maker who takes responsibility for promoting and strengthening bilateral economic relationships needs access to well-structured information that is relevant to his/her decisions.

1.1 Current Challenges

Unfortunately, in reality, the basic concept of political and investment information is a linguistic variable, that is, a variable with values in words rather than numbers. This makes it extremely difficult for the decision-maker to understand the concepts that exist in these domains. For example, Alshayji et al. [5] identified some concepts that influence decisions to strengthen economic relationships with other countries, such as the agreements concept [4], the nuclear affairs concept, and the peace in the Middle East concept; these ideas are also presented by Alshayji et al. [7]. The decision maker who is considering whether to strengthen economic relationships requires structured information. Examples of information that may be assessed in the decision-making process include competency questions such as "What is the result of the bilateral meeting? The types of answers may involve a description such as "strongly positive," "positive," "neutral," and "negative." In this situation, the political decision-maker could describe the bilateral relationships between the two nations in several phases, such as "very good" at a specific time, "good" at another time, and "weak" at the current time.

1.3 Problem formulation

A serious problem that the political or investment decision-maker faces is the difficulty of building an efficient political decision support system (DSS) with heterogeneous and vague information in the political and investment domains, especially regarding the decision to strengthen bilateral economic relationships with friendly nations. Typically, these critical decisions are influenced by

heterogeneous and vague information from different domains. Most of the political decision maker's documents use linguistic variables whose values are words rather than numbers and therefore are closer to human intuition. A natural language is needed to describe such information, which requires more human knowledge for interpretation.

Political centers

Political and diplomatic research centers (also known as think tanks) play a very large role in driving and shaping a country's domestic and international policy issues. Alshayji et al. [7] highlighted the need to establish a center for the political decision makers in the government. As such, many governments around the world depend on such think tanks to provide analysis and recommendations that help policymakers make domestic and foreign policy decisions. In the gulf region of the Middle East, such policy-oriented research centers are finding themselves under pressure and are being heavily scrutinized by the Gulf Cooperation Council (GCC) governments due to the range of sensitive topics that think tanks are debating. The uprisings and revolts brought about by the Arab Spring have undoubtedly led to much dialogue and discourse about change among the citizens of the Arab world. Research think tanks are a relatively new phenomenon in the Middle East. Alshayji et al. previously explored the roles of prominent think tanks in the Middle East [9]. The tiny number of think tanks in the gulf region when compared to think tank figures around the world is an indication. Emphasis was placed on the urgent need to increase the role of think tanks in the region. Dr. Gidon Windecker [22] has argued that "scientific research and decision-making are still worlds apart in the region" and that the time has come to "bridge the gap through a more active role of think tanks." There is a need for political research centers in the GCC region to be more advanced and brought to the awareness of the public.

Think tanks in the Arab world are perceived as being repressed and carefully chosen by authoritarian regimes to push their agenda. At the same time, numerous prominent foreign policy think tanks are taking millions of dollars in donations from foreign governments in the Middle East seeking favorable research and connections to U.S. policymakers. Such is the case with Qatar. For example, the Washington, DC-based Brookings Institution received a $14.8 million donation from the government of Qatar [13]. Some scholars argue that these donations have led to implicit agreements that the think tanks would refrain from criticizing the donor governments. Although Qatar is the least restrictive Gulf state in terms of its treatment of its treatment of academic and media debates; and is home to more than 10 think tanks, the range of sensitive political topics in the nation, is highly controlled [3]. Saleem Ali, who served as a visiting fellow at the Brookings Doha Center, has said that he was explicitly told that he could not criticize the Qatar government [13]; Ali argues, "If a member of Congress is using the Brookings reports, they should be aware, they are not getting the full story. They may not be getting a false story, but they are not getting the whole story." An internal report commissioned by the Norwegian Foreign Affairs Ministry, clearly states that, "In Washington, it is difficult for a small country to gain access to powerful politicians, bureaucrats and experts. Funding powerful think tanks is one way to gain such access, and some think tanks in Washington are openly conveying that they can service only those foreign governments that provide funding" [11]. Thus, one can argue that these prominent think tanks are merely marketing propaganda to the highest bidder.

As think tanks in the Middle East are put under serve pressure, many are having a hard time executing their research effectively. The United Arab Emirates (UAE) has exercised extensive control over public debates, and therefore The Dubai School of Government had a difficult time operating its research center [22]. The government of Dubai has recently reduced funding to the center due to the sensitive and controversial political topics that were being discussed. Khalil Shikaki brings to light this issue by contending, "In the many authoritarian countries in the Middle East, ideas coming from outside the political elite are not considered important and can easily be silenced. Therefore, think tanks do not play a significant role in either making policy decisions or even formulating policy options. Nevertheless, they are still capable of having an impact" [20].

Proposed solutions

A popular way to handle scattered data is to construct the so-called fuzzy ontology as presented by Inyaem et al. [15]. The fuzzy membership value μ is used for the relationship between the objects in question, where $0<\mu<1$ and μ corresponds to fuzzy membership relationships such as "low," "medium," or high" for each object. The purpose of fuzzy control is to influence the behavior of a system by changing the inputs to that system according to the rule or set of rules under which that system operates. The purpose of applying fuzzy systems is to enable one to weigh the consequences (rule conclusions) of certain choices based on vague information.

Contribution knowledge

The fuzzy inference system contributes to understanding the context and perspectives that are important to the impact of political variables on strengthening bilateral economic relationships. The proposed technique efficiently utilizes algorithms to access, integrate, and manage the contributed information at the international level. Using object paradigm ontology and Protégé-OWL methods to contribute to understanding the domain as well as the relation between objects, the technique also contributes significantly to simplifying the concept by extracting the main variables that affect the decision process [5]. These methods facilitate implementation. In addition, they enhance the clarity of the natural concepts and encourage us to shed light on other,

more difficult domains, such as a parliament. Utilizing fuzzy logic contributes to the understanding of linguistic and imprecise data. The utilization of the fuzzy cognitive mapping (FCM) scheme provides insight into the interdependency variables (vague data). FIS is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computing. Its contribution lies in the secret of the calculations that automate dealing with imprecise language and vague information.

Methodology

1.2 Proposed ontology

Ontology facilitates the communication between the user and the system, and the success of the information systems is based on integration of information. Different methodological approaches for building ontology have been proposed in the literature [10, 12, 14, 18].

Two approaches are described in this paper, adopted from the ontology modeling approach of Noy and McGuinness [18] and Fernandez-Lopez [14]. The process of construction of fuzzy ontology is adopted from Inyaem et al. [15]. The main framework is to complete the construction of fuzzy ontology for a specific domain involves the following steps: 1) input unstructured data; 2) specify the definition of related concepts in the domain and their relationships; 3) clarify the generation of domain ontology; 4) extend the domain ontology to fuzzy ontology; and 5) apply the fuzzy ontology to the specific domain.

We will use the same developed model of fuzzy ontology for two reasons: 1) the authors used this model in the terrorism domain, which is considered an integral part of the political domain because terrorism undermines political stability; the model includes political variables such as "stability" and "terrorism" and 2) the author used linguistic variables and ambiguous concepts that are roughly equivalent to vague variables used in the political domain.

However, more sub-steps (processes) will be added within the main steps used by the Inyaem model [15]. The five new processes (sub-steps) are as follows: 1) construct object paradigm (OP) ontology; 2) apply ontology language OWL-editor from the World Wide Web Consortium (W3C); 3) construct fuzzy cognitive map theory FCM; 4) apply fuzzy causal algebra method; and 5) apply fuzzy inference system FIS.

Mechanism for using new sub-steps of fuzzy construction

Alshayji et al. [6] used an object paradigm (OP) ontology to identify important concepts and capture a high level for ontological conceptualization of knowledge to facilitate the work of decision processes [4, 5, 6, 8]. More details of OP were presented by Alasswad et al. [2]. Accordingly, this paper first presents the concept of the bilateral meeting concept by using an OP ontology, and the OWL editing tools ontology, and then proceeds to integrate fuzzy logic with ontology. Alshayji et al. [8, 9] used OWL to present the concept in the political domain [4, 5]. More justification for using Protégé was presented by Alshayji et al. [5], Islam et al. [16], and Noy & Guinness [18]. On the other hand, the third and fourth processes, which involve FCM and causal algebra, are especially applicable in the soft knowledge domains (e.g., political science, military science, international relations, and political elections at government levels). Alshayji et al. [7] demonstrated the causal inter-relationships between certain variables in the domain, such as "stability" and "terrorism," in addition the processes of fuzzy ontology construction presented in investment domains and the agreement ontology in political domains, respectively [4, 5, 6, 8].

Justification for using new sub-steps of fuzzy construction

In this regard, and coinciding with the previously mentioned process, the new sub-steps are added for two key reasons: to accelerate the application process for the construction of fuzzy ontology and to simplify the extraction of the most variables that in some way affect the political decision-making process. Political decision-makers would thus be aided by a system that would allow them to formulate constructive rule conclusions by dealing with vague variables as described and drawing rule conclusions in the form of an IF-THEN statement- an if-antecedent (input) and then-consequent (output). Because of this situation, and along with FCM and causal algebra propagation, the fuzzy inference process includes displaying what is going on in the political mind in the form of a calculation. This calculation uses fuzzy sets and linguistic models that consist of IF-THEN fuzzy rules. Fuzzy systems enable one to weigh the consequences (rule conclusions) of certain choices based on vague information. Rule conclusions follow from rules composed of two parts: the "if" (input) and the "then" (output). The fuzzy logic graphical user interface (GUI) toolbox enables us to build an FIS to aid in decision-making processes. For the purposes of the ontology, we refer the readers to Alshayji et al. [8]. Figure 1 depicts the complete process of the construction of fuzzy ontology.

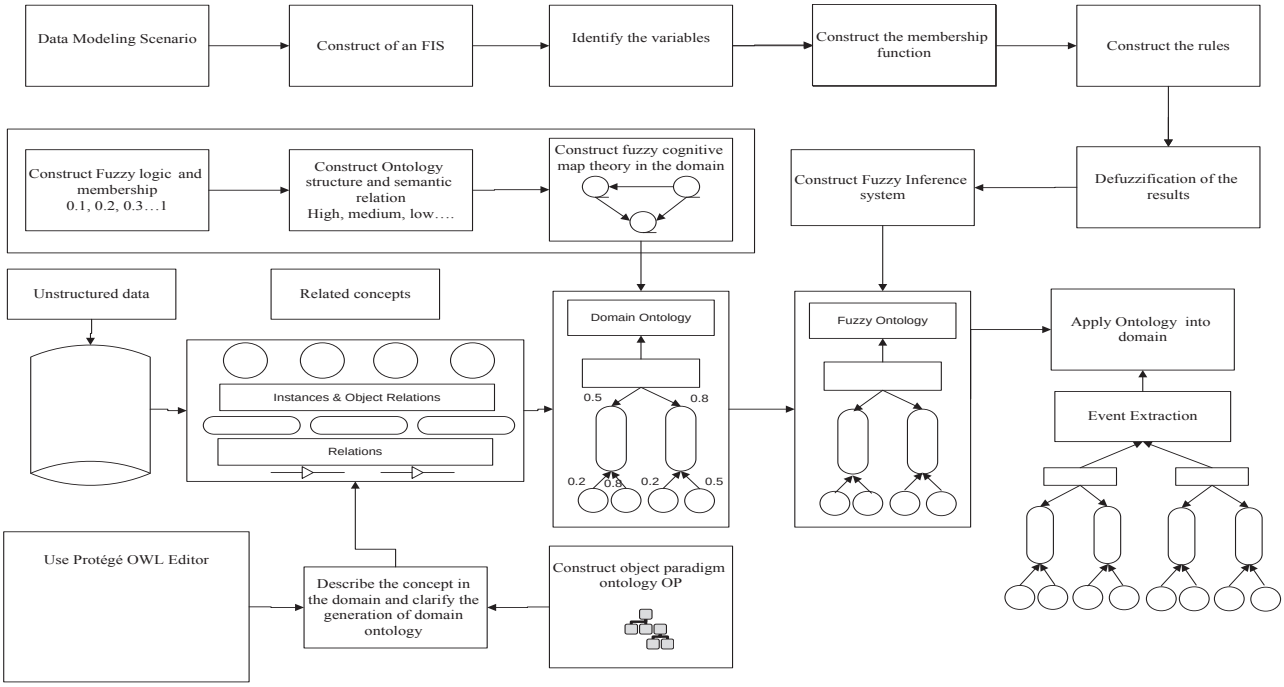


Figure 1: Process of Construction of Fuzzy Ontology with inference system for the Specific Domain

Specifying the definition of related concepts – Illustrative case: Bilateral meetings

Bilateral meetings (minutes of meetings) occur between friendly countries through the meetings of heads of state or envoys, whether the prime minister or the minister of foreign affairs. Such silent thoughts are found in the documents of bilateral meetings, so that we can understand the formal talks between friendly countries. In a simple manner, when the subject changed during the speech, this meant that the speaker had an unwillingness to talk about the topic. In the modern formulation, this indicates the rejection of the desires required, in other words, the refusal of friendly state requests. We consider that it is possible to convert this matter to such a parameter, but most of the time, the terminology is too difficult to analyze and understand. For instance, it is important not to limit the answers to sentence such as "we will think about it," "we have no decision in this matter," "it must be passed through the parliament thread,". " The parliament is an obstacle standing in front of me," and "we all suffer from this problem." Those answers create difficult challenges for the programmers and analysts in information systems. All such previous answers must be converted into other parameters and in different degrees so that we can finally get accurate results for the outcome of the final interview. In other words, we can depict the power of the political relationship by understanding such variables. Undoubtedly, this change was made for plural agreement with variables. This sheds light on the proper planning needed to build and strengthen the investment relationship between nations.

Specifying the definition of related concepts

In this section we will apply the proper ontology to identify some concepts of silent thoughts found in bilateral meetings. Such silent thoughts are found in the documents of bilateral meetings, so we can understand the formal talks between friendly countries. The proper ontology was identified by Alshayji et al. [5], who used the loan as case study, in this section, we will use the bilateral meetings as a case study. To capture all concepts of bilateral meetings, the "result" class is linked to the "negative" class through the "hasWeak" tuple type. In addition, to capture the "positive" result, the "P-strong" class is linked to the "positiveAnswer" class. To be more semantically precise, the engineering process links link with all concepts that related to bilateral meetings.

According to the OP, the process starts with the selection of the concept, followed by the analysis of its spatial and temporal dimensions. The aim is to have a clear conceptualization of the bilateral concept while extracting the concepts that exist within the bilateral meetings document. This is done by considering all official talks in order to create a bilateral meetings ontology. The result of the bilateral meetings concept will be identified by using OP ontology, through which we will analyze the “results” concept. Mainly, each result has a different type such as “negative answer,” “positive answer,” “neutral,” etc. In addition, the result is submitted on a specific date. This date requires analysis in order to track change over time, as OP considers the temporal dimension, thus enabling changes over time (see Figure 2).

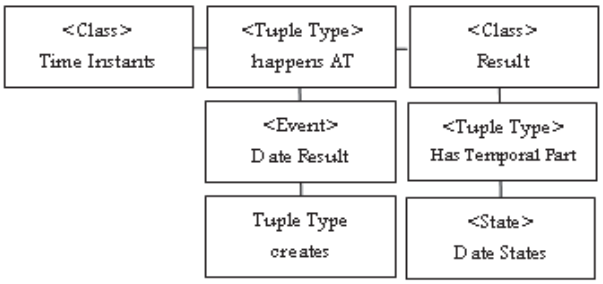


Figure 2: Engineering relationships of the result ontology

We considered the important information in the bilateral meetings concept. This information enhances the semantic presentation, and such enhancements may also significantly affect the quality and performance of the implemented software system. Thus, more details enable an ontology to provide a more faithful presentation.

Using OWL ontology

The construction of the bilateral relation "result" is also presented by the Protégé OWL editing tool in Figure 3.

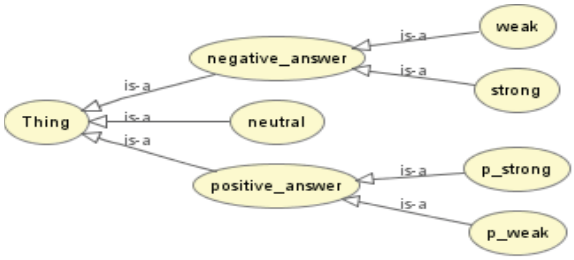


Figure 3: Result of bilateral meetings ontology by Protégé OWL

At this stage we have 1) input unstructured data; 2) specified the definition of related concepts in the domain and their relationships; and 3) clarified the generation of domain ontology. In the next section, we will extend the domain ontology to fuzzy ontology.

Extending ontology to fuzzy ontology

At this point, it is important to understand and specify the classes in the bilateral meetings domain and generate fuzzy ontology.

Fuzzy set and membership

In this section, we will integrate fuzzy logic in our ontology. Fuzzy logic, as presented by Abulaish and Dey [1] and also Alshayji et al. [5, 8, 9] has different properties. More fuzzy concepts in the same domain have been presented [5]. Integrating information with rich concepts undoubtedly helps political decision-makers make correct decisions. Answering whether to "prevent" or "redirect" the bilateral economic relationships requires also considering the concept of an "investment indicator."

Fuzzy Cognitive Map Theory

FCM is a fuzzy-graph structure for representing causal reasoning with a fuzzy relationship to a causal concept [7]. Justification for its use is described in subsections 1.6 and 2.3; more justification can found in the literature [19, 7]. Signed fuzzy non-hierarchic digraphs and metrics can be used for further computations, and causal conceptual centrality in cognitive maps can be defined with an adjacency matrix [7, 17].

Use of Fuzzy Casual Algebra

This work seeks to clarify the relationships between concepts and to elucidate the positive or negative effects on each concept while clarifying knowledge of the relationships. Furthermore, an FCM structure allows systematic causal propagation, Arrows sequentially contribute to the convenient identification of the causes, effects, and affected factors [7, 17]. Figure 7 has seven variables that describe the impact of some conditions on bilateral economic relationships and causal variables. For example, (C1→C2, C1) is said to impact C4. This is apparent because C1 is the causal variable, whereas C4 is the effect variable. Suppose that the causal values are given by p {none ≤ some ≤ much ≤ a lot}. The causal relationship between concepts and the effect of these relations were presented by Alshayji et al. [9]. The FCM appears below in figure 4.

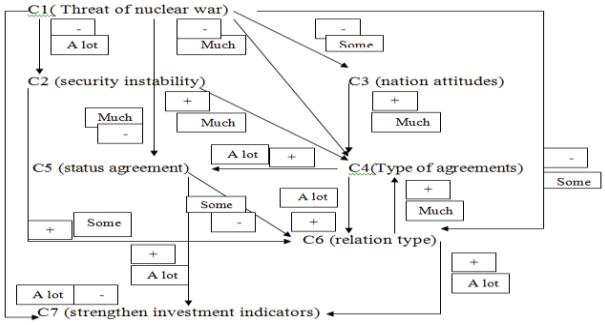


Figure 4: A fuzzy cognitive map on the impact of strengthening economic bilateral relationship

Inference system in the political domain

Incorporating the concept of the specific domain, this step applies a method that can deal with dismantling each variable to several parameters. Decision-makers would be aided by a system that would allow them to formulate constructive rule conclusions by dealing with several parameters (membership) for each variable. Alshayji et. al previously described the advantage of GUI tools in MATLAP [9] and the capability of building a productive graphical fuzzy inference system . There are five primary GUI tools for building a fuzzy inference systems: 1) the FIS editor; 2) the membership function editor (MFE), which allows users to define and shape the membership

function associated with the input and output variables of the FIS; 3) the rule editor, for editing the list of rules that define the behavior of the system (IF-THEN); 4) the rule viewer, a technical computing environment for diagnosing the behavior of specific rules and viewing the details; and 5) the surface viewer, which generates a 3D surface from two input variables and displays their dependencies. Further figures were presented by Alshayji et al. [9].

Data and modeling scenario

As mentioned in section 2.2 we need to collect all input/output data in a form that can be used by inference. Alshayji et al. [9] also presented the five primary GUI tools for building, editing, and monitoring FISs. Therefore, a need emerges for giving different interpretations according to the context. Table 1 presents the proposed "InvestmentIndicatorName" class with linguistic and semantic properties.

Country name	Investment Indicator
A	Encourage strongly
B	Encourage
C	Encourage weakly
D	Prevent
E	Caution
F	Redirect

Table 1. Fuzzy logic assigned to "CountryName" and "InvestmentIndicator"

For example, input 1 is "Bilateral Meetings result." In this step, we need to add the parameters for the "Bilateral Meetings" input, so we need to define all inputs and their values. The second step uses the membership function editor. In the third step, which involves the rule editor, we need to construct the rules; for example, we construct the first two rules as follows:

if (Bilateral Meetings result is strong negative), then (investment is redirect) (1).

if (Bilateral Meetings is strong positive), then (investment is encourage strongly)

These rules are verbose. The result is an extremely compressed version of the rules in a matrix where the number of rows is the number of rules and the number of columns is the number of variables, as follows:

1 1 0 0 0 0 0 0 0 0 0 0 0 3, 1 (1): 1

2 2 2 0 0 0 0 0 0 0 0 0 0 3, 3 (1): 1

Using such functions in the political domain provides the opportunity to choose a membership value with infinite accuracy. Reading across the first row, a literal interpretation of rule 1 is "input 1 is MF1" (the first value for the membership function associated with input 1). This means that from the first input (bilateral meeting result) we select {strong negative}, the value for the membership function associated with input 1. Continuing across, MF1 from input 2 was selected, and so on. Obviously, the functionality of this system does not depend on how well the operator named the variables and membership functions and does not even bother with variable names. The next step is to use the rule viewer to display the whole fuzzy inference process. The construction of the rules editor is presented in Figure 5.

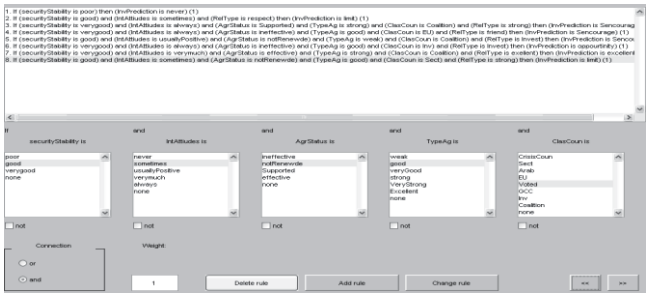


Figure 5. The rules in the Rule Editor, in the verbose form

The decision will depend on the input values for the system. The defuzzified output (value) and the fuzzy inference diagram containing the calculation were displayed previously by Alshayji et al. [9]. The fourth step, using the rule viewer, was presented by Alshayji et al. [9]. Some information about inputs, memberships, and output (variables in the system) is presented in the following system:

ows:Name="Investment3,"Type="mamdani,"Version=2.0, NumInputs=13,NumOutputs=1,NumRules=8,AndMethod="min, "OrMethod="max,"ImpMethod="min,"AggMethod="max,"Def uzzMethod="centroid,"(Input1),Name="bilateral meeting,"Range=(01),NumMFs=3,MF1="weaknegative": "trimf"(-0.5 0 0.5),MF2="positive":"trimf"(0 0.5 1), MF3="strongpositive":"trimf"(0.511.5),(Input2),Name="Politic alStability."

The fifth step is using a surface viewer that generates a 3D surface from two input variables and displays their dependencies.

Conclusion and Future Work

This paper focuses on developing a fuzzy inference system in the political domain to handle imprecise data by controlling information uncertainty. We have built a fuzzy inference system that has stronger abilities for expressing uncertain linguistic variables. Our further research lies in the automatic generation of fuzzy ontology from more fuzzy systems. In addition, more research needs to be done about silent thought, the idea of applying the political centers' work in order to support top decision-makers, In further research, we need to improve the

application and development of technical political centers. We will present the intended functions of such political centers and the challenges that hinder their work.

References

[1] Abulaish, M. and Dey, L. "Interoperability among distributed overlapping ontologies: A fuzzy ontology framework". Proceedings of the 2006 IEEE/IWC/ACM International Conference on Web Intelligence, 2006.

[2] Al Asswad, M. M., Al-Debei, M. M., de Cesare, S. and Lycett, M. "Conceptual modeling and the quality of ontologies: A comparison between object-role modeling and the object paradigm". Proc. 18th European Conf. Information Systems, Pretoria, 2010.

[3] Al-Ibrahim, H. "In Qatar, do think tanks matter?". Brookings Doha Center, Doha, Qatar, 2011.

[4] Alshayji, S., El Kadhi, N. and Wang, Z. "Building Fuzzy-Logic for Political decision-maker". D 20, E 814, Naun.org/International journal on Semantic Web, Romania, 2011a.

[5] Alshayji, S., El Kadhi, N. and Wang, Z. "Building ontology for political domain". 2011 International. Conference on Semantic Web and Web Services. SWWS'11'. Inspec/IET/The Institute for Engineering & Technology; DBLP/CS Bibliography; CNRS, INIST databases, Las Vegas, USA, 2011b.

[6] Alshayji, S., ElKadhi, N. and Wang Z. "Fuzzy-based ontology intelligent DSS to strengthen government bilateral economic relations". Kcess'11 (Second Kuwait Conference on e-System and e-Services), April 9, ACM 978-1-4503-0793-2/11/04, 2011c.

[7] Alshayji, S., ElKadhi, N. and Wang, Z. "Fuzzy Cognitive Map Theory for the political Domain". Federated Conference on Computer Science and Information System, The Fed CSIS'2011 September 19-21, Szczecin, Poland, IEEE Digital library CEP1185N-ART, 2011d.

[8] Alshayji, S., El Kadhi, N. and Wang Z. "On fuzzy-logic-based ontology decision support system for government sector." 12th WSEAS International Conference on Fuzzy Systems, Brasov, 34, 2011e.

[9] Alshayji, S., Al-Sabah, N. and Al-Sabah A. "Building Fuzzy Inference System in the Political Domain." 12th International Conference on Scientific Computing (CSC'14), Las Vegas, USA, 2014.

[10] Beck, H. and Pinto, H. S. "Overview of approach, methodologies, standards, and tools for ontologies". Agricultural Ontology Service (UN FAO), 2003.

[11] Bjorgaas, T. "From Contributor to Partner". Noref, Norweigan Peacebuilding Resource Centre. May 2012.

[12] Calero, C., Ruiz, F., and Piattini M. "Ontologies for software engineering and software technology". Springer-Verlag, Berlin, Heidelberg, New York, USA, 2006.

[13] Confessore, N. Lipton, E. Williams, B. "Foreign Powers Buy Influence at Think Tanks". New York Times, Sept 6, 2014.

[14] Fernandez-Lopez, M. "Overview of methodologies for building ontologies". Journal Data & Knowledge Engineering, 46, 2003.

[15] Inyaem, U., Meesad, P. and Haruechaiyasak C. "Dat Tran: Construction of fuzzy ontology-based terrorism event extraction". Third International Conference on Knowledge Discovery and Data Mining, IEEE. DOI 10.1109/WKDD.113, 2010.

[16] Islam N., Abbasi A. Z. and Zubair A. "Semantic Web: Choosing the right methodologies: Tools and Standards". IEEE, 2010.

[17] Kosko, B. "Fuzzy cognitive maps". London: Academic Press Inc., 65-75, 1986.

[18] Noy, N. and McGuinness, D. "Ontology development 101: A guide to creating your first ontology", 2001.

[19] Sharif, A.M. and Irani Z. "Knowledge dependencies in fuzzy information systems evaluation", 2005.

[20] Shikaki, K. "Ideas and Influence in Middle East Politics: The Role of Think Tanks". The Washington Institute of New East Policy. Policy #207, 1996.

[21] Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Stanford Medical Informatics Technical Report SMI-0880.

[22] UPI. "Gulf states tighten grip on think tanks". United Press International, 2011.

[23] Windecker, G. "The role of think tanks in the gulf region: potential, challenges, and benefits". Konrad-Adenauer-Stiftung, Berlin, Germany, 2011.

An Effective Methodology for Processing and Managing Massive Spacecraft Datasets

Haydar Teymourlouei
Department of Computer Science
Bowie State University,
Bowie, MD, USA

Abstract - *The emergence of enormous and complex datasets has made existing data processing methods more strenuous. The growth in datasets continues to increase vastly. Despite the Interpolation technologies out there to manipulate efficient large data searching methods, the task to search for datasets expeditiously is still an obstacle. However, this research offers a more effective method to quickly search a large dataset within a timely manner. For this method to work, we used a technique where we create a directory file to catalogue and to retrieve data. The directory file is where the file can be acquired based on its time and date. The proposed method is intended to alleviate the process of searching a data's content entirely and to scale down the search time in order to find the data file.*

Keywords: Big data, data processing, data set, interpolation search, raw data, unsorted data

1 Introduction

As technology is expanding, so is the rapid acceleration of complex and diverse types of data results in the emergence of a fast paced algorithm. The sheer amount of data generated that must be ingested, analyzed, and managed is of relevant importance and must be considered when attempting to propose useful tools. The speed in which data must be received and be processed must also be considered. The rise of information coming from new sources has taken a toll on IT . Therefore, data management is a much more difficult task using only the traditional methodologies. Data has attained the form of continuous data streams rather than finite stored data sets, posing barriers to users that wish to obtain results at a preferred time. Data prescribed in this manner displays no bounds or limitations; thus, a delay in the retrieval of data can be expected.

With today's overflowing datasets, data management and analysis challenges are on a rise. Large data is described as a substantial amount of data accumulated over time that makes it difficult to examine and to process using various algorithms. Data analysis is the process of understanding the meaning of the data we have acquired, catalogued, and exhibited in representation's form such as a table or a line graph. Working with millions or even billions of datasets has become problematic for researchers. If we can sort, approach, allocate, and evaluate the datasets competently, we can alleviate the trouble in searching for data. Many researchers believe that using various indexing methods to search for data expedites the search mechanism [2].

2 Data Processing

The term data is typically described as information. Data processing is basically conversion of raw data into meaningful information through process. Data is first gathered, then it gets processed. Large datasets usually refer to voluminous data beyond the capabilities of the current database technology. Data is used to refer to the vast amounts of information in a standardized format. Generally, data can include numbers, letters, equations, images, dates, figures, maps, documents, media files, and much more information. Particularly, data processing is a distinctive step in the information processing cycle. In information processing, "data is acquired, entered, validated and processed, stored and outputted, either in response to queries or in the form of routine reports. Data processing refers to the act of recording or otherwise handling one or more sets of data" [5]. The processing of data requires to be displayed in an understandable and efficient form. The levels must be given consecutively in order from the pathway to the result where is its readable to the reader. Nonetheless, large datasets are transforming the way research is carried out, resulting in the emergence of a fast-paced algorithm.

Data has to be effectively processed in order to convert raw data into meaningful information. See Figure below for details.

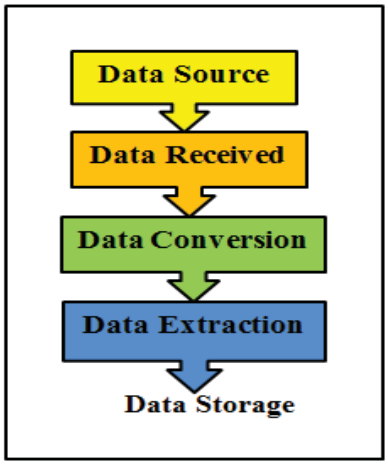


Figure 1: Raw Data Conversion

To obtain large data from satellite and to transform raw data into a simple and utilizable form requires tedious data processing. In order to perform such procedures, dynamic component of satellite operations are needed. It is potent to articulate how to analyze the data in order to extract intriguing trends and patterns as huge amounts of data are being formed. Level 1 and level 2 data processing proceeds by operating on the raw data and edited data. Level 3 and level 4 edited data is being calibrated then resampled. Level 5 and level 6 data is obtained from maps, reports, or graphics, etc. then ancillary data is calibrating or resampling data sets. Level 7 and level 8 used correlative data to interpret space-based data sets and then user description data is available for secondary user to extract information from the data.

Table 1: Data Processing Level [7]

Level	Type	Data Processing Level Description
1	Raw Data	Telemetry data with data embedded.
2	Edited Data	Corrected for telemetry errors and split or decommutated into a data set for a given instrument. Sometimes called Experimental Data Record. Data are also tagged with time and location of acquisition. Corresponds to NASA Level 0 data.
3	Calibrated Data	Edited data that are still in units produced by instrument, but that have been corrected so that values are expressed in or are proportional to some physical unit such as radiance. No resampling, so edited data can be reconstructed. NASA Level 1A.
4	Resampled Data	Data that have been resampled in the time or space domains in such a way that the original edited data cannot be reconstructed. Could be calibrated in addition to being resampled. NASA Level 1B.
5	Derived Data	Derived results, as maps, reports, graphics, etc. NASA Levels 2 through 5.
6	Ancillary Data	Nonscience data needed to generate calibrated or resampled data sets. Consists of instrument gains, offsets, pointing information for scan platforms, etc.
7	Correlative Data	Other science data needed to interpret space-based data sets. May include ground-based data observations such as soil type or ocean buoy measurements of wind drift.
8	User Description	Description of why the data were required, any peculiarities associated with the data sets, and enough documentation to allow secondary user to extract information from the data.

If the volume of raw data is too large, making more than a single pass over the data may not be achievable. Processing levels offer adequate approaches to dissect useful information from huge data sets by generating small passes over the data. Lots of information can be gathered simply from making a single pass over the data or a small number of passes over the data. The extent of data processing enforced to a data product establishes massive significant characteristics of the product. It also establishes if the particular metadata elements or data services are suitable to the product. For georectified example, geographic coordinate - based data sub-setting can be easily implemented in the georectified raster data, but the same type of service is difficult to be provided to raw remote sensing images. In order to facilitate the data management and standardize the metadata and data services, data products in EOSDIS are classified into five levels according to the degree of processing. The higher the level, the higher the degree of processing [3].

2.1 Unsorted Data

Accessing unsorted data is not the issue anymore; instead it is about extracting valuable information from the unsorted data. Before data can become "information," the data needs to be extracted, organized, and at times analyzed and formatted to be presented. Unsorted data is new and original data that has yet to be touched or modified, in other words unprocessed and unorganized. Raw data can be anything from a series of numbers, the way those numbers are sequenced, even the way they are spaced, but they can yield very important information. A computer interprets this information in a way that attempts to make sense to the reader [9].

Raw data is a data that has been captured from spacecraft which is not in presentable form. It needs to be processed to gather meaningful and relevant information and is also known as source or atomic data. The data is completely unrecognized and needs to be processed manually or by the machine. Raw data could be either hex, binary data, or characters. A computer may interpret this information and give a readout that then may make sense to the reader. Once raw data is collected, it goes into the database which later becomes available for supplementary processing and examining. A good example would be hex unsorted data. As you see below, hex unsorted data is very complex to read. To make the hex data readable, the data has to be processed through several levels.

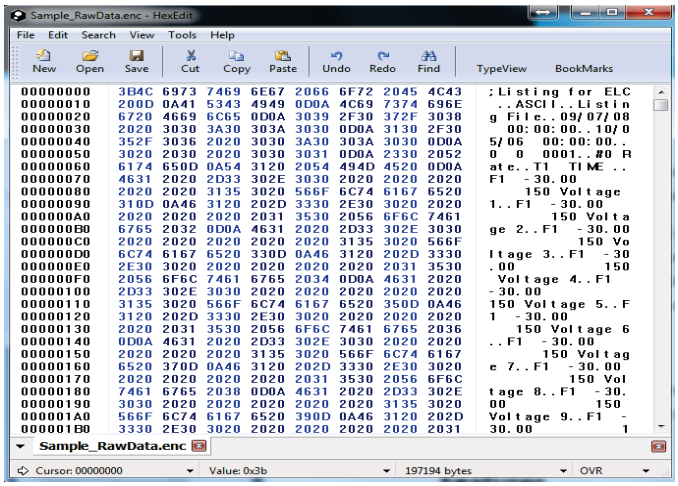


Figure 2: Example of Hex Unsorted Data

Once data has been generated, processed, and stored, it can then be made available in a more useful form for scientists' research.

2.2 Massive Data Challenges

Due to the massive amount of data that is being collected, storage space, management, and analyzing the data are big challenges. Other problems also include the increasing of volume, velocity (speed), and variety (data type). For these problems to be resolved, scalable computing and analyzing methods must improve. Massive data is the size in which common software is not capable of handling. According to NASA, they gathered approximately 1.73 gigabytes of data from our nearly 100 currently active missions! We do this every hour, every day, every year and the collection rate is growing exponentially. Handling, storing, and managing this

data is a massive challenge. Big data is very simply a collection of data sets so large and complex that your legacy IT systems cannot handle them. Approaching the big data challenge often necessitates Interpolation algorithms, infrastructure, and frameworks [8].

Data gathering comes in a variety of different sources. Typically, there are two major concerns involved with analyzing massive datasets: (i) the period of time required to search through data files and (ii) how to effectively identify relationships between data. Certainly, this requires knowledge of what the scientist is looking for; for instance, what is considered to be an anomaly (e.g. calibration problems with instruments). Therefore, the algorithms must be able to search through and efficiently manipulate massive data sets [9]. In order for such certainty to vanish, data has to be analyzed reasonably rapid in order to present users with the capability to gather data in a short amount of time.

NASA faces major challenges throught out daily activities's every day. NASA's big data challenge is not just a global challenge, but often appears as an unorthodox challenge. NASA describes many of their "Big Data" sets to be substantial metadata. The term metadata is often referred to as data that describes other data, which can make finding and working with particular data more efficient and easier. When NASA engages in spacecraft missions, they have two very different types of spacecraft. One deep space spacecraft that allows them to send back data in MB/s, which is the equivalent speed of 1,000,000 bits of information being sent back per second. When NASA is not using a spacecraft to retrieve data, they also send out earth orbiters, allowing them to send back data in GB/s per second. NASA typically uses these two types of spacecraft's because they regularly engage in missions where data is continually streaming from one spacecraft on Earth and in space, faster than they can store, manage, and interpret it. In NASA's current mission, they are allowing data to be transferred through radio frequency, which is currently the most inefficient frequency due to its slow speed. NASA is working on employing a new type of technology that uses Optical Laser communication to increase their transfer, which would result in 1000x (times) more increase in the volume of data. Allowing this type of data transfer is much more than what they can handle today, but in preparing for new technology today, NASA will be prepared for the future. NASA is planning future missions today that will allow them to stream more than 24 terabytes a day. For example, if NASA has the ability to easily stream 24 TB a day, that's the equivalent speed of roughly traveling 2.4 times the entire Library of congress every day just for one mission.

Allowing data to be transferred at those speeds, it is still relatively expensive to transfer one bit of information from a spacecraft. Once data travels to their data centers, being able to store, mange, visualize and analyze the data becomes a concern. For example, since everything changes with time, the projected number of the climate change data sources are expected to grow nearly 350 Petabytes in 2030, which is only 15 years away. Even though that may seem like a long time from now, a change in five petabytes a year is equivalent to the number of letters delivered by the US postal service in one year.

One awesome sample of the remarkable test that we confront with overseeing space information is simply beginning to be exhibited by the Australian Square Kilometer Array Pathfinder (ASKAP). The venture, which is an extensive show made up of 36 receiving wires, every 12 meters in distance across, spread out more than 4,000 square meters however; cooperating as a solitary instrument to open the riddles of our universe. Furthermore, the spacecraft is by all account not the only source of our information, because of a perpetually developing supply of cell phones, ease sensors, and online stages. The size of the enormous information challenge for NASA, in the same way as other associations, is overwhelming. As you can likely foresee, the expanding information volumes are not our only difficulties. As our abundance of information expands, the test of indexing, seeking, and exchanging, thus on all increment exponentially also. Moreover, the expanding multifaceted nature of instruments and algorithm's, expanding rate of innovation invigorate, and the diminishing plan environment, all play a critical factor in our approach.

3 Interpolation Search

Interpolation search is a method of retrieving a preferred data by key in an ordered file by using the value of the key and the statistical distribution of keys. It is an algorithm for discovering a given key in a sorted array. It tries to anticipate where the key would lie in the search space through a straight insertion, decreasing the search space to the part before or after the assessed position if the key is not found there. This technique will work if figuring on a distinction between key qualities are sensible. It is a modified algorithm of binary search, reducing the complexity. While in binary search, algorithm finds the position of a specified input value "Search "key" within an array sorted by the key value. For binary search, the array should be arranged in ascending or descending order. Binary search constantly selects the middle number for comparison, discarding one half of the search space. It is an algorithm to efficiently find the indexed array that has been ordered by the values of the key. In each search step it calculates where in the remaining search space the wanted item might be, based on the key values at the bounds of the search space and the value of the wanted key. The key value found at the estimated position is then compared to the key value being searched. The remaining search space is then reduced to the part before or after the estimated position based on the comparison if it is not equal.

Interpolation search is an alternative to the binary search that exploits information about the primary supply of data that is being searched. By utilizing this extra data, interpolation search can be as quick as O (log(log (n))), where n is the size of the array. Interpolation search models how people seek a word reference better than a binary search. For example, on the grounds that if a human were to search "Yellow", they

would instantly flip towards the end of the dictionary to find that statement, instead of flipping to the center. This is the basic thought of how interpolation search functions.

Interpolation search is a search algorithm impacted by binary search that, for some information sets, performs asymptotically better. Both binary and interpolation searches oblige the information to be sorted, and utilize the sordidness to rule out sections of the data from consideration. They work by picking a component at an arbitrary position, contrasting it with the component being referred to, then choosing whether to proceed with the search on the left or the right. The key distinction is that binary search always works by splitting the input range perfectly in half, which guarantees a runtime of O (lg n). Interpolation search works by assuming that the data is distributed uniformly, then doing a linear interpolation between the endpoints to guess where the element ought to be. Assuming the data is distributed uniformly, it can be shown that interpolation search runs in expected O (lg lg n) time, exponentially faster than binary search.

3.1 Interpolation Search Algorithm

1. **Inputs:** Positive integer n and sorted array of n elements and the key to be searched
2. **Bottom:** = array[i]
3. **Top:** = array[j]
4. **if** size < Bottom **then**
5. **Return** 0
6. **if** size ≥ Top then i: = j
7. loop invariant: size ≥ Bottom and size ≤ Top
8. **While** (Bottom <= Top && Top >= key) do
9. **Middle:** = Bottom + (Top-Bottom)*((Key-Array [Bottom]) / (Array [Top] - Array [Bottom]))
10. **if** (key == Array [Middle]) **then**
11. **Return** Middle
12. **else if** (key < Array [Middle])) **then**
13. Top = Middle-1
14. **else**
15. Bottom: = Middle +1
16. **End** while loop
17. **Return** i

3.2 Example of Interpolation Search

1. Array [] = { 0,2,4,7,8,10,12,15,16,18,20,23,24,27,28,40}
2. Middle = Bottom + (Top-Bottom)*((Key-Array[Bottom]) / (Array[Top] - Array[Bottom]))
3. Middle = 0 + ceil (15 – 0)* ((20 – 0) / (30 – 0)) = 0 +10 =10
4. Middle = 10 index key is 20

Search for the key 20 in the following array using Interpolation search

3.3 Interpolation Complexity Search

Complexity is often divided in two ways, time complexity and space complexity. Time complexity is the amount of time the computer requires executing the algorithm. Space complexity is an algorithm that computes the amount of memory space the computer requires. The complexity of an algorithm is a function g(n) that gives the greater guarantee number of operations performed by an algorithm when the input size is n. In the case of algorithm searching, this is the process to finding the location of the given data elements in the data structure. The Interpolation algorithm search for random order data and conquer running time is O(lg lg n), that is better than binary search.

Best/Average case complexity is O(log log N) where N is the number of keys if they are uniformly distributed.

Worst case complexity is O(n) example searching for 1000 in 1,2,3…..,888,1000,109

The figure below shows the comparison of the interpolation and binary search complexity algorithms. The results show there is a huge time difference between the two searches of finding the desired value. The Interpolation complexity search algorithm uses log(log (n)) where the result was found at about five seconds, meanwhile binary search uses log (n) where the desired key was found at about 27 seconds.

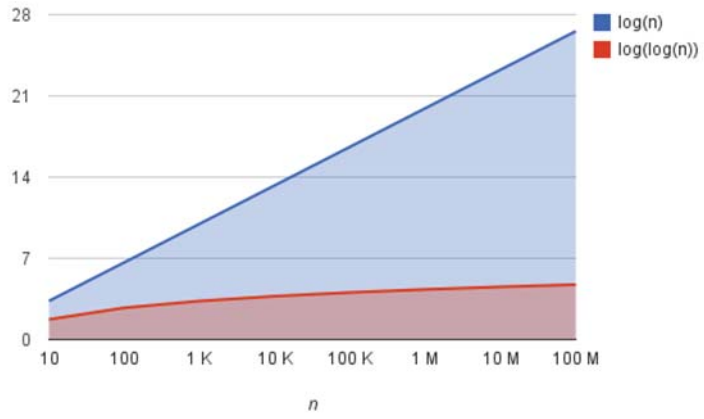


Figure 3: Interpolation vs. Binary Search Complexity

4 Methodology

The motivation behind the proposed algorithm is to present the user with the ability to search specific data in a timely fashion. This method is efficient for obtaining only a selected partition of the scientific data. In other words, after data has been reduced to selected files, the algorithm carries out a search on these files. One is looking for a certain information from numerous files where each item is stored as a different entity of files. Followed by gathering feedbacks from reviewers, one then wants to examine all incidents of the specific item that appears in the multiple files. To simplify the searching, a directory file will be necessary. A directory file can point to the file index start and end time of the search data within a matter of time. A directory file can help catalogue and retrieve data much faster. A file directory is a place where files are stored in a computer. File retrieval is found based on its time and data stored in a directory file. With this method, data retrieval process is reduced and the amount of time to locate the selected file is condensed. Each data file holds a start time and an end time.

Table 2: Sample of Directory File

Files	File Name Nu	Start Time	End Time	Start Block. number	End Block number
Index 1	Filename 1	YYYY MM DD 00:00	YYYY MM DD 20:59	0	Total Num block
Index 2	Filename 2	YYYY MM DD 20:59	YYYY MM DD 22:59	Total Num block +1	Total Num block
Index 3	Filename 3	YYYY MM DD 22:59	YYYY MM DD 23:59	Total Num block +1	Total Num block
:	:				
Index +N	Filename +N	YYYY MM DD 00:00	YYYY MM DD 00:00	Total Num block +N	Total Num block
		Sample Index file			
1	C:\Data_2014_136.dat	2014 May 16 00:00	2014 May 16 23:59	0	1438
2	C:\Data_2014_137.dat	2014 May 17 00:00	2014 May 17 23:59	1439	2877
3	C:\Data_2014_138.dat	2014 May 18 00:00	2014 May 18 23:59	2878	4316
4	C:\Data_2014_139.dat	2014 May 19 00:00	2014 May 19 23:59	4317	5755
5	C:\Data_2014_140.dat	2014 May 20 00:00	2014 May 20 23:59	5756	7194
6	C:\Data_2014_141.dat	2014 May 21 00:00	2014 May 21 23:59	7195	8633
7	C:\Data_2014_142.dat	2014 May 22 00:00	2014 May 22 23:59	8634	10072
8	C:\Data_2014_143.dat	2014 May 23 00:00	2014 May 23 23:59	10073	11511
9	C:\Data_2014_144.dat	2014 May 24 00:00	2014 May 24 23:59	11512	12950
10	C:\Data_2014_145.dat	2014 May 25 00:00	2014 May 25 23:59	12951	14389
11	C:\Data_2014_146.dat	2014 May 26 00:00	2014 May 26 06:15	14390	14764

More specifically, this study proposes the development of an algorithm that generates a directory file that contains the following information:

- File index;
- File name;
- Start and end time for every file;
- Start address and end address (position of the file).

This will provide a tool for rapidly accessing data within these parameters.

4.1 Experiment (Binary Search, Linear Search, and Interpolation Search)

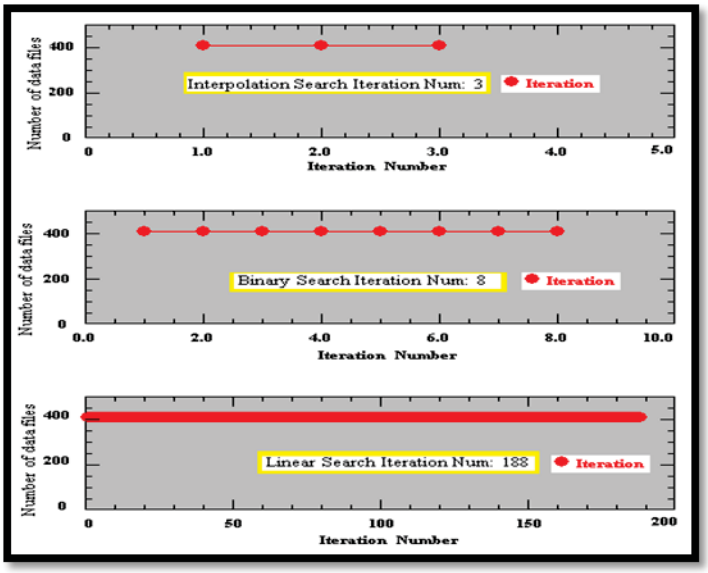
Search algorithms are used to check and find an element from a very large list of elements. There are many different search algorithms but we will be doing comparison between binary, linear, and interpolation search algorithm.

Binary search: search the complete sorted list which is divided into two parts. First compare an input value to the middle element of the array. This limits us to check only the second list in subject to if the input value comes from the right or left of the middle element. This will decrease the length that has to be searched to search for an element from the complete sorted list. This algorithm searches minimum possible comparisons. This makes the binary search more efficient than the linear search.

Linear search: is a basic and simple way of searching by finding a certain value in a list that contains of checking every one of its elements, one at a time and in order, until the search is found. Mainly, each element in an array is read sequentially and then compared with others elements. A successful search will be once all the elements are read and the preferred element is not found.

Interpolation search: also refer as extrapolation search. This algorithm searches for a given key value in an indexed array which has been ordered by values of the key. It uses doubly logarithmic with the values in {ai} distributed relatively equal, to have good time complexity.

Table 3: Comparisons of Search Algorithms Result

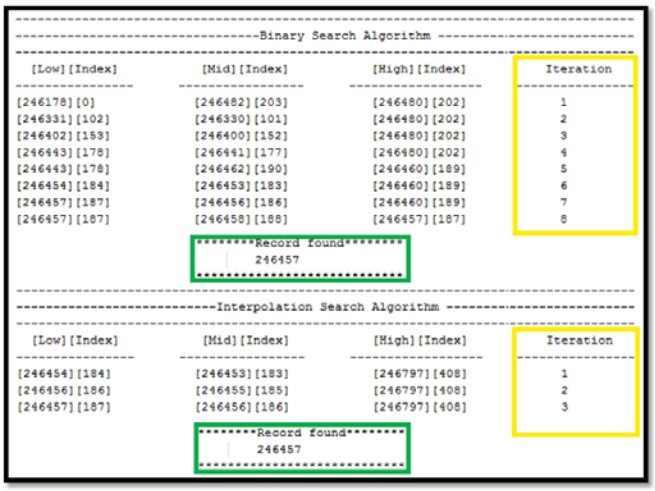


5 Results

The proposed algorithm can adequately extract relevant information from a vast quantity of data, and this is done with less iteration than the existing methods. When examining the linear search algorithm, we took into consideration the number of iterations that occurred and determined ways to decrease them. The greater the number of iterations, the longer the delay for the retrieval of data; therefore, the number of iterations must be reduced to provide quicker replies. The attained outcomes suggest that this algorithm is effective in making data available at any time. The table above compares three search algorithms: binary search, linear search, and interpolation search. Both linear and binary search algorithms generate a massive number of iterations. However, the proposed interpolation search algorithm has successfully reduced the number of iterations.

Table 4: Sample Data File

Table 5: Interpolation vs. Binary Iteration Results



Binary Search Algorithm

[Low][Index]	[Mid][Index]	[High][Index]	Iteration
[246178][0]	[246682][203]	[246480][202]	1
[246331][102]	[246330][101]	[246480][202]	2
[246402][153]	[246400][152]	[246480][202]	3
[246443][178]	[246441][177]	[246480][202]	4
[246443][178]	[246462][190]	[246460][189]	5
[246454][184]	[246453][183]	[246460][189]	6
[246457][187]	[246456][186]	[246460][189]	7
[246457][187]	[246458][188]	[246457][187]	8

********Record found********
246457

Interpolation Search Algorithm

[Low][Index]	[Mid][Index]	[High][Index]	Iteration
[246454][184]	[246453][183]	[246797][408]	1
[246456][186]	[246455][185]	[246797][408]	2
[246457][187]	[246456][186]	[246797][408]	3

********Record found********
246457

The above table shows the proposed algorithm takes less time and iteration than binary search, which is more appropriate for researchers' purpose. Binary search algorithm is used widely to search and sort data in an efficient manner. The binary search algorithm took ten iterations to arrive at the selected data whereas the proposed algorithm took only two iterations to arrive at the desired data.

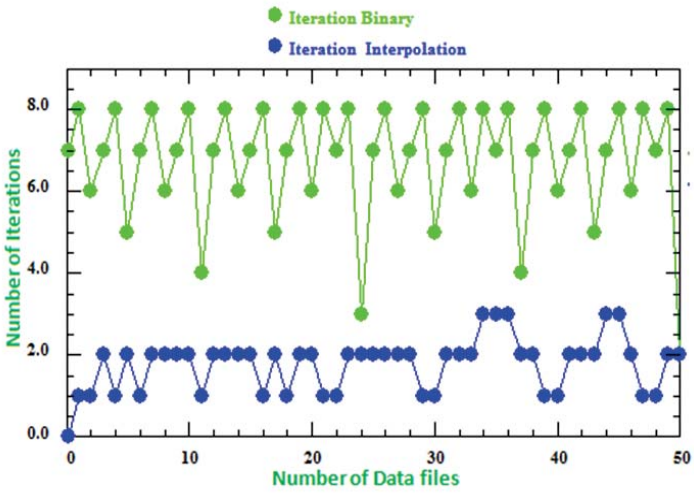


Figure 4: Interpolation vs. Binary Multiple Data Iteration Results

The figure above shows that the number of iterations in binary search are much higher than interpolation search. The results presented by the interpolation algorithm shows that the selected files were found easily and effectively. The algorithm was instructed to locate the closest points that match the user's searched query with vast amounts of data. Interpolation search took less iteration to find the data file whereas binary search took many iterations. Researchers want to access the results in a timely manner and get the most relevant data. Therefore, the proposed algorithm is quite suitable for such a case. This algorithm has made a significant advancement from existing binary search algorithm.

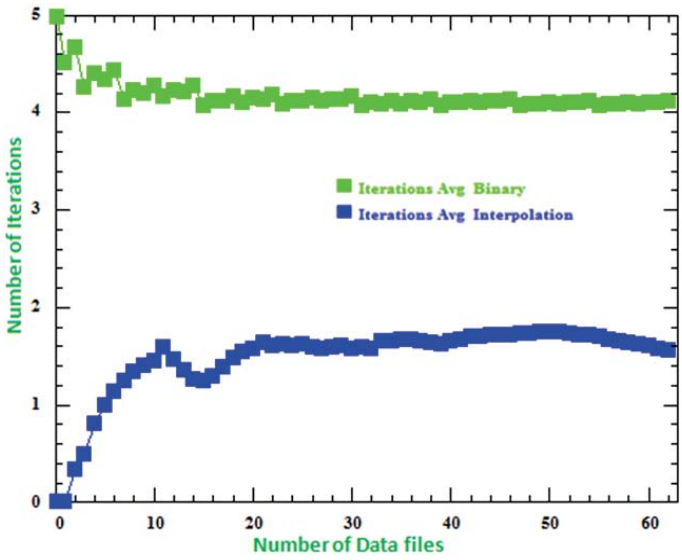


Figure 5: Interpolation vs. Binary Multiple Data Average Iteration Results

Figure 5 compares the binary and interpolation search algorithm's average number of iterations. Essentially, choosing the right program for data analysis can save more time and frustration. Working with the right program not only helps scientists gather more suitable results and more revealing graphics, but it also allows researchers to organize their data effectively. Thus, the recommended method helps researchers gain such ability.

6 Conclusion

The proposed algorithm presented will help researchers to develop a range of tools for searching, retrieving, and processing data. Large datasets continue to rapidly increase in size with time. Therefore, through better analysis of the large volumes of data, there is a potential of making faster advances and improving the profitability and success of many enterprises. Due to a significant reduction of processing time achieved by the proposed algorithm, researchers can manage and obtain the desired data at a preferred time in the field of computing. This algorithm is not limited to studies conducted by NASA or scientists in general. It can also be utilized in several data centers as well as in the medical field. For instance, in the field of medicine, the processing of medical data is playing an increasingly important role, e.g. computer tomography, magnetic resonance imaging, and so forth. These data types are produced persistently in hospitals and are increasing at a very high rate. Therefore, the need for systems that can provide efficient retrieval of medical data that is of a particular interest is becoming very high. The suggested algorithm can be utilized. In this case, to ease the burden of data retrieval and to assess the relevant data retrieval process. The algorithm can manage data in all of its aspects, including

data in ASCII formats, binary codes, compressed data, uncompressed data, and so forth.

7 References

[1] Bustrace technologies llc. (n.d.). Retrieved from http://www.bustrace.com/bustrace7/manual/HTML/06_buscapture/3_layout/5_io_details/4_raw_data.htm

[2] Chou, J. (2011). Parallel index and query for large scale data analysis large scale data analysis. Manuscript submitted for publication, Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=06114446&tag=1

[3] Di, L. (2000). Nasa standards for earth remote sensing data. International archives of photogrammetry and remote sensing, Amsterdam. Retrieved from http://www.isprs.org/proceedings/&iii/congress/part2/147_XXXIII-part2.pdf

[4] Facts, K. A. (2002). Key Aqua Facts. Delta, 73-93.

[5] Leung, A. W. (2009). Organizing, Indexing, and Searching Large-Scale File Systems.

[6] Member, S., Ortega, A. & Shen, G. (2010). Transform-Based Distributed Data Gathering.

[7] Rossi, R., & Witasse, O. (n.d.). Introduction to pds and esa data archives. Informally published manuscript, Retrieved from http://sci.esa.int/science-e/www/object/doc.cfm?fobjectid=46886

[8] Skytland, N. (2012, October 04). [Web log message]. Retrieved from http://open.nasa.gov/blog/2012/10/04/what-is-nasa-doing-with-big-data-today/

[9] Teymourlouei, H. (2013, March). An effective methodology for processing and analyzing massive datasets. 2nd international conference on computational techniques and artificial intelligence, Dubai. Retrieved from http://psrcentre.org/images/extraimages/313057.pdf

Optimum Singularity Size in Data Deduplication Technique

Matrazali Noorafiza, Mayuko Hirose, Mizuki Takaya,
Itaru Koike, Toshiyuki Kinoshita

School of Computer Science, Tokyo University of Technology
1404-1 Katakura, Hachioji Tokyo, 192-0982, Japan

Abstract *Recently, massive data growth and duplicate data in enterprise systems have led to the use of deduplication techniques. Since we keep multiple versions of files, there may be a large volume of exactly or mostly identical data. Deduplication is a powerful storage optimization technique that can be adopted to manage maintenance issues in data growth. The target files for deduplication are divided into several parts (each part is called a block) and any duplicate blocks are eliminated. In the variable-length block method, we use a particular bit-pattern (called a singularity) to decide the breakpoint of the block. We analyzed how does the singularity size and the ratio of the minimum block length to the maximum block length affect the effect of deduplication. By the experiment, we traced the change in the deduplication rate, that indicates the reduce ratio of file data volume, by changing the singularity size from 10 bits to 18 bits and changing the minimum / maximum block length ratio from 1:2 to 1:5. The result shows that the optimum singularity size is 15 bits and the minimum / maximum block length ratio is optimum at 1:4 or1:5.*

Keywords *data deduplication, variable-length block, Rabin-Karp algorithm, optimum singularity size*

1. Introduction

In recent years, the volume of file data in enterprise systems has greatly increased due to the growing popularity in handling multimedia data including animation or video, etc. In these file systems, multiple versions of a file that are exactly or mostly identical might exist and deduplication techniques may be used to minimize the file data volume by eliminating redundant data. Data deduplication is one of file compaction techniques that is commonly used in general enterprise systems by removing duplicates within and across a file. This general concept has been successfully applied to file backup, virtual machine storage, and WAN replication and so on.

Data deduplication is a process that calculates the similarity in record pairs and merges them if similarity is detected. It can reduce a huge amount of data by eliminating overlapping data (redundant data) in large-scale servers or data storage. Using data deduplication, only one representative of two or more overlapping data files or the same areas of similar data is preserved, while the overlapping data is replaced with links that point to the representative data (as shown in Fig. 1). By replacing multiple overlapping data with links, data storage size can be extremely reduced. As a result, the efficiency of data storage can be highly improved and cost in data maintenance and storage can be also decreased.

In deduplication technique, the target files are divided into several parts (each part is called a block) and any duplicate blocks are eliminated. By dividing the files into blocks and finding the duplicate part by the blocks, files are not required to be strictly the same and high deduplication efficiency can be achieved.

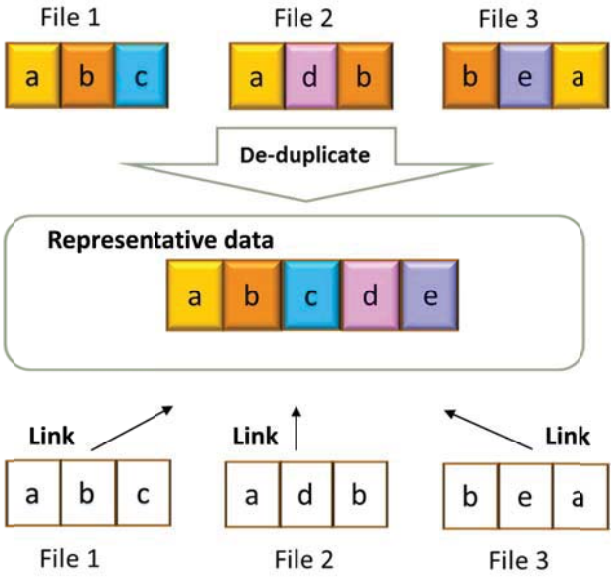
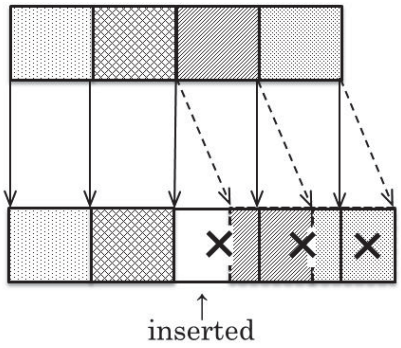


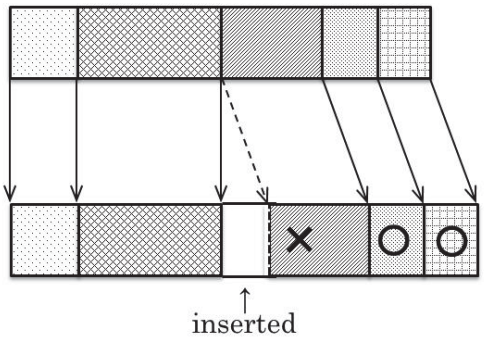
Fig.1 Concept of data deduplication

2. Two types of block

In the deduplication technique, there are two types of block; one is fixed-length block whose length is constant and the other is variable-length block whose length can be changed. In the fixed-length block method, when some data are inserted into the file and the blocks after inserted position are shifted, they cannot be recognized as duplicate blocks in the original data (Fig. 2 (a)). On the other hand, in the variable-length block method, the block length can be changed and adjusted for the insertion. The blocks after inserted position can be recognized as duplicate blocks and considered for deduplication (Fig. 2 (b)). Thus, in the variable-length block method, the effect of deduplication can be maintained even if some data have been inserted or deleted.



(a) Fixed-length block



(b) Variable-length block

Fig. 2 Tow types of block

In order to efficiently generated variable-length blocks, a hash value of a window, that is a small part with constant length from the candidate of the breakpoint of the block, is calculated. If a particular bit-pattern, that is called a singularity, is included in the hash value, the candidate becomes a real breakpoint of the block. If the singularity is not included in the hash value, the candidate does not become a breakpoint. The effect of deduplication can be changed by the singularity; especially the singularity size mainly affects the effect of deduplication. The purpose of this study is to analyze the optimum singularity size to maximize the effect of deduplication.

3. Related Works

In the variable-length block method, the block length can be changed, and the maximum and minimum block length is set not to generate an extremely large or small block. The effect of deduplication is also affected by this maximum / minimum block length.

The effect of deduplication in the fixed-length block method when the block length is set to 4 ~ 16 K bytes is investigated in [1] and the efficiency of the variable-length method is discussed in [3]. The difference of the effect of deduplication between in the variable-length block and in the fixed-length block when the block length is larger than 4 K bytes is reported in [2] and when the block length is smaller than 4 K bytes is discussed in [4]. These studied studies have investigated the relationship between the block length and the effect of deduplication. In our previous work [5], we analyzed the relationship between the singularity size and the deduplication rate. In this study, we clarified how does not only singularity size but also the ratio of skip part length to search part length affect the effect of deduplication.

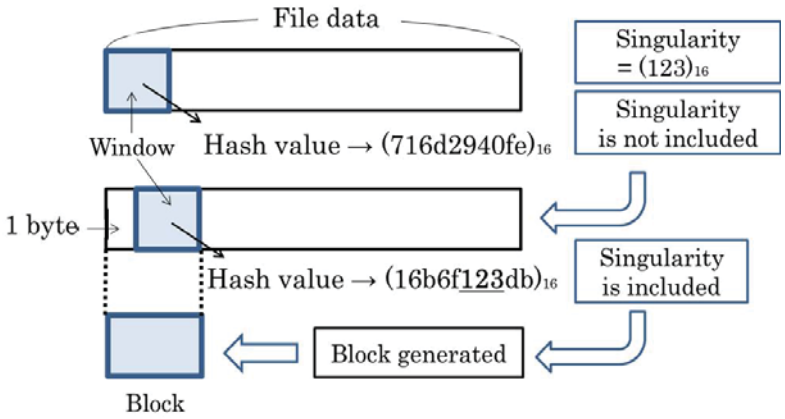


Fig.3 Breakpoint search

4. Variable-length block

The Rabin-Karp string search algorithm is used for finding the singularity in the hash value of the window. The following parameters are used in the algorithm.

(1) Minimum file size (default is 40 bytes)
Files that are smaller than this size will not be targeted for deduplication.
(2) Minimum block length (default is 4,000 bytes)
(3) Maximum block length (default is 16,000 bytes)
(4) Window size (default is 32 bytes)
Window is a unit for calculating a hash value.
(5) Singularity
When the singularity is included in the bit-pattern of the hash value of the window, a block is generated at the position of the window.

For a file whose size is between the minimum file size and the minimum block length, the whole file is generated as a block. If a file is larger than the minimum block length, a breakpoint will be determined. As shown in Fig. 3, in searching for the breakpoint, a hash value is first created for the window at the location of the minimum length block, and is checked up if it includes the singularity, or not. If the hash value includes the singularity, a breakpoint is found and a block is generated at the position. On the other hand, if the hash value does not include the singularity, the window is shifted one byte and the breakpoint search is repeated. If the breakpoint is not found until the maximum block length, a maximum length block is generated at this position.

Since the common default block length is between 4,000~16,000 bytes, files having lengths of 4,000 bytes or less are not deduplication targets. When the minimum block length becomes smaller, the deduplication rate can be improved since the smaller files will be included in deduplication. However, once the block length becomes smaller, not only the number of blocks but also the time for generating blocks will increase. With the traded-off relationship between the effect of deduplication and the number of generated blocks, we can expect that the optimal singularity length, which maximizes deduplication efficiency, exists.

Since a smaller part than the minimum block length is excluded from the target of breakpoint search, we call this range a skip part. On the other hand, the range from the minimum block length to the maximum block length is the target of breakpoint search, and we call this range a search part. When the minimum block length is changed and keeping the ratio of skip part length to search part length constant (therefore, the maximum block length is also changed), the block length can be changed while the type of block length distribution remains unchanged (see Fig. 4)(if the block length becomes smaller, many small blocks are generated, and if the scale of block length is enlarged, very few large blocks are generated). We denote R_s as the ratio of skip part length to search part length.

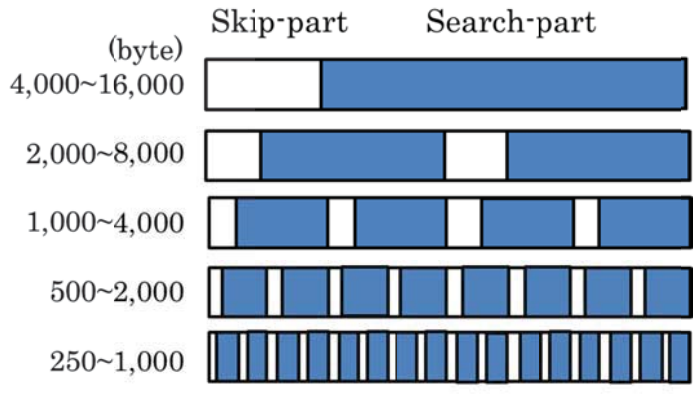


Fig. 4 Skip and Search parts (for R_s=1:3)

5. Verification of optimum singularity size

5.1 Verification experiment

As an indicator of the effect of deduplication, we use "the deduplication rate = the data size reduced by deduplication / the original data size". The large deduplication rate shows better effect of deduplication. Experiment for verification was performed using the variable-length block method and testing was done on the file systems in our research laboratory in Tokyo University of Technology. The total size of the file system is 7.0 G bytes and the target file system includes text data (data types are doc, xls, ppt), image data (pdf, gif, bmp, jpg), etc. The maximum / minimum block length is set to 4K bytes and 16K bytes, which are usually used in almost all deduplications [1].

5.2 Experimental Results

Fig. 5 and Fig. 6 show the relationship between the singularity size and the deduplication rate, and the relationship between the singularity size and the number of blocks respectively at each ratio R_s of skip part length to search part length. We can see that the highest deduplication rate achieves when the singularity size is 15 bits in any R_s=1:2 ~ 1:5. (in Fig. 5) and the number of blocks decreases along with the increase of singularity size since the singularity more hardly exists in the hash value of the window when the

singularity size is larger (in Fig. 6).
According to Fig.7, the peak of the number of blocks at R_s=1:3 occurs when the range of the minimum block length is 5,001 – 6,000 bytes, while it is 4,001 – 5,000 when R_s=1:2, 1:4, 1:5. When the ratio R_s is larger, skip part becomes relatively larger and the number of files whose size is smaller than skip part size increases. This causes that the peak position of the number of blocks when R_s=1:2 is in smaller range of block length than that of when R_s=1:3. When R_s=1:4 and 1:5, many moderate length blocks are generated in the smaller range of block length. Then the peak position of the number of blocks when R_s=1:4 and 1:5 is also in smaller range of block length than that of when R_s=1:3.

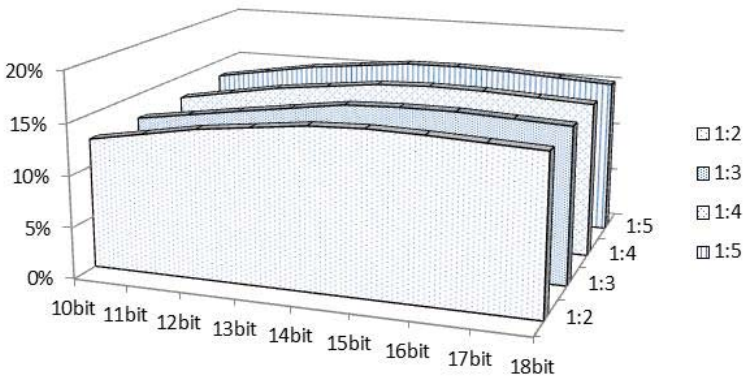


Fig. 5 Singularity size vs deduplication rate

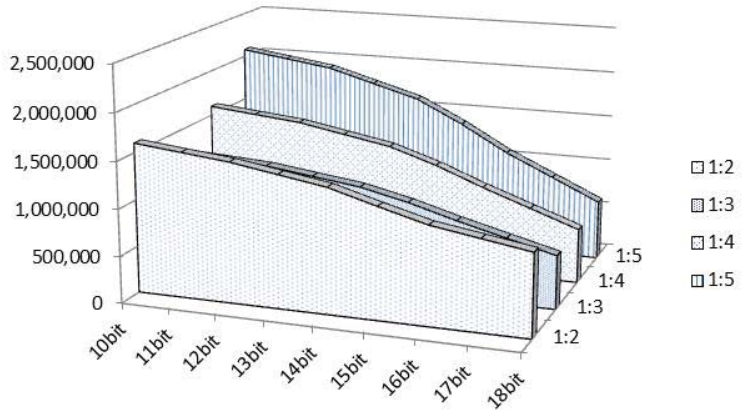


Fig. 6 Singularity size vs number of blocks

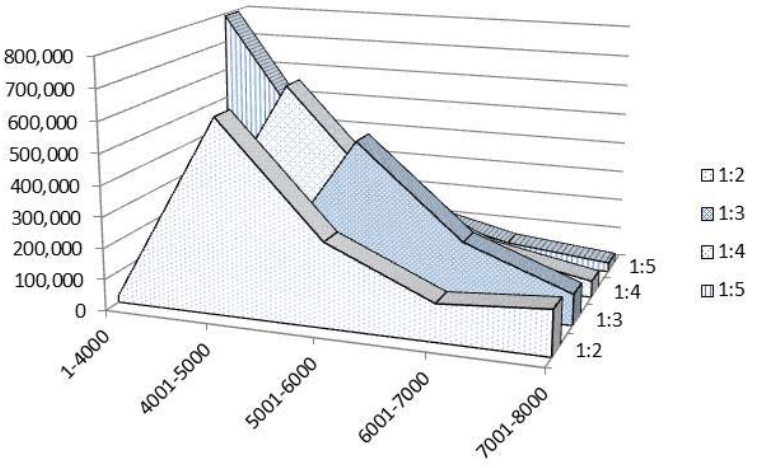


Fig. 7 Minimum Block length vs number of blocks

6. Conclusion

In the variable-length method in the deduplication technique, the singularity size and the ratio of skip part length to search length affect the effect of deduplication. When the singularity size is larger, the singularity more hardly exists in the hash value of the window and the moderate size of block is hardly found.

In this study, we clarified that the highest deduplication rate can be achieved at the singularity size 15 bits, and more moderate length blocks are generated at the ratio of skip part length to search part length is 1:4 or 1:5.

As our future works, we will investigate the optimum singularity size and the ratio of the skip part length to the search part length for the specified file type.

References

[1] Q. He, Z. Li, X. Zhang, "Data deduplication techniques," Future Information Technology and Management Engineering (FITME) 2010, vol. 1, pp. 430-433, Oct. 2010

[2] C. Constantinescu, J. Glider, D. Chambliss , "Mixing Deduplication and Compression on Active Data Sets," Data Compression Conference (DCC) 2011, pp. 393-402, March 2011

[3] A. N. Yasa, P. C. Nagesh, "Space savings and design considerations in variable length deduplication," ACM SIGOPS Operating Systems Review, Vol. 46 Issue 3, pp. 57-64, Dec. 2012
[4] M. Noorafiza, I. Koike, H. Yamasaki, A. Rizalhasrin, T. Kinoshita, "Block Length Optimization in Data Deduplication Technique," Proceedings of the 10th International Conference on Scientific Computing (CSC2013), pp.216-220, July 2013
[5] M. Ogiwara, M. Takaya, T. Kasuya, I. Koike, T. Kinoshita, "Singularity Size Optimization in Data Deduplication Technique," Proceedings of the International Parallel and Distributed Processing Techniques and Applications 2014, (PDPTA2014), pp.171-175, July 2014

Semantic Web Improved with the Weighted IDF Feature and the Class Information

A. Mrs. Jyoti Gautam[1], **B. Dr. Ela Kumar**[2]
[1] Department of Computer Science and Engineering, Uttar Pradesh Technical University, NOIDA, Uttar Pradesh, INDIA
[2] Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, Delhi, INDIA

Abstract - *The development of search engines is taking at a very fast rate. Different algorithms have been tried and tested. Still the results are not precise. Social networking sites are developing at tremendous rate and their growth has given birth to the new interesting problems. The social networking sites use semantic data to enhance the results. This provides us with a new perspective on how to improve the quality of information retrieval. As we are aware, many techniques of text classification are based on TFIDF algorithm. Term weighting has a significant role in classifying a text document. In this paper, firstly, we are extending the queries by "keyword+tags" instead of keywords only. In addition to this, secondly, we have developed a new ranking algorithm (JEKS algorithm) based on semantic tags from user feedback that uses CiteUlike data. The algorithm enhances the already existing semantic web by using the weighted IDF feature of the TFIDF algorithm. The suggested algorithm provides a better ranking than Google and can be viewed as a semantic web service in the domain of academics. The algorithm can be modified by including class information.*

Keywords: *Text classification; Semantic Web with weighted idf feature; Expanded query; New Semantic Web Algorithm with class information; Ranking Algorithm.*

1 Introduction

A lot of information is available on the Internet. Search engines remain as the primary infrastructure for Information Retrieval. The relevance of the result-sets is not as desired by the user. This leads to the requirement of a good ranking algorithm to put the best results on the front.

Many popular Web services like Delicious, Citeulike and flickr.com rely on folksonomies (Gautam and Kumar, 2012).Some websites such as CiteUlike (Research Paper Recommender), Delicious (online bookmarking), Flickr (online photo management and sharing application), Furl (File Uniform Resource Locators), Blinklist (links saver), Diigo (collect and organize anything e.g. bookmarks, highlights, notes, sceenshots etc.), Otavo (collaborative web search), Stumbleupon (discovery engine), Blummy (tool for quick access to favorite web services), and Folkd (saves bookmarks and links online) etc. which contain these tag information.

Various difficulties are encountered while doing research on folksonomies. In spite of all this, the growth is tremendous in this area. Researches based on social-bookmarking have become increasingly popular, which lets users specify their keywords of interest, or tags on web resources. Social tagging, also known as social annotation or collaborative tagging is one of the major characteristics of Web 2.0. Social-tagging systems allow users to annotate resources with free-form tags. The resources can be of any type, such as Web pages (e.g., delicious), videos (e.g., YouTube), photographs (e.g., Flickr), academic papers (e.g., CiteULIke), and so on.

In this paper, we utilize the semantic tag information with web page. This information is obtained from CiteUlike (Research Paper Recommender and online Tagging System).When users submit their query; they also submit some semantic description to disambiguate the query. Then, by matching the semantic description between the query and web page, user's query intent can be well understood. The better understanding of the user's query leads to better ranking results in academic domain.

In this paper, the following approach has been adopted. We have tried to use the metadata available in the form of user feedback and semantic tags from CiteUlike.

a. A new ranking algorithm has been developed. The algorithm utilizes the weighted IDF feature of the TFIDF algorithm.
b. The query was expanded. The idea was to use "keyword + tags" instead of keywords only, so that it carries some semantic description along with it.
c. The data was obtained through CiteUlike.
d. The performance analysis was done by comparing the approach with Google by several evaluation methods.

The paper is organized by an introduction to the existing ranking methods, then the new optimized JEKS algorithm followed by significance of the algorithm. Thereafter, the experiments and analysis is done followed by significance and relevance of the research work. In the end, finally the paper is concluded.

2 The Existing Ranking Methods

Tf-idf, term frequency-inverse document frequency is a numerical statistic which reflects how important a word is to a

document in a corpus. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus.

The literature (S. Lu, X. Li, S. Bai and S. Wang., 2000) provides an improved approach named tf.idf.IG to remedy this defect by Information Gain from Information Theory.

The literature (S. Lu, X. Li, S. Bai and S. Wang., 2000) provides an improved approach named tf.idf.IG to remedy this defect by Information Gain from Information Theory.

The Lingo algorithm proposed by Osinski and Weiss (2005) combines common phrase discovery and latent semantic indexing techniques to separate search results into meaningful groups. It looks for meaningful phrases to use as cluster labels and then assigns documents to the labels to form groups.

(Wu, Zhang and Yu, 2006) explored the technique of Social Annotations for the Semantic Web. These annotations are manually made by normal web users without a predefined formal ontology. The evaluation of the approach shows that the method can effectively discover semantically related web bookmarks that current social bookmark service cannot discover easily.

(Farooq, Kannampallil and Song, 2007) The authors use six tag metrics to understand the characteristics of a social bookmarking system. Possible design heuristics was suggested to implement a social bookmarking system for Cite Seer using the metrics.

The authors Cilibrasi and Vitanyi (2007) described a technique for calculating the Google similarity distance.

Jin, Lin and Lin (2008) proposed the architecture of a semantic search engine and an improved algorithm based on TFIDF algorithm. The algorithm considers crawling of static web pages. The algorithm can be considered for crawling of dynamic web pages and for parallel crawling also.

A personalized search framework was proposed by Shenliang, Shenghua and Fei (2008).It utilizes folksonomy for personalized search.

(Jiang, Hu, Li, and Wang 2009).The other method of basic TFIDF model uses supervised term weighting approach. The model uses class information to compute weighting of the terms. The approach is based on the assumption that low frequency terms are important, high frequency terms are unimportant, so it designs higher weights to the rare terms frequently.

Jomsri, Sanguansintukul and Choochaiwattana (2010) proposed a framework for Tag-Based Research Paper Recommender system. User self-defined tags were used for creating a profile for each individual user and cosine similarity was used to compare a user profile and research paper index. The recommender system demonstrated an encouraging preliminary result with the overall accuracy percentage up to 91.66%. The number of subjects is considered to be small in the experiment.

(Zhao and Zhang, 2010) proposed a new viewpoint on how to improve the quality of information retrieval. The queries are extended by "keywords+tags" instead of keywords only. A new tag based ranking algorithm (OSEARCH) was proposed and the results obtained were also compared with Google by several evaluation methods.

The authors Leung and Lee (2010) focussed on search engine personalization and developed several concept-based user profiling methods that are based on both positive and negative preferences. The proposed methods were evaluated against the previously proposed personalized query clustering method.

(Kaczmarek, 2010) introduced a novel approach to interactive query expansion. When a user executes a query, the algorithm shows potential directions in which the search can be continued.

Another supervised term weighting method, proposed by the authors (Zhanguo, Jing, Liang, Xiangyi and Yanqin, 2011), provides an improved tf-idf-ci model to compute weighting of the terms. The method uses intra and inner class information.

Various variations of the tf–idf weighting scheme are often used by search engines. Search engines use these weighted measures as a central tool in scoring and ranking a document's relevance given a user query. The tf-idf is improved by many literatures. The proportion of distribution of terms in text collection is one of the most important factors of expressing the content of text, but it is beyond tf-idf's power (Zhanguo, Jing, Liang, Xiangyi and Yanqin, 2011).

The paper proposed by (Yoo, 2011) suggests a hybrid query processing method for the effective retrieval of personalized information on the semantic web. When individual requirements change, the current method of query processing requires additional reasoning for knowledge to support personalization.

(Halpin and Lavrenko, 2011) proposed the method of relevance feedback between hypertext and semantic web search. The paper proposed investigates the possibility of using semantic web data to improve hypertext web search.

In this paper, the authors (Gracia and Mena, 2012) presented the web's natural semantic heterogeneity problems – namely, redundancy and ambiguity. The authors' ontology matching, clustering, and disambiguation techniques aim to bridge the gap between syntax and semantics for Semantic Web construction.

The authors Zhong, Li and Wu (2012) proposed an effective pattern discovery method for text mining. The paper presents an innovative and effective pattern discovery technique which includes the processes of pattern deploying and pattern evolving, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information.

The paper (Lee, Kim and Park, 2012) proposes searching and ranking method of relevant resources by user intention on the semantic web. There are more limitations in information searching as the information on the Internet dramatically increases. To overcome the various limitations, the Semantic Web must provide search methods based on the different relationships between resources.

This paper proposed by (Gautam and Kumar, 2012) proposes a framework for a tag-based Academic Information Sharing and Recommender System which shares information such as question papers, assignments, tutorials and quizzes on a specific area.

(Shaikh, Siddiqui and Shahzadi, 2012) proposed the Semantic Web based Intelligent Search Engine. SWISE required including domain knowledge in the web pages to answer intelligent queries. The layered model of Semantic Web provides solution to this problem by providing tools and technologies to enable machine readable semantics in current web contents.

(Lee, Kim, and Park 2012) presented some proposals to improve and extend the semantic approach based on conceptual neighborhood's graphs in order to best preserve the proximity between the adapted and original documents and to deal with models that define delays and distances.

3 User Query Intent and Storage of Tags

3.1 Metadata Information in the web pages and expansion of the query

While talking about semantic web, metadata comes into picture. What is this semantic? How is it related to metadata? Semantic Web is something that implies the content, meaning or the metadata related to the web. This metadata information is hidden in the web pages. There are different websites which are working upon it since a long time. We have sites like Delicious, CiteUlike, Flickr etc., which allow different users to create their accounts. After creating the accounts, the users can add metadata for the different websites. This metadata conveys the content of the website as interpreted by different users.

The method should be such that which tries to capture the user's real query intent. The primary purpose of the search engines is to return the optimal results. But before returning the results, it should be able to analyze the query clearly. The simple keywords can't express user's real query intent. In order to analyze the query, some metadata information is added along with the query. The metadata information is added by expanding the query .i.e., keyword+tags instead of the keywords only.

So, the idea is to consider utilizing metadata which is available in the form of semantic tags .One area that arises is to consider utilizing the semantic tag information with web page. When users submit their query, they can also submit some simple semantic description to narrow down the query. Then by matching the semantic information between query and web page metadata, we can understand user's query intent better and return better result.

So, the idea is to utilize this semantic tag information. Here, we are proposing the development of a new algorithm based on semantic tags and the weighted IDF feature of the TFIDF algorithm.

3.2 Storage of Semantic Tags on Web Pages

The semantic tags of a web page are some object properties that reflect the content of the web page, such as marked with "semantic web", which signifies that the page contains information about the object of "semantic web". Of course, there may be multiple tags on a page, because the pages always contain multi information. These tags carry the metadata information along with them.

In our case, we are storing the tags from CiteUlike. A popular website in academia is CiteULike (www.CiteULike.org). CiteUlike is a free service for managing and discovering scholarly references.

- Easily store references you find online
- Discover new articles and resources
- Automated article recommendations
- Share references with your peers
- Find out who's reading what you are reading
- Store and search your PDF's

CiteULike has a filing system based on tags. Tags provide an open, quick and user-defined classification model that can produce interesting new categorizations.

Additionally, it is also capable to:

- 'tag' papers into categories.
- Add your own comments on papers.
- Allow others to see your library

The semantic tags are retrieved from CiteUlike. The URLs along with their tags are stored in a local database. For the semantic tags, each URL is opened in CiteUlike and the tags with their numeric values are stored in the database. We add tags' values in the MYSQL database. The data was retrieved from April, 2012 to June, 2013 from CiteUlike for the 50 queries. A total of 5000 URLs were opened in CiteUlike and the database was created.

4 A New Optimized Ranking Algorithm

4.1 Utilizing the Weighted Inverse Document Frequency and class information

In this paper, we are proposing a new algorithm based on semantic tags in the web pages. An enhanced semantic web algorithm is proposed. The algorithm is based on utilizing the metadata information available with the web pages by integrating in the algorithm some good features of weighted IDF and the class information.

Pseudo-Code for the JEKS (Jyoti and Ela Kumar Search) Algorithm [4]:

Step1: Start

Step2: Declare variables query, keyword, tag, user_tag, r_tag, google_score, tg_score, IDF score, q, i, j.

Step3: Query ← {Keyword1, keyword2,…., tag1, tag2,….}

Step4: v_usrt ← {user_tag1, user_tag2,….}

Step5: v_rest ← {r_tag1, r_tag2,….}

Step6: TotalScore ← Google_score+Score (1)

Step7: Score ← Tg_score × IDF score × weighting score (2)

Step8: Calculation of Google_score

{Here p is set to 100, q varies from 1 to 100}

For (q=1; q ≤ 100; q++)

Google_score ← (p – q + 1)/p

END of for loop

Step9: Calculation of Tg_score

For (i=1; i ≤ 100; i++)

For (j=1; j ≤ 5; j++)

Calculate sim(v_usrt[i], v_rest[k])

= 1, V_usrt[i] and V_rest[k] have the same root,

= 1, V_usrt[i] and V_rest[k] have the same meaning,

= 0, V_usrt[i] and V_rest[k] does not have a semantic relation,

= 0.5, even if half of the V_usrt[i] tag resembles with the V_rest[k] tag.

Find freq(V_rest[i])

Tg_score←

$$\frac{\sum_{i=1}^{|V\_usrt|}\sum_{k=1}^{|V\_rest|}(freq(V\_rest[i])*sim(V\_usrt[i]),V\_rest[k]))}{\sum_{k=1}^{|V\_rest|}freq(V\_rest[k])} \quad (3)$$

END for inner for loop

END for outer for loop

Step10: Calculation of IDF score

IDFscore ← $\log(|D|/f_{w,D})$ (4)

D ← 100

N ← 0

For (int i = 1; i ≤ D; i++)

{if (r_tag = user_tag)

N ← N+1

Else

N remains same}

$f_{w,D}$ ← N

END of for loop

Step11: Calculation of weighting score

Weightingscore← A_i/C_i (5)

A_i/C_i ← $f_{w,D}/(D - f_{w,D})$

Step12: END

Time-Complexity of the JEKS algorithm

Time-Complexity of Google Score –

It will be proportional to n.

Time-Comlexity of Tg_Score

It will be proportional to n×m.(If n=m, then n^2)

Time-Complexity of IDF Score

It will be proportional to n.

The algorithm takes f(n) operations, where

$f(n)=n^2+ n + n,$ (6)

Since, the polynomial grows at the same rate as n^2.

So, the time-complexity of the complete algorithm is proportional to n^2.

Finally, the time-complexity can be represented as $O(n^2)$.

The JEKS algorithm can be modified by including in the algorithm the class information.

5 Modification suggested in the algorithm

The suggested algorithm can be modified to include the following changes. The score in eq. (2) can be extended to include class information also. We know from eq. (2) that the score is:-

Score=Tg_score × IDFscore × weighting score

Equation (2) can be modified like,

Score= Tg_score×IDFscore×weighting score×ci (7)

Where, the ci is the class information.

The class information contains two parts. One part is intra class information, and the other is inner class information. That is:

ci = Cit X Cii (8)

where, Cit is intra class information, Cii is inner class information.

Intra Class Information

$$Cit = \frac{P(ti|Cj)}{\sum_{k=1,\ k\neq j}^{m} P(ti|Ck)} \quad \text{if} \quad \sum_{k=1,\ k\neq j}^{m} P(ti|Ck) \neq 0 \qquad (9)$$

$$Cit = \frac{P(ti|Cj)}{\beta} \quad \text{if} \sum_{k=1,\ k\neq j}^{m} P(ti|Ck) = 0 \qquad (10)$$

Where, $P(ti|Cj)$ is the probability of documents containing tag t_i in the class Cj of the training set? The value of the parameter β is determined through actual situation. Generally, β = 0.001. The number of classes taken is m.

It is quite evident that the Cit is a monotone increasing with the number of documents in the class Cj containing the tag t_i increasing. The Cit is increasing with the sum of documents beyond the class Cj containing the tag t_i decreasing. Only when the documents of two classes containing the tag t_i , and the document numbers of containing the tag t_i are almost same the value of Cit approximate to 1. The largest value is achieved when the sum of documents beyond the class Cj containing the tag t_i is zero.

Inner Class Information

Inner class divergence can be represented by the term Cii. It is very important to classify when the tag t_i appears evenly in the documents of one class.

$$Cii = \frac{tfavg\ (ti,Cj)}{\sum_{k=1}^{N(Cj)} [\,|\,tfik - tfavg\ (ti,Cj)\,|\,]}$$

$$\text{If} \sum_{k=1}^{N(Cj)} [\,|\,tfik - tfavg\ (ti,Cj)\,|\,] \neq 0 \qquad (11)$$

$$Cii = \frac{tfavg\ (ti,Cj)}{\gamma}$$

$$\text{If} \sum_{k=1}^{N(Cj)} [\,|\,tfik - tfavg\ (ti,Cj)\,|\,] = 0 \qquad (12)$$

Where, γ is a parameter, the value is determined which is based on the actual situation. Generally, γ = 1. The tf_{ik} is the frequency of the tag t_i in document k. The tf_{avg} (t_i ,Cj) is the average term frequency of the tag t_i in the documents of the class C_j :

$$tf_{avg}(t_i,Cj) = \frac{\sum_{k=1}^{N(Cj)} tfik}{N(Cj)} \qquad (13)$$

The largest value of Cii is achieved when the tag t_i appears evenly in the documents of the class C_j. If the difference of the tag t_i appeared in the documents of the class C_j is larger, then the denominator of the function is larger, then the value obtained of Cii is less. So, it is representative and important for classification purposes when the tag ti appears evenly in the documents of one class.

The algorithm is suggested, but has not been implemented. The experimental results are based on the new optimized algorithm, which has been proposed in section 4.

5.1 Experimental Results

First, we determine the relevance between each query intent and each result page. Each result is assigned a relevance score according to its relevance, which ranges between 0 to 3 (totally irrelevant, basically irrelevant, basically relevant, and totally relevant).

We obtain normalized DCG values for our algorithm and Google. For the 50 chosen queries, the average nDCG values for our algorithm are .9227 and for Google, it is .911833.The values of recall remain the same i.e. 1.Wheraes, the average values of precision and F1-score for our algorithm is .6872 and .790979. For Google, the average values of precision and F1-score is .6808 and .786715. The results are better for our algorithm.

6 Relevance of My Research Work

The relevance of the research work is that the entire work has been done using semantic tags from CiteULike(which provides tags in a fully uncontrolled environment). The algorithm is entirely based on tags, which are the essence of semantic web. So, it can be taken as an application or a web service in Academics Domain using semantic web. The algorithm can be extended for more queries.

7 Conclusion

In this paper, we have analyzed some existing ranking methods and proposed a new algorithm based on the previous methods. Semantic tag of a web page is the metadata information associated with it and depicts a lot about the information associated with it. The match degree between user's real query intent and web page content is determined by calculating the similarity between query and web page tag.

We have proposed the new algorithm using the already existing semantic web algorithm which basically calculates the weighted score of the tags. We have utilized the IDF feature of TFIDF algorithm to improve the semantic web which uses tags. In addition to this, we have used a weighting score. In experiments, we have collected the data from Citeulike and implemented the above algorithm. The relevance scores to the different web links have been given by a group of users. Comparing with Google search results, we find that JEKS algorithm acquires better ranking results, and can put more relevant results in front. Our algorithm acquires higher values of DCG for 40 queries when compared to Google. Our

algorithm acquires higher precision in comparison to Google throughout the varying levels of K for all the 50 queries. We have proposed extension of the algorithm by including class information.

In the future work, the suggested changes in the algorithm can be implemented.

8 References

[1] Cilibrasi, R.L., & Vitanyi, P.M.B. (2007) The Google similarity distance, in IEEE Transactions on Knowlegde and Data Engineering, vol. 19, no.3.

[2] Farooq, U., Kannampallil, T.G., & Song, Y. (2007) Evaluating Tagging Behaviour in Social Bookmarking Systems: Metrics and design heuristics, *in the international ACM Conference on Supporting Group Work*.

[3] Gautam, J., & Kumar, E. (2012) An Improved Framework for Tag-Based Academic Information Sharing and Recommender System, in *Proc. of the World Congress on Engineering*, Vol. 2, 2012, 845-850.

[4] Gautam, J., & Kumar, E.(2015) "Semantic Web improved with weighted IDF feature of the TFIDF," IJACSA, vo.6, issue2, pages164-173.

[5] Gracia, J., & Mena, E. (2012) Semantic Heterogeneity Issues on the Web, *IEEE Internet Computing*, pages 60-67.

[6] Halpin, H., & Lavrenko, V. (2011) Relevance feedback between hypertext and Semantic Web search, *Journal of Web Semantics*, vol. 9, 2011, pages 474-489.

[7] Jiang, H., Hu, X., Li, P., & Wang S.(2009) An improved method of term weighting for text classification, in *International Conference on Intelligent Computing and Intelligent Systems*, IEEE, Vol.1, 2009, pages 294-298.8

[8] Jin Y., Lin Z., & Lin H., The Research of Search Engine Based on Semantic Web, in *proc. of International Symposium on Intelligent Information Technology Application Workshops (IITAW), IEEE*, 2008, pages 360-363.

[9] Jomsri P., Sanguansintukul S. & Choochaiwattana W., A Framework for Tag-Based Research Paper Recommender System: An IR Approach, in *proc. of the 24th International Conference on Advanced Networking and Applications Workshops, IEEE*, 2010, pages 103-108.

[10] Kaczmarek, A.L. (2011) Interactive Query Expansion with the Use of Clustering-by-Directions Algorithm, *IEEE Transactions on Industrial Electronics*, VOL. 58, No. 8, pages 3168-3173.

[11] Lee, M., Kim, W., & Park, S. (2012) Searching and ranking method of relevant resources by user intention on the Semantic Web, Expert Systems with Applications, vol. 39, pages 4111- 4121.

[12] Leung, K.W.T., & Lee, D.L. (2010) Deriving concept-based user profiles from search engine logs, in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7.

[13] Lu C., Hu X., & Park J. (2011) Exploiting the Social Tagging Network for Web Clustering, *(Systems, Man, and Cybernetics – Part A: Systems and Humans)*, vol. 41, pp. 840-852.

[14] Maredj A., & Tonkin N. (2013) Semantic Adaptation of Multimedia-Documents, International Arab Journal of Information Technology, vol. 10, No. 6, pages 579-586.

[15] Osinski, S., & Weiss, D. (2005) A Concept-Driven Algorithm for Clustering Search Results, *IEEE Intelligent Systems*, Volume 20, Issue 3, pp. 48-54.

[16] Shaikh, F., Siddiqui, U.A. & Shahzadi, I. (2012) Semantic Web based Intelligent Search Engine, in *proc. of International Conference on Information and Emerging Technologies*, pp. 1-5.

[17] S. Lu, X. Li, S. Bai & S. Wang., (2000) An improved approach to weighting terms in text. *Journal of Chinese Information Processing*, 14(6), pp. 8-13.

[18] Shenliang X., Shenghua B. and Fei, B., *Exploring Folksonomy for Personalized Search*, in proc. of the 31st annual international ACM SIGIR conference on Research and Development in information retrieval, 2008, pp. 155-162.

[19] Yoo, D. (2012) Hybrid Query Processing for Personalized Information Retrieval on the Semantic Web, *Knowledge-Based Systems*, vol 27, pages 211-218.

[20] Wu, X., Zhang, L., & Yu Y. (2006) Exploring Social Annotations for the Semantic Web, in *proc. of the 15th International Conference on World Wide Web (WWW 06), ACM*, pages 417-426.

[21] Zhanguo, M., Jing, F., Liang, C., Xiangyi H., & Yanqin, S. (2011) An improved approach to terms weighting in text classification , in *proc. of the International Conference on Computer and Management*, IEEE, pages1-4.

[22] Zhao, C., & Zhang, Z. (2010) A New Keywords Method to Improve Web Search, in *12th International Conference on High Performance Computing and Communications, IEEE*, pages 477-484.

[23] Zhong, N., Li, Y., & Wu. S.T. (2012) Effective Pattern Discovery for Text Mining, IEEE Transactions on Knowledge and Data Engineering, vol. 24, no1.

A Particle Swarm Optimization and Fuzzy Based Algorithm for Solving Classical Travelling Sales Person Problem

Azmath Mubeen[1], D.Hemalatha[2], D.Rama Krishna Reddy[3]

[1]Asst.Professor, Dept. of Computer Science, Osmania University College for Women, Koti, Hyderabad, Telangana, India

[2]Asst. Professor, Dept. of Computer Science, Osmania University College for Women, Koti, Hyderabad, Telangana, India

[3]Asst. Professor in Computer Science, Dept. of Mathematics, UCS, Osmania University, Hyderabad, Telangana, India

Abstract— *Travelling Sales Person Problem is one of the classical combinatorial optimization problems that belong to the NP-complete class. It is the problem of finding the optimized path for a given set of cities. The path is drawn in such a way that the salesperson has to visit each city exactly once. This paper provides an efficient method for solving the classical Travelling Sales Person Problem by using Particle Swarm Optimization (PSO) based on fuzzy logic. A particle is represented using a particle encoding/decoding scheme for the Travelling Sales Person Problem (TSPP). The searching ability of the PSO is expanded here by hybridizing the PSO with fuzzy logic. The local bests will not be the point of meeting for the particles and the global targeted goals can be searched in a shorter span of time if the PSO is correctly adjusted for the particles with the help of fuzzy logic rules. Numeric values for random weights have been taken to illustrate the efficiency of the system proposed for solving the Travelling Sales Person Problem.*

Keywords: Travelling Sales Person Problem, Particle Swarm Optimization, Fuzzy Logic, evolutionary algorithm, Velocity Clamping, Construction Factor Method(CFM).

1 Introduction

The Classical Travelling Sales Person Problem has been solved by various methods and techniques to find an optimized solution. This paper present an approach which combines the Particle Swarm Optimization with Fuzzy logic rules. Particle Swarm Optimization(PSO) has been used to solve Optimization problems since it was proposed by Kennedy and Eberhart in 1995. The algorithm of PSO has the behavior of animal societies that don't have any leader in their group or swarm such as bird flocking or fish schooling. If the group of animals does not have leaders will find the food by random and follow tone of the members of the group that has the closest position of the food source and find the better solution. Animals which have better solution will inform it to its flocks and others will move accordingly. This process happens repeatedly until the best condition or the food source is discovered.

Travelling Sales Person Problem is the most basic computational problems for finding the optimized route in a network. This paper provides a novel approach to find the optimized solution for the single source and finding the shortest path by applying the fuzzy rules to the Particle Swarm optimization technique. The Travelling Sales Person Problem (TSPP) is one of the important basic computational problems in graph theory, and of greatest importance in communication networks. This TSP problem is concerned with exploring the shortest path from a particular origin to a specific target in a specified network however minimalizing the cost and perhaps taking particular limits into consideration. This problem has many varied applications, such as route scheduling in robotic systems [1], vehicle routing in transportation systems [2], sequence alignment in molecular biology [3], and traffic routing in communication networks [4], has made this significant computational problem the focus of interest in the scientific and research communities. The performance of a computer network is mostly motivated by routing, particularly in multi-hop networks, such as the Internet and mobile Ad-hoc Networks. An appropriate routing algorithm must be able to find an optimal path for communication within a specified period of time to fulfill the Quality of Service (QoS) [5, 6, 7]. Different distinguished known deterministic algorithms, such as Dijkstra [8] and Bellman-ford [9] are usually used to solve the

Travelling Sales Person Problem. Nevertheless, these classic algorithms experience some severe limitations, one of which is that they may not be used for networks with negative weights of edges. For example, in some communications networks, the weights can characterize the transmission line capacity, and the negative weights depict the links with gain rather than loss. Another problem of these algorithms is the point that they need complicated calculations for simultaneous communications involving rapidly changing network topologies such as the earlier-mentioned wireless ad-hoc networks [10]. Therefore, there is an evident requirement for more competent optimization algorithm for the Travelling Sales Person problem. In recent times, there has been a huge interest in the Particle Swarm Optimization (PSO) due to its huge capability as an evolutionary algorithm, which is built on the regular social activities of flocks of birds and schools of fish [11]. In this paper, a modified version of the PSO, based on the use of fuzzy logic, is proposed for computation of the single source Travelling Sales Person Problem, which can be of great use in improving the routing in multi-hop communication networks. However, the PSO itself is not flawless. It can fall into the local optimum trap and converges slowly. By combining the PSO with fuzzy logic [12], these problems can be solved.

2 Background

A number of scientists have created simulations of various interpretations of the movement of organisms in a bird flock or a fish school. Particle Swarm Optimization works on the swarm of candidate solution called Particle, each having a velocity that is update recurrently and added to the particle's current position to move it to a new position. Two different entities can hold identical manners and theories without hitting together, but two birds can occupy the same position in space without bumping into each other. Bird and Fish adjust their physical movement to avoid predators, optimize environmental parameters such as temperature , seek food and mates etc. The swarm of particles were initialized with a population of candidate solutions through d-dimensional problem space to search the new solutions.

3 Particle Swarm Optimization

The Particle Swarm Optimization algorithm is grounded on specific social behaviors noted in flocks of birds, schools of fish, etc., from which specific features of intelligence emerge. After its development by Kennedy and Eberhart [13] in 1995, this evolutionary model has been genuinely studied on and developed in the past era. The standard PSO model comprises of a swarm of particles, moving interactively across the practical problem space to discover new solutions. Every particle has a position characterized by a position vector; where i is the index of the particle, and a velocity represented by a velocity vector. Every particle retains its particular best position so far in the vector pbest and the best position vector among the swarm is stored in a vector gbest. The exploration to find the optimal position (solution) advances as the particles' velocities and locations are updated. In each iteration, the fitness of each particle's position is calculated using a pre-defined strength function and the speed of each particle is updated using the gbest and pbest which were previously defined.

$V_{id}=WV_{id} + c_1r_1 (P_{Best} - X_{id}) + c_2r_2 (g_{Best} - X_{id})$;

i=1,2,3,……N and

d=1,2,3,……..D--------(1)

$X_{id} = X_{id} + V_{id}$ ----------(2)

c1 and c2 are two learning factors that control the effects of pbest and gbest on the way the particles travel along the exploration space. In many of the researches done on the PSO, c1 and c2 are given the value of 2. Nevertheless, mainly the particles far from the global best reach velocities with large values; hence have enormous position updates, and may leave the limits of the search space as a result. Therefore, the speed of the particles must be controlled. Velocity clamping, which could
be used in (1), gives a particle in a dimension the velocity of $V_{id,}$ if the right side of (1) for that particle goes beyond the maximum value in that dimension. It must be observed that various other improvements have been made into this algorithm. Manrice [14] proposed the use of a constriction factor χ to prevent the velocity from increasing out of bounds so that there would be no need for clamping. In the Constriction Factor Method (CFM), (1) is modified as follows:

$V_{id} = \chi[v_{id} + c_1r_1(P_{Best} - X_{id}) + c_2r_2 (g_{Best} - X_{id})]$---(3)
$\chi = 2 (| 2- \phi - \sqrt{\phi^2 - 4\phi}|)^{-1}$ if $\phi = c_1 + c_2 > 4$------(4)

Where w is the inertia weight which generally drops linearly in the interval [0,1]. c1 and c2 are positive constants, called acceleration coefficients, N is the total number of particles in the swarm, D is the dimension of the search space or in other words,

number of the factors of the function which are optimized, and r1 and r2 are two independently generated random numbers in the interval [0,1]. In (1), w is the inertia weight, which as stated earlier decreases in the interval [0,1]. w is one of the elements that regulates the velocity of the particles and hence their position updates. The larger the w, the more globally the particles search the space; and the smaller the w, the more locally the particles search the space. Thus, by reducing the w as the iterations move on, the global exploration transforms into a local search slowly.

3.1 PSO Algorithm

PSO Algorithm is a population based set of potential solutions evolves to approach a convenient solution for a problem. It is based on three factors:

1. The knowledge of the environment (its fitness value).
2. The individual's previous history (its memory).
3. The previous history of the states of individual's neighborhood.

In PSO algorithm, each individual is called a "particle." Particles have memory and are subjected to a movement in a multi-dimensional space that represents the belief space. Each particle's movement is the composition of an initial random velocity and to randomly waited influences that tends to return to the particles best previous position and also socially tends to move towards the neighborhood's best previous position. There are two kinds of basic PSO algorithms that is continuous and binary. This algorithm uses a real-valued multi-dimensional space such as belief space and evolves the position of each particle in that space using the following equations:

$$v(t+1) = (w * v(t)) + (c1 * r1 * (p(t) - x(t)) + (c2 * r2 * (g(t) - x(t))$$

$$x(t+1) = x(t) + v(t+1)$$

where:

V(t+1): Component in the dimension d of the ith particle velocity in the iteration t.

X(t+1): Component in the dimension d of the ith particle position in the iteration t.

C1, C2: Constant weight factors.

P(t): Best position achieved so long by the particle i.

G(t): Best position found by the neighbors by the particle i.

W : Inertia weight.

Algorithm is initialized with the particles at random positions and then it explores the search space to find better solution. In every iteration, each particle adjust it's velocity to follow two best solutions.

1. The first is the cognitive part where the particles follow its own best solution found so far which is called as P-best (particles best value).

2. The other best value is the current best solution of the swarm that is the best solution of any particle of the swarm, which is called as G best (global best value).

3.2 The Optimal Set Of PSO Parameters

The PSO algorithm is well-selected parameter that can set good performance, In our TSP problem, we use n number of swarms, the number of particles in each swarm and inertia weight according to Eberhart and Shi [3]., the acceleration coefficients c1, c2 represents the stochastic acceleration that force each particle towards G-best and P-best position. The fitness f can be calculated as the quality measures. Each particle has the position represented by I that is the index of the particle and a velocity represented by velocity vector i. After every iteration, the best position vector among the swarm is stored in a vector. The update of the velocity from the previous velocity to the new velocity is been determined.

How it works:

The birds are the solutions which are termed as particles.

1. Each particle has a fitness value and our aim is to find out the fitness value as evaluated by the fitness function.
2. Now, each particle will have its own velocity and position which is calculated by the velocity and position function respectively.
3. Now, initially PSO is initialized with the group of particles whose parameters are altered during each iteration.
4. In each iteration, every particle updates its fitness value and it's personal best value.
5. Meanwhile, during the process of each iteration the PSO reviews g-best (i.e., the best p-best value obtained by any particle to that point).

4 Modified PSO And Fuzzy-Based Method For Travelling Sales Person Problem

This section proposes a modified PSO algorithm with Fuzzy-based rules. Two main components of the proposed method are particle representation and fuzzy inference system, which are discussed in details.

4.1 Particle Representation

One of the highly significant issues in solving the Travelling Sales Person Problem and the ones similar to it, is how to encode a path in a network graph into a particle (or a chromosome). The way in which this encoding is prepared wholly influences the efficiency of the search process. In the method proposed in this paper, the position vector of a particle in the PSO is denoted by a priority vector, which comprises some guiding information about the nodes that represent the path in the graph. This technique of encoding, which was first used by Gen et al. [16] in a GA-based method, involves the significances of several nodes in the network. These priorities are primarily assigned randomly. The path is created as a sequence of nodes starting with the source node and ending at the destination node. According to the nature of the Travelling Sales Person Problem, as a path is being constructed, there are usually several nodes available for consideration, at each step of the path construction. In this approach, the node with the highest priority is chosen and the process continues until the destination node is reached.Fig. 1 illustrates a typical 20-node random network [17],on which the Travelling Sales Person Problem solving methods can be applied. The described encoding scheme is depicted in Fig. 2. p1, p2,… are the priorities of the nodes 1,2, …,
respectively. Fig. 2(b) shows a simple example of the encoding method explained above for the graph in Fig. 1.The path creation begins from node 1, and from the node adjacency relations, the node with the highest priority (node 4) is chosen as the next node in the path. Then, out of all the possible non-visited nodes that can be visited from node 4, the node with the highest priority (node 9) is chosen. The method is repeated until a complete path (1, 4, 9, 15,14, 20) is obtained.

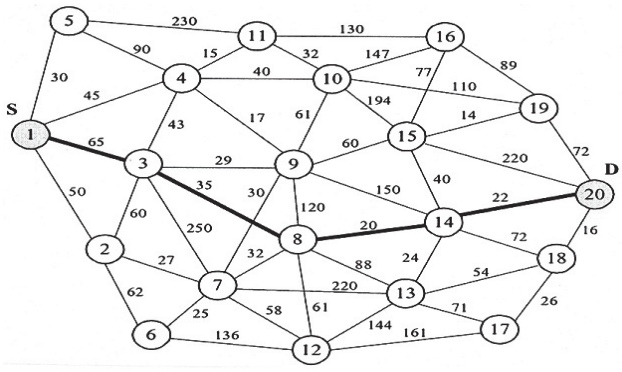


Figure 1. A typical 20-node random network. node numbers are encircled.The weights of the connecting edges are also shown adjacent to the
corresponding edges [17].

4.2. Fuzzy Inference System

Fuzzy inference is the process of making the mapping from a given input to an output using fuzzy logic. The process of fuzzy inference involves membership functions, logical operations, and If-Then rules. There are two types of fuzzy inference systems, Mamdani-type [18] and Sugeno-type [19]. These two types of inference systems differ slightly in the way outputs are established. Mamdani's fuzzy inference method, which is used in this paper, is the most regularly seen fuzzy methodology. Mamdani's approach was amongst the first control systems built using fuzzy set theory. Sinking into the trap of local optimum and slow merging are of the most important limitations of the PSO. There have been many

schemes proposed to solve the first problem, all of which comprise detecting the local optimum and preventing it. In [20], to prevent from obtaining the local optimum, when the velocity of the particle is lower than a specific level, but the fitness is not appropriate, a function is used to give a jolt to the particle and increase its velocity. In [21,22], a non-linear function for reducing the inertia weight is used to rise the velocity of a particle when the inertia weight is small, but the fitness is undesirable. All these methods prevent the particles to converge to a local optimum and some even speed up the convergence. In this paper, a fuzzy-based method proposed by Noroozi and Meybodi [12] is used to overcome the above-mentioned shortcomings of the PSO. In this method, a variable called the CBPE (Current Best Performance Evaluation), which indicates the fitness level of a particle at the moment, is used. CBPEmin is the best fitness attained so far and CBPEmax is the worst fitness attained so far. In (5), a normalized value NCBPE in the interval [0,1] is obtained using the three mentioned-above variables:(5)

$$NCBPE = \frac{CBPE - CBPE_{min}}{CBPE_{max} - CBPE_{min}} \text{ -------(5)}$$

In this method, a fuzzy function is defined with the parameters d1, d2 and NCBPE as its input and w as its output (Fig. 3).

$$d1 = |p_{best} - x| \text{ , } d2 = |g_{best} - x| \text{ ---------- (6)}$$

In (6), d1 and d2 represent the distance between the current position of the particle and its local best, and the global best, respectively. The lingual values "low","medium", and "high" are used to describe the parameters.d1 and d2 are determined with respect to hitthe size of the search space; therefore, these three parameters are the basis for the fuzzy system to decide the value of w, which is in the interval [0,1]. Choosing the correct fuzzy rules hjas a direct influence on the obtained results. A number of the rules used in this system are illustrated in Table 1.
It should be noted that a large number of rules in the system can not affect the result significantly, but the quality of the chosen rules is what produces accurate results

Figure 3. Fuzzy inference system for solving shortest path problem.

TABLE 1. FEW OF FUZZY INFERENCE SYSTEM RULES.

Rules	Input			Output
	d1	d2	NCBPE	W
1	LOW	LOW	LOW	LOW
2	LOW	LOW	NOT LOW	HIGH
3	LOW	NOT LOW	MEDIUM	HIGH
4	HIGH	HIGH	HIGH	HIGH

- Rule 1: if d1, d2, and NCBPE are low, the particle is near to the optimal best and the fitness is acceptable; therefore w is given a low value so that the search continues around the global optimum.
- Rule 2: if d1 and d2 are low, but NCBPE is not low,it means that the particle is close to the optimum, but the fitness is not acceptable (local optimum); therefore w is given a high value to increase the particle's velocity and change its position.
- Rule 3: if d1 is low, d2 is not low, and NCBPE is medium, it means that the particle is close to the local optimum but not close to the global optimum; hence w is given a high value to increase the particle's velocity and change its position
- Rule 4: if both d1 and d2 are not low; the particle's velocity must increase; therefore w is given a high value. Conferring to these fuzzy inference rules, the inertia weight w is modified.

5 Experimental Results

Several networks with many numbers of nodes are used to assess the functioning of the proposed approach. Networks with many numbers of nodes are created to investigate the quality of solution and the convergence speed of the proposed method. The highly apparent advantage of PSO is that the convergence speed of the swarm is very high. The weight variance of the present position of the particle swarm and the best position of the swarm Pbest will also be added to velocity vector for adjusting the next population velocity. These two alterations will enable particles to search around two bests. These Networks are arbitrarily generated with the maximum number 40 of nodes and edges are given random values between (0,100). To have a improved assessment, the proposed algorithm is run 100 times for each network. The other PSO parameters are chosen as: w reduces linearly from 0.9 to 0.2; c1 and c2 are chosen to be 2. The performance of the algorithm is assessed by success rate which is defined as the number of

times the shortest path is found over the number of runs. The hit rate for different swarm sizes between 10 and 40 is found to contrast the PSO with the proposed method which is illustrated in Fig. 4. In Fig. 5, the Average Best So Far is shown in various iterations of both algorithms. As it can be seen evidently, the proposed approach presents more precise results. Although both algorithms seem to be similar in the beginning iterations, the proposed method merges to a improved solution as the time passes. A unique tour for each of the test is found. Here The testing process using randomly chosen cities is more objective. The use of the fuzzy logic along with PSO helps in finding random city sets which leads to find an optimized solution. All statistics are generated after 100 runs on each city set. When number of iterations is taken as 100, the average results show considerable difference. The tours of the cities with 100 iterations obtained leads to finding optimized results by generating a tour in relatively short time. The experiments show that PSO with fuzzy logic finds better solutions for instances with up to 100 cities. Both average and best results are better than other algorithms. For city sets with 50 or less, PSO with fuzzy rules finds optimum routes in every execution. Corresponding to Fig. 4 and 5 , it is easy to see that by using a fuzzy inertia weight the performance of PSO can be enhanced and have similar or improved results than that of PSO with a linearly dropping inertia weight. Figure 5. Hit rate vs. Swarm Size for a network of 30 nodes

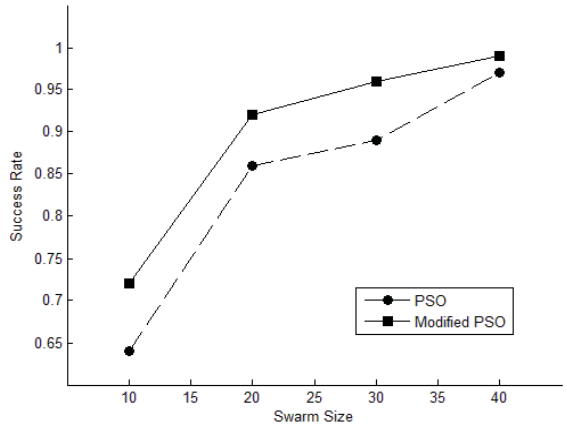


Figure 4. Hit rate vs. Swarm Size for a network of 30 nodes

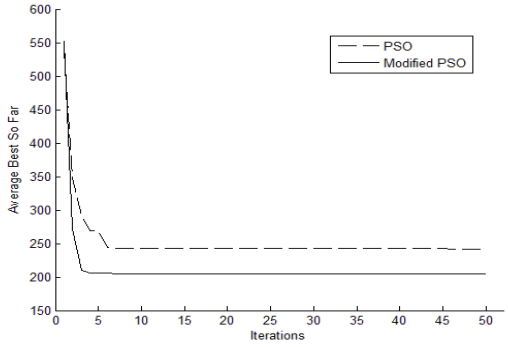


Fig 5. Average Best So Far over iterations for PSO and the proposed modified PSO.

6 Conclusion

Particle swarm optimization is a simple algorithm that seems to be effective for optimizing a wide range of operations. Significantly, it lies between genetic algorithm and evolutionary programming. It is highly dependent on stochastic processes with the variation towards the t-best and g-best by particles swarm optimizer to crossover operation utilized by Travelling Sales Person Problem. Travelling Sales Person Problem provides an optimized solution for networks to find the optimized route. A hybrid PSO –fuzzy search algorithm for solving the single source optimized path for TSP problem is presented in this paper. The method takes advantage of an efficient encoding mechanism in the PSO so as to include the parameters of the path graph in the representation itself. Additionally, in order to enhance the search efficiency, the inertia weight whose right values can prevent the search from falling in the trap of local optima, is determined using fuzzy rules. The results illustrate that the variation in values improves the performance of the algorithm significantly by achieving a success rate of 0.99.

7 References

[1] G. Desaulniers, and F. Soumis, "An efficient algorithm to find a shortest path for a car-like robot," IEEE Trans. Robot Automat. 11 (6), pp. 819–828, 1995.

[2] F.B. Zahn, and C.E. Noon, "Shortest path algorithms: An evaluation using real road networks", Transport. Sci. 32, pp. 65–73, 1998.

[3] N. Deo, and C. Pang, "Shortest-Path Algorithms: Taxonomy and Annotation," Networks, vol. 14, pp. 275-323, 1984.

[4] Moy, J., 1994. Open Shortest Path First Version 2. RFQ 1583,Internet Engineering Task Force. http://www.ietf.org.

[5] M. K. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," IEEE Trans. Neural Networks, vol. 4, pp. 941–954, 1993.
[6] D. C. Park and S. E. Choi, "A neural network based multi-destination routing algorithm for communication network," in Proc. Joint Conf.
Neural Networks, pp. 1673–1678, 1998.
[7] C. W. Ahn, R. S. Ramakrishna, C. G. Kang, and I. C. Choi, "Shortest path routing algorithm using hopfield neural network," Electron. Lett., vol. 37, no. 19, pp. 1176–1178, 2001.
[8] E.W Dijkstra. "A note on two problems in connection with graphs,"Numerische Mathematik, 1, pp. 269-271, 1959
[9] E.L. Lawler, "Combinatorial Optimization: Networks and Matroids", Holt, Rinehart, and Winston, New York, pp. 59–108, 1976.
[10] Ammar W. Mohemmed, Nirod Chandra Sahoo , and Tan Kim Geok, "Solving Shortest Problem using Particle Swarm Optimization," Applied Soft Computing, 8(4), pp. 1643-1653, 2008.
[11] J. Kennedy and R. C. Eberhart., "Particle swarm optimization.," Proceedings of the IEEE Int. Conf. on Neural Networks, Perth, Australia, pp. 1942-1948, 1995.
[12] M. H. Noroozi Beyrami, and M. R. Meybodi, , "Improving Particle Swarm Optimization using Fuzzy Logic," Proceedings of the Second Iranian Data Mining Conference, Amirkabir University of Technology, Tehran, Iran, pp. 108-120, 2008.
[13] J. Kennedy and R. C. Eberhart., "Particle swarm optimization.,"Proceedings of the IEEE Int. Conf. on Neural Networks, Perth,Australia, pp. 1942-1948, 1995.
[14] C. Maurice, "The swarm and queen: Towards a deterministic and adaptive particle swarm optimization," Proceedings of the IEEE Congress on Evolutionary Computation, Washington, pp. 1951 – 1957, 1999.
[15] R. C. Eberhart and Y. Shi, "Comparing inertia weight and constriction factors in particle swarm optimization," Proceedings of the IEEE Congress on Evolutionary Computation., San Diego, CA,pp. 84-88, 2000.
[16] M. Gen, R. Cheng, D.Wang, Genetic Algorithms for solving shortest path problems, in: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 401–406. 1997.
[17] C.W. Ahn, R.S. Ramakrishna, A genetic algorithm for shortest path routing problem and the sizing of populations, IEEE Trans. Evol.Comput. 6 (6), pp. 566–579, 2002.
[18] Mamdani, E.H. and S. Assilian, "An experiment in linguistic
synthesis with a fuzzy logic controller," International Journal of Man-
Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.
[19] M. Sugeno, "Industrial Applications of Fuzzy Control", Elsevier ,New York, 1985.
[20] L. Hongbo, and M. Abraham, "Fuzzy Adaptive Turbulent Particle Swarm Optimization", IEEE Fifth International Conference on Hybrid Intelligent Systems (HIS'05), pp.445-450, 2005.
[21] K. P. Wang, L. Huang, C. G. Zhou and W. Pang, "Particle swarm optimization for traveling salesman problem," Proceedings of International Conference on Machine Learning and Cybernetics, pp.1583-1585, 2003.
[22] B. Waxman, "Routing of multipoint connections," IEEE J. of Selected Areas in Communications, Vol. 6, No. 9, pp. 1622–1671,1988.
V6

SESSION

SIMULATION AND COMPUTATIONAL MODELING METHODS AND RELATED ISSUES

Chair(s)

TBA

Coupled Dam Erosion Analysis using DAKOTA and WinDAM

Mitchell L. Neilsen, Quan Kong, Matthew Bulleigh, and Geordy Williams
Dept. of Computing and Info. Sciences
Kansas State University
234 Nichols Hall
Manhattan, KS, USA

Abstract

Windows™ Dam Analysis Modules (WinDAM) is a set of modular software components that can be used to analyze overtopping and internal erosion of embankment dams. These software components are being developed in stages. Initial computational modules address the routing of floods through a reservoir with dam overtopping and evaluation of the potential for vegetation or riprap to delay or prevent failure of the embankment. Subsequent modules incorporate dam breach analysis and internal erosion analysis. Future modules will address analysis of non-homogeneous, zoned embankments, and analysis of various other forms of embankment protection.

The focus of this paper is on the development of new interfaces to conduct coupled analysis over a wide range of input parameters including both structural and flow properties. The Sandia National Laboratories' Design Analysis Kit for Optimization and Terascale Applications (DAKOTA) provides a flexible and extensible toolkit to enhance existing WinDAM analysis codes and supports several iterative systems analysis methods including reliability analysis, sensitivity analysis, uncertainty quantification, and many different types of parameter studies. The focus of this paper is on parameter studies, but plans are underway to extend the interface to handle other types of analyses.

Keywords: Dam erosion, hydraulic modeling, hydrology, parameter study, simulation.

1. Introduction

Windows™ Dam Analysis Modules (WinDAM) is a set of modular software components that can be used to analyze overtopped earthen embankments and internal dam erosion. The development of WinDAM is staged. The initial computational model addresses routing of the flood through the reservoir with dam overtopping and evaluation of the potential for vegetation or riprap to delay or prevent failure of the embankment. The first module, WinDAM A+, also incorporates the auxiliary spillway erosion technology used in SITES. However, unlike SITES, it allows a user to simultaneously analyze up to three auxiliary spillways and embankment erosion on the dam. The next computational model, WinDAM B, incorporates dam breach analysis; i.e., the breach failure of a homogeneous embankment through overtopping and drainage of stored water in the reservoir. The third model, WinDAM C incorporates internal erosion analysis. In addition, work is currently underway to include analysis of non-homogeneous embankments, and analysis of other forms of embankment protection. The two most common causes of earthen embankment and levee failure are overtopping and internal erosion. An example of the empirical analysis of dam overtopping in the lab is shown below in Figure 1.

WinDAM is designed to address the dam safety concerns facing the national legacy infrastructure of over 11,000 small watershed dams constructed with US Federal involvement over a seventy-year period. The US Department of Agriculture -Agricultural Research Service (USDA-ARS) Hydraulic Engineering Research Unit (HERU), the US Department of Agriculture-Natural Resources Conservation Service (USDA-NRCS), and Kansas State University are working jointly to develop and refine this software.

Public Law 78-534 – Flood Control Act of 1944 started the small watershed program, and it was followed by Public Law 83-566 – Watershed Protection and Flood Prevention Act of 1954. Starting in 1958, an average of one dam per day was constructed over a period of twenty years.

Figure 1: Empirical dam analysis at USDA HERU

In 1983, the USDA-SCS-ARS Emergency Spillway Flow Study Task Group (ESFSTG) was formed to develop better technology for earth spillway analysis. The ESFSTG collected data on dams that experienced either emergency spillway flow at least three feet deep or significant damage during a storm event. Approximately one hundred sites were selected for more in-depth evaluation and data collection, and data analysis began in 1990 from the field spillway data initially collected. Tests were conducted at the USDA-ARS outdoor Hydraulic

Engineering Research Unit (HERU) Laboratory near Stillwater, Oklahoma, to further understand spillway performance processes such as vegetal cover failure, flow concentration, surface detachment, and headcut migration as shown in Figure 2. These findings were incorporated into the DAMS2 software, and then into Stability and Integrity Technology for Earth Spillways (SITES) software in 1994. The bulk length concept was replaced by SITES spillway erosion modeling technology in other USDA-NRCS references. Although SITES may be used to analyze existing dams and spillways, it was developed primarily for design and it was developed over a period in which computational capability was much more limited than today. The legacy infrastructure of aging structures also means there has been a transition from design of new structures to analysis of existing structures. For example, existing structures may overtop as a result of watershed changes or sediment deposition within the flood pool leading to inadequate spillway capacity. WinDAM builds on and extends the existing technology in SITES to provide the needed capability for these types of analyses.

Figure 2: Large flume test at USDA HERU

Windows™ Dam Analysis Modules (WinDAM) is a collection of modular software components that can be used to design and analyze the performance of earthen dams. The focus of the initial collection of computational modules is to evaluate earth dams subjected to flooding that may result in overtopping of the dam embankment and auxiliary spillway(s) [1]. The reservoir routing model incorporated into the software includes outflow from a principal spillway, up to three auxiliary spillways, and over the top of the dam embankment. For conditions where overtopping of the embankment is predicted, the hydraulic attack on the downstream face can also be evaluated using the initial software modules in WinDAM A+. The downstream face of a dam is typically protected using vegetation or riprap. WinDAM A+ has been extended to include erosion and breach computations for conditions where the hydraulic attack exceeds that which can be withstood by the vegetal or riprap lining, and the resulting modules are in WinDAM B. The next version, WinDAM C, will incorporate analysis of failures caused by internal erosion or piping failures. To evaluate erosion in each auxiliary spillway, the SITES Spillway Erosion Analysis module with Latin Hypercube Sampling (SSEA+LHS) is integrated with WinDAM A+. The Embankment Erosion Module is extended to include a Breach Analysis Module. The current model assumes the dam has a homogeneous embankment. It is most applicable for the analysis or design of embankments constructed from cohesive soil materials. It is anticipated that the model will be expanded to handle zoned embankments in WinDAM D. The breach technology enabling this expansion is currently under development.

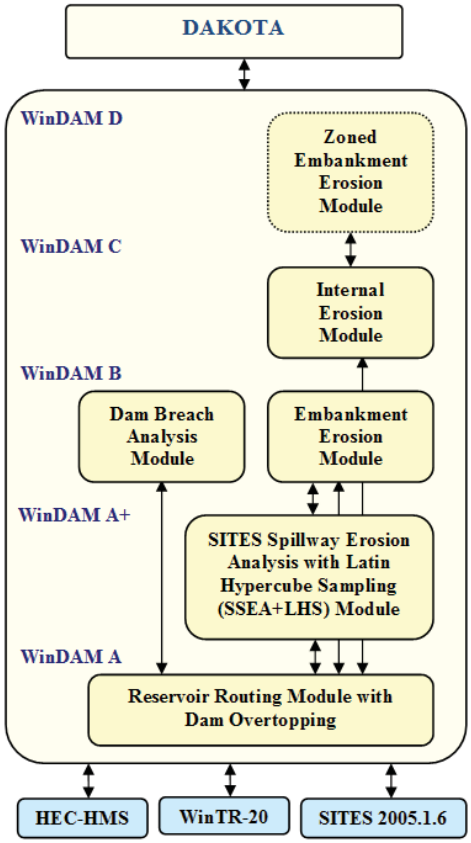


Figure 3: WinDAM software architecture

Inputs to WinDAM include a description of the reservoir inflow hydrograph, reservoir storage capacity, all spillway properties, the dam cross section and profile, properties of the embankment, and input parameters for the breach analysis module. Inflow hydrographs can also be obtained automatically from other reach routing software, such as SITES 2005.1.6, SSEA+LHS [2], HEC-HMS [3], HEC-RAS, or WinTR-20 as shown in Figure 3.

Outputs include a description of the reservoir water surface variation with time, the hydrographs associated with outflow through each of the spillways and over the top of the embankment, and a description of the attack on the dam embankment and downstream embankment face. Output hydrographs can be directed to external reach routing software. Output information is generated in both text and graphical format. The software generates ASCII text and/or XML control files for the model simulator which performs the model calculations. Output from the

simulator is written to intermediate XML and/or fixed-format ASCII text files that can be read by a Graphical User Interface (GUI) to display results in both text and graphical format. Due to the well-defined interfaces that automatically convert data to and from different forms, it is easy for software developers to interface the system with existing analysis software and with software under development. Templates that can be used in conjunction with DAKOTA are also automatically generated.

In the DAKOTA system, a strategy is used to create and manage iterators and models [4]. A model contains a set of variables, an interface, and a set of responses, and an iterator operates on the model to map the variables into responses using the interface. The WinDAM system is used to automatically generate DAKOTA input files. For parameter studies, the user indirectly specifies these components through strategy, method, model, variables, interface, and responses keywords. Then, DAKOTA is invoked to iterate on the WinDAM simulation models, or vice versa, as needed to generate output. Instead of having WinDAM drive the analysis, we can also allow DAKOTA to be used to drive the analysis in an iterative fashion [6].

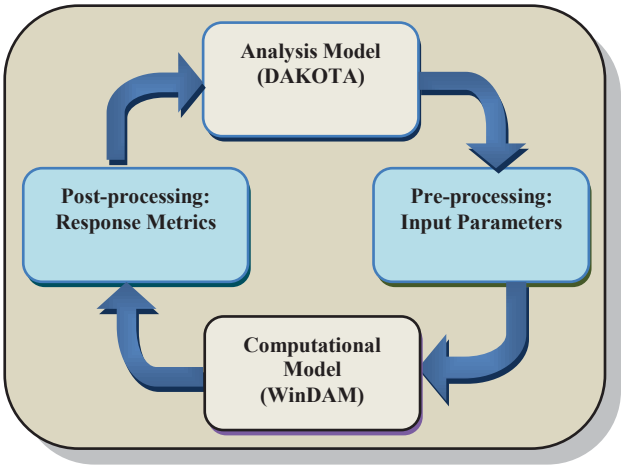


Figure 4: Iterative analysis with DAKOTA

DAKOTA supports several different options for the Design and Analysis of Computer Experiments (DACE):

- *Sensitivity Analysis (SA)* - determine which inputs have the most influence on the output.
- *Uncertainty Analysis (UA)* - compare the relative importance of model input uncertainties on output.
- *Response Surface Approximation (RSA)* - use sample input and output to create an approximation to the simulation output; e.g., neural net, etc.
- *Uncertainty Quantification (UQ)* - take a set of distributions on the inputs and propagate them through the model to obtain distributions on the outputs.
- *Parameter Studies* – specify a range of input parameters and compute the corresponding output which can be displayed in text or graphical format.

In this paper, we describe an innovative, new DAKOTA User Interface to create the input file used by DAKOTA to drive the analysis as shown below in Figure 5.

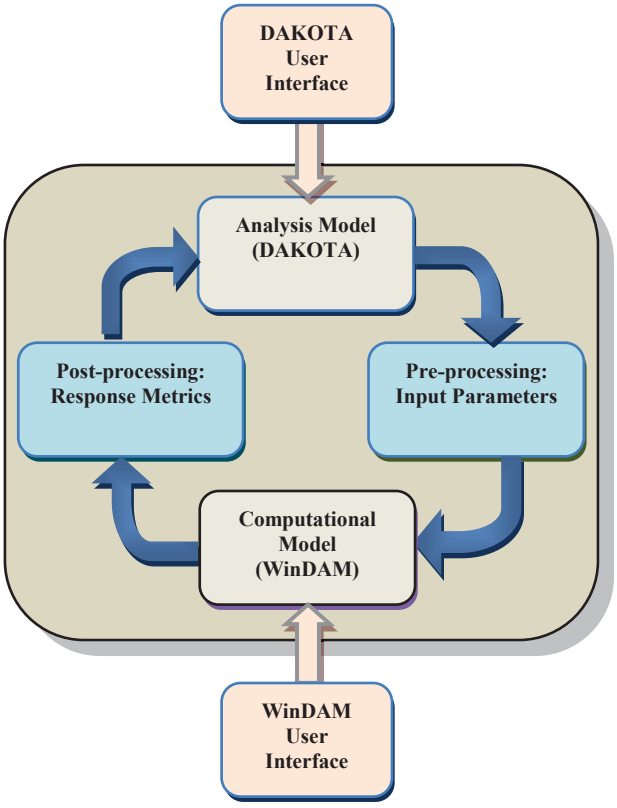


Figure 5: User interfaces to create models

Within the Computational Model, the input parameters generated by DAKOTA are combined with the dam model generated using WinDAM. A mesh is generated and the hydrodynamic properties are computed to determine the flow through each spillway and over the top of the dam using WinDamSim as shown in Figure 6.

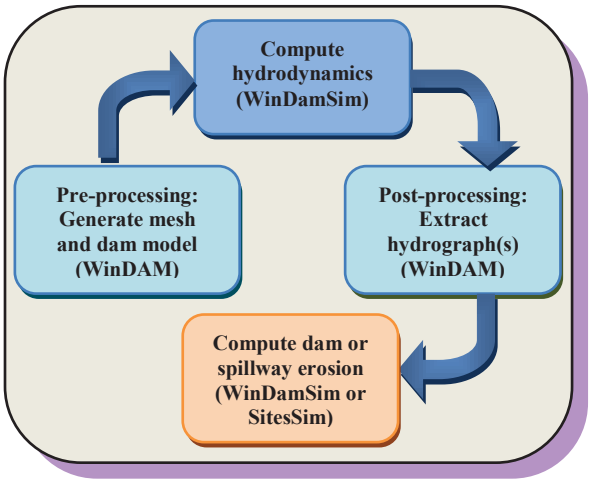


Figure 6: Current computational model

Then, the hydrographs generated are passed as input to the erosion models to compute the amount of erosion generated through each auxiliary spillway and over the top of the dam. At the same time, we are working to refine the models used within WinDAM to leverage advances in fluid flow models. We have recently shown

how to generate new open-source computational fluid dynamics (CFD) models from existing WinDAM models, and to compute the resulting flows using OpenFOAM [7]. The next step will be to use these new flow models to conduct coupled analysis at the particle-fluid level by coupling refined erosion models based on the existing WinDAM erosion models with these new CFD models as shown in Figure 7.

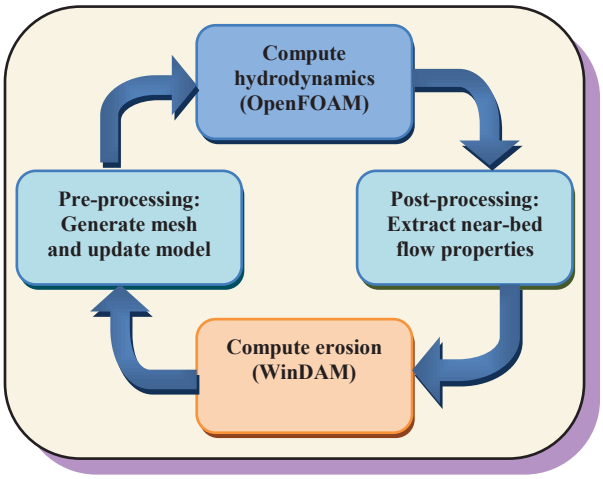


Figure 7: Coupled computational model

In WinDAM, hydraulic flow is routed through the reservoir by balancing inflow, outflow, and storage under the assumptions of a level reservoir surface with all outflow being a function of reservoir water surface elevation. Stage-storage properties of the reservoir are entered in tabular format with elevation in feet and the corresponding surface area in acres or storage volume in acre-feet. Reservoir inflow hydrographs are entered into WinDAM as series of time-discharge pairs with time in hours and flow in cubic feet per second (cfs) or cubic meters per second (cms).

In what follows, Section 2 describes WinDAM and the development of DAKOTA input control files from existing WinDAM models using the new DAKOTA User Interface as shown in Figure 5. These models can be used for both overtopping breach and internal erosion analysis; e.g., WinDAM B and WinDAM C models. The user interface can also be used to initiate the analysis and visualize the results. Section 2 describes the integration of WinDAM with other tools to perform coupled analysis. Section 3 covers the use of DAKOTA to perform simple parameter studies and visualization of the results using a graphical user interface developed using jFreeChart and JZY3D [11]. Finally, Section 4 concludes the paper and describes some directions for future research.

2. Computational Models

The computational model currently incorporated into WinDAM assumes stepwise steady-state flow and a level water surface in the reservoir. The mass balance equation governing flow through the reservoir for any given time step may be obtained by averaging conditions over the time step. The inflow to the reservoir is a known function of time only, and is obtained through application of appropriate hydrologic models such as SITES 2005.1.8, HEC-HMS [3], or WinTR-20. The outflow from the reservoir is the sum of the outflow from all spillways and the flow over the top of the dam. Using the assumptions of a level water surface in the reservoir and stepwise steady flow, each of the individual outflows may be treated as a unique function of the reservoir water surface elevation. Likewise, the storage volume in the reservoir also becomes a unique function of the reservoir water surface elevation.

2.1 WinDAM B

The primary purpose of WinDAM B is threefold:

- Hydraulically route one input hydrograph through, around, and over a single earthen dam.
- Estimate auxiliary spillway erosion in up to three earthen or vegetated auxiliary spillways.
- Estimate erosion of the earthen embankment caused by overtopping of the dam embankment.

Since WinDAM B does not include any specific hydrology component, the user must create the input hydrograph using other software. This allows the user the flexibility to choose the hydrologic software most suitable for analysis of site conditions; e.g., HEC-HMS, etc.

WinDAM B assumes the embankment of the dam is a homogenous earthen material. Many USDA-NRCS dams are homogenous earthen fill, so the WinDAM B model applies. Future versions of WinDAM will address zoned embankments where each zone exhibits different erosion resistance from other zones. Most existing USDA-NRCS dams are built with a single earthen auxiliary spillway. In rehabilitation of old USDA-NRCS-designed dams, it is more common to utilize additional auxiliary spillways. As a result, WinDAM B allows the user to input up to three auxiliary spillways, each spillway can be designed with a zoned, heterogeneous embankment with different physical characteristics for each zone.

Computation of the discharge through the area of the breach, if any, is unit discharge based on the effective width. If breach is to be evaluated, the associated erosion is assumed to be initiated in an area corresponding to maximum unit discharge over the top of the dam.

Following breach initiation, the unit discharge is computed assuming negligible energy loss from the reservoir to the hydraulic control and critical flow conditions with hydrostatic pressure at the hydraulic control. The processes that determine the erosion during embankment breach are dependent on the breach geometry and the breach area discharge. The way in which the erosion will progress depends on the local geometry and discharge. Initially, the headcut (local vertical) may not be sufficiently high to generate the plunging action that is associated with typical headcut advance. Likewise, during latter stages of the process, the headcut may become submerged resulting in a slower rate of erosion.

2.2 WinDAM C

The primary purpose of WinDAM C is to extend prior models to include internal erosion models developed as a result of empirical analysis at the USDA-ARS HERU as shown in Figure 8.

Figure 8: Internal erosion analysis at USDA-ARS HERU

Users can specify a range of different material properties and different erosion models as shown in Figure 9.

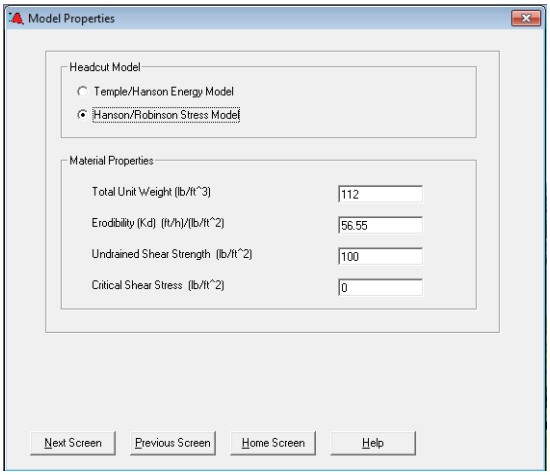


Figure 9: Silty sand (SM) material properties

Here are the inputs for the synthetic erosion model which was empirically evaluated in the lab as shown in Figure 8. The material is an erodible silty sand (SM). Once the properties have been entered, the user can evaluate the amount of internal erosion that is expected to occur and the time before failure. In contrast, if the dam consists of a much stronger clay material, then the input might be as shown in Figure 10. Not surprisingly, this results in much less erosion, in fact the same inflow resulted in no failure even after 72 hours.

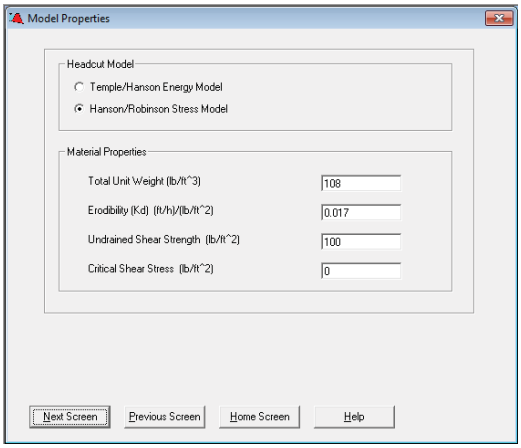


Figure 10: Clay (CL) material properties

Some model output showing a cross section of the dam from WinDAM C is shown below in Figure 11.

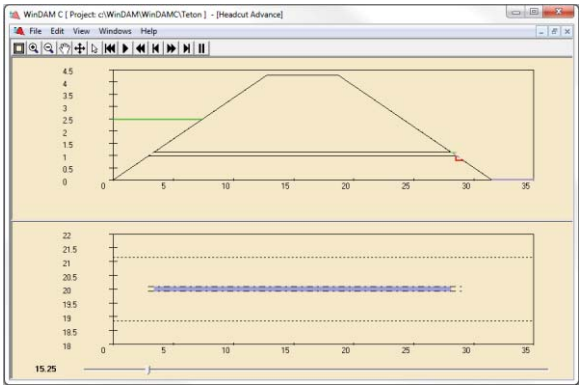


Figure 11: WinDAM C model output

Next, we turn our attention to the construction of input control files for DAKOTA to vary the model inputs between these two extremes.

3. Analysis Models

Spillway designs are compared by determining both the stability and integrity of the spillway when it is subjected to a given design storm. A *stability design hydrograph*, when routed through a reservoir, generates the maximum auxiliary spillway outflow that the reservoir will be expected to pass without erosion damage, but a *freeboard hydrograph* represents the maximum flow for which the structure is designed. Naturally, this is the most important consideration in designing an earth (soil, rock, or both) spillway. Even though extremely large discharges may cause significant erosion, the spillway must not breach during passage of the *freeboard hydrograph*.

Our new integrated development system for water resource site analysis is designed to fully integrate the simulation models in WinDAM with the uncertainty quantification, sensitivity analysis, and parameter studies capabilities in DAKOTA. This novel, new development environment interactively guides user input, invokes the sampling and simulation models in the background, and

parses the results to automatically generate output hydrographs, summary tables, and graphs.

A new DAKOTA user interface is developed to create the analysis model and specify the type of study requested. For example, consider the two cases shown above, we may want to analyze the range of parameters between these two extremes. The WinDAM + DAKOTA User Interface allows us to open an existing WinDAM project directory, select parameter study, and then specify the type of parameter study as shown in Figure 12.

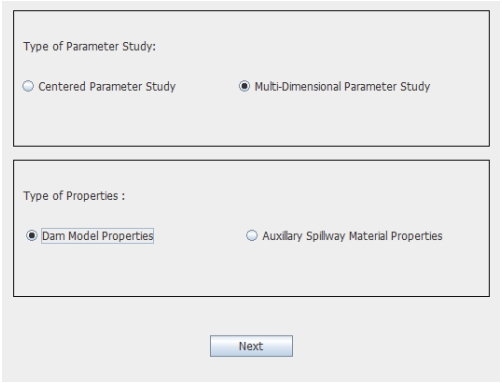


Figure 12: Parameter study input screen

Then, the user can specify a range of parameters denoting material from silty sand to clayey sand, and the user can also specify the number of partitions in each dimension, as shown below in Figure 13.

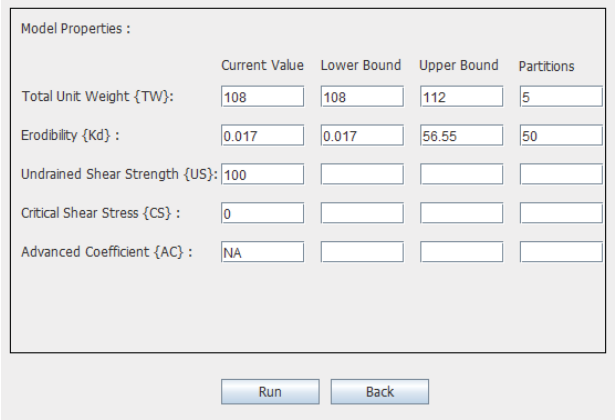
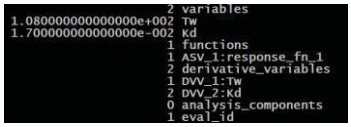


Figure 13: Parameter study properties

Finally, the user can simply click on the Run Button to generate the DAKOTA input file shown in Figure 17, and fire up DAKOTA and WinDamSim to perform the analysis using iterative analysis on 6 x 51 = 306 input data sets. The values for Total Unit Weight (Tw) and Erodibility (Kd) are computed by DAKOTA using fixed intervals at the end of each partition. Then, the selected values are parsed to generate decimal values in 10-column format using a simple Perl script, called parseWinDamIn, as shown in Figure 14. A Sandia double-precision pre-processing tool, dprepro, is used to replace the tokens for {Tw} and {Kd} in the template file shown in Figure 15 with the values in the parameter input file to generate one simulation input file as shown in Figure 16. The template file is generated automatically from the original input control file, similar to the one shown in Figure 16.



params.in

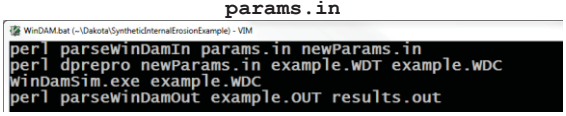


Figure 14: WinDam.bat script

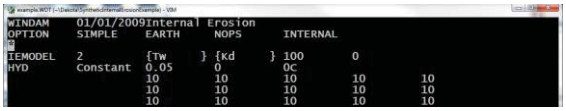


Figure 15: WinDAM template file

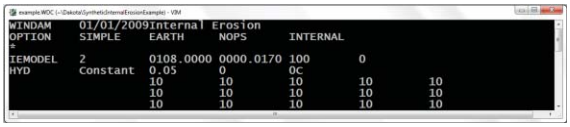


Figure 16: WinDAM input file

Once the iterative analysis has been completed on all 306 input files, the output can be viewed in text or graphical format. Text data is stored in the file *dakota_multidim.dat* based on the file generated by the interface as shown below in Figure 17.

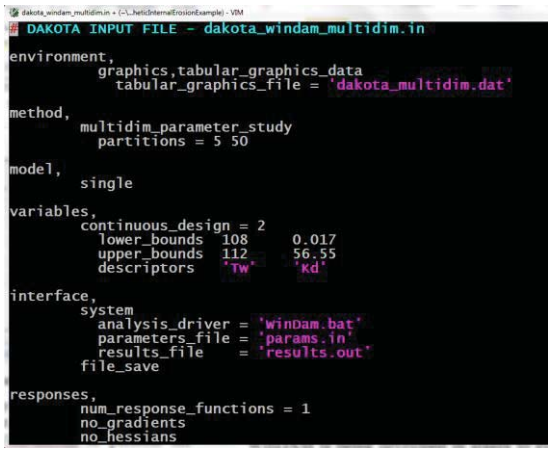


Figure 17: DAKOTA input file

The interface can also be used to display the data in graphical format. Graphs can be displayed in both 2-D and 3-D format. The 3-D graphs are generated using JZY3D [11] as shown in Figure 18.

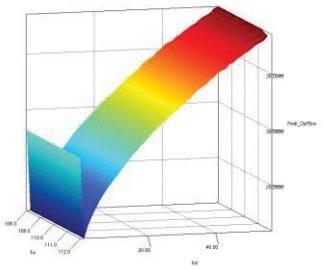


Figure 18: JZY3D graph generated

Since there is very little difference based on changes in Tw, but much change in the interval from 0.017 to 5 for Kd, a user might decide to change the inputs and re-run the analysis. This can also be done directly by modifying the text control file as shown below in Figure 19 and re-running the analysis from the command line using the command: $ **dakota dakota_windam_multidim.in**

```
method,
        multidim_parameter_study
          partitions = 2 150

model,
        single

variables,
        continuous_design = 2
          lower_bounds   108      0.017
          upper_bounds   112      5.0
          descriptors    'Tw'     'Kd'
```

Figure 19: Updated DAKOTA input file

Then, after opening the output data file, a user can specify the fields to be used for each parameter, and the type of graph to be generated – scatter plot – as shown below in Figure 20.

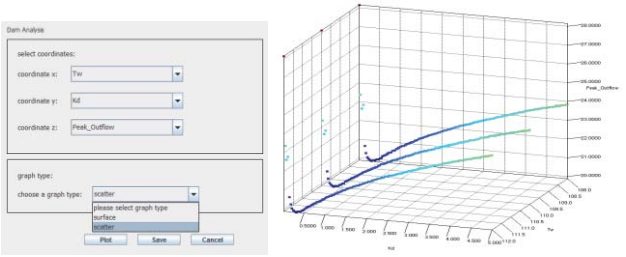


Figure 20: Output graph interface

Again, with more activity near 0, the user could adjust the Kd range to go from 0.01 to 0.05 resulting in the output shown below in Figure 21.

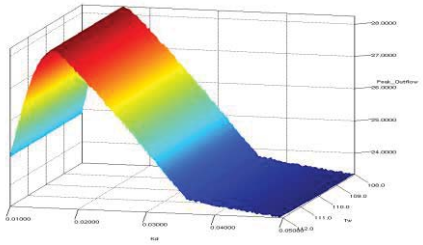


Figure 21: Output peak flow with Kd in [0.01, 0.05]

This type of iterative analysis can continue as needed to allow hydraulic engineers to focus in on the areas of interest and complete a complex analysis in a short time.

4. Conclusions

WinDAM is being developed in stages to evaluate the performance of earth dams. Existing modules with well-defined interfaces enable efficient integration of existing legacy software with new innovations. The system provides tools that can be used to better understand the structure, function, and dynamics of water control structures. This paper describes how WinDAM models can be analyzed using a novel new DAKOTA interface. The next step will be to couple CFD flow models with physical models to model the erosion that results from the given flows. High-level analysis can still be performed by combining these new models with DAKOTA.

Acknowledgements

We would like to thank the USDA-ARS and USDA-NRCS for use of the images used in this paper and the ongoing joint cooperative research that has enabled this work.

References

[1] D.M. Temple, G.J. Hanson, and M.L. Neilsen, "WinDAM -- Analysis of overtopped earth embankment dams", In *Proc. of the ASABE Annual Conference*, Paper Number 062105, 2006.

[2] M.L. Neilsen, D.M. Temple, and J.L. Wibowo, "A distributed hydrologic simulation environment with latin hypercube sampling", In *Proc. of the Intl. Conf. on Env. Modelling and Simulation*, No. 432-032, St. Thomas, USVI, Nov. 22-24, 2004.

[3] United States Army Corps of Engineers, "Hydrologic modeling system HEC-HMS User's Manual", CPD-74A, Ver. 3.5, USACE, HEC, 2010.

[4] B.M. Adams, W.J. Bohnhoff, K.R.Dalbey, J.P. Eddy, M.S. Eldred, D.M. Gay, K. Haskell, P.D. Hough, and L.P. Swiler, "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual," Sandia Technical Report SAND2010-2183, Dec. 2009. Updated Dec. 2010 (Ver. 5.1) Updated Nov. 2014 (Ver. 6.1)

[5] D.M. Temple and G. J. Hanson, "Earth dam overtopping and breach outflow", In *Proc. of the World Water and Environmental Resources Congress*, Anchorage, Alaska, ASCE, 8 pp., 2005.

[6] M.L. Neilsen, "Global sensitivity analysis of dam erosion models", in Proceedings of the 10th International Conference on Scientific Computing, Paper No. CSC-3502, July 22-25, 2013.

[7] M.L. Neilsen, "Computational fluid dynamics models for WinDAM", In Proceedings of the 30th International Conference on Computers and their Applications, Honolulu, Hawaii, March 9-11,2015.

[8] D.M. Temple, J. Wibowo, M.L. Neilsen, "Erosion of earth spillways", In *Proc. of 23rd United States Society on Dams (USSD) Annual Meeting and Conference*, pp. 331-339, 2003.

[9] M.L. Neilsen and D.M. Temple, "A concurrent simulation model for analysis of water control structures at the watershed scale", In *Proc. of the Intl. Conf. on Par. and Dist. Proc. Tech. and Apps.*, (PDPTA 2010), pp. 1565-1570, June 26-29, 2000.

[10] OpenFOAM 2.3.x - software and documentation retrieved from www.openfoam.org, 2015.

[11] JZY3D, http://jzy3d.org/download-0.9.1.php, 2015.

[12] K. Visser, R. Tejral, M. Neilsen, "WinDAM C Earthen Embankment Internal Erosion Analysis Software" in Proc. of the 2015 SEDHYD 5th Federal Interagency Hydrologic Modeling Conference, Reno, NV, March 23-27, 2015.

Application of SimulationX® – based Simulation Technique to the Design of Opening Area for a Valve Plate of Swash Plate Type Piston Pump

A. Jun Hyeong Bae[1], B. Won Jee Chung[1], and C. Seong Bhin Kim[1]
[1]School of Mechatronics, Changwon National University
Changwon, 641-773, South Korea

Abstract - ***Hydraulic system is a device that converts energy into fluid energy and mechanical energy when a closed space, which is formed by the static pressure of a fluid, moves or changes. Of these hydraulic systems, swash plate type piston pump, as shown in Fig.1, is the most important and basic component composing the hydraulic system. In this paper, we will investigate the design factors of opening area in order to consider the kinematic stability of the valve plate (main component of swash plate type piston pump), conducting an analysis of the reduction effects of pressure pulsation and flow ripple depending on design factors using SimulationX® (Germany), hydraulic analysis program. Further we will perform a structure analysis to confirm the kinematic stability of the valve plate in swash plate type piston pump. Finally, this paper will analyse the effects of pulsation of 1-step V notch type and 2-step V type notch and 2-step U type notch, which were not discussed in Jang et al. [6] for finding out the effects of pulsation reduction.***

Keywords: Swash plate type variable piston pump, Pulsation, Structure analysis, Opening area, Notch design of valve plate, SimulationX®

1 Introduction

Hydraulic system is a device that converts energy into fluid energy and mechanical energy when a closed space, which is formed by the static pressure of a fluid, moves or changes. Although it is inferior to electrical and electronic systems in terms of controllability, noise, and price, it has been used in broader fields such as construction equipment, vessel, machine tool, automobile, and industrial machine because it is advantageous to miniaturization and weight reduction due to its high power density [1].

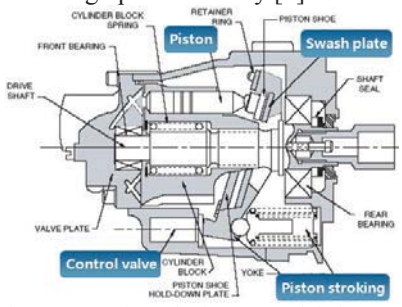


Fig. 1 Basic structure of swash plate type axial piston pump [2]

Of these hydraulic systems, swash plate type piston pump, as shown in Fig.1, is the most important and basic component composing the hydraulic system, which plays a role of converting mechanical energy into fluid energy [3]. In addition, its capacity can become variable easily by the changes of a swash plate angle, compared to the bent-axis type axial piston pump, it has a simple structure, and it can control high speed because the moment of inertia is small at the portion of variable capacity and so its use is increasing [4]. A swash plate type variable piston pump is largely used as a main pump for heavy construction equipment and recently, its research and development is being conducted in the direction of reinforcing performance, environment, low noise, and regulations of hydraulic system containing a swash plate type variable piston pump [1]. Thus weight reduction and miniaturization of hydraulic system, high speed and high pressure control system, and low pulsation should be considered together.

A valve plate, the main component of swash plate type piston pump, can have binding post-tensioning force and separating force due to its relative movement with cylinder block [5]. If its binding post-tensioning force is high, friction occurs between the two components and there is a concern for mechanical fracturing. In the meanwhile, if the separating force is high, leakage flow occurs from the gap. Therefore, designing both the opening area and the notch shape according to the performance and specification of pump is a very important point of design because flow ripple and pressure pulsation in cylinder occur according to the opening area and notch shape of valve plate.

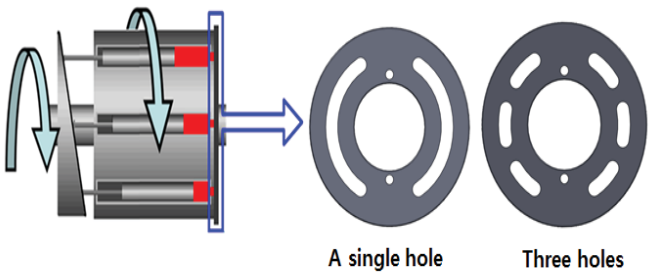


Fig. 2 Valve plate of swash plate type piston pump [6]

Jang *et al.* [6] has investigated the pressure pulsation and the flow ripple according to the shape of notch of valve plate, suggesting kinematics analysis of swash plate type piston pump. It has been confirmed in [6] that the pulsation of V notch type is effectively reduced compared to that of circular notch type. However, Jang *et al.*'s paper only performed an analysis of circular type and V type simply; they neither made an analysis depending on the design factor of opening area nor conducted a research on the interval between opening holes, and did not study the other notch types. There are several holes to the valve plate, as shown in the valve plate (right picture) of Fig, 2. For a single hole, the flow at the suction and discharge part is constant, but the valve plate cannot lead to kinematic stability. Additionally, for the three holes, the kinematic stability of the valve plate can be secured and thereby its durability can be extended.

In this paper, we will investigate the design factors of opening area in order to consider the kinematic stability of the valve plate (main component of swash plate type piston pump), conducting an analysis of the reduction effects of pressure pulsation and flow ripple depending on design factors using SimulationX® (Germany), hydraulic analysis program. Further we will perform a structural analysis to confirm the kinematic stability of the valve plate in swash plate type piston pump. Finally, this paper will analyse the effects of pulsation of 1-step V notch type and 2-step V type notch and 2-step U type notch, which were not discussed in Jang *et al.* [6] for finding out the effects of pulsation reduction.

2 Theoretical Approach and Structure analysis

The general movement of piston pump can be explained by pumping of a constant amount of flow while the piston is doing a repeating motion at the suction and discharge ports of the pump. At this time, when transition is made from the area of low-pressure suction to that of high-pressure discharge, in other words, when pressure pulsation and flow ripple occurs at the part that there is a big difference in pressure pulsation, this pulsation has a considerable influence on hydraulic system. Reducing such pressure pulsation and flow ripple is very important because it can extend the durability of hydraulic pump and thus improve reliability. As shown in Fig. 3, when suction and discharge begins, the opening area changes rapidly and so a notch should be designed so that the pressure pulsation and flow ripple cannot be occurred. Regarding this notch design, Jang *et al.* [6] has already found that pulsation can be reduced in V notch type compared to circular notch type.

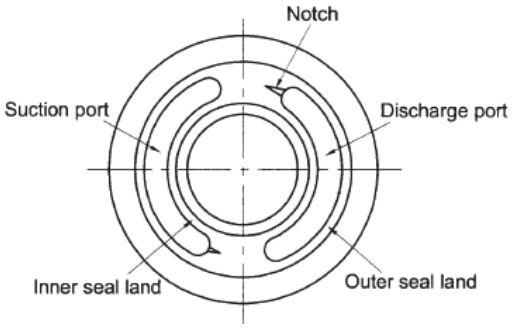


Fig. 3 Basic valve plate [4]

Figure 4 represents the factors for designing the valve plate in which, in general, 2~4 small kidney-shaped holes are machined for suction and discharge ports, respectively. In Fig. 4, '*a*' (distance between suction and discharge ports) is designed equal to or larger than the long radius of the cylinder port and '*b*' is usually designed 1/4 of *a* [7]. In order to consider kinematic stability, a valve plate can be classified into 3 types according to the number of factor '*b*' as shown in Fig. 5. For structural analysis, 8 cases are illustrated by Table 1, based on number of opening areas, angle of '*b*' and distance of '*b*'.

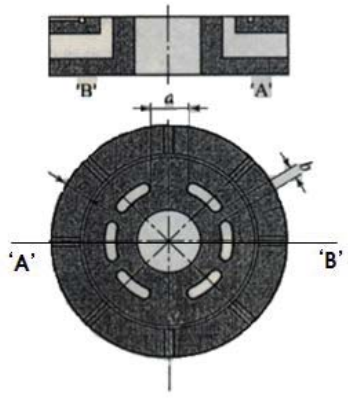


Fig. 4 Structure of valve plate [7]

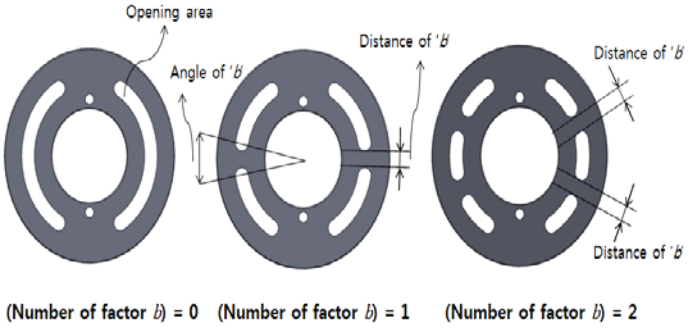


Fig. 5 3 Types of valve plate according to factor '*b*'

Table 1 Cases of Valve Plate

Case	Number of opening areas	Angle of *b*	Distance of *b*
1	1	0	0
2	2	24	7.9
3	2	20	5.3
4	2	16	2.8
5	2	14	1.5
6	3	24	8
7	3	19.5	5.3
8	3	15	2.1

The material for valve plate is set as SPHC (Steel Plate Hot Commercial) as shown in Table 2, while the boundary conditions for structure analysis using ANSYS workbench® are 1) pump pressure = 300 bar, 2) force pushed on the valve plate by a cylinder block = 28GN, and 3) fixed condition on the centre of valve plate. Then the analysis results are presented in Fig.6. In Case 1 without '*b*' factor, the maximum stress is 60MPa, which has occurred in both sides of opening hole, and the safety factor is found to be 4.5.

Table 2 Matrial properties of SPHC

SPHC	Value
Density (kg/m^3)	7,850
Young's modulus (GPa)	210
Poisson's ratio	0.3
Tensile yield strength (MPa)	270

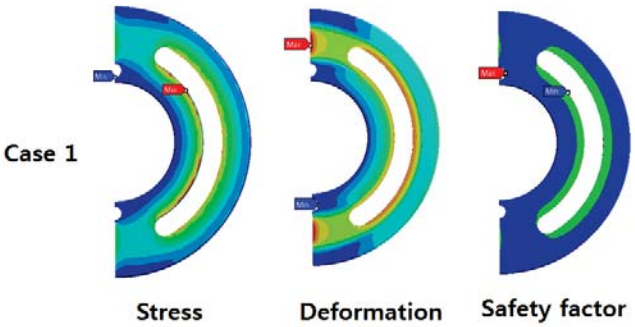


Fig. 6 Result of structure analysis for Case 1

Figure 7 shows that, as it goes from Case 2 to Case 5 (in other words, as the angle and interval of '*b*' factor becomes shorter), the maximum stress exerted on the both sides of opening hole is concentrated on the region corresponding to '*b*' of opening hole. The maximum stress was found to be 45~55MPa and the safety factor 4~4.5. In Fig. 8, Cases 6 to 8 are related to the analyses of two '*b*' factors and three opening holes. Similarly in Cases 2 to 5, it is found that the angle and distance of '*b*' factor becomes shorter as it goes from Case 6 to Case 8. In the similar manner to Cases 2 to 5, the stress is concentrated on the region corresponding to '*b*' of opening hole. For the analysis of three opening holes, the maximum stress is 40~50MPa and the safety factor 4.5~5. Thus it can be concluded that the stress of valve plate is reduced by designing '*b*' factor for Cases 6 to 8 rather than Cases 2 to 5. Consequently kinematic stability is improved based on safety factor.

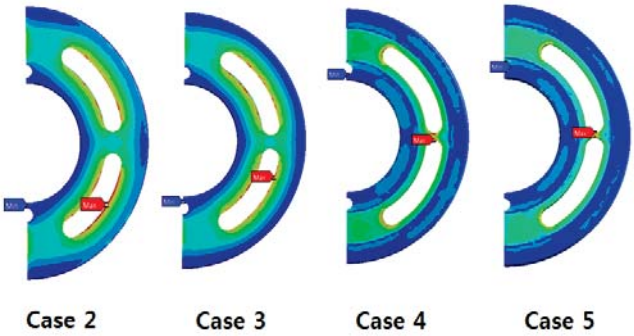


Fig. 7 Result of structure analysis for Case 2 to 5

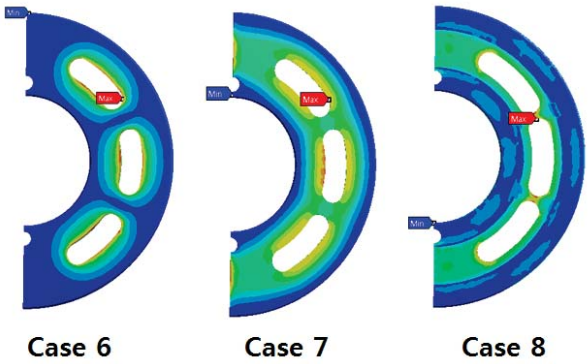


Fig. 8 Result of structure analysis for Case 6 to 8

3 Application of SimulationX®

To analyse the effects of pulsation of the '*b*' factor, which is a design factor for valve plate, first, it is necessary to make a kinematic analysis of the swash plate type piston pump. Figure 9 represents a basic schematic for the swash plate type piston pump in order to design a single piston pump model using SimulationX®, a hydraulic analysis program. When using fixed angle frame method [9], the result can be obtained as shown in Eq. (1). In [6], Jang *et al.* have already conducted

the kinematic analysis on which the single piston pump modelling is presented in Fig. 10.

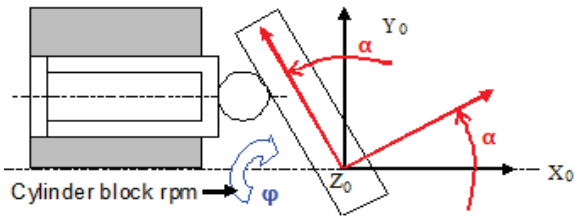


Fig. 9 Coordinate system in piston and swash plate [3]

$$XO_{P1} = -Rsin(\alpha)\cos(\varphi) \quad (1)$$

In Eq. (1), R represents a pitch circle diameter, while α and φ represent a swash plate angle and a rotation angle of cylinder block, respectively.

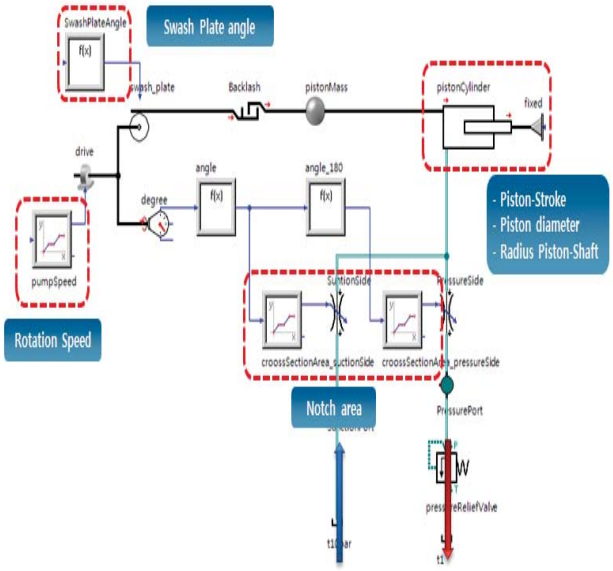


Fig. 10 Single piston pump modelling using SimulationX®

The parameters for analysing the effects of pulsation of '*b*' factor in the valve plate are shown in Table 3. In order to find out the effects of pulsation depending on the number of '*b*', this paper has conducted an analysis of three cases such as Case1, Case 2, and Case 6 in Table 1. In addition, to confirm the effects of '*b*' factor only, this paper has made an analysis of the opening areas (whose graphs are shown in Fig. 11) for the valve plate without notch. The results of SimulationX®, are as shown in Figs. 12 to 14. It can be noticed that there is no difference in pulsation between Case 1 and Case 2. But, in Case 6 where three opening holes are constructed, it is found from Fig. 12 to 13 that pressure pulsation was reduced, compared to both Case 1 and Case 2. Figure 14 shows that Case 6 has less pressure fall than both Case1 and Case 2. Therefore, in designing a valve plate, '*b*' factor is found to be a factor not only to maintain kinematic stability but also to reduce pulsation especially for 3 opening holes compared to 1 or 2 opening holes.

Table 3 Parameters of single piston modelling

Variable	Value
Swash plate angle (deg), α	15
Pump speed (RPM)	2000
Piston diameter (mm), *d*	16.5
Stroke (mm), *L*	2Rtanα
The number of piston (ea)	9
Pitch circle diameter (mm), *R*	37
Piston mass (gr)	100

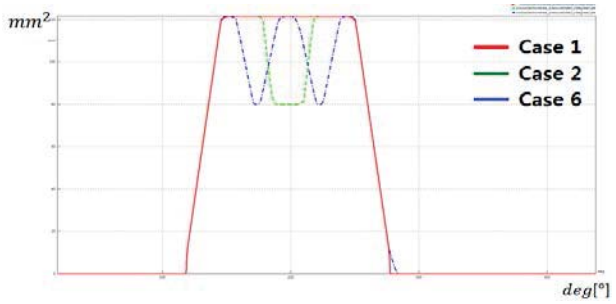


Fig. 11 The opening area according to the factor '*b*'

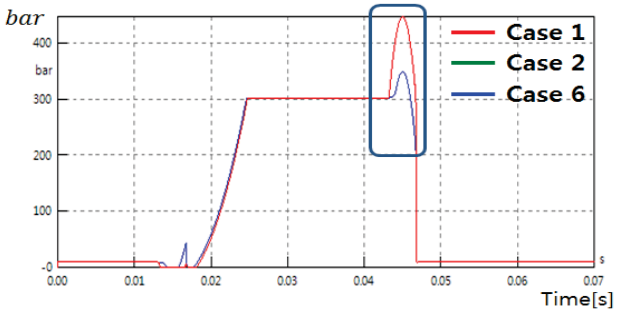


Fig. 12 Pressure inside the cylinder to the factor 'b'

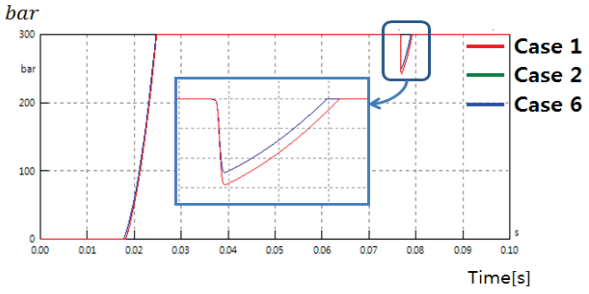


Fig. 13 Pressure pulsation

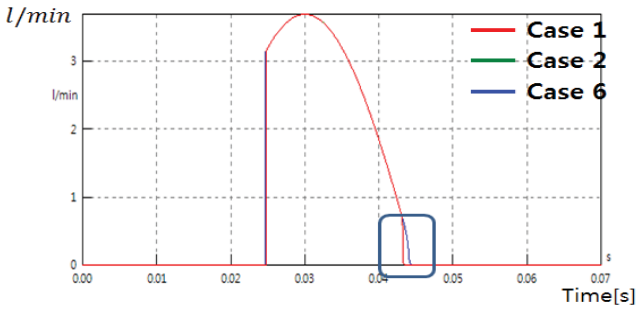


Fig. 14 Pressure fall

4 Simulation Analysis according to Notch Type

Earlier, this paper has performed the SimulationX®-based analysis of '*b*' factor, *i.e.*, the design factor for valve plate in a single piston pump, to simulate the effects of pulsation on the number of '*b*'. As a result, it has been found that pulsation can be more reduced more for three opening holes at the suction and discharge port, rather than one or two opening holes. In this section, notch type will be considered as a factor to reduce pulsation which can be caused by the rapid changes in opening area in both the suction and the discharge. It is well known that pulsation can be reduced through smooth increase and decrease in pressure. Jang *et al.* has conducted a notch analysis of circular type and V type regarding pressure pulsation. In this paper, we will investigate pulsation in 2-step V type and 2-step U type, as well as 1-step V type which has been already shown to have good pulsation reduction effect. Figure 15 represents the notch shape in 1-step V type, 2-step V type, and 2-step U type.

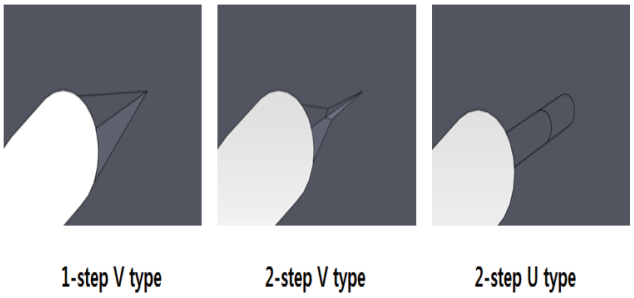


Fig. 15 Notch shapes (1-step V type *vs.* 2-step V type *vs.* 2-step U type)

Figure 16 shows the opening area of Case 6 according to three notch types shown in Fig. 15. Figure 17 is an enlarged figure of the notch part shown in Fig.16. Analysis conditions are the same as Table 3. The results of SimulationX® for the single piston pump model (see Fig. 10) are as follows. Figure 18 is a graph showing the pressure (inside the piston cylinder) depending on notch types, and Figure 19 is a graph showing the pulsation for outlet pressure. Figure 20 is an enlarged figure of the part where pressure pulsation has occurred in Fig.19. Pulsation has occurred significantly in 2-step U type notch compared to 1-step V type notch. Especially pulsation is slightly reduced in 2-step V type notch compared to 1-step V type notch.

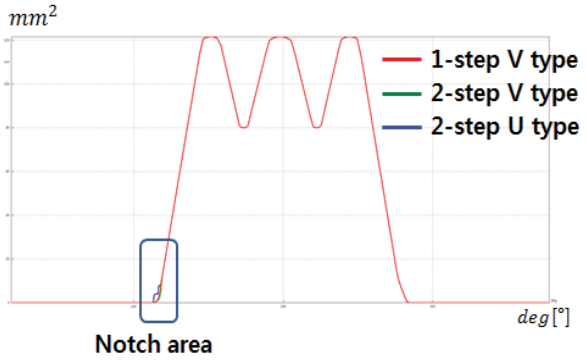


Fig. 16 Opening area of Case 6 according to notch types

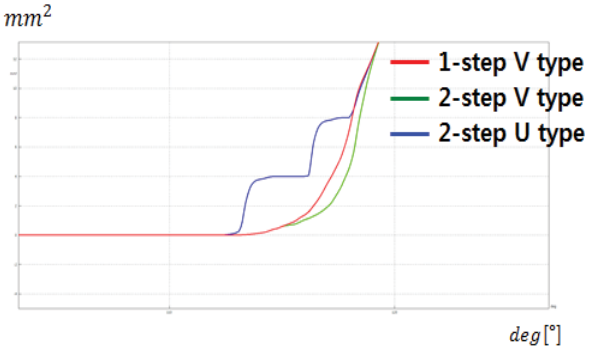


Fig. 17 Notch area (1-step V type *vs.* 2-step V type *vs* 2-step U type)

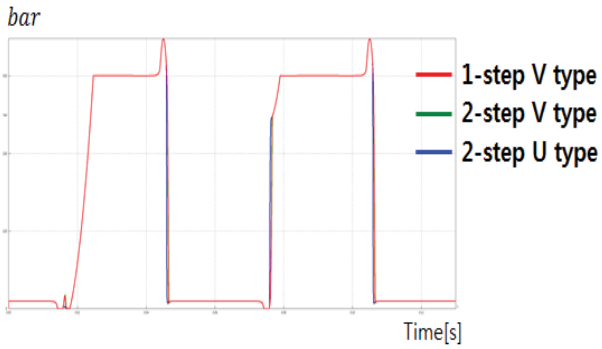


Fig. 18 Pressure inside the cylinder according to notch types

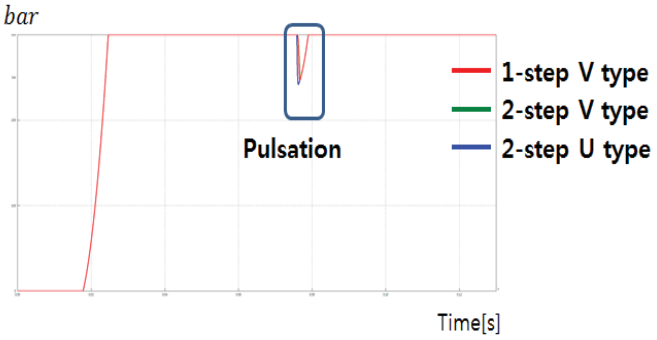


Fig. 19 Outlet pressure

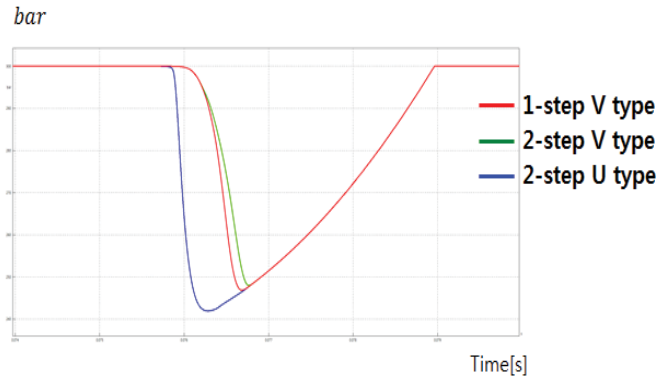


Fig. 20 Pulsation at outlet pressure

Table 4 illustrates the summarized reduction level of pulsation shown in Fig. 20. When designing the shape of notch of the valve plate in a swash plate type variable piston pump, 1-step V type notch is better for reducing pulsation rather than 2-step U type. And it is more efficient for reducing pulsation if designing 2-step type notch rather than 1-step V type.

Table 4. Comparison of pressure pulsation according to three notch types

	Value			Improvement Rate
	2-step U type	1-step V type	2-step V type	
Pressure [bar]	58.1	53.2	52	***10.5%*** *(2-step U and 2-step V)* , ***2.3%*** *(1-step V and 2-step V)* ***(decrease)***

5 Conclusions

This paper has investigated the factors for designing opening area of the valve plate in a swash plate type variable piston pump which is mainly used for a main pump especially for an excavator. First, we have conducted a structural analysis to confirm the kinematic stability on the distance and the number of '*b*' factor, as a design factor for valve plate. In addition, this paper has analysed the effects of '*b*' factor on pulsation and found making three opening holes better than making 1 to 2 opening holes for suction and discharge port, respectively. And after comparative analysis of 1-step V type notch, 2-step U type notch, and 2-step V type notch, it is found that 2-step V type notch is efficient for reducing the pressure pulsation compared to the other two notches. Finally, this paper has presented '*b*' factor as a design factor that has affected on the pulsation in the swash plate type piston pump and then maintained a kinematic stability going beyond Jang *et al.*'s study [6]. Finally we have analysed the pulsation effects of 1-step V type, 2-step V type and 2-step U type, which are mainly used for valve plates at present, using SimulationX®, hydraulic analysis program. It can be concluded that 1-step V type notch is better for reducing pulsation rather than 2-step U type. In addition, it is more efficient for reducing pulsation if designing 2-step type notch rather than 1-step V type.

6 References

[1] I. H. Baek, J. H. Kim, I. S. Cho, J. Y. Jung, S. H. Oh, "Stress Analysis of Valve Plate of Swash Plate Type Hydraulic Piston Pump", Journal of The Korea Society of Tribologists and Lubrication Engineers KSTLE, 2005

[2] Y. H. Yoon, J. S. Jang, and Y. B. Lee, "An analysis of Dynamic Characteristics for Variable Swash Plate Type Axial Piston Pump", Journal of The Korea Society for Fluid Power and Construction Equipments KFSC, 2012

[3] D. H. Jang, S. K. Lee, J .H. Kwon and S. H. Park, "A Study on Pressure, Flow Fluctuation and noise in the Cylinder of Swash Plate Type Piston Pump", Journal of The Korea Society for Fluid Power and Construction Equipments KSFC, 2012

[4] J. G. Kim, "Performance Characteristics with Valve Plate Shapes in Swash Plate Type Oil Piston Pumps", Ph.D thesis of Chonbuk National University, 2003

[5] H. W. Choi, J. K. Kim and J. Y. Jung, "Pressure Characteristics at the Land of Valve Plate in the Oil Hydraulic Axial Piston Pump", Journal of The Korea Society of Tribologists and Lubrication Engineers KSTLE, 2000

[6] J. H. Jang, W. J. Chung, D. S. Lee, Y. H. Yoon, "Application of SimulationX based Simulation Technique to Notch Shape Optimization for a Variable Swash Plate Type Piston Pump, Journal of The 2013 International Conference on Scientific Computing, 2013

[7] Y. Y. Lee, "Hydraulic Engineering", 2012

[8] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, "ROBOTICS, control, Sensing, Vision, and Intelligence"

[9] D. H. Jang, S. G. Lee, J. H. Kwon, S. H. Park, "A Study on Pressure, Flow Fluctuation and Noise in the Cylinder of Swash Plate Type Axial Piston Pump"

[10] Y. H. Yoon, and J. S. Jang, "SimulationX, Multi-domain Simulation and Modeling tool for the Design, Analysis, and Optimization of Complex systems", Journal of The Korea Society for Fluid Power and Construction Equipments KSFC, 2012

[11] S. L. Choi, "A basic study on simulation of flow ripple in piston pump", 2011

[12] D. K. Noh, and J. S. Jang, "Shape Design Sensitivity analysis of the Valves installed in the Hydraulic Driving Motor" Journal of The Korea Society for Fluid Power and Construction Equipments KSFC, 2012

[13] SimulationX user manual and library manual, ITI GmbH, 2011

Analysis of Microstrip Line using Markov Chain Monte Carlo

Adebowale E. Shadare, Matthew N.O. Sadiku and Sarhan M. Musa
College of Engineering
Prairie View A&M University, Prairie View, TX 77446
Email: shadareadebowale@yahoo.com, sadiku@ieee.org, smmusa@pvamu.edu

***Abstract*— Advances in the design of microwave solid-state integrated circuits in recent years have attracted pronounced research interests in microstrip lines technology. The sublime usefulness of microstrip line has made a modest demand for a more holistic and novel approach that can demystify intricacies associated with analytical methods of its analysis. In this paper, we considered the deployment of Markov chain Monte Carlo method to a high grade microstrip line consisting of a track of alumina as an insulating substrate. Also, we obtained the values of capacitances with and without the substrate in place, characteristic impedance, and potential distribution along the air-dielectric interface as well as that along the line of symmetry. We then compared our results with those obtainable from analytical and finite element methods using the same parameters and we found them to be close.**

Keywords: **Shielded microstrip lines, capacitance per unit length, Markov chain, characteristic impedance, Modeling and simulation.**

I. INTRODUCTION

Advances in the design of microwave solid-state integrated circuits in recent years have attracted pronounced research interests in microstrip lines technology. Microstrip lines have received wide endorsement as suitable transmission lines for the design of microwave integrated circuits for high frequency applications [1]. Their sublime usefulness in circuit components such as filters, couplers, resonators and antennas and their attendant importance in sensitive communication systems such as high-speed digital applications, power radars and satellite communications have made a modest demand for a more holistic approach in their analysis. Owing to the intricacies involved in analytical methods for calculating capacitance of shielded microstrip transmission lines, there is a need to explore a simple and novel approach. There has been a rash of impressive research efforts by a plethora of authors to analyze microstrip lines for capacitance and characteristic impedance using diverse approaches. Methods such as finite element method [2], variational method [3], equivalent electrode [4], conformal method [5], spectral analysis [6], etc have all been used till date with dazzling results. Fusco et al [7] applied Markov chain for static field analysis. Thus, the drive to investigate a novel approach for the analysis of microstrip lines other than those already widely reported in the literature has motivated this research work. Because of the nature of the electromagnetic field surrounding the microstrip lines, we consider that the deployment of absorbing Markov Chains Monte Carlo method will be suitable for whole field potential calculations for the microstrip.

II. ABSORBING MARKON CHAIN

Suppose that this method is to be applied in solving Laplace's equation

$$\nabla^2 V = 0 \quad \text{in region R} \tag{1}$$

subject to the Dirichlet boundary condition

$$V = V_p \ \text{on boundary B} \tag{2}$$

The region R is divided into a mesh (as in finite difference). Equation (1) is replaced by its finite difference equivalent as follows:

$$\begin{aligned} V(x,y) = {} & p_{x+}V(x+\Delta, y) + p_{x-}V(x-\Delta, y) + p_{y+}V(x, y+\Delta) \\ & \vdots + p_{y-}V(x, y-\Delta) \end{aligned} \tag{3}$$

where $$p_{x+} = p_{x-} = p_{y+} = p_{y-} = \frac{1}{4} \tag{4}$$

We can define a Markov chain as a sequence of random variables $X^{(0)}, X^{(1)},..., X^{(n)}$, where the probability distribution of $X^{(n)}$ is determined by the probability distribution $X^{(n-1)}$ [8, 9]. A Markov process is a type of random process that is characterized by the memoryless property. It is a process evolving in time that remembers only the most recent past and whose conditional probability distributions are time invariant. Markov chains are mathematical models of this kind of process. The Markov chains of interest to us are discrete-state, discrete-time Markov chains. The transition probability P_{ij} is the probability that a random-walking particle at node i moves to node j. It is expressed by the Markov property

$$P_{ij} = P\left(x_{n+1} = j \mid i = x_0, x_1, ..., x_n\right) = P\left(x_{n+1} = j \mid i = x_n\right), \quad i, j \in X, n = 0, 1, 2, ... \tag{5}$$

The Markov chain is characterized by its transition probability **P,** defined by

$$\mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \cdots \\ P_{10} & P_{11} & P_{12} & \cdots \\ P_{20} & P_{21} & P_{22} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \tag{6}$$

P is a stochastic matrix, meaning that the sum of the elements in each row is unity, that is,

$$\sum_{j \in X} P_{ij} = 1, \quad i \in X \tag{7}$$

In our case, the Markov chain is the random walk, and the states are the grid nodes. If we assume that there are n_f free (or non-

absorbing) nodes and n_p fixed (absorbing) nodes, the size of the transition matrix **P** is n, where

$$n = n_f + n_p \tag{8}$$

If the absorbing nodes are numbered first and the non-absorbing states are numbered last, the $n \times n$ transition matrix becomes

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{9}$$

where the $n_f \times n_p$ matrix **R** represents the probabilities of moving from non-absorbing nodes to absorbing ones; the $n_f \times n_p$ matrix **Q** represents the probabilities of moving from one non-absorbing node to another; **I** is the identity matrix representing transitions between the absorbing nodes $(P_{ii} = 1 \; and \; P_{ij} = 0)$; and **0** is the null matrix showing that there are no transitions from absorbing to non-absorbing nodes. Given that from equation (4)

$$Q_{ij} = \begin{cases} \frac{1}{4}, & if \; i \; is \; directly \; connected \; to \; j \\ 0, & if \; i = j \; or \; i \; is \; not \; directly \; connected \; to \; j \end{cases} \tag{10}$$

The same applies to R_{ij} except that j is an absorbing node. For any absorbing Markov chain, $\mathbf{I} - \mathbf{Q}$ has an inverse. This is usually referred to as the fundamental matrix

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} \tag{11}$$

where N_{ij} is the average number of times the random-walking particle starting from node i passes through node j before being absorbed. The absorption probability matrix **B** is

$$\mathbf{B} = \mathbf{N}\,\mathbf{R} \tag{12}$$

where R_{ij} is the probability that a random-walking particle originating from a non-absorbing node i will end up at the absorbing node j. **B** is an $n_f \times n_p$ matrix and is stochastic, similar to the transition probability matrix, that is

$$\sum_{j=1}^{n_p} B_{ij} = 1, \quad i = 1, 2, ..., n_f \tag{13}$$

If $\mathbf{V}_f$ and $\mathbf{V}_p$ contain potentials at the free and fixed nodes, respectively, then

$$\mathbf{V}_f = \mathbf{B}\mathbf{V}_p \tag{14}$$

III. PROBLEM FORMULATION

In this research work, absorbing Markov Chain Monte Carlo method was used to analyze microstrip lines. In order to achieve this, a high grade microstrip line consisting of a track of alumina as an insulating substrate was deployed. The microstrip line was divided into equal halves to take advantage of symmetry and only one half was analyzed. The microstrip was first analyzed without the substrate in place and the procedure repeated for the case where the substrate is in place. For the two cases, an artificial shielded height was created such that a finite domain was formed. The domain was then discretized into a number of grids. The size of the grid was carefully selected to ensure that the microstrip structure was entirely taken care of. From Figure 1, a grid of free and fixed nodes was formed as shown in Figure 4. We generated the grid using the parameters shown in Table 1. We then developed a matrix from the grid based on probability and interactions of the random walking particles within the free nodes and between the free nodes and the absorbing nodes.

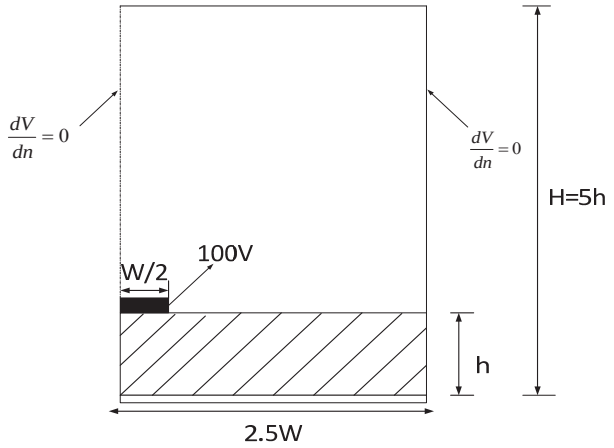


Fig. 1 Half sized microstrip structure

Table 1 shows the detailed parameters of the microstrip line that are used for our simulations with their explicit pictorial representation shown in Figure 1.

Table 1: Parameters of microstrip line

Parameter	Value
h	1.0cm
W	1.6cm
V_d	100V
H	5h
Size of dielectric	5W
Dielectric constant for alumina	9.6
Step size (Δ)	0.1 and 0.2cm

Case 1: Microstrip without the dielectric in place

From Figure 4, we obtained a matrix for **Q** that represents the probabilities of moving from one non-absorbing node to another. The transient probabilities are determined using equation (4) and on the line of symmetry, the condition $\frac{\partial V}{\partial n} = 0$ is imposed. The finite difference equivalent for line of symmetry is given by:

$$V_0 = p_{x+}V_1 + p_{y+}V_3 + p_{y-}V_4 \quad (18)$$

where $p_{x+} = \frac{1}{2}, \; p_{y+} = p_{y-} = \frac{1}{4}$ (19)

The probabilities of the particles lying on the line of symmetry are determined using equation (19).

Case 2: Microstrip with the dielectric in place

From Figure 4, we obtained a matrix for **Q** that represents the probabilities of moving from one non-absorbing node to another. In order to achieve this, a boundary condition $D_{1n} = D_{2n}$ is imposed. Consider the finite difference equivalent of the boundary condition at the interface given as:

$$V_o = p_{x+}V_1 + p_{x-}V_2 + p_{y+}V_3 + p_{y-}V_4 \quad (20)$$

where the transient probabilities are given by:

$$p_{x+} = p_{x-} = \frac{1}{4}, \; p_{y+} = \frac{\varepsilon_1}{2(\varepsilon_1 + \varepsilon_2)}, \; p_{y-} = \frac{\varepsilon_2}{2(\varepsilon_1 + \varepsilon_2)} \quad (21)$$

where in our case, ε_1 = dielectric constant of air and ε_2 =dielectric constant of alumina. So, the probabilities and interactions of the random walking particles within the free nodes are determined as in case 1 except that particles beginning at the interface and moving to any immediate nodes have their probabilities determined as shown in equation (21).

IV. EVALUATION OF CHARACTERISTIC IMPEDANCE

(a). Evaluate the values of potential at free nodes as in Equation (14) for the case without the dielectric substrate

(b). Determine the value of the resultant charge, q[10] from Equation (21) and Figure 3.

$$q = \varepsilon_o \left[\sum \varepsilon_{ri} V_i \text{ for nodes i on external rectangle GHJMP} \right] - \varepsilon_o \left[\sum \varepsilon_{ri} V_i \text{ for nodes i on internal rectangle KLN} \right] \quad (22)$$

Note: corners such as J are not counted and corners such as L are counted twice

(c). Then, evaluate the capacitance C_o without the dielectric in place $C_o = \frac{4q_o}{V_d}$, where q_o is the resultant charge when the dielectric is not in place and V_d is the potential on the conductor

(d). Repeat steps (a) and (b) (with the dielectric substrate in place) and evaluate $C = \frac{4q}{V_d}$, where q is the resultant charge with the dielectric in place and V_d is the potential on the conductor

(e). Finally, calculate $Z_o = \frac{1}{u\sqrt{CC_o}}$, where $u = 3 \times 10^8 \, ms^{-1}$

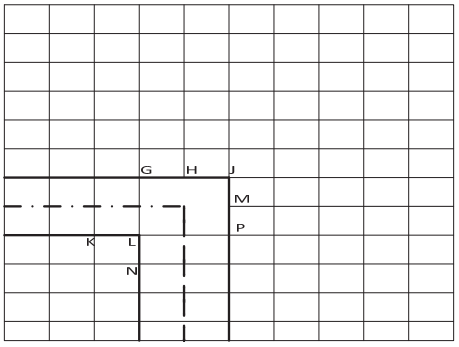


Fig. 3 Rectangular shield around the strip conductor

V. RESULTS

After we carried out our analysis, we obtained the following results. We then compare our result with those obtained using analytical method as detailed in the Table 2. In the table, some results are not shown as they are not a requirement for obtaining the characteristic impedance through the analytical method.

Table 2: Analytical result versus Simulation result

	Analytical Result	Finite element method	Markov chain Step size (0.2)	Markov chain Step size (0.1)
Capacitance without dielectric(F/m)	-	3.543×10^{-11}	4.1486×10^{-11}	3.7473×10^{-11}
Capacitance with dielectric (F/m)	-	2.2462×10^{-10}	2.4448×10^{-10}	2.4780×10^{-10}
Characteristic Impedance(ohms)	38.77	37.3653	33.0985	34.5916

Table 2 shows the analytical result compared with the simulation results. The capacitance per unit length without the dielectric substrate in place is as expected less than the capacitance per unit length when the dielectric substrate is in place. Similarly, the charge, Q without the dielectric substrate in place is obtained to be less than the case in which the dielectric substrate is in place. The characteristic impedance obtained in the simulation shows close agreement with that obtained from analytical and finite element methods.

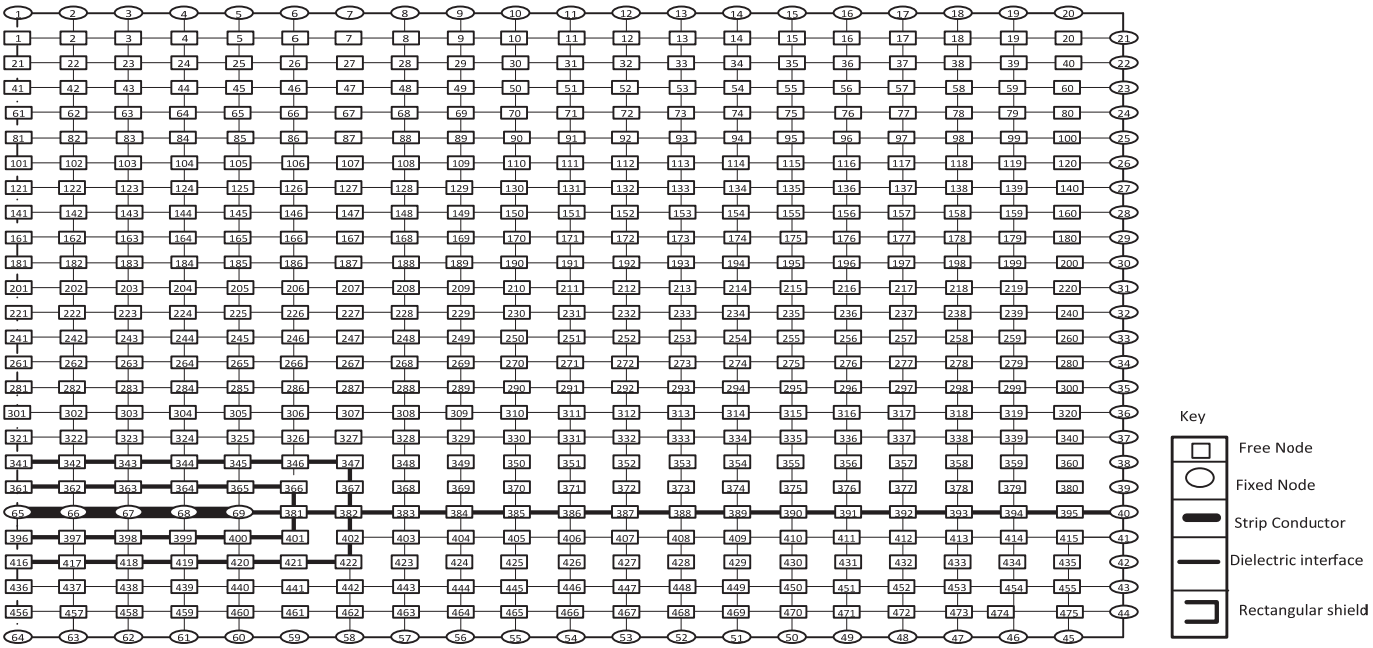


Fig. 4 Free and fixed nodes (grids) generated for microstrip

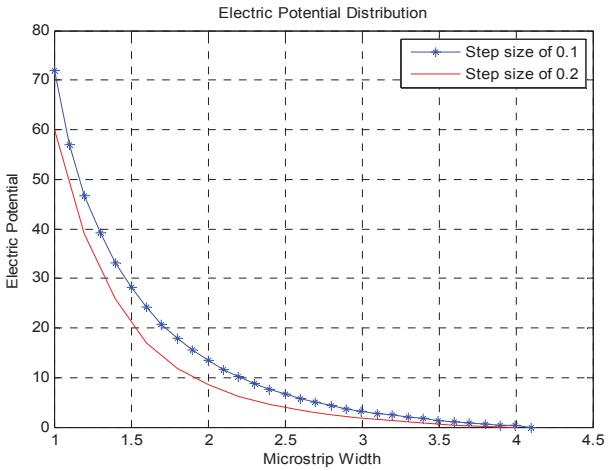


Fig. 5 Potential distribution along the air-dielectric interface for the half-sized microstrip line

The Figure 5 shows the potential distribution along the air-dielectric interface for the half-sized microstrip line. From Figure 5, it is obvious that the electric potential along the dielectric interface decreases exponentially. It is zero at the point where the interface coincides with an absorbing node.

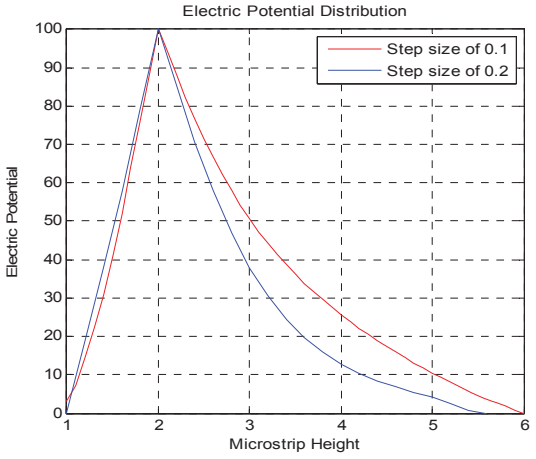


Fig. 6 Potential distribution along the line of symmetry for the half-sized microstrip line

The Figure 6 shows potential distribution along the line of symmetry for the microstrip line case where the dielectric substrate is in place and the case where the dielectric substrate is not in place. The graph is peaked at the nodes where the strip conductor exists.

VI. CONCLUDING REMARKS

This research work analyzes microstrip lines using Markov Chain Monte Carlo method. The characteristic impedance value for the parameters of the microstrip line was obtained and compared closely with that obtained from analytical and finite element methods. Plots for the potential distribution along the dielectric-air interface and that along the line of symmetry were also obtained. It is hoped that this work will ease some identified and conceivable challenges that are associated with other methods previously deployed to the analysis of microstrip lines.

REFERENCES

[1] M.A. Kolbehdari, and M.N.O. Sadiku, "Finite and Infinite Element Analysis of Coupled Cylindrical Microstrip Line In A nonhomogeneous Dielectric Media,"*IEEE Proceedings of Southeastcon'95*, pp. 269-273, March 1995.

[2] S.M. Musa and M.N.O. Sadiku, "Modeling and Simulation of Shielded Microstrip Lines," *The Technology Interface*, Fall 2007.

[3] E. Yamashita and R. Mittra, "Variational Method for the Analysis of Microstrip Lines," *IEEE Transactions on Microwave Theory and Techniques*, vol.MTT-16, no.4, pp. 251 -256, April 1968.

[4] N.B. Raicevic and S.S. Ilic, "Equivalent Electrode Method Application on Anisotropic Microstrip Lines Calculation," *Int. Conf. on Electromagnetics Advanced Applications*, pp. 998-1001, February 2007.

[5] J. Scvacina, "Analysis of Multilayer Microstrip Lines by a Conformal Mapping Method," *IEEE Transactions on Microwave Theory and Techniques,* vol. 40, no.4, pp. 769-772, April 1992.

[6] F. Medina and M. Horno, "Spectral and Variational Analysis of Generalized Cylindrical and Elliptical Strip and Microstrip lInes," *IEEE Transactions on Microwave Theory and Techniques,* vol. 38, no.9, pp.1287-1293, Sept. 1990.

[7] V.F. Fusco and P.A. Linden, "A Markov Chain Approach For Static Field Analysis," *Microwave and Optical Technology Letters*, vol. 1, no. 6, pp. 216 -220, 1988.

[8] M.N.O. Sadiku, *Monte Carlo Methods for Electromagnetics*, Boca Raton, Fl: CRC Press, Taylor & Francis Group, pp. 162-174, 2009.

[9] M.N.O. Sadiku and R. C. Garcia, "Whole Field Computation Using Montel Carlo Method," *Int. Jour of Numerical Modeling: Electronic Networks, Devices and Fields,* vol. 10, pp. 303-312, 1997.

[10] M.N.O. Sadiku, *Elements of Electromagnetics*, New York: Oxford University Press, 6th ed., pp. 535-613, 2015.

Rover Trajectory Planning via Simulation Using Incremented Particle Swarm Optimization

Ayman Hamdy Kassem
Aerospace Engineering Department, Cairo University

Abstract – *This paper presents an off-line optimal trajectory planning for differential-drive rover through simulation of the dynamic model. The paper starts with the model dynamics of an actual rover built in our space science and technology lab (SSTLab) and controlled by simple PD controllers. Next, the proposed optimization technique used is presented which is called Incremented Particle-Swarm Optimization (IPSO) where the number of variable increases incrementally if the goal is not satisfied to minimize the time and CPU usage running the cost function. Different trajectories and cost functions were tested with obstacles and without it. The results show that the trajectory can be optimized efficiently using IPSO and a simple cost function based on total time and distance to final destination.*

Keywords: *Modeling, optimization, trajectory planning.*

1 Introduction

Differential-drive rover is a two wheels configuration where the wheels are in-line with each other. The wheels are independently powered and controlled so the desired motion will depend on how these wheels are controlled. It is a simple robot which gained a lot of popularity for its many uses applications such outer space exploration and deep sea excavation, etc. It also gained a lot of popularity in robotic competitions such as NASA rover challenge and Robotic Rover Competition by The Mars Society. It also gained popularity in academic research [1-5].

For so many applications on land and under sea, there is always a need to have a planned trajectory optimized for time, distance, or obstacle-avoidance to save resources in a mission while improving its value. Off-line trajectory optimization will work perfectly for well-known terrains and will improve sensors readings for real-time trajectory planning with partially-known terrains.

In this paper, the model of an experimental differential-drive rover, built in our lab, will be presented and will be used for simulating the rover for trajectory optimization. The resulting system takes voltage as a motor input, hence the rotational speed of the wheels may vary and so will the robot's position whose central point is represented in a Cartesian plane (XY).

A new proposed optimization technique is introduced. It is based on particle swarm optimization starting with small number of variables and then incrementing of the number of variables to improve computational cost and reach the optimum solution faster.

2 Rover Model

The differential-drive rover uses the two different controllers to control the speed of each wheel. If the speeds of both wheels are equal the rover will go straight. If they are unequal, the rover will rotate. Figure 1 shows counter clockwise rotation mechanism.

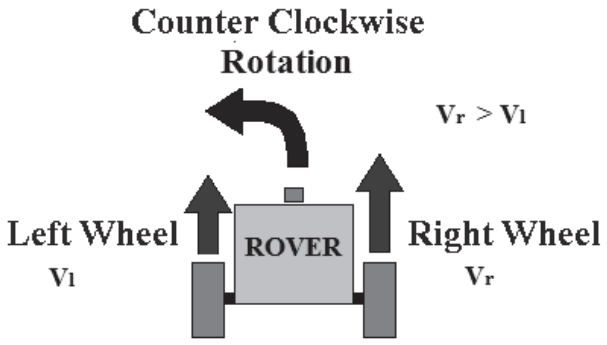


Figure 1: Rotational mechanism for differential-drive rover

The dynamics equations for a single wheel motor are given by:

$$u_e(t) = R\ i\ (t) + L\ \frac{di\ (t)}{dt} + e_b(t)$$

$$e_b(t) = k_b \omega_m(t) \qquad (1)$$

$$T_m(t) = k_i i\ (t)$$

$$T_m = J_m \frac{d^2\theta_m}{dt^2} + B_m \frac{d\theta_m}{dt}$$

Where:
R: motor resistance.
L: motor inductance.
e_b : back emf.
T_m: motor torque.
i: electric current.
k_b, k_i: motor constants.
θ_m: motor rotational angle.
J_m: motor inertia.
B_m: motor damping coefficient.
ω_m: motor rotational speed.

Figure 2 shows the electromechanical circuit for the motor

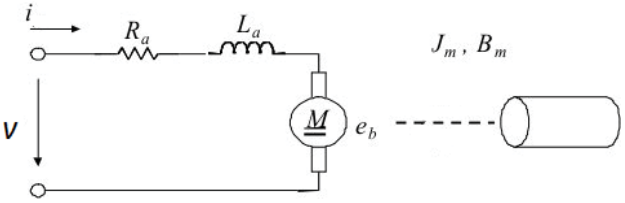


Figure 2: DC motor circuit

The rover uses two identical motorized wheels and its average translational speed V and rotational speed ω are given by the following simple equations:

$$V = \frac{V_l + V_r}{2} \quad , \quad \omega = \frac{V_r - V_l}{d} \tag{2}$$

where d is the distance between the wheels.

Manipulating equations 1,2 and substituting numerical values, the rover model can be put in the following state space form [6]

$$\dot{z} = Az + Bu \quad , \quad y = Cz \tag{3}$$

Where $z = [v, \omega, i_r, i_l, \psi]$, u = [$v_r$ v_l]

V: the translational velocity of the rover.
ω: heading angular rate.
i_r : current in right motor.
i_l : current in left motor.
ψ: heading angle.
v_r: volt to right motor.
v_l: volt to left motor.

$$A = \begin{bmatrix} -0.45 & -0.007 & 0.51 & 0 & 0 \\ 0 & -1.58 & -5.31 & 0 & 0 \\ -11.74 & 4.11 & -12.97 & 0 & 0 \\ -11.74 & -4.11 & 0 & -12.97 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 3.432 & 0 \\ 0 & 3.432 \\ 0 & 0 \end{bmatrix} \tag{4}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The model is controlled by two PD controllers for the speed and heading rate channels. The actual rover is shown in figure 3

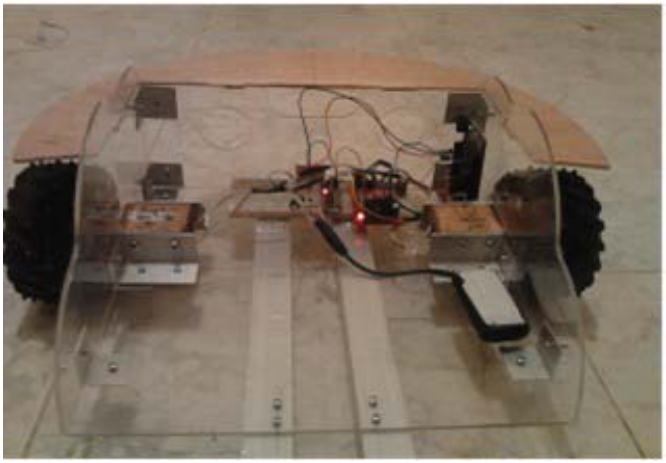


Figure 3: Test rover at SSTLab

Adding integrators to the angular and translational rates and using transformation (sine and Cosine), we can have the position co-ordinates or the rover (X, Y) as shown in figure 4.

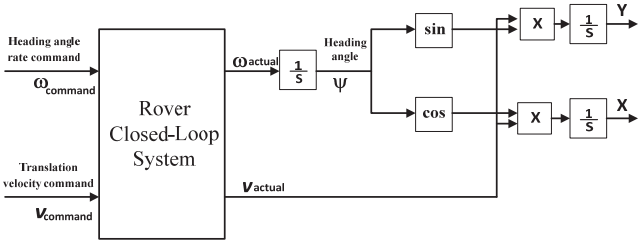


Figure 4: Rover Simulator block diagram

Setting the heading angle rate command and translation velocity command, the simulator will calculate the position of the rover. By fixing the rover translation velocity command, the heading angle rate command is used to control the position of the rover and it will be used as the optimization parameters as shown in the following section.

3 Optimization

The optimization process is shown in figure 5. It starts with specifying the destination for the rover and any constraints (obstacles) and the rover model. The PSO module will pass randomly generated input to the rover model simulator and then the error budget is calculated and passed to the cost function to the PSO module.

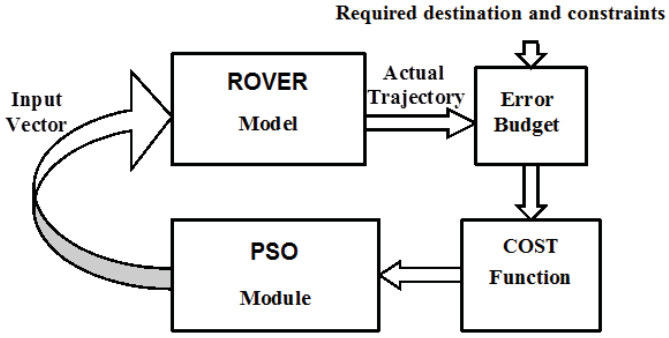


Figure 5: Trajectory optimization process diagram

PSO is an algorithm proposed by James Kennedy and R. C. Eberhart in 1995 [7, 8, 9] who were motivated by the social behavior of organisms such as bird flocking and fish schooling.

PSO shares many similarities with evolutionary computation techniques such as GAs. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike the GA, PSO has no evolution operators such as crossover and mutation.

PSO mimics the behaviors of bird flocking in algorithmic form. Suppose the following scenario: a flock of birds is randomly searching for food in an area. There is only one piece of food in the area being searched. All of the birds do not know where the food is, but they know which bird is the closest to food in each iteration. The effective strategy is to follow the bird which is nearest to the food.

```
For each particle
   Initialize particle
End
Do
   For each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pBest) in
history
set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the
gBest
For each particle
Calculate particle velocity according to Eq. 2
Update particle position according to Eq. 3
End
While maximum iterations or minimum error criteria is not attained
```

Figure 6: The pseudo-code for PSO

PSO learned from this scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it a "particle." All of the particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flight of the particles. The particles fly through the problem space by following the current optimum particles.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In each iteration, particles are updated by following two "best" values. The first one is the best solution (fitness) that has been achieved so far. This value is called "particle best (pbest)." Another "best" value that is tracked by the PSO program is the best value, obtained so far by any particle in the population. This best value is a global best and called "gbest." When a particle takes only a part of the population as its topological neighbors, the best value is a local best and is called "local best (lbest)", and it is used instead of (gbest) in Eq. (2) for a large population to save execution time. After finding the two best values, the particle updates its velocity and position using Eqs. (5) and (6) assuming unit time steps for each update.

$$v = v + c1 * rand * (pbest - present) + c2 * rand * (gbest - present) \quad (5)$$

$$present = persent + v \quad (6)$$

Here v is the particle velocity and *present* is the current particle (solution). *pbest* and *gbest* are defined as stated before. *rand* is a random number between (0,1). *c1* and *c2* are learning factors,[7] and usually $c1 = c2 = 2$.

PSO is an extremely simple algorithm that seems to be effective for optimizing a wide range of functions.[7,9]. It also uses the concept of *fitness*, as do all evolutionary computation paradigms. The pseudo code is shown in figure 4.

The proposed incremented particle swarm optimization IPSO is a modified version of PSO. It is best suited when the optimization parameters are functions in time like direction commands for robots. These types of optimization problems were solved numerically by assuming higher order polynomials or splines etc. and trying to find the coefficients of these polynomials or splines that satisfy the optimization problem. In IPSO, few numbers of values (control points) are assumed for design variable (in this case it is the heading angular speed). These control points, as it control the shape of a function, are equally distributed over time and the design variable is assumed piecewise linear in between using direct linear interpolation as seen in figure 7-a. In figure 7-a, four values define the shape of the design variable over time F(t). It is the PSO task to find these values which satisfy the minimization process. If PSO failed, the number of control points is increased which allow more freedom for the shape of the design variable. Figure 7-b shows a function F(t) after incrimination with three intermediate control points (seven control points total). It can be seen that F(t) in figure 7-b has the ability to represent more complicated functions with seven control points than F(t) in figure 7-a with only four control points as shown with the dotted lines. The same procedure of increasing the number of variables is applied repeatedly. Each time the search-space representing the variables is expanded (i.e. more control points are added) and the added values are linear interpolation of the adjacent old ones. In another words, the procedure starts with a coarse grid of control points and

uses the solutions of this coarse grid as an initial guess (population) for a finer grid and continues this process until a stop criterion occurs. The idea of linear interpolation, besides simplifying the calculations, guaranties that the initial population of the expanded search-space has the same best fitness of the search-space of smaller dimensions. This guarantees convergence to better solution faster than classical PSO. It also helps in selecting the correct number of optimization variables for the trajectory as you start with small number and increase till you reach the optimum.

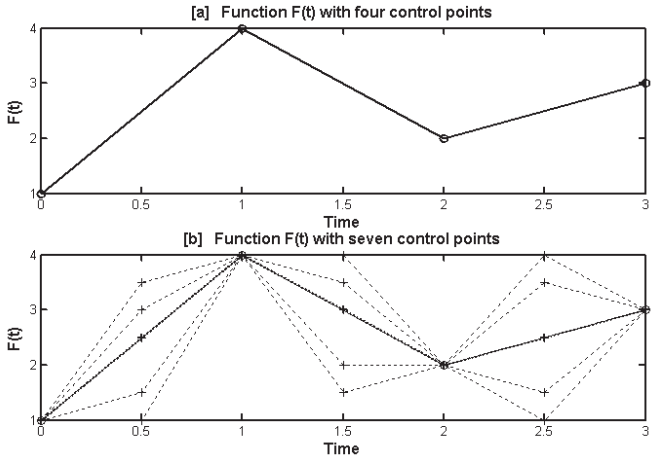


Figure 7: Piecewise linear function approximation using control points

The cost function has three parts one for the trajectory which is the difference between the required final position and the actual final position for the rover and the second part is time which is used as a measure of power consumption. The third part is a penalty for constraints usually obstacles. The cost function is shown in table 1

Table 1 Cost Function

1- Cost function for trajectory optimization: $F_1 = \left\| P_{final} - P_{required} \right\|$ where P_{final} represents rover actual final position. $P_{required}$ represents the required final position
2- Cost function for power minimization $F_2 = \int T dt$ where T is the time in seconds.
3- Penalty for constraints F_3 defined based on the constraint and will be shown in the test cases
3- Cost fitness function $F = \alpha_1 F_1 + \alpha_2 F_2 + \alpha_3 F_3$ where α_1, α_2, α_3 are weighting constants to be selected by user.

Table 2 IPSO procedure

1- Start with n control points for each variable.
2- Run the PSO procedure (figure 4) and call the simulator to calculate the cost function as in table 1.
3- Repeat PSO procedure for m step. If satisfied go to 7
4- Increment the number of control points for input variable through linear interpolation.
5- Adjust problem size accordingly.
6- Go to step (2) until stop criterion is reached.
7- Stop.

The procedure for incrementing the variables is simple. The number of variables starts with two variables and adding additional one in between it became three variables then five, nine, seventeen, and so on as shown in table 3.

Table 3 Variables distribution over time span

Two variables	* *
Three variables	* * *
Five variables	* * * * *
Nine variables	* * * * * * * * *
	←Time----------Span→

4 Test Cases

Three test cases will be shown in this section, one with no obstacles and the second one with one obstacle and the third with two obstacles.

4.1 No obstacle trajectory

In this case, the rover is required to move from point (0, 0) to point (10, 10). The speed is fixed to 1 m/s and the optimization problem is set for heading angular speed. The problem starts with two variables and the minimum is satisfied with 17. It is clear that optimal trajectory is to make a single turn to an angle 45 deg. then straight to the final destination. The following figures show the trajectory and the command for heading angular speed. The heading angle rate command has 1.57 in the first element and then zeros. So we have this triangle shape rate command which when integrated, it will give the required 45 deg. turn. This test case shows that the optimization technique is working correctly.

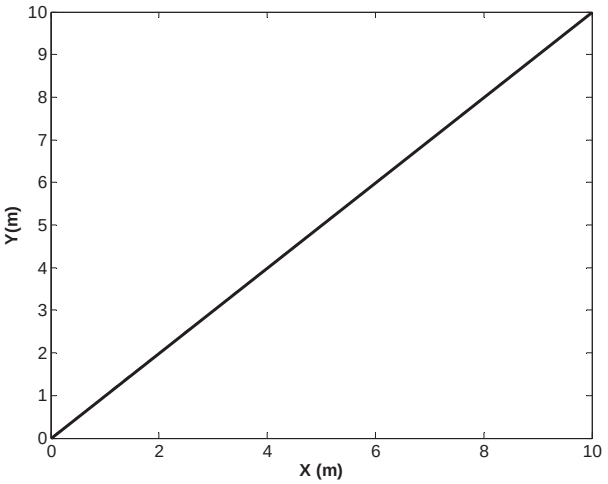


Figure 8: optimum trajectory

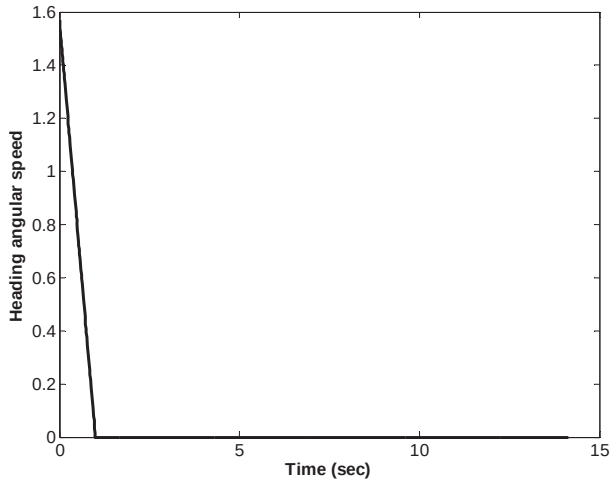


Figure 9: Heading angle rate command

4.2 Circular obstacle trajectory

In this case, the rover is required to move from point (0, 0) to point (10, 10) with a circular obstacle with radius 3m in its way. The translational speed is fixed to 1 m/s and the optimization problem is set for heading angular speed. The problem starts with two variable and incremented to three to reach the goal. The circular constraint is defined as:

If sqrt{ $(x-4)^2 + (y-4)^2$} < 9 then set F_3 to a non-zero value in the cost function ($F_3 = 50$ in this case)

The following figures show the trajectory and the command for heading angular speed.

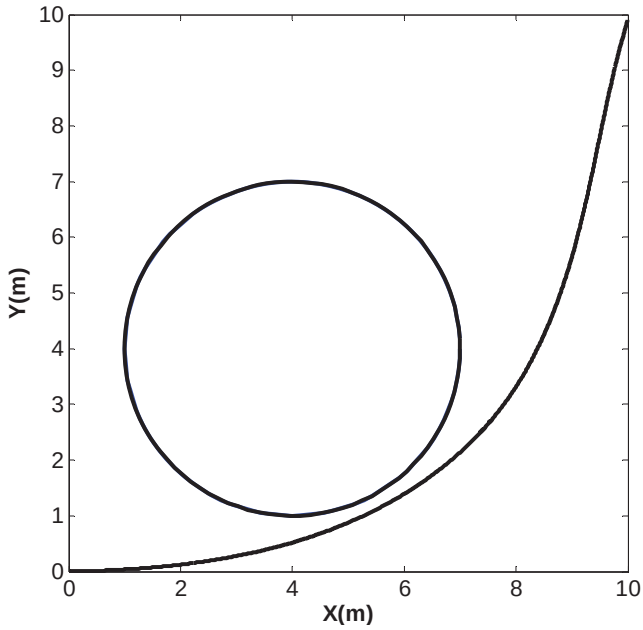


Figure 10: optimum trajectory

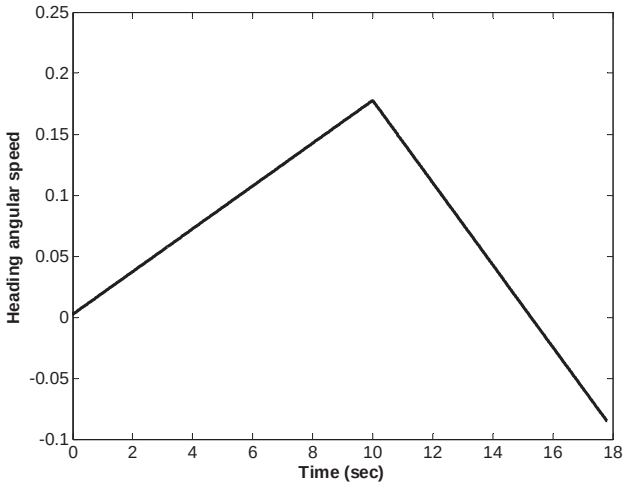


Figure 11: Heading angle rate command

4.3 Two Circular obstacles trajectory

In this case, the rover is required to move from point (0, 0) to point (10, 10) with two circular obstacles with radii 3m and 1 m respectively, in its way. The translational speed is fixed to 1m/s and the optimization problem is set for heading angular speed. The problem starts with two variable and incremented to five to reach the goal. The circular constraints are defined as:

If sqrt{ $(x-2)^2 + (y-1)^2$} < 1 then set F_{31} to a non-zero value in the cost function ($F_{31} = 50$ in this case)

If sqrt{ $(x-7)^2 + (y-7)^2$} < 4 then set F_{32} to a non-zero value in the cost function ($F_{32} = 50$ in this case)

Where $F_3 = F_{31} + F_{32}$
The following figures show the trajectory and the command for heading angular speed.

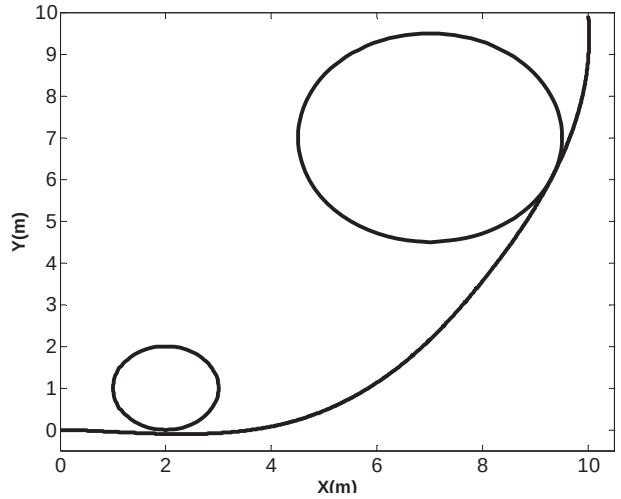


Figure 12: optimum trajectory

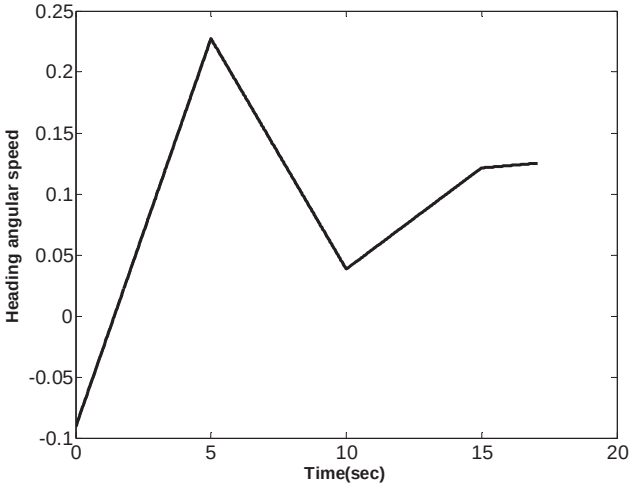


Figure 13: Heading angle rate command

4.4 Comparison between PSO and IPSO convergence

Hundreds of runs were carried out to find the minimum number of particles and number of iterations combination which guarantees a near optimal solution for a maximum of 17 optimization variables. It was found that a number of particles of 100 with 200 iterations work reasonably well as upper limit for the three tested trajectories.

To make the comparison between the two methods, the following procedure is followed:

1- Run IPSO starting with 2 variables then 3, 5, 9, and 17 as shown in table 3 with 20 iterations at each stage so the total number of iterations will sum up to 100 and record the convergence curve.

2- Run PSO with 17 variables 100 iterations and record the convergence curve.

3- Run steps 1 and 2 100 times for each case and plot the average convergence curve.

The incremented case showed better convergence performance in reaching the optimal value. Figures 14, 15, and 16 show average convergence-curves for the two optimization methods for the three test cases.

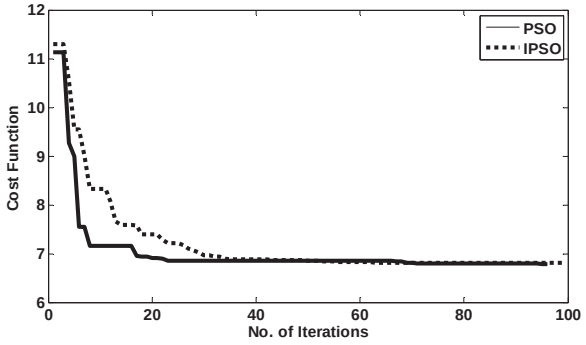


Figure 14: A comparison between PSO and IPSO convergence for case 1

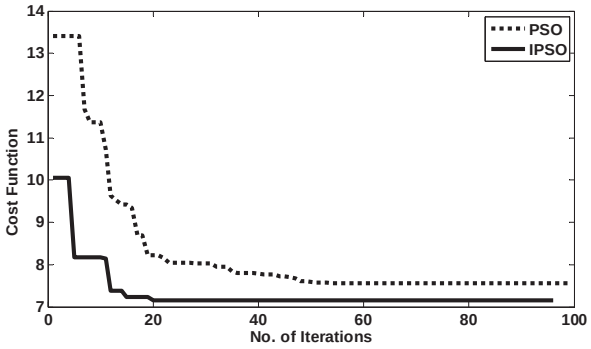


Figure 15: A comparison between PSO and IPSO convergence for case 2

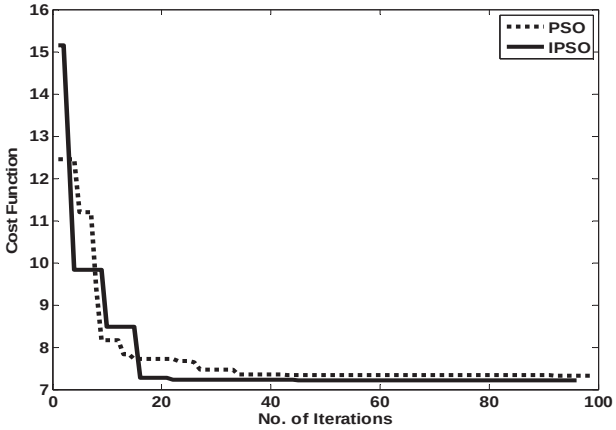


Figure 16: A comparison between PSO and IPSO convergence for case 3

5 Conclusions

A new method for off-line optimal trajectory planning for differential-drive rover through simulation is proposed. The method combines the global characteristics of particle swarm optimization with better convergence characteristics by incrementing the number of variables. It also uses a special cost function based on time and distance to final destination. The method is tested on two-dimensional rover trajectories and the optimized solution showed fast convergence and good obstacle avoidance with satisfaction of the imposed constraints.

6 References

[1] Mohd Azlan Shah Abd Rahim and Illani Mohd Nawi, "Path Planning Automated Guided Robot", Proceedings of the World Congress on Engineering and Computer Science WCECS, San Francisco, USA, 665-670, 2008.

[2] Graciano, J., and E. Chester. "Autonomous rover path planning and reconfiguration", 11th Symposium on Advanced Space Technologies in Robotics and Automation ESTEC, April 2011.

[3] Elshamli, A.; Abdullah, H.A.; Areibi, S. "Genetic algorithm for dynamic path planning", Canadian Conference on Electrical and Computer Engineering, 677 – 680, 2004.

[4] Zaheer, S.; Jayaraju, M.; Gulrez, T., "A trajectory learner for sonar based LEGO NXT differential drive robot", International Electrical Engineering Congress (iEECON), 1 – 4, 2014.

[5] Bhattacharya, S., Murrieta-Cid, R. , Hutchinson, S., "Path planning for a differential drive robot: minimal length paths - a geometric approach", Proceedings of 2004 IEEE/ RSJ international conference on Intelligent Robots and Systems, (IROS 2004), 2793-2798, 2004.

[6] Ali, H., Abdelhady, M., Deif, T., "Modelling and Control Design of Rover Vehicle Using Classic and Adaptive Control", International Review of Aerospace Engineering (IREASE), 7(4), pp. 116-123, 2014.

[7] Kennedy J., Eberhart R., Shi Y., "Swarm Intelligence", Morgan Kaufmann Academic Press, 2001.

[8] Kennedy, J. and Eberhart, R. C., "Particle Swarm Optimization", Proc. IEEE Int'l Conf. on Neural Networks Vol. IV, pp. 1942-1948. IEEE Service Center, Piscataway, NJ, 1995.

[9] Eberhart, R. C. and Shi, Y., "Particle swarm optimization: developments, applications and resources", Proc. Congress on Evolutionary Computation 2001 IEEE Service Center, Piscataway, NJ., Seoul, Korea., 2001.

[10] Ahmed Al-Garni and Ayman H. Kassem, "On the Optimization of Aerospace Plane Ascent Trajectory", Transactions of the Japan Society for Aeronautical and Space Sciences, Vol. 50, No. 168, August, 2007.

SESSION

FINITE ELEMENT METHODS AND FINITE DIFFERENCE METHODS + CELLULAR AUTOMATA

Chair(s)

TBA

Nonlinear Vibration of Single-Walled Carbon Nanotubes with Magnetic Field Based on Stochastic FEM

Tai-Ping Chang[1] **and Quey-Jen Yeh**[2]

[1] *Department of Construction Engineering, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan*

[2] *Department of Business Administration, National Cheng-Kung University, Tainan, Taiwan*

Abstract - *In this paper, we study the statistical dynamic behaviors of nonlinear vibration of the single-walled carbon nanotubes (SWCNTs) subjected to longitudinal magnetic field by considering the effects of the geometric nonlinearity. We consider both the Young's modulus of elasticity and mass density of the SWCNTs as stochastic with respect to the position to actually characterize the random material properties of the SWCNTs. By using the Hamilton's principle, the nonlinear governing equations of the single-walled carbon nanotubes subjected to longitudinal magnetic field are derived. We utilize the stochastic finite element method along with the perturbation technique to compute the statistical response of the SWCNTs. Some statistical dynamic response of the SWCNTs such as the mean values and standard deviations of the midpoint deflections are computed and checked by Monte Carlo Simulation, in addition, the effects of the small scale coefficients, magnetic field and the elastic stiffness of matrix on the statistical dynamic response of the SWCNTs are studied and discussed.*

Keywords: Nonlinear vibration; Carbon nanotubes; Magnetic field; Nonlocal elasticity theory ; Small scale effect.

1 Introduction

Carbon nanotubes (CNTs) has attracted worldwide attention due to their potential use in the fields of chemistry, physics, nano-engineering, electrical engineering, materials science, reinforced composite structures and construction engineering [1-3]. Therefore, a lot of work has been done on the mechanical properties of nanotubes with both theoretical and experimental methods [4-6]. Both experimental and atomistic simulation studies show that when the dimensions of structures become very small, the size effect is important. Due to this fact, the size effect plays an important role on the mechanical behavior of nanostructures. The nonlocal elasticity theory, which was introduced by Eringen [7] to consider the scale effect in elasticity, was used to study lattice dispersion of elastic waves, wave propagation in composites, dislocation mechanics, fracture mechanics and surface tension fluids. According to the Eringen's nonlocal elasticity theory, the stress at a reference point is considered to be a function of the strains at all other points in the body. After the first several studies on mechanical properties of nanotubes with the nonlocal continuum theory [8-9], lots of researches have been reported on vibration [10-12] of nanotubes. Previous theoretical and experimental studies show that the mechanical behavior of nanostructures is nonlinear in nature when they are subjected to large external loads [13]. Among others, several investigations on nonlinear problems with nonlocal continuum theories have been reported [14-15]. By using AFM test on clamped–clamped nanoropes, Salvetat et al. [16] estimated the flexural Young's modulus and shear modulus and obtained the values with 50% of error. Besides, Krishnan et al. [17] presented the histogram distribution of the flexural Young's modulus by performing AFM test on 27 CNTs. The Young's modulus was measured observing free-standing vibrations at room temperature using transmission electro-microscope (TEM), with a mean value of 1.3 TPa - 0.4 TPa/+0.6 TPa. In addition, the stochastically averaged probability amplitude for the vibration modes is calculated to get the root-mean-square vibration profile along the length of the tubes in Ref. [18]. Especially in the engineering and materials science communities, uncertainty is also related to the equivalent atomistic-continuum models. Therefore, to be realistic, the Young's modulus of elasticity of carbon nanotube (CNTs) should be considered as stochastic with respect to the position to actually describe the random property of the CNTs under certain conditions. Very recently, Chang [19] adopted the stochastic finite element method to perform the nonlinear vibration analysis on fluid-loaded double-walled carbon nanotubes subjected to a moving load based on nonlocal elasticity theory. In his work, the Young's modulus of elasticity of the DWCNTs was considered as stochastic with respect to the position, however, the mass density of the DWCNTs was considered deterministic. In the present study, we investigate the stochastic dynamic behaviors of nonlinear vibration of the single-walled carbon nanotubes (SWCNTs) subjected to longitudinal magnetic field by considering the effects of the geometric nonlinearity. Not only the Young's modulus of elasticity of the SWCNTs is considered as stochastic with respect to the position, but also the mass density is considered as stochastic with respect to the position to actually characterize the random material properties of the SWCNTs. In addition, the small scale effects on the nonlinear vibration of the SWCNTs are considered by using the theory

of nonlocal elasticity. Based on the Hamilton's principle, the nonlinear governing equations of the single-walled carbon nanotubes subjected to longitudinal magnetic field are formulated. The stochastic finite element method along with the perturbation technique is utilized to study the statistical response of the SWCNTs; in particular, the Newton-Raphson iteration procedure in conjunction with Newmark scheme is utilized to solve the nonlinearity of the dynamic governing equation of the SWCNTs. The effects of the small scale coefficients, magnetic field and the viscous matrix stiffness on the statistical dynamic response of the SWCNTs are studied.

2. The governing equation of nonlinear vibration

As shown in Fig. 1, the single-walled carbon nanotubes (SWCNTs) embedded in the viscous elastic matrix with longitudinal magnetic field is modeled as a single-tube pipe which has the radius R. The thickness of the tube is h, the length is L, the Young's modulus of elasticity is $E(x)$ and the mass density of SWCNTs is $\rho(x)$. It is noticed that the Young's modulus of elasticity $E(x)$ and the mass density of SWCNTs $\rho(x)$ are assumed as stochastic with respect to the position to actually describe the random material property of the SWCNTs. The geometric shapes and sizes of the SWCNTs, the spring constant k_w, the damping coefficient c of the matrix, applied load and the longitudinal magnetic field are considered as deterministic. In addition, the boundary conditions of the SWCNTs are considered as simply-supported at both ends.

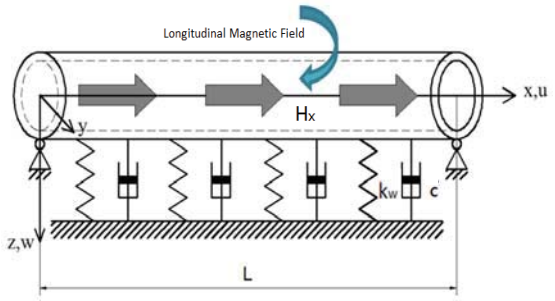


Fig.1. Single-walled carbon nanotubes embedded in viscous elastic matrix with magnetic field.

Based on the Euler–Bernoulli beam theory, the displacement field of any point of the SWCNTs is given as

$$u_x(x,z,t) = u(x,t) - z\frac{\partial w(x,t)}{\partial x} \tag{1}$$

$$u_y(x,z,t) = 0 \tag{2}$$

$$u_z(x,z,t) = w(x,t) \tag{3}$$

where u and w are the axial and the transverse displacement of any point on the neutral axis. The von-Kármán's nonlinear strain–displacement relationship based on assumptions of large transverse displacements, moderate rotations and small strains for a straight SWCNTs are given by

$$\varepsilon_{xx} = \varepsilon_{xx}^0 - z\kappa_x \tag{4}$$

$$\varepsilon_{xx}^0 = \frac{\partial u(x,t)}{\partial x} + \frac{1}{2}\left(\frac{\partial w(x,t)}{\partial x}\right)^2,\ \kappa_x = \frac{\partial^2 w(x,t)}{\partial x^2} \tag{5}$$

where ε_{xx} is the longitudinal strain, ε_{xx}^0 is the nonlinear membrane strain, κ_x is the curvature of the SWCNTs. In this study, the equations of motion are derived by using Hamilton's principle. This principle can be expressed as

$$\delta\int_0^t \left[K - (U - W)\right]dt = 0 \tag{6}$$

where K is the kinetic energy, U is the strain energy and W is the work done by the external applied forces. Based on nonlocal elasticity theory [7], the nonlocal governing equations of the SWCNTs in terms of the displacements can be obtained as follows:

$$\begin{aligned} &\mathrm{I}\frac{\partial^2(E(x)(\partial^2 w/\partial x^2))}{\partial x^2} + \left[\int_0^L \left(\frac{\partial w}{\partial x}\right)^2 \frac{\mathrm{E}(x)\mathrm{A}}{2L}dx\right]\frac{\partial^2}{\partial x^2}\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] \\ &= \rho(x)\mathrm{A}\frac{\partial^2}{\partial t^2}\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] + k_w\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] \\ &+ c\frac{\partial}{\partial t}\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] + q_w - (e_0a)^2\frac{\partial^2 q_w}{\partial x^2} \end{aligned} \tag{7}$$

where q_w is distributed transverse load, e_0 is a constant appropriate to each material, a is an internal characteristic length (e.g., length of C–C bond, lattice parameter, and granular distance). Now consider the SWCNTs is subjected to an externally applied longitudinal magnetic field, based on the formulations derived by Murmr et al. [20], the governing equation of motion of the system can be expressed as follows:

$$\begin{aligned} &\mathrm{I}\frac{\partial^2(E(x)(\partial^2 w/\partial x^2))}{\partial x^2} + \left[\int_0^L \left(\frac{\partial w}{\partial x}\right)^2 \frac{\mathrm{E}(x)\mathrm{A}}{2L}dx\right]\frac{\partial^2}{\partial x^2}\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] \\ &- f(x,t) + (e_0a)^2\frac{\partial^2 f}{\partial x^2} = \rho(x)\mathrm{A}\frac{\partial^2}{\partial t^2}\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] \\ &+ k_w\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] + c\frac{\partial}{\partial t}\left[(e_0a)^2\frac{\partial^2 w}{\partial x^2} - w\right] + q_w - (e_0a)^2\frac{\partial^2 q_w}{\partial x^2} \end{aligned} \tag{8}$$

where $f(x,t)=\int_A \bar{f}_z dz=\xi A H_x^2 \frac{\partial^2 w}{\partial x^2}$, ξ is the magnetic field permeability and H_x is the longitudinal magnetic field.

3. Solutions by finite element method

In the present study, the finite element method is utilized to determine the solutions to Eq. (8). Using the finite element formulation, we can get the governing matrix equation of the structure after assembly as follows:

$$[\mathbf{M}]\ddot{\mathbf{U}}+[\mathbf{C}]\dot{\mathbf{U}}+\mathbf{S}(\mathbf{U})=\mathbf{F} \tag{9}$$

where $[\mathbf{M}]$ is the global consistent mass matrix of the structure, $[\mathbf{C}]$ is the global damping matrix of the structure, $\mathbf{U}$ is the global displacement vector of the structure, $\dot{\mathbf{U}}$ is the global velocity vector of the structure, $\ddot{\mathbf{U}}$ is the global acceleration vector of the structure, $\mathbf{F}$ is the global external force vector of the structure and $\mathbf{S}(\mathbf{U})$ is the global vector of restoring forces of the structure that depends on the displacement field. Eq. (9) can be solved by any direct time integration method even it is nonlinear. Needless to say, it is necessary to carry out the equilibrium iteration in each time step. In this study, the Newton-Raphson method in conjunction with Newmark scheme is adopted to perform the numerical analysis.

4. Perturbation technique

In this study, both the Young's modulus of elasticity $E(x)$ and the mass density $\rho(x)$ of SWCNTs are assumed as stochastic with respect to the position to actually describe the random material property of the SWCNTs, the geometric shapes and sizes of the structure and the longitudinal magnetic field are assumed to be deterministic. Applying the perturbation technique, the randomly fluctuating Young's modulus of elasticity $E(x)$ and the mass density $\rho(x)$ can be assumed to be of the form:

$$E(x)=E^{(0)}\left[1+\alpha(x)\right]=E^{(0)}+E^{(0)}\alpha(x) \tag{10}$$

$$\rho(x)=\rho^{(0)}\left[1+\beta(x)\right]=\rho^{(0)}+\rho^{(0)}\beta(x) \tag{11}$$

where $E^{(0)}$ and $\rho^{(0)}$ are the mean value of the Young's modulus of elasticity and mass density, respectively; $\alpha(x)$ and $\beta(x)$ are random variables with zero mean; and $E^{(0)}\alpha(x)$ and $\rho^{(0)}\beta(x)$ are homogeneous stochastic fields that represent the fluctuation of the Young's modulus of elasticity and mass density to their mean values. Assuming the random variables α and β are uniform within each element, then the stochastic nodal displacement vector can be expanded about α and β by using Taylor series as:

$$\begin{aligned}\mathbf{U}^{t+\Delta t}=\mathbf{U}^{(0)t+\Delta t}+\sum_{i=1}^{NE}\mathbf{U}_i^{(1\alpha)t+\Delta t}\alpha_i+\sum_{i=1}^{NE}\mathbf{U}_i^{(1\beta)t+\Delta t}\beta_i+\frac{1}{2}\sum_{i=1}^{NE}\sum_{j=1}^{NE}\mathbf{U}_{ij}^{(2\alpha)t+\Delta t}\alpha_i\alpha_j \\ +\frac{1}{2}\sum_{i=1}^{NE}\sum_{j=1}^{NE}\mathbf{U}_{ij}^{(2\beta)t+\Delta t}\beta_i\beta_j+\sum_{i=1}^{NE}\sum_{j=1}^{NE}\mathbf{U}_{ij}^{(2\alpha\beta)t+\Delta t}\alpha_i\beta_j+\ldots\end{aligned} \tag{12}$$

where the superscript (0) represents the mean value term, both i and j denote the element numbers, NE is the total number of the element and Σ means the merging with respect to element. Similarly, the restoring force vectors and the tangent stiffness matrix can be written in terms of Taylor series. Substituting these equations into equation (9) and applying the perturbation technique to equation (9), the higher order terms are truncated, and comparing equal order terms for the random variable α and β, the zero, first, and second order equations for the problem can be obtained, respectively. The solutions of these equations are achieved by using the procedures described in the previous section. The statistical dynamic responses of SWCNTs can be obtained after calculating the zero, first and second order equations.

5. Numerical results and discussions

In the numerical computations, the simply supported boundary conditions are considered for the SWCNTs with longitudinal magnetic field. The numerical values of the parameters are adopted as follows:

Mean value of Young's modulus E=1 Tpa, mean value of mass density $\rho=2300Kg/m^3$, tube thickness h=0.34 nm, the tube radius $R=0.7nm$ and the parameters for the random variable α and β are assumed as $\sigma_\alpha=0.1,\ \sigma_\beta=0.1,\ b_\alpha=1.0,\ b_\beta=1.0$. The length of the SWCNTs is considered as $L=40nm$. The elastic matrix is polymer and the corresponding elastic stiffness of the matrix $k_w=1$ G Pa. The value of e_0 should be determined from experiments or by matching dispersion curves of plane waves with the atomic lattice dynamics in the systems. In the present study, e_0a is considered as the whole scale coefficient and is smaller than 2.0 nm for the nano structures based on the previous study [21]. The damping coefficient of matrix is assumed as 3×10^{-7} Pa s [22]. Besides, no applied load is considered here except the longitudinal magnetic field, the initial displacement at the midpoint of the SWCNTs is assumed as 1 nm and the scale coefficient e_0a is 2 nm. For simplicity, let us define dimensionless magnetic field (DMF) as the ratio between ξAH_x^2 and $E^{(0)}I/(\pi/L)^2$ to conveniently describe the effect of magnetic field. In Figs. 2-3, the nonlinear vibration responses of the SWCNTs are presented, in particular, the mean value and standard

deviation of the midpoint displacement are depicted with respect to time, respectively. It can be seen from the Fig. 2, the amplitude decays for the nonlinear damping structure, which is quite reasonable. Furthermore, the computed numerical results from the present study are matched with those by Monte Carlo Simulation. Similar results can be detected from Fig. 3 except the amplitude of the standard deviation of the midpoint displacement is much smaller. For simplicity, from now on we only present the mean values of the midpoint displacement despite the other statistical dynamic response such as the standard deviation or the correlation function of the displacement can be obtained without any difficulty. Figs. 4 show the mean value of the midpoint displacement with or without the magnetic field. As it can be seen from Figs. 4, the natural frequency of the SWCNTs increases when the longitudinal magnetic field intensity increases, especially when dimensionless magnetic field (DMF) gets larger, besides, the amplitude of mean value of the midpoint displacement gets smaller with the magnetic field. Therefore, it can be concluded that the magnetic field has significant effect on the natural frequency and displacement of the SWCNTs. In Fig. 5, the mean value of midpoint displacement of the SWCNTs is presented for different values of elastic stiffness k_w. It can be found that the nonlinear vibration behaviors can be influenced by the elastic stiffness, furthermore, as the matrix elastic stiffness increases, the mean value of the midpoint displacement gets smaller, which is fairly reasonable. Figs. 6 depict the nonlinear vibration history by considering the small scale effects. As it can be seen from Fig. 6, the nonlinear vibration response of the SWCNTs can be influenced by the small scale coefficient.

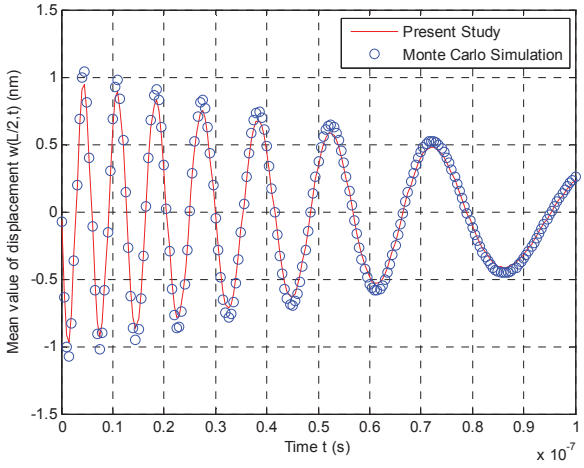


Fig. 2. Mean value of displacement w(L/2,t) of SWCNTs versus time t. ($k_w = 1\,GPa,\ e_0a = 2\,nm,\ DMF = 1$)

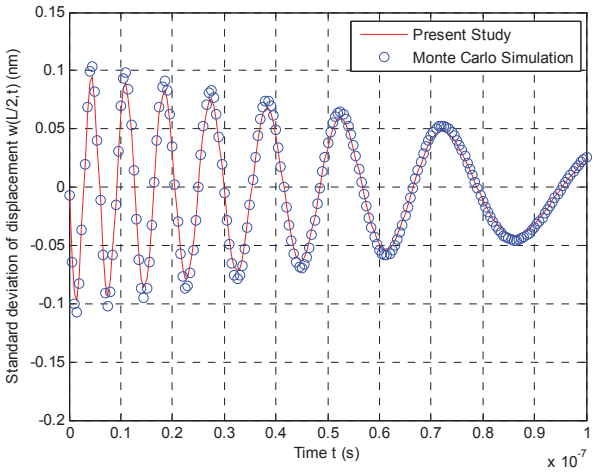


Fig. 3. Standard deviation of displacement w(L/2,t) of SWCNTs versus time t. ($k_w = 1\,GPa,\ e_0a = 2\,nm,\ DMF = 1$)

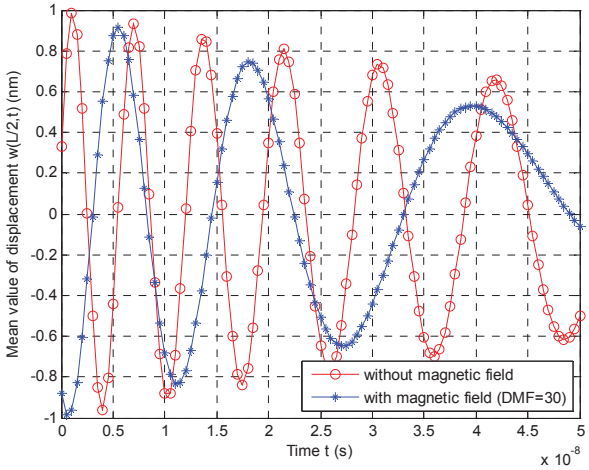


Fig. 4. Mean value of displacement w(L/2,t) of SWCNTs versus time t. ($k_w = 1\,GPa,\ e_0a = 2\,nm,\ DMF = 30$)

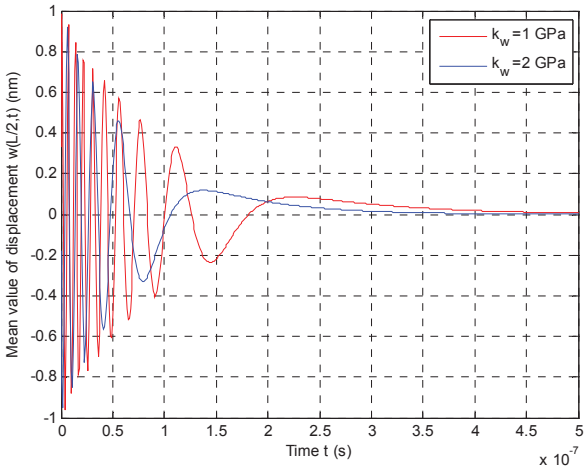


Fig. 5. Mean value of displacement w(L/2,t) of SWCNTs versus time t. ($e_0a = 2\,nm,\ DMF = 1$)

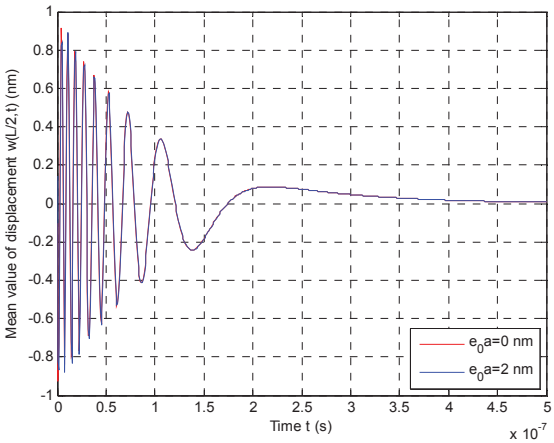


Fig. 6. Mean value of displacement w(L/2,t) of SWCNTs versus time t. $(k_w = 1\,GPa,\ DMF = 1,\ w(L/2,0) = 1\,nm)$

6. Conclusions

In the present study, we study the statistical dynamic behaviors of nonlinear vibration of the single-walled carbon nanotubes (SWCNTs) subjected to longitudinal magnetic field by considering the effects of the geometric nonlinearity. We consider both the Young's modulus of elasticity and mass density of the SWCNTs as stochastic with respect to the position to truly characterize the random material properties of the SWCNTs. Besides, we adopt the theory of nonlocal elasticity to investigate the small scale effects of the nonlinear vibration of the SWCNTs. By using the Hamilton's principle, the nonlinear governing equations of the single-walled carbon nanotubes subjected to longitudinal magnetic field are derived. We utilize the stochastic finite element method along with the perturbation technique to compute the statistical response of the SWCNTs; in particular, we adopt the Newton-Raphson iteration procedure with Newmark scheme to solve the nonlinearity of the dynamic governing equation of the SWCNTs. Some statistical dynamic response of the SWCNTs such as the mean values and standard deviations of the midpoint deflections are computed and checked by Monte Carlo simulation, besides, the effects of the small scale coefficients, magnetic field and the elastic stiffness of matrix on the statistical dynamic response of the SWCNTs are studied and discussed. It is concluded that the small scale coefficients has influences on the statistical dynamic response of the SWCNTs. Besides, the natural frequency of the SWCNTs increases when the magnetic field intensity increases, and the amplitude of mean value of the midpoint displacement gets smaller with the magnetic field. Moreover, the matrix elastic stiffness can influence the nonlinear vibration properties during the whole period, in particular, the mean value of the midpoint displacement gets smaller as the matrix elastic stiffness increases.

Acknowledgments This research was partially supported by the National Science Council in Taiwan through Grant No. MOST-103-2221-E-327-011. The authors are grateful for this support.

References

[1] S. Iijima, Helical microtubules of graphitic carbon, Nature 354 (1991) 56–58.
[2] K.T. Lau, C. Gu, D. Hui, A critical review on nanotube and nanotube/nanoclay related polymer composite materials, Compos. Part B 7 (2006) 425–436.
[3] Z. Spitalsky, D. Tasis, K. Papagelis, C. Galiotis, Carbon nanotube-polymer composites: chemistry, processing, mechanical and electrical properties, Prog. Polym. Sci. 35 (2010) 357–401.
[4] R.F. Gibson, E.O. Ayorinde, Y.F. Wen, Vibrations of carbon nanotubes and their composites: a review, Compos. Sci. Technol. 67 (2007) 1–28.
[5] L. Sun, R.F. Gibson, F. Gordaninejad, J. Suhr, Energy absorption capability of nanocomposites: a review, Compos. Sci. Technol. 69 (2009) 2392–2409.
[6] M.M. Shokrieh, R. Rafiee, A review of the mechanical properties of isolated carbon nanotubes and carbon nanotube composites, Mech. Compos. Mater. 46 (2010) 155–172.
[7] A.C. Eringen, On differential equations of nonlocal elasticity and solutions of screw dislocation and surface waves, J. Appl. Phys. 54 (1983) 4703–4710.
[8] L.J. Sudak, Column buckling of multiwalled carbon nanotubes using nonlocal continuum mechanics, J. Appl. Phys. 94 (2003) 7281–7287.
[9] J. Peddieson, G.R. Buchanan, R.P. McNitt, Application of nonlocal continuum models to nanotechnology, Int. J. Eng. Sci. 41 (2003) 305–312.
[10] W. Xia, L. Wang, Vibration characteristics of fluid-conveying carbon nanotubes with curved longitudinal shape, Comput. Mater. Sci. 49 (2010) 99–103.
[11] M. Simsek, Vibration analysis of a single-walled carbon nanotube under action of a moving harmonic load based on nonlocal elasticity theory, Phys. E 43 (2010) 182–191.
[12] T.P. Chang, Thermal–mechanical vibration and instability of a fluid-conveying single-walled carbon nanotube embedded in an elastic medium based on non-local elasticity theory, Appl. Math. Modell. 36 (2012) 1964–1973.
[13] J. Yang, L.L. Ke, S. Kitipornchai, Nonlinear free vibration of single-walled carbon nanotubes using nonlocal Timoshenko beam theory, Physica E, 42 (2010) 1727–1735.
[14] T.P. Chang, Nonlinear thermal-mechanical vibration of flow-conveying double-walled carbon nanotubes subjected to random material property, Microfluidics and Nanofluidics 15 (2013) 219-229.

[15] M. Simsek, Large amplitude free vibration of nanobeams with various boundary conditions based on the nonlocal elasticity theory, Compos. Part B 56 (2014) 621–628.
[16] J.P. Salvetat, J.A.D. Briggs, J.M. Bonard, R.R. Bacsa, A.J. Kulik, T. Stöckli, N.A. Burnham, L. Forró, Elastic and shear moduli of single-walled carbon nanotube ropes, Phys. Rev. Lett. 82 (5) (1999) 944-947.
[17] A. Krishnan, E. Dujardin, T.W. Ebbesen, P.N. Yianilos, M.M.J. Treacy, Young's modulus of single-walled nanotubes, Phys. Rev. B 58 (20) (1998) 14013-14019.
[18] A. J. Mieszawska, R. Jalilian, G. U. Sumanasekera, F. P. Zamborini, The synthesis and fabrication of one-dimensional and nanoscale heterojunctions, Small 3 (2007) 722-756.
[19] T.P. Chang, Stochastic FEM on nonlinear vibration of fluid-loaded double-walled carbon nanotubes subjected to a moving load based on nonlocal elasticity theory, Composites: Part B 54 (2013) 391-399.
[20] T. Murmu, M.A. McCarthy, S. Adhikari, Vibration response of double-walled carbon nanotubes subjected to an externally applied longitudinal magnetic field: A nonlocal elasticity approach, J. Sound Vib. 331 (2012) 5069-5086.
[21] B. Arash, Q. Wang, A review on the application of nonlocal elastic models in modeling of carbon nanotubes and graphenes, Comput. Mater. Sci. 51(2012) 303–313.
[22] E. Ghavanloo, F. Daneshmand, M. Rafiei, Vibration and instability analysis of carbon nanotubes conveying fluid and resting on a linear viscoelastic Winkler foundation, Phys. E 42 (2010) 2218–2224.

Finite Element Analysis of Microstrip Transmission Lines on Silicon Substrate

Sarhan M. Musa, Matthew N. O. Sadiku, and A.E. Shadare
Roy G. Perry College of Engineering
Prairie View A&M University
Prairie View, TX 77446
Email: smmusa@pvamu.edu, sadiku@ieee.org, shadareadebowale@yahoo.com

***Abstract*- Numerical techniques have proven to be accurate and efficient in designing microstrip in microwave monolithic integrated circuits (ICs) and interconnections in high speed digital ICs. In this paper, we apply the finite element method (FEM) to design microstrip transmission lines on silicon substrate for microwave and digital IC in future wireless technology. We mainly focus on computing the capacitance per unit length for two types structures on silicon substrate: a single microstrip on two layer media (oxide and silicon substrate) and another with two microstrips, one microstrip interacts in oxide layer and the other on top of the oxide layer all above the silicon substrate. Also, we illustrate the potential distribution of the models**

***Keywords*- Finite element method, Capacitance matrices, Microstrip, Microwave, Digital integrated circuit, Silicon technology.**

I. INTRODUCTION

Due to the complexity of electromagnetic modeling, researchers and scientists always look for development of accurate and fast methods to extract the parameters of electronic interconnects and package structures. In recent years, we have observed a magnificent application and development in the complexity, density, and speed of operations of integrated circuits (ICs), multichip modules (MCMs), and printed circuit-boards (PCBs). For examples, MCMs are extensively used to reduce interconnection delay and crosstalk effects in complex electronic systems. Multiconductor transmission lines embedded in multilayered dielectric media are known as the basic interconnection units in ICs and MCMs, and have been characterized with the distributed circuit parameters such as capacitance C and inductance L matrices under quasi-TEM conditions. Also, these distributed circuit parameters are very important factor in the electrical behavior and performance of other microwave integrated circuits (MIC) and very large scale integration (VLSI) chips. Today, in the advances for fabrication of high speed integrated circuits, it is essential to examine the limitations due to the parasitic coupled mechanisms present in silicon-integrated circuit processes. To optimize electrical properties of IC interconnects such as minimization of the length of the interconnection lines, an adequate attention must be given to the geometrical size of their transverse cross sections; the estimation of the transmission line parameters requires being accurate for system design.

Multiconductor multilayered structures are essential for ICs, MCMs, and PCBs systems, due to the important effects on the transmission characteristics of high-speed signals. Also, the transmission lines effect on the IC interconnects become extremely important for the transmission behavior of interconnect lines on silicon oxide-silicon semiconducting substrate. The conducting silicon substrate causes capacitive and inductive coupling effects in the structure. Therefore, in this work, we apply the FEM for parameter extraction for electrostatic modeling of single and coupled interconnects lines on silicon-silicon oxide substrate.

Many researchers have presented various kinds of methods for solving the problem. These include equivalent source and measured equation of dielectric Green's function and boundary integral equation approach [1-4],

CAD and quasi-static spectral domain [5-8], complex image method [9-10], quasti-stationary full-wave analysis and Fourier integral transformation [11], and conformal mapping method [12]. We illustrate that our method using FEM is suitable and effective as other methods for modeling of inhomogeneous quasi-static multiconductor interconnects in multilayered dielectric media [13].

In this work, we design of a single microstrip on two layer media (oxide and silicon substrate) and another with two microstrips, one microstrip interact in oxide layer and the other on top of the oxide layer all above the silicon substrate using FEM. We focus on the calculation of the capacitance per unit length of the models and we determine the quasi-static spectral for the potential distribution of the silicon-integrated circuit.

II. RESULTS AND DISCUSSIONS

The models designed with finite elements are unbounded (or open), meaning that the electromagnetic fields should extend towards infinity. This is not possible because it would require a very large mesh. The easiest approach is just to extend the simulation domain "far enough" that the influence of the terminating boundary conditions at the far end becomes negligible. In any electromagnetic field analysis, the placement of far-field boundary is an important concern, especially when dealing with the finite element analysis of structures which are open. It is necessary to take into account the natural boundary of a line at infinity and the presence of remote objects and their potential influence on the field shape [11]. In all our simulations, the open multiconductor structure is surrounded by a W X H shield, where W is the width and H is the thickness.

The models are designed in 2D using electrostatic environment in order to compare our results with the other available methods. In the boundary condition of the model's design, we use ground boundary which is zero potential ($V=0$) for the shield. We use port condition for the conductors to force the potential or current to one or zero depending on the setting. Also, we use continuity boundary condition between the conductors and between the conductors and left and right grounds.

In this paper, we consider two different models. Case A investigates the designing of four-transmission lines embedded in two-layered dielectric media. For case B, we illustrate the modeling of six-transmission lines interconnect in three-layered dielectric media. The results from both models are compared with some other results in the literature such as MoM, MoL, and SAGF methods and found to be close.

The dimension of the coefficient capacitance matrix is proportional to the sum of widths of every dielectric layer and the parameters of all conductors. This results in long computing time and large memory especially when the structure to be analyzed has many layers and conductors [3].

We use one port at a time as the input to evaluate all the matrix entries. With the Forced Voltage method, the capacitance matrix entries are computed from the charges that result on each conductor when an electric potential is applied to one of them and all the others are set to ground. The matrix is defined as follows:

$$\begin{bmatrix} Q_1 \\ Q_2 \\ \cdots \\ Q_N \end{bmatrix} = \begin{bmatrix} C_{11}V_1 + C_{12}V_2 + \cdots + C_{1N}V_N \\ C_{21}V_1 + C_{22}V_2 + \cdots + C_{2N}V_N \\ \cdots \\ C_{N1}V_N + C_{N2}V_2 + \cdots + C_{NN}V_N \end{bmatrix}$$

For example, using port 2 as the input will provide the entries of the second column: C_{12} , C_{22} , ..., C_{N2} .

A. A Single Microstrip on Two Layers Media Oxide and Silicon Substrate

Figure 1 shows the cross section of a single microstrip on two layers media oxide and silicon substrate.

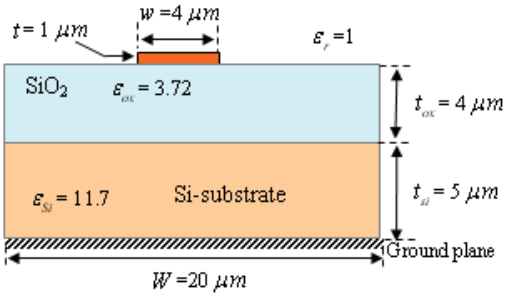


Figure 1. Cross-section of a single microstrip on two layers media oxide and silicon substrate

The geometry is enclosed by a 20 X 25 μ m shield. From the model, we generate the finite elements mesh as in Fig. 2. Table I shows the statistically properties of the mesh. Figure 3 shows the two-directional (2D) surface potential distribution. While, the counter of electrical potential (V) and streamline of electric field plots of the model are presented in figures 4 and 5, respectively.
The value of the single line capacitance is 7.0332 x 10^{-11} F/m.

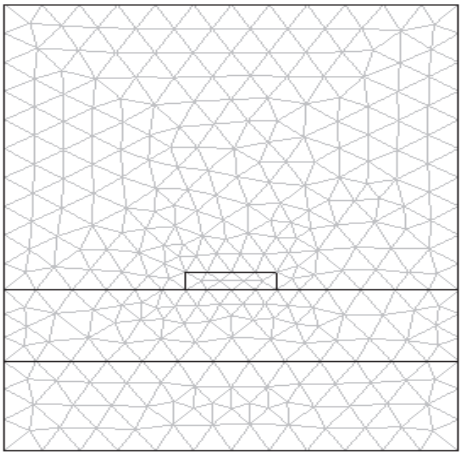


Figure 2. Mesh of open single microstrip on two layers media oxide and silicon substrate

Table 1. Mesh statistics of the single microstrip on two layers media oxide and silicon substrate

Items	Value
Number of degrees of freedom	1389
Total number of mesh points	329
Total number of elements	600
Triangular elements	600
Quadrilateral elements	0
Boundary elements	91
Vertex elements	12
Minimum element quality	0.8294
Element area ratio	0.1503

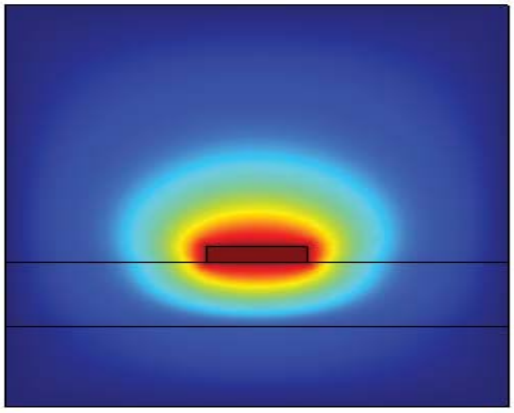


Figure 3. 2D surface of open single microstrip on two layers media oxide and silicon substrate

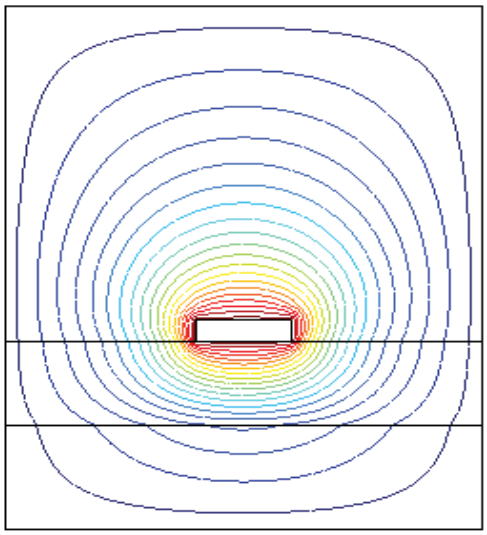


Figure 4. Contour plot of open single microstrip on two layers media oxide and silicon substrate

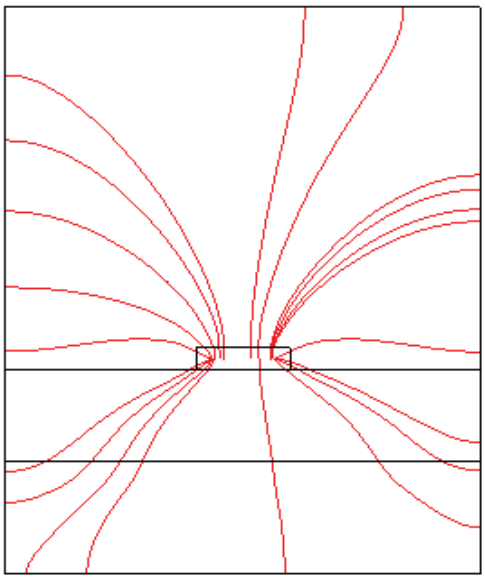


Figure 5. Streamline plot of open single microstrip on two layers media oxide and silicon substrate

Figure 6 shows the comparison analysis of potential distribution of the model with and without dielectric substrate along the line from (x,y) =(0,0) to (x,y) =(20,25) μ m .

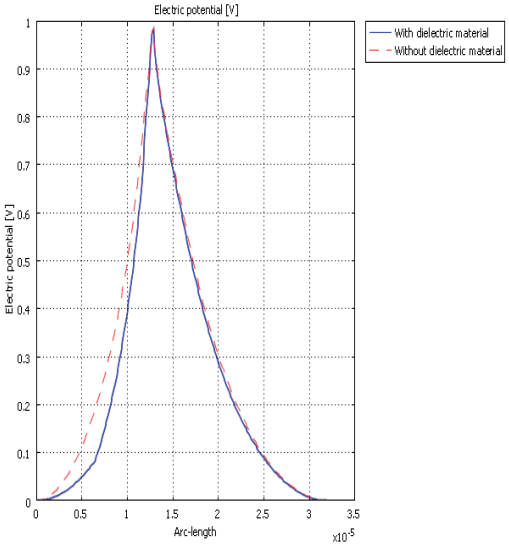


Figure 6. Potential distribution of the model with and without dielectric substrate

It observed from Fig. 5 that the peak value of electric potential is approximately same as the different dielectric in placed in the substrate.

B. Two Microstrips in Interconnect Layers

Figure 7 shows the cross section two microstrips, one microstrip interact in oxide layer and the other on top of the oxide layer all above the silicon substrate.

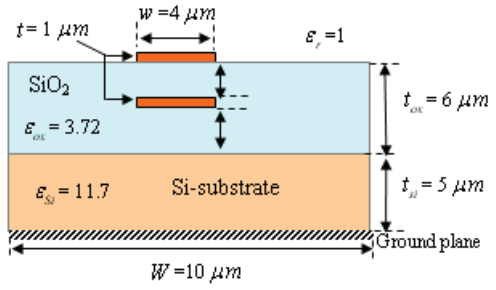


Figure 7. Cross-section of the two microstrips in interconnect layers

The geometry is enclosed by a 10 X 15 μm shield. From the model, we generate the finite elements mesh as in Fig. 8. Table II shows the statistically properties of the mesh. Figure 9 shows the two-directional (2D) surface potential distribution with port 1 as input. While, the counter of electrical potential (V) and streamline of electric field plots of the model are presented in Figures 10 and 11, respectively.

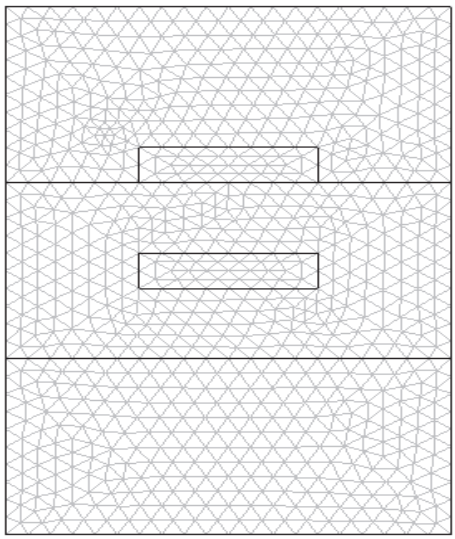


Figure 8. Mesh of the two microstrips in interconnect layers

Table 1I. Mesh statistics of the two microstrips in interconnect layers

Items	Value
Number of degrees of freedom	3805
Total number of mesh points	907
Total number of elements	1712
Triangular elements	1712
Quadrilateral elements	0
Boundary elements	172
Vertex elements	16
Minimum element quality	0.8308
Element area ratio	0.1915

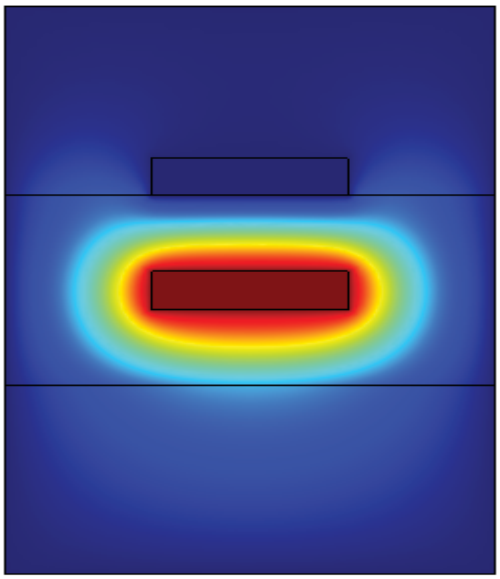


Figure 9. 2D surface of the two microstrips in interconnect layers

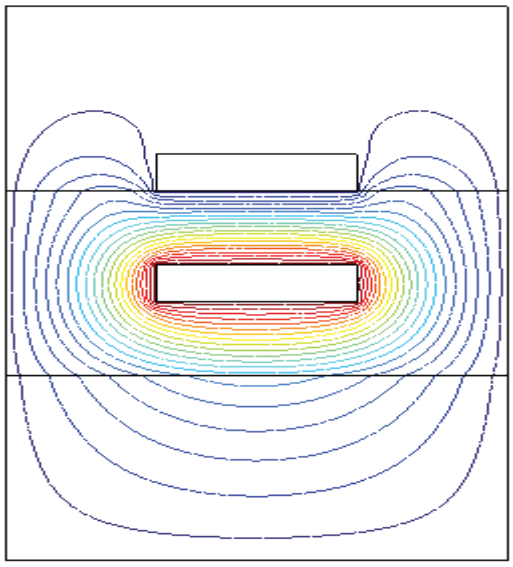


Figure 10. Contour plot of the two microstrips in interconnect layers

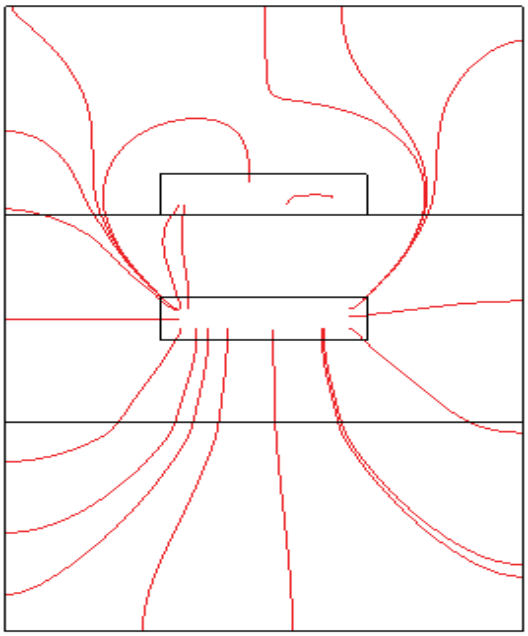


Figure 11. Streamline plot of the two microstrips in interconnect layers

Figure 12 shows the comparison analysis of potential distribution of the model with and without dielectric substrate along the line from (x,y) =(0,5) to (x,y) =(10,15) μ m .

It observed from Fig. 12 that the peak value of electric potential is approximately same as the different dielectric in placed in the substrate.

The value of the matrix of capacitance per unit length of the model is

$$[C]=\begin{bmatrix} 1.269\times10^{-10} & -8.134\times10^{-11} \\ -8.134\times10^{-11} & 1.269\times10^{-10} \end{bmatrix} F/m\,.$$

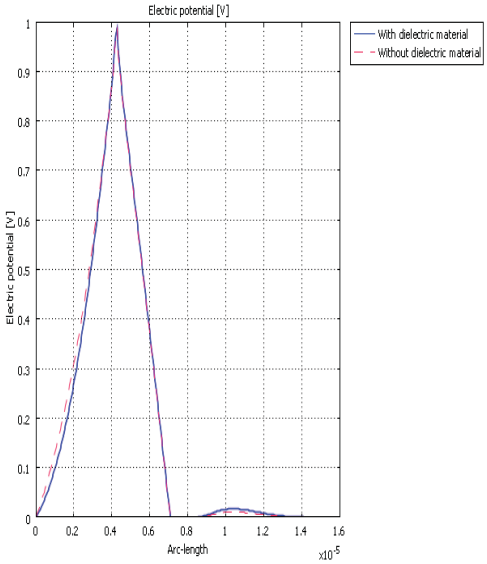


Figure 12. Potential distribution of the model with and without dielectric substrate

III. CONCLUSION

In this paper, we have presented the modeling in 2D of transmission lines on silicon substrate using FEM as suitable and effective method. We obtained for models the capacitance per unit length and the potential distribution.

IV. REFERENCES

[1] H. Ymeri, B. Nauwelaers, and K. Maex, "On the modeling of multiconductor multilayer systems for interconnect applications,"*Microelectronics Journal,* vol. 32, pp. 351-355, 2001.

[2] H. Ymeri, B. Nauwelaers, and K. Maex, "On the frequency-dependent line admittance of VLSI interconnect lines on silicon-based semiconductor substrate," *Microelectronics Journal,* vol. 33, pp. 449-458, 2002.

[3] H. Ymeri, B. Nauwelaers, and K. Maex, "On the capacitance and conductance

calculations of integrated-circuit interconnects with thick conductors," *Microwave and Optical Technology Letters,* vol. 30, no. 5, pp. 335-339, 2001.

[4] W. Delbare and D. De Zutter, "Accurate calculations of the capacitance matrix of a multiconductor transmission line in a multilayered dielectric medium," *IEEE Microwave Symposium Digest,* Long Beach, CA, pp. 1013-1016, 1989.

[5] J. Zhang, Y-C. Hahm, V. K. Tripathi, and A. Weisshaar, "CAD-oriented equipment-circuit modeling of on-chip interconnects on lossy silicon substrate," *IEEE Transaction on Microwave Theory and Techniques*, vol. 48, no. 9, pp. 1443-1451, 2000.

[6] H. Ymeri, B. Nauwelaers, and K. Maex, "Distributed inductance and resistance per-unit-length formulas for VLSI interconnects on silicon substrate," *Microwave and Optical Technology Letters,* vol. 30, no. 5, pp. 302-304, 2001.

[7] H. Ymeri, B. Nauwelaers, K. Maex, S. Vandenberghe, and D. D. Roest, "A CAD-oriented analytical model for frequency-dependent series resistance and inductance of microstrip on-chip interconnects on multilayer silicon substrates," *IEEE Transaction on Advanced Packaging* , vol. 27, no. 1, pp. 126-134, 2004.

[8] E. Groteluschen. L. S. Dutta, and S. Zaage. "Full-wave analysis and analytical formulas for the line parameters of transmission lines on semiconductor substrates," *Integration, the VLSI Journal*, vol. 16, pp. 33-58, 1993.

[9] C-N. Chiu, "Closed-form expressions for the line-coupling parameters of coupled on-chip interconnects on lossy silicon substrate," *Microwave and Optical Technology Letters,* vol. 43, no. 6, pp. 495-498, 2004.

[10] J.J. Yang, G. E. Howard, and Y.L. Chow, "Complex image method for analyzing multiconductor transmission lines in multilayered dielectric media", *International Symposium Digest of Antennas and Propagation,* pp. 862-865, 1991.

[11] H. Ymeri, B. Nauwelaers, and K. Maex, "Frequency-dependent mutual resistance and inductance formulas for coupled IC interconnects on an Si-SiO_2 substrate," *Integration, the VLSI Journal*, vol. 30, pp. 133-141, 2001.

[12] E. Chen and S. Y. Chou, "Characteristics of coplanar transmission lines multilayer substrates: modeling and experiments," *IEEE Transaction on Microwave Theory and Techniques*, vol. 45, no. 6, pp. 939-945, 1997.

[13] S. M. Musa, M. N.O. Sadiku, and P. H. Obiomon, "Integrated circuit interconnect lines on lossy silicon substrate with finite element method," *International journal of Engineering Research and Applications*, vol. 4, no. 1, pp. 243-247, 2014.

Time Adaptivity Stable Finite Difference for Acoustic Wave Equation

A. J. M. Antunes[1], R. C. P. Leal-Toledo[2], E. M. Toledo[3], O. T. S. Filho[2], and E. Marques[2]
[1] SEEDUC-FAETEC, Rio de Janeiro, Brazil
[2] Federal Fluminense University, Niteroi, Brazil
[3]LNCC - UFJF, Petropolis, Brazil

Abstract—*Acoustic wave modeling is widely used to synthesize seismograms theoretically, being the basis of the reverse time migration strategy. Explicit Finite Difference Method (FDM) is often employed to find numerical solution of this problem and in this case, spatial discretization is related to the shortest wavelength to be captured and temporal discretization is determined by stability condition. In this case small grid size has to be used to assure a stable and accurate solution and algorithms with locally adjustable time steps can be of advantageous use when treating heterogeneous domains. In this paper, we are concerned with a temporal adaptivity algorithm: Region Triangular Transition algorithm (RTT), discussing its accuracy and its computational efficiency when applied to complex heterogeneous domains. To evaluate computational efficiency of this algorithm we present, in this work, how computational cost varies with the subregions sizes ratio of the heterogeneous medium when compared with computational cost of the conventional algorithm using only one time step, showing how this adaptivity algorithm can be used in complex domains to reduce the amount of values to be calculated. Some discussion are made concerning how dispersion error can be reduced when adaptive schemes are used.*

Keywords: time adaptivity, finite difference, seismic analysis

1. Introduction

Acoustic wave modeling is widely used to theoretically synthesize seismograms and is also the basis of the reverse time migration. Explicit Finite Difference Method (FDM) is frequently used in the numerical solution of this problem and spatial discretization is related to the shortest wavelength to be captured and temporal discretization is restricted by some stability condition. A small grid size helps to increase the accuracy but the substantial growth in memory and computational cost may become these methods prohibitive for realistic modeling.

Several proposals have been presented to improve the accuracy and stability of the FDM, including staggered-grid finite-difference method [1], [2], variable grid difference schemes [3], higher order operators to approximate the derivatives [4], [5], and variable time step.

In the application of FDM in heterogeneous media in its conventional form, to ensure numerical stability of the solution, temporal discretization is given by the smallest time step defined by the subregion of the heterogeneous domain with the highest wave propagation velocity. In this context algorithms using locally adjustable time steps have been proposed to optimize computational cost as it occurs in numerical seismic modeling, adjusting the time increment for each subregion with different physical characteristics of the considered domain.

Falk et al. [6] proposed, in 1998, an algorithm for local time step adjustment for the solution of elastic wave equation in a domain composed of two subregions, obtaining a reduction of 77% in process time if compared to conventional algorithm. Meanwhile in this algorithm the time steps needed to be proportional to 2^n, being n an integer value, a new algorithm was suggested by Tessmer [7], extending this proportional limiting relationship to any integer value n. Although these formulations work well in heterogeneous domains, reducing the computing effort, accuracy may be affected with the appearance of a *noise* in the signal wave which was identified for time steps close to the stability threshold of the used method as shown in [8] and in [9]. In these references we presented a new adjustable time step algorithm, named RTT, that remains stable even when using time steps close to the stability condition for each sub-region of the heterogeneous domain. The amount of reduction in computational cost is directly related to the domain to be analyzed, either through the relationship between the values of different existing propagation velocities as well as with the size of each subdomain associated with these propagation velocities.

To evaluate computational efficiency of this algorithm we present in this work how computational cost varies with the heterogeneous medium subregions sizes ratio when compared to the computational cost of the conventional algorithm using only one time step. We show also, with a complex domain example how this adaptivity algorithm can be used in such cases to reduce the amount of values to be calculated. Some discussion is also made concerning how dispersion error can be reduced when adaptive schemes are used.

2. Acoustic Wave Equation

The acoustic wave equation for a two-dimensional $(2D)$ problem can be described as:

$$\frac{\partial^2 u}{\partial t^2} - c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = f(x, y, t) \tag{1}$$

where $u = u(x, y, t)$ is a function of space and time and c is the medium wave velocity propagation, $f(x, y, t)$ the time variation of the source distribution. Initial conditions are $u(x, y, 0) = g(x, y)$ and $\frac{\partial u}{\partial t}(x, y, 0) = h(x, y), \forall x, y \in \Omega$ and suitable boundary conditions are adopted.

For the two-dimensional acoustic wave, using Finite Difference Method (FDM) with temporal and spatial second order approximations (2-2) one obtain an explicit method with the following expression:

$$\begin{aligned} \mathbf{u}_{i,j}^{t+\Delta t} &= 2\mathbf{u}_{i,j}^{t} + C^2(\mathbf{u}_{i+1,j}^{t} - -2\mathbf{u}_{i,j}^{t} + \\ &+ \mathbf{u}_{i-1,j}^{t} + \mathbf{u}_{i,j+1}^{t} - 2\mathbf{u}_{i,j}^{t} + \mathbf{u}_{i,j-1}^{t}) - \\ &- \mathbf{u}_{i,j}^{t-\Delta t} + \Delta t^2 f(x, y, t) + O(\Delta t^2, h^2) \end{aligned} \tag{2}$$

with Courant number $C = \frac{c\Delta t}{\Delta h}$, where t designates the actual time, i and j indexes designate spatial mesh points in directions x and y while Δt and h define time and spatial discretizations respectively, with $\Delta x = \Delta y = h$. Next we can see in Eq. 3 the expression of the explicit Finite Difference Method(FDM) when using time second order and spatial fourth order approximations (2-4).

$$\begin{aligned} \mathbf{u}_{i,j}^{t+\Delta t} &= 2\mathbf{u}_{i,j}^{t} + \frac{C^2}{12}\left[-\mathbf{u}_{i+2,j}^{t} - \right. \\ &- \mathbf{u}_{i-2,j}^{t} - \mathbf{u}_{i,j+2}^{t} - \mathbf{u}_{i,j-2}^{t} + \\ &+ 16(\mathbf{u}_{i+1,j}^{t} + \mathbf{u}_{i-1,j}^{t} + \mathbf{u}_{i,j+1}^{t} + \\ &+ \left. \mathbf{u}_{i,j-1}^{t}) - 60\mathbf{u}_{i,j}^{t}\right] - \mathbf{u}_{i,j}^{t-\Delta t} + \\ &+ \Delta t^2 f(x, y, t) + O(\Delta t^2, h^4). \end{aligned} \tag{3}$$

The finite difference approximations previously described are conditionally stable and their stability limits for homogeneous domains given in terms of Courant number (C), can be found in [10].

Conventional explicit finite-difference algorithms in heterogeneous media are based on a global and constant time step to ensure the stability of the method. The use of locally adjustable time steps can help to save a great deal of computing time, using intermediate increments in time that are adjusted depending on the local wave velocity propagation. In this case, the choice of these intermediate time intervals, must satisfy the stability limits for each subregion of the heterogeneous domain.

Close to these subregions separating boundaries, spatial derivatives approximations have to be calculated by accessing mesh points of finite differences that could not have their values defined in the same instant of time due to the use of different temporal discretizations. These points work like $ghostpoints$ for subregions using these intermediate time steps. The way proposed to calculate these values in this *transition region*, is that differentiate these adaptive algorithms. Thus, the *transition region* depends on the algorithm used and of finite difference order to approximate the spatial derivatives. To preserve accuracy, temporal adaptivity algorithms use finite difference approximation of the same order to calculate values for the points in the *transition region* but differ in how to calculate these values.

As mentioned in Section 1, Falk et al. [6] proposed an algorithm where time increments needed to be multiples of 2^n, where n is a positive integer while Tessmer [7] suggested a modification to this algorithm which lets you use any previous time discretization satisfying $\Delta t_{subreg} = \Delta t/(n + 1)$. This last algorithm has a constant transition zone width, which varies only with the order of the finite difference approximation used to approximate the spatial derivatives. Both these algorithms utilize different time steps (Δt) in the definition of the *transition region*, ranging between some maximum (Δt_{max}) and the minimum (Δt_{min}) and produce good results when discretizations are far below the stability limit of the methods [6], [7], [8]. However, in these schemes a *noise* appears in the solution close to the border of the physical discontinuities for temporal discretizations near to the stability limit of the method, and this *noise* spread throughout the domain over time [9]. In [8] we proposed an algorithm named RTT that uses the same time steps in all *transition region* and its efficiency is analyzed here in the present paper.

2.1 Region Triangular Transition Algorithm (RTT)

The Region Triangular Transition (RTT) temporal adaptivity algorithm we proposed in [9], enables the adoption of the same time steps ratio subdivision presented by Tessmer algorithm in [7] with the advantage of the attenuation of the undesirable effects at the interface where wave speed changes.

The RTT algorithm construction is shown in Fig. 1 to Fig. 3 considering time second-order and space fourth order approximation FDM (named RTT 2-4) applied to a 1D domain composed of two sub-regions: $Region1$ with propagation velocity V_1 and locally stable time step Δt and mesh points represented by black triangles and retangles and $Region2$ with velocity $V_2 = 4V_1$ and time step $\Delta t/4$, and mesh points indicated by white retangles.

Although with this scheme the transition region is enlarged compared to others adaptive schemes it preserves the same time discretization to calculate all points values that are directly used to approximate values near two subregions separating boundary. This transition zone forms a triangular

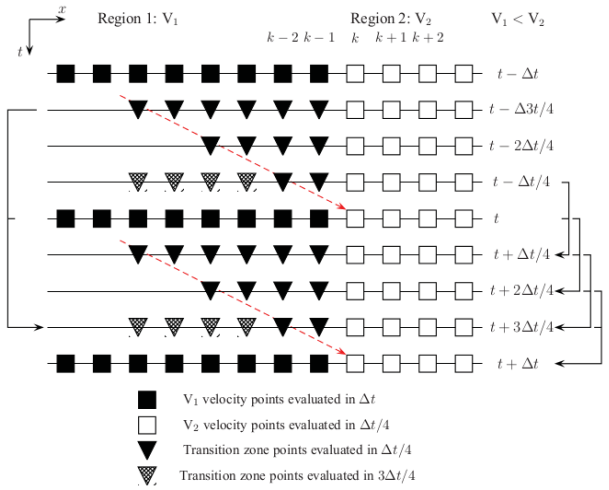


Fig. 1: Triangular Scheme - 1D - approximations 2-4 - 4 subdivisions of time step.

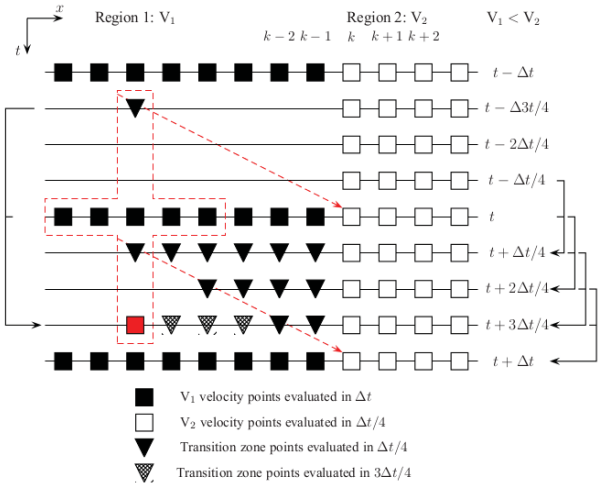


Fig. 3: Triangular Scheme - 1D - approximation 2-4 - 4 subdivisions of time step - Additional Points.

region composed by the black triangles in the four intermediate times between t and $t + \Delta t$ as shown in Fig. 1, since we set here $V2 = 4V1$.

Points in Fig. 2 and Fig. 3 show the new values to be calculated in time $t + \Delta t$. Points of the transition region which are calculated in the first level of time $t + \Delta t/4$, require values in two moments of earlier times and that are outside the triangular region. Only those values of the transition region are calculated $\Delta T = 3\Delta t/4$, as shown in Fig. 3.

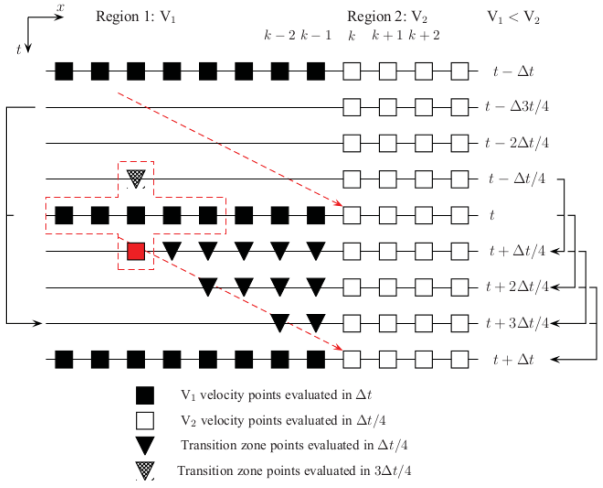


Fig. 2: Triangular Scheme - 1D - approximation 2-4 - 4 subdivisions of time step - Triangular Region.

This scheme can be easily generalized to any integer ratio between different time steps, as Δt and $\Delta t/2$ or Δt and $\Delta t/3$; for combinations of different temporal discretizations provided time steps have integer ratios, as Δt, $\Delta t/2$ and $\Delta t/4$ or Δt, $\Delta t/3$ and $\Delta t/6$; and also for different spatial dimensions ($2D$ and $3D$ domains).

2.2 Considerations about Dispersion Error

In FDM the analysis of the dispersion error are made considering homogeneous media taking constants spatial and temporal discretizations. In this section we present results to show what influence the use of different time discretizations can cause, in dispersion error. To illustrate this let us take the 1D scalar acoustic wave equation.

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}$$

with properly boundary and initial conditions.

For explicit FDM with temporal and spatial second order approximations (2-2) we obtain the following expression:

$$\begin{aligned} \mathbf{u}_i^{t+\Delta t} &= 2\mathbf{u}_i^t + C^2(\mathbf{u}_{i+1}^t - 2\mathbf{u}_i^t + \mathbf{u}_{i-1}^t) \\ &- \mathbf{u}_i^{t-\Delta t} + O(\Delta t^2, h^2). \end{aligned} \tag{4}$$

and dispersion error [11] is given by:

$$\omega = \frac{2}{\Delta t} asin\left[c\frac{\Delta t}{\Delta x} sin\left(\frac{\Delta x}{2}\right)\right]. \tag{5}$$

One obtains a better view about dispersion error using normalized group velocity defined as $\frac{v_g}{c} = \frac{1}{c}\frac{d\omega}{dk}$ and $\overline{K} = \frac{k\Delta x}{2}$, which applied to Eq. 5 produces:

$$\frac{v_g}{c} = cos\left(\overline{K}\right)\left[1 - c^2\frac{\Delta t^2}{\Delta x^2}sin^2(\overline{K})\right]^{-1/2} \quad (6)$$

Alford [12] showed that dispersion on finite differences is smaller for Courant numbers close to stability limit and we explore this result on our analysis. Although this analysis treats only homogeneous domains it was used here to evaluate the dispersion error behaviour in two domains using the same spatial discretization assuming one domain has a wave velocity propagation twice the other ($V_1 = 2V_2$). As an example Fig. 4 shows dispersion error results when we use the same time step in both regions which means different Courant numbers such as $C_1 = 0.8$ and $C_2 = 0.4$. We can see in this figure different dispersion errors and note more dispersion on the region with the lower velocity. One can expect that, in this case, the dispersion error in one region influences the results for the whole domain. This result suggests a smaller dispersion on adaptive algorithms in comparison with classical algorithm when the time step is adjusted to keep the lowest dispersion error which, in this case, is $\Delta t_2 = 2\Delta t_1$.

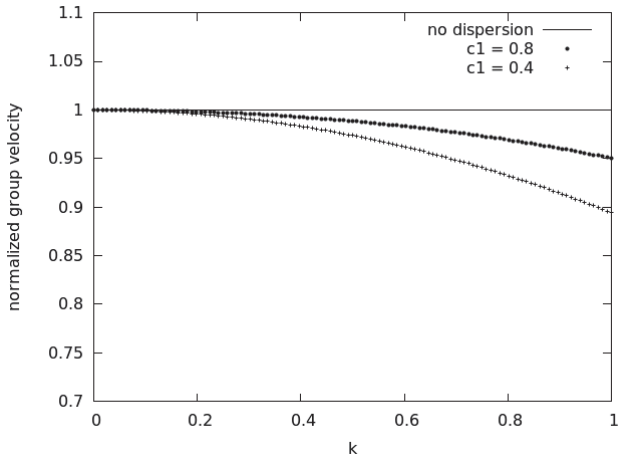


Fig. 4: Dispersion in classic algorithm

3. Numerical Results

3.1 Comparing RTT with Conventional Algorithm

To compare results obtained with the RTT algorithm with conventional FDM and reference results [8], Equation (1) was solved in a 2D square (Ω), consisting of an heterogeneous field, subdivided in two subregions: in $Subregion1$ (Ω_1), we assumed the lower propagation velocity, $V_1 = 750$ m/s while $Subregion2$ (Ω_2) had the greatest velocity, $V_2 = 4V_1 = 3000$m/s. The interface with velocity discontinuity were at $x = 3250$m and initial condition was given by:

$$u(x, y, 0) = \frac{1}{12800\pi}e^{\left(\frac{-(x^2+y^2)}{12800}\right)} \quad (7)$$

and $\frac{\partial u}{\partial t}(x, y, 0) = 0., \forall x, y$. As boundary condition we considered homogeneous Dirichlet, ie, $u(\Gamma, t) = 0., t > 0$, being Γ the boundaries of the considered domain, with $x \in [0.; 6000.]$ and $y \in [0.; 6000.]$. To stress the effects of RTT algorithm in the numerical result precision we show results obtained with different spatial discretizations, including coarse meshes.

Figure 5, Figs. 6 and 7 show results in $y = 4500$m and Fig. 8, Figs. 9 and 10 show results in $x = 4500$ m both at $t = 0.90$s for finite difference method with time second order and spatial fourth order approximations(2-4). We consider $h = \Delta x = \Delta y = 10$m; 20m and 40m with $C = 0.61$, (close to the stability limit), using conventional FDM taking this Courant number for the subregion where the propagation velocity is $V_2 = 4V_1$.

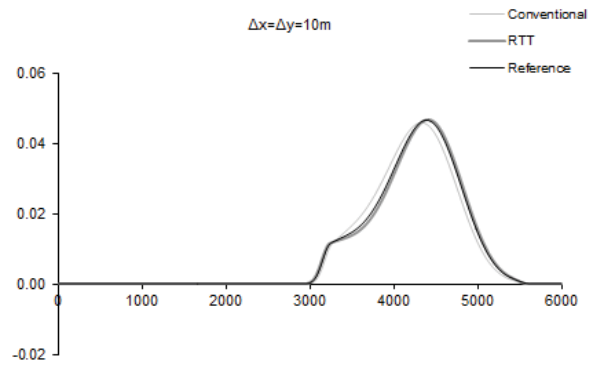


Fig. 5: Solution in $y = 4500$m at $t = 0.90$s and $h = 10$m

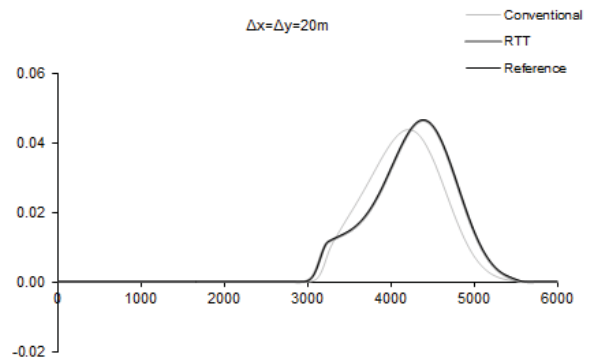


Fig. 6: Solution in $y = 4500$m at $t = 0.90$s and $h = 20$m.

It can be seen, for all displayed graphics, that RTT algorithm gives a better approximation to the adopted reference result than those obtained with conventional methods.

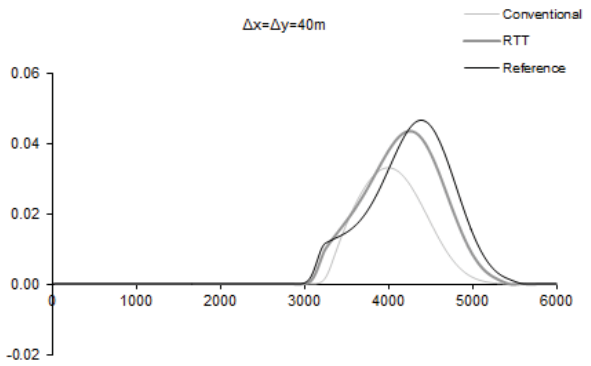


Fig. 7: Solution in $y = 4500$m at $t = 0.90$s and $h = 40$m

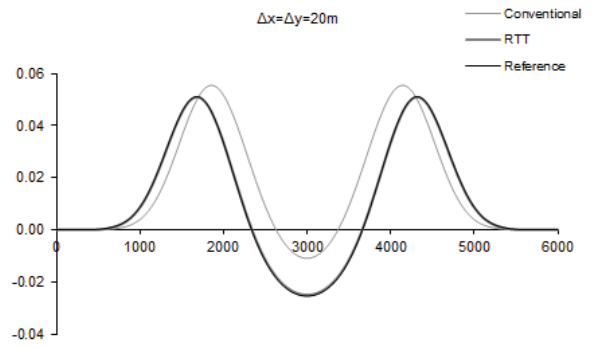


Fig. 9: Solution in $x = 4500$m at $t = 0.90$s and $h = 20$m.

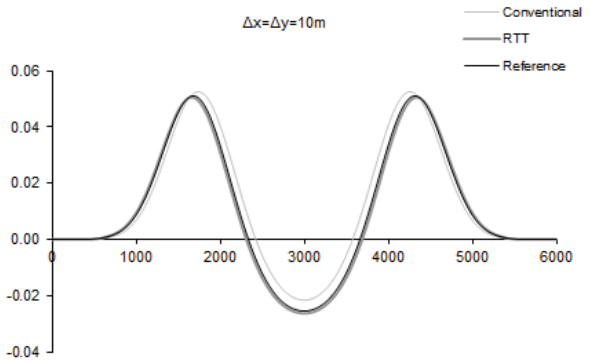


Fig. 8: Solution in $x = 4500$m at $t = 0.90$s and $h = 10$m

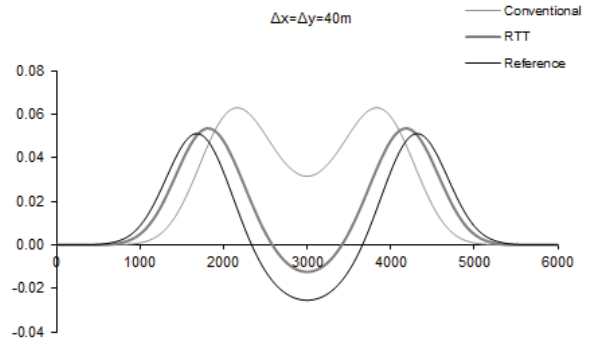


Fig. 10: Solution in $x = 4500$m at $t = 0.90$s and $h = 40$m

3.2 Computational Cost Comparison

Temporal adaptivity always reduces the number of floating-point operations and this reduction depends on the properties of each heterogeneous media considered, being directly related to the subregions size ratios and to their wave propagation velocities relation. To evaluate the efficiency of RTT algorithm we show in Tables 1 and 2, results obtained for the same 2D domain described in Section 3.1 varying the interface subregion boundary. The first column shows the position (y) of the interface between these two subregions, Ω_1 and Ω_2 while second and third columns present the percentages of the whole domain that each subregion occupies. Processing times comparisons are displayed as the ratio of RTT time processing to conventional algorithm time processing, for each one of these six cases. These resuts are shown in columns 4 and 5 of these tables: on Table 1 when using the RTT 2-2 and on Table 2 for RTT 2-4 scheme. Both algorithms, conventional and RTT, used mesh sizes with $h = 10$ m and $h = 20$ m and, while RTT adaptive scheme employed two time steps, one for each subregion, conventional algorithm used only the smallest time step of that in RTT scheme.

3.3 Treatment of Complex Domains - Adaptivity Possibilities

Next, to show an even more practical implication of the strategy presented here, we evaluated the use of different time steps combinations possibilities in the adaptivity scheme. To do this we used the well-known 2D synthetic acoustic Marmousi model, shown in Fig. 11 which contains, due to its heterogeneity, a complex 2-D wave velocity distribution field in a square domain with $(4602 \times 1502)m$, where the great wave velocity is 4.3 km/h.

Time adaptivity in this kind of data can be executed by joining distinct subregions, which can have the same time step, defining what we call here $\Delta t's$ map as shown in Fig. 12 and Fig. 13. This subdivision depends on the desired adaptivity scheme to be used. The most refined temporal discretization Δt_{min}, associated to the higher wave velocity propagation, defines all others time increments to be used for others regions defined in this map.

For the comparative analysis, here taken in effect, we used a spatial discretization of $\Delta x = \Delta y = 1m$, totalizing 6,912,204 values to be calculated per time step, considering two different temporal discretizations combinations in RTT

Table 1: Time processing comparisons - Conventional Versus RTT Scheme 2-2

Interf. (coord. y)	Subregions size rate *(area)* (Ω_1/Ω)	(Ω_2/Ω)	RTT/conv.*(time)* $(h=10)$	$(h=20)$
1125	15 %	85 %	46.86 %	48.80 %
2250	30 %	70 %	40.64 %	41.59 %
3375	45 %	55 %	34.51 %	36.29 %
4500	60 %	40 %	28.80 %	30.88 %
5625	75 %	25 %	25.00 %	26.58 %
6750	90 %	10 %	17.75 %	20.52 %

Table 2: Time processing comparisons - Conventional Versus RTT Scheme 2-4

Interf. (coord. y)	Subregions size rate *(area)* (Ω_1/Ω)	(Ω_2/Ω)	RTT/conv.*(time)* $(h=10)$	$(h=20)$
1125	15 %	85 %	36.84 %	42.74 %
2250	30 %	70 %	34.78 %	39.24 %
3375	45 %	55 %	32.00 %	35.34 %
4500	60 %	40 %	29.88 %	33.48 %
5625	75 %	25 %	32.84 %	34.58 %
6750	90 %	10 %	25.41 %	32.98 %

algorithm: (a) Δt and $\Delta t/2$; (b) Δt, $\Delta t/2$ and $\Delta t/4$.

For this first configuration one obtains the distribution shown in Fig. 12, $\Delta t_{min} = \Delta t/2$ and $\Delta t_{max} = \Delta t = 2\Delta t_{min}$ while for the second one obtains Fig. 13, $\Delta t_{min} = \Delta t/4$ and $\Delta t_{max} = \Delta t = 4\Delta t_{min}$. In this case we considered $\Delta t_{min} = 0.000164$ s

For a simulation time of only $1s$, corresponding to 6098 time integration steps with conventional algorithm, without any adaptivity one has to adopt $\Delta t = \Delta t_{min}$ and the amount of values needed to be calculated is $42,150,619,992$. In Table 3 and Table 4 we present the quantity of values to be evaluated for each time discretization using RTT adaptive scheme for configurations showed in Fig. 12 and Fig. 13 displaying their relations with the number of values to be calculated with conventional FDM.

Table 3: Number of operations - Time-steps used: Δt_{min} and $2\Delta t_{min}$.

Values of Δt	Number of Intervals	Values number
0.000164	6098	201,459,626
0.000328	3049	20,974,580,183
Total RTT		21,176,039,809
Perc. values RTT/Convent.		50.24%

4. Conclusion

Acoustic wave equation employed to model problems in seismic modeling applications involving heterogeneous media by conventional processing with explicit finite difference

.

Fig. 11: Wave velocity distribution - Marmousi data set

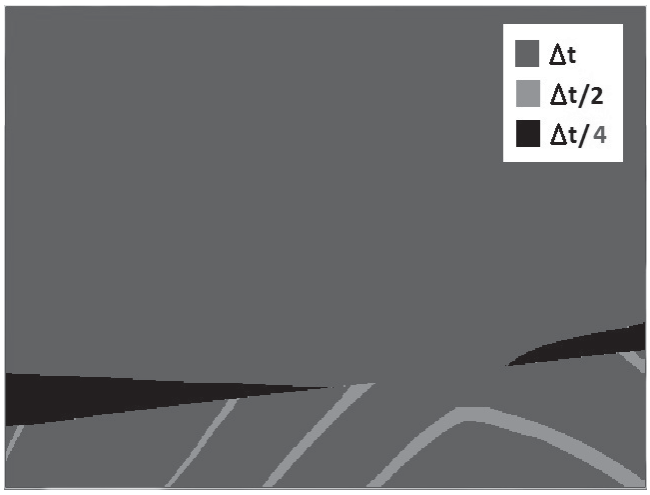
.

Fig. 12: Δt map ($\Delta t_{min} = \Delta t/2$ and $2\Delta t_{min} = \Delta t$)

uses constant increment of time across the whole domain, which implies in large computational effort.

When high contrasts in the values of physical characteristics are present, as it occurs with the variation in wave propagation speed values in the subsurface model, temporal adaptive procedures are a good choice for the reduction of this computational effort. Working with big problems these savings with RTT algorithm, can reach considerable values as it was shown here for not so large problem.

We also showed that with RTT we can obtain less dispersion error than using conventional algorithm and exemplify how to apply it in more complex domains with the use of a pre-processing step constructing a $\Delta t's$ map. In this case, pre-processing are required with some computational extra cost, and an efficient algorithm can make this task easily classifying and ranking the different subregions according to their wave speed propagation which, with the spatial mesh characteristics, defines the maximum time step to be



.

Fig. 13: Δt map ($\Delta t_{min} = \Delta t/4$, $2\Delta t_{min} = \Delta t/2$ and $4\Delta t_{min} = \Delta t$)

Table 4: Number of operations - Time-steps used: Δt_{min}, $2\Delta t_{min}$, $4\Delta t_{min}$

Values of Δt	Number of Intervals	Values number
0.000164	6098	199,185,072
0.000328	3049	71,050,849
0.000656	1524	10,448,905,188
Total RTT		10,719,141,109
Perc. values RTT/Convent.		25.43%

used in each one of those. This RTT procedure follows the same methodology for different temporal discretizations possibilities and it can be easily extended to tridimensional domains. Finally as in seismic problems the same model domain is solved several times changing only the seismic source location, this $\Delta t's$ map can be generate only once for all analysis what makes RTT an attracting algorithm.

References

[1] Q. Jin, W. Shigui, and C. Ruofei. Accuracy of the staggered-grid finite-difference method of the acoustic wave equation for marine seismic reflection modeling. *Chinese Journal of Oceanology and Limnology*, 31:169–177, 2013.

[2] Y. Wang, W. Liang, Z. Nashed, and X. Li. Seismic modeling by optimizing regularized staggered-grid finite-difference operators using a time-space-domain dispersion-relationship-preserving method. *Geophysics*, 79:T277–T285, 2014.

[3] C. Chua and P. L. Stoffab. Nonuniform grid implicit spatial finite difference method for acoustic wave modeling in tilted transversely isotropic media. *Journal of Applied Geophysics*, 76:44–49, 2012.

[4] J. Chen. High-order time discretization in seismic modeling. *Geophysics*, 72:SM115–SM122, 2007.

[5] W. Liao. On the dispersion, stability and accuracy of a compact high-order finite difference scheme for 3d acoustic wave equation. *Journal of Computational and Applied Mathematics*, 270:571, 2014.

[6] J. Falk, E. Tessmer, and D. Gajewski. Efficient finite-difference modelling of seismic waves using locally adjustable time steps. *Geophysical Prospecting*, 46:603–616, 1998.

[7] E. Tessmer. Seismic finite-difference modeling with spatially varying time steps. *Geophysical Prospecting*, 65:1290–1293, 2000.

[8] A. J. M. Antunes. Finite difference method for acoustic wave equation using locally adjustable time steps (in portuguese). Master's thesis, UFF - Federal Fluminense University - Niteroi, 2012.

[9] A. J. M. Antunes, A. J. Lima, R. C. Leal-Toledo, O. T. Silveira Filho, and E. M. Toledo. Finite difference method for solving acoustic wave equation using locally adjustable time-steps. *Procedia Computer Science*, 29:627–636, 2014.

[10] Laurence R. Lines, Rafael Slawinski, and Phillip R. Bording. A recipe for stability analysis of finite-diference wave equations computations. *CREWES Research Report*, 10:6 p., 1998.

[11] G. C. Cohen. *Higher Order Numerical Methods for Transient Wave Equations*. Springer, 2002.

[12] R. M. Alford, K. R. Kelly, and D. M. Boore. Accuracy of finite-difference modeling of the acoustic wave equation. *Geophysics*, 39:834–842, 1974.

Complex Dynamics of Hybrid Cellular Automata Composed of Two Period Rules

Bo Chen, Fangyue Chen, Zhongwei Cao and Xia Zhang
Department of Mathematics, School of Science, Hangzhou Dianzi University, Hangzhou, China

Abstract— *The members of Chua's period rules, which are considered to have the simplest dynamic behaviors before, actually define chaotic subsystems by introducing the hybrid mechanism. Through exploiting the mathematical definition of hybrid cellular automata (HCAs), this work presents an analytical method of symbolic dynamics of the HCA rule 77 and 168 as well as the HCA rule 28 and 33. In particular, the two HCAs are all topologically mixing and possess the positive topological entropy on their subsystems. This result therefore naturally argues that they are chaotic in the sense of both Li-Yorke and Devaney on the subsystems. Finally, it is worth mentioning that the method presented in this paper is also applicable to other HCAs therein.*

Keywords: Hybrid cellular automata, symbolic dynamics, chaos, topologically mixing, topological entropy

1. Introduction

Cellular automata (CAs) are a class of spatially and temporally discrete, deterministic mathematical systems with large degrees of freedom characterized by local interactions and an inherently parallel form of evolution [1-5]. Basing on previous work, L. O. Chua et al. provided a nonlinear dynamics perspective to Wolfram's empirical observations and grouped elementary cellular automata (ECAs) into six classes depending on the quantitative analysis of the orbits [6-10]. These six classes are established as period-1, period-2, period-3, Bernoulli σ_τ-shift, complex Bernoulli-shift and hyper Bernoulli-shift rules. It is worth mentioning that some of their work is consistent with previous studies of other authors.

In view of an one-dimensional CA, when the evolution of all its cells is dependent on the one and only global function, it is called uniform, otherwise it will be called hybrid, i.e. hybrid cellular automata (HCA) [11,12]. For instance, denoted by HCA(N,M), HCA rule, composed of ECA rule N and ECA rule M, is specified to obey the ECA rule N at odd sites of the cell array and obey the rule M at even sites of the cell array. There are much research on HCAs which have been applied in cryptographically secure, see [13-15] and references therein.

Though HCAs are endowed with simple hybrid rules and evolve on the same square tile structures, the evolution of HCAs may exhibit rich dynamical behavior with local interactions. More accurately, it can be asserted that the dynamics of the CAs might be changed from simple to complex and vice versa by just introducing the hybrid mechanism. Noting that the dynamics of Chua's period rules are extremely simple, we have opted for two of these rules to compose the HCAs and discovered that several HCAs ultimately produce behavior of complexity. Although ECA rule 77 and ECA rule 168 are belong to Chua's period-1 rules, ECA rule 28 and ECA rule 33 are belong to Chua's period-2 rules, it is found that HCA(77,168) and HCA(28,33) are endowed with glider phenomena.

2. Preliminaries

First and foremost, several terminology and notations are the necessary prerequisite to the rigorous consideration in the following. The set of bi-infinite configurations is denoted by $S^Z = \cdots S \times S \times S \cdots$ and a metric d on S^Z is defined as $d(x,\overline{x}) = \sum_{i=-\infty}^{+\infty} \frac{1}{2^{|i|}} \frac{\widetilde{d}(x_i,\overline{x}_i)}{1+\widetilde{d}(x_i,\overline{x}_i)}$, where $S = \{0,1,\ldots,k-1\}$, $x,\overline{x} \in S^Z$ and $\widetilde{d}(\cdot,\cdot)$ is the metric on S defined as $\widetilde{d}(x_i,\overline{x}_i) = 0$, if $x_i = \overline{x}_i$; otherwise, $\widetilde{d}(x_i,\overline{x}_i) = 1$. As for a finite symbol S, a word over S is finite sequence $a = (\alpha_0,...,\alpha_n)$ of elements of S.

In S^Z, the cylinder set of a word $a \in S^Z$ is $[a]_k = \{x \in S^Z | x_{[k,k+n]} = a\}$, where $k \in Z$. It is apparent that such a set is both open and closed (called clopen) [17]. The cylinder sets generate a topology on S^Z and form a countable basis for this topology. Therefore, each open set is a countable union of cylinder sets. In addition, S^Z is a Cantor space. The classical right-shift map σ is defined by $[\sigma(x)]_i = x_{i-1}$ for any $x \in S^Z$,$i \in Z$. A map $F : S^Z \to S^Z$ is a CA if and only if it is continuous and commutes with σ, i.e., $\sigma \circ F = F \circ \sigma$. For any CA, there exists a radius $r \geq 0$ and a local rule $N : S^{2r+1} \to S$ such that $[F(x)]_i = N(x_{[i-r,i+r]})$. Moreover, (S^Z, F) is a compact dynamical system.

A set $X \subseteq S^Z$ is F invariant if $F(X) \subseteq X$ and strongly F invariant if $F(X) = X$. If X is closed and F invariant, then (X,F) or simply X is called a subsystem of F. A set $X \subseteq S^Z$ is an attractor if there exists a nonempty clopen F-invariant set Y such that $\bigcap_{n\geq O} F^n(Y) = X$. Thus, there always exists a global attractor, denoted by $\Lambda = \bigcap_{n\geq O} F^n(S^Z)$, which is also called the limit set of F. For instance, let $\mathcal{A}$ denote a set of some finite words over S, and $\Lambda = \Lambda_\mathcal{A}$ is the set which consists of the bi-infinite configurations made up of all the words in $\mathcal{A}$. Then, $\Lambda_\mathcal{A}$ is a subsystem of (S^Z, σ), where $\mathcal{A}$ is said to be

the determinative block system of Λ. For a closed invariant subset $\Lambda \subseteq S^Z$, the subsystem (Λ, σ) or simply Λ is called a subshift of σ.

3. Dynamics of HCA(77,168)

Among 256 ECA rules, 67 belong to period-1 rules since most random initial bit strings converge to a period-1 attractor. The 25 dynamically-independent period-1 rules are N={0, 4, 8, 12, 13, 28, 32, 40, 44, 72, 76, 77, 78, 104, 128, 132, 136, 140, 160, 164, 168, 172, 200, 204, 232}. And the dynamics of the remaining rules can be trivially predicted and determined from one of the corresponding equivalent rules in N [7]. Here, we construct the HCAs composed of two period-1 rules in N. Thus, an empirical glimpse on all spatio-temporal patterns of above HCAs reveals that glider phenomena are discovered in HCA(13,40), HCA(13,104), HCA(13,160), HCA(13,168), HCA(13,232), HCA(77,160), HCA(77,168) and HCA(77,232).

Figure 1 provides an emblematic example of spatio-temporal pattern of HCA(77,168). By exploiting the mathematical definition of HCAs, this section is devoted to an in-depth study of the symbolic dynamics of HCA(77,168) in the bi-infinite symbolic sequences space [18-20]. For bi-infinite, one-dimensional ECAs, $r = 1$ and S is denoted by $\{0, 1\}$. Each local rule of ECAs can be endowed with a boolean function [16]. Then, the boolean function of rule 77 is expressed as $[F_{77}(x)]_i = x_i\overline{x}_{i+1} \oplus \overline{x}_{i-1}\overline{x}_i\overline{x}_{i+1} \oplus \overline{x}_{i-1}x_ix_{i+1}, \forall i \in Z$, where $x_i \in S$, "$\cdot$", "$\oplus$" and "$-$" denote "AND", "XOR" and "NOT" logical operations, respectively. The Boolean function of rule 168 is expressed as $[F_{168}(x)]_i = x_{i-1}x_{i+1} \oplus \overline{x}_{i-1}x_ix_{i+1}, \forall i \in Z$. Consequently, the Boolean function of HCA(77,168) is induced as

$$[F(x)]_i = \begin{cases} [F_{77}(x)]_i & \text{if i is odd,} \\ [F_{168}(x)]_i & \text{if i is even.} \end{cases}$$

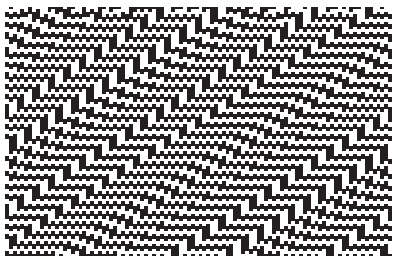


Fig. 1: The spatio-temporal pattern of HCA(77,168).

Proposition 1: For HCA(77,168), there exists a subset $\Lambda_{\mathcal{A}}$ of S^Z, such that $F^4(x)|_{\Lambda_{\mathcal{A}}} = \sigma_L^4(x)|_{\Lambda_{\mathcal{A}}}$, where $\Lambda_{\mathcal{A}} = \{x = (\cdots, x_{-2}, x_{-1}, x_0, x_1, x_2, x_3, \cdots) \in S^Z, (x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{i+3}) \in \mathcal{A}, \forall i \in Z\}$ and $\mathcal{A}$ = {(0,0,0,0,0,0),(0,0,0,0,0,1),(0,0,0,1,0,1),(0,0,0,1,1,0),(0,0,0,1,1,1),(0,1,0,1,0,1),(0,1,0,1,1,0),(0,1,0,1,1,1),(0,1,1,0,0,0),(0,1,1,0,0,1),(0,1,1,0,1,0),(0,1,1,1,1,0),(0,1,1,1,1,1),(1,0,0,0,0,0),(1,0,0,0,0,1),(1,0,0,1,0,1),(1,0,0,1,1,0),(1,0,0,1,1,1),(1,0,1,0,0,0),(1,0,1,0,0,1),(1,0,1,0,1,0),(1,1,1,0,0,0),(1,1,1,0,0,1),(1,1,1,0,1,0),(1,1,1,1,1,0),(1,1,1,1,1,1)}.

***Proof*:** The proof of above proposition can be obtained through computer-aided method according to the definitions of F and σ_L, the details are omitted here.

Proposition 2: $\Lambda_{\mathcal{A}}$ in Proposition 1 is a subshift of finite type of (S^Z, σ).

The following objective is to investigate the complexity and chaotic dynamics of F^4 on its subsystem $\Lambda_{\mathcal{A}}$. It follows from proposition 2 that dynamical behaviors of F^4 on S^Z can be characterized via a subshift $\Lambda_{\mathcal{A}}$, which is a subshift of finite type. As the topological dynamics of a subshift of finite type is largely determined by the properties of its transition matrix, it is helpful to briefly review some definitions.

A matrix D is positive if all of its entries are non-negative, irreducible if $\forall i, j$, there exist n such that $D_{ij}^n > 0$, and aperiodic if there exists N, such that $D_{ij}^n > 0, n > N, \forall i, j$. If Λ is a two-order subshift of finite type, then it is topologically mixing if and only if D is irreducible and aperiodic, where D is its associated transition matrix with $D_{ij} = 1$, if $(i, j) \prec \Lambda$; otherwise $D_{ij} = 0$.

Then, the topologically conjugate relationship between $(\Lambda_{\mathcal{A}}, \sigma)$ and a two-order subshift of finite type is established. The dynamical behaviors of F^4 on $\Lambda_{\mathcal{A}}$ are discussed based on existing results.

Let $\widehat{S} = \{r_0, r_1, \cdots, r_{24}, r_{25}\}$ be a new symbolic set, where $r_i, i = 0, \ldots, 25$, stand for elements of $\mathcal{A}$ respectively. Then, one can construct a new symbolic space $\widehat{S}^Z$ on $\widehat{S}$. Denote by $\mathcal{B} = \{(rr')|r = (b_0b_1b_2b_3b_4b_5), r' = (b'_0b'_1b'_2b'_3b'_4b'_5) \in \widehat{S}, \forall 2 \leq j \leq 5 \; such \; that \; b_j = b'_{j-1}\}$. Further, the two-order subshift $\Lambda_{\mathcal{B}}$ of σ is defined by $\Lambda_{\mathcal{B}} = \{r = (\cdots, r_{-1}, r_0^*, r_1, \cdots) \in \widehat{S}^Z | r_i \in \widehat{S}, (r_i, r_{i+1}) \prec \mathcal{B}, \forall i \in Z\}$. Therefore, it is easy to calculate the transition matrix D of the subshift $\Lambda_{\mathcal{B}}$ as follow:

$$\begin{pmatrix}
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1
\end{pmatrix}$$

Proposition 3: $(\Lambda_{\mathcal{A}}, \sigma_L^4)$ and $(\Lambda_{\mathcal{B}}, \sigma_L^4)$ are topologically conjugate; namely, $(\Lambda_{\mathcal{A}}, F^4)$and $(\Lambda_{\mathcal{B}}, \sigma_L^4)$ are topologically conjugate.

***Proof*:** Define a map from $\Lambda_{\mathcal{A}}$ to $\Lambda_{\mathcal{B}}$ as follows: $\pi : \Lambda_{\mathcal{A}} \to \Lambda_{\mathcal{B}}, x = (..., x_{-1}, x_0^*, x_1, ...) \mapsto (..., r_{-1}, r_0^*, r_1, ...)$, Where $r_i = (x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}, x_{i+5}), \forall i \in Z$. Then, it follows from the definition of $\Lambda_{\mathcal{B}}$ that for any $x \in \Lambda_{\mathcal{A}}$, one has $\pi(x) \in \Lambda_{\mathcal{B}}$; namely,$\pi(\Lambda_{\mathcal{A}}) \subseteq \Lambda_{\mathcal{B}}$. One can easily check that π is a homeomorphism and $\pi \circ \sigma_L^4 = \sigma_L^4 \circ \pi$. Therefore, $(\Lambda_{\mathcal{A}}, \sigma_L^4)$ and $(\Lambda_{\mathcal{B}}, \sigma_L^4)$ are topologically conjugate.

Proposition 4: F^4 is topologically mixing on $\Lambda_{\mathcal{A}}$.

***Proof*:** It follows from [17] that a two-order subshift of finite type is topologically mixing if and only if its transition matrix is irreducible and aperiodic. Meanwhile, it is easy to verify that D^n is positive for $n \geq 4$, where D is the transition matrix of the two-order subshift $\Lambda_{\mathcal{B}}$. This implies that D is irreducible and aperiodic.

Proposition 5: The topological entropy of $F^4|_{\Lambda_{\mathcal{A}}}$ is $log\lambda^* = log(2.61803) = 0.962424$, where λ^* is the maximum positive real root of equation $\lambda^{22}(\lambda^2 - 3\lambda + 1)(\lambda^2 - \lambda + 1) = 0$.

***Proof*:** Recall that two topologically conjugate systems have the same topological entropy and the topological entropy of σ on $\Lambda_{\mathcal{B}}$ equals $log\rho(D)$, where $\rho(D)$ is the spectral radius of the transition matrix D of the subshift $\Lambda_{\mathcal{B}}$.

Proposition 6: F is topologically mixing on $\Lambda_{\mathcal{A}}$.

***Proof*:** To prove $F|_{\Lambda_{\mathcal{A}}}$ is topologically mixing, it is necessary to check that for any two open sets $U, V \subset \Lambda_{\mathcal{A}}$, $\exists N$ such that $F(U) \cap V \neq \varnothing$, for $n \geq N$. Since $F|_{\Lambda_{\mathcal{A}}}$ is topologically mixing, it immediately follows that for any two open sets $U, V \subset \Lambda_{\mathcal{A}}$, there exists N_0 such that $(F^4)^k(U) \cap V = F^{4k}(U) \cap V \neq \varnothing$, for $k \geq N_0$, we consider two situations separately.

Case 1: $n = 4k$. It is obvious that $F^n(U) \cap V = F^{4k}(U) \cap V \neq \varnothing$.

Case 2: $n = 4k+1, 4k+2$ or $4k+3$. Firstly, we need to prove that $F : \Lambda_{\mathcal{A}} \to \Lambda_{\mathcal{A}}$ is a homeomorphism. It is evident that $F|_{\Lambda_{\mathcal{A}}}$ is surjective. Suppose that there exist $x, x' \in \Lambda_{\mathcal{A}}$ such that $F(x) = F(x')$. Thus, $F^4(x) = F^4(x')$ holds, i.e., $\sigma_L^4(x) = \sigma_L^4(x')$, which implies $x = x'$. Hence, $F|_{\Lambda_{\mathcal{A}}}$ is injective. Since $\Lambda_{\mathcal{A}'}$ is a compact Hausdorff space, and $F|_{\Lambda_{\mathcal{A}}}$ is one-to-one, onto and continuous, F^{-1} exists and is continuous. Consequently, $F : \Lambda_{\mathcal{A}} \to \Lambda_{\mathcal{A}}$ is a homeomorphism. This implies that $F^i(U)$ is also an open set, thus, one has $F^n(U) \cap V = F^{4k} \circ (F^i(U)) \cap V \neq \varnothing$, where $i = 1, 2, 3$.

It follows from [17] that the positive topological entropy implies chaos in the sense of Li-Yorke. Meanwhile, the topological mixing is also a very complex property of dynamical systems. A system with topologically mixing property has many chaotic properties in different senses. For instance, the chaos in the sense of Li-York can be deduced from positive topological entropy. More importantly though, both the chaos in the sense of Devaney and Li-York can be deduced from topologically mixing.

In conclusion, the mathematical analysis presented above provides the rigorous foundation for the following theorem.

Theorem 1: F is chaotic in the sense of both Li-Yorke and Devaney on the subsystem $\Lambda_{\mathcal{A}}$.

4. Dynamics of HCA(28,33)

Among 256 ECA rules, 25 belong to period-2 rules since most random initial bit strings converge to a period-2 periodic orbit. Moreover, the 13 dynamically-independent period-2 rules are M={1, 5, 19, 23, 28, 29, 33, 37, 50, 51, 108, 156, 178} [8]. An empirical glimpse on all spatio-temporal patterns of HCAs composed of two period-2 rules reveals that shift phenomena are discovered in HCA(23,28), HCA(28,33), HCA(28,37), HCA(28,50), HCA(28,178), HCA(33,156), HCA(37,156) and so on. Remarkably, HCA(29,33) and HCA(29,37) are endowed with more complicated phenomena.

Figure 2 provides the example of spatio-temporal pattern of HCA(28,33). Its symbolic dynamics on the space of bi-infinite symbolic sequences is also briefly discussed in the following. The boolean function of rule 28 is expressed as $[F_{28}(x)]_i = x_{i-1}\overline{x}_i\overline{x}_{i+1} \oplus \overline{x}_{i-1}x_i, \forall i \in Z$, where $x_i \in S$. The Boolean function of rule 33 is expressed as $[F_{33}(x)]_i = x_{i-1}\overline{x}_i x_{i+1} \oplus \overline{x}_{i-1}\overline{x}_i\overline{x}_{i+1}, \forall i \in Z$. Consequently, the Boolean function of HCA(28,33) is induced as

$$[\widetilde{F}(x)]_i = \begin{cases} [F_{28}(x)]_i & \text{if i is odd,} \\ [F_{33}(x)]_i & \text{if i is even.} \end{cases}$$

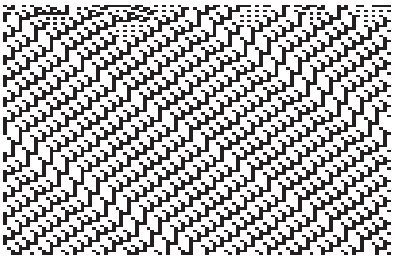


Fig. 2: The spatio-temporal pattern of HCA(28,33).

Proposition 7: For HCA(28,33), there exists another subset $\Lambda_{\widetilde{\mathcal{A}}}$ of S^Z, such that $\widetilde{F}^4(x)|_{\Lambda_{\widetilde{\mathcal{A}}}} = \sigma_R^4(x)|_{\Lambda_{\widetilde{\mathcal{A}}}}$, where $\Lambda_{\widetilde{\mathcal{A}}} = \{x = (\cdots, x_{-3}, x_{-2}, x_{-1}, x_0, x_1, x_2, x_3, x_4 \cdots) \in S^Z, (x_{i-3}, x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}) \in \widetilde{\mathcal{A}}, \forall i \in Z\}$ and $\widetilde{\mathcal{A}}_2 = \{$(0,0,0,0,1,0,0,0),(0,0,0,0,1,0,0,1),(0,0,0,0,1,0,1,0),(0,0,0,0,1,0,1,1),(0,0,0,0,1,1,1,0),(0,0,1,0,0,1,0),(0,0,0,1,0,0,1,1),(0,0,0,1,0,1,0,1),(0,0,0,1,0,1,1,1),(0,0,1,0,0,0,0,0),(0,0,1,0,0,0,0,1),(0,0,1,0,0,0,1,0),(0,0,1,0,0,0,1,1),(0,0,1,0,0,1,0,0),(0,0,1,0,1,0,0,0),(0,0,1,0,1,0,0,1),(0,0,1,0,1,1,1,0),(0,0,1,1,

1,0,0,0),(0,0,1,1,1,0,0,1),(0,1,0,0,1,0,0,0),(0,1,0,0,1,0,0,1),(0,1,0,0,1,0,1,0),(0,1,0,0,1,0,1,1),(0,1,0,0,1,1,1,0),(0,1,0,1,0,1,0,1),(0,1,0,1,0,1,1,1),(0,1,0,1,1,1,1,0),(0,1,1,1,1,0,0,1),(1,0,0,0,0,0,1,0),(1,0,0,0,0,0,1,1),(1,0,0,0,0,1,0,0),(1,0,0,0,0,1,0,1),(1,0,0,0,1,0,0,0),(1,0,0,0,1,0,0,1),(1,0,0,0,1,0,1,0),(1,0,0,0,1,0,1,1),(1,0,0,0,1,1,1,0),(1,0,0,1,0,0,1,0),(1,0,0,1,0,0,1,1),(1,0,1,0,0,0,0,0),(1,0,1,0,0,0,0,1),(1,0,1,0,0,0,1,0),(1,0,1,0,0,0,1,1),(1,0,1,0,0,1,0,0),(1,0,1,1,1,0,0,0),(1,1,1,0,0,0,0,0),(1,1,1,0,0,0,0,1),(1,1,1,0,0,0,1,0),(1,1,1,0,0,0,1,1),(1,1,1,0,0,1,0,0)}.

$\Lambda_{\widetilde{\mathcal{A}}}$ is a subshift of finite type of (S^Z, σ_R). Further, the relevant two-order subshift $\Lambda_{\widetilde{\mathcal{B}}}$ also can be introduced by the above similar methods. Therefore, it is easy to calculate the transition matrix $\widetilde{D}$ of the subshift $\Lambda_{\widetilde{\mathcal{B}}}$ as follow:

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Meanwhile, $\widetilde{D}^n$ is positive for $n \geq 10$, which implies $\widetilde{D}$ is irreducible and aperiodic. Thus, it can be obtained immediately that $\widetilde{F}^4$ is topologically mixing on $\Lambda_{\widetilde{\mathcal{A}}}$. Therefore, $\widetilde{F}$ is topologically mixing on $\Lambda_{\widetilde{\mathcal{A}}}$. The topological entropy of $\widetilde{F}^4|_{\Lambda_{\widetilde{\mathcal{A}}}}$ is $log\lambda^* = log(2.139374) = 0.760513$ as λ^* is the maximum positive real root of $\lambda^{40}(\lambda^9 - 2\lambda^8 + \lambda^7 - 4\lambda^6 + 2\lambda^5 + \lambda^4 + \lambda^3 - \lambda - 1)(\lambda + 1) = 0$ which is the characteristic equation of $\widetilde{D}$. In conclusion, the mathematical analysis presented above provides the rigorous foundation for the following theorem.

Theorem 2: $\widetilde{F}$ are chaotic in the sense of both Li-Yorke and Devaney on the subsystem $\Lambda_{\widetilde{\mathcal{A}}}$.

Acknowledgments

This research was supported by NSFC (Grants no. 11171084 and 60872093).

References

[1] J. von-Neumann, *Theory of Self-reproducing Automata*, Urbana, University of Illinois Press, 1966.

[2] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Rev. Mod. Phys.*, vol. 55, pp. 601-644, 1983.

[3] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D*, vol. 10, pp. 1-35, 1984.

[4] S. Wolfram, *Theory and Applications of Cellular Automata*, Singapore, World Scientifc, 1986.

[5] S. Wolfram, *A New Kind of Science*, Champaign, Wolfram Media, 2002.

[6] L. O. Chua, V. I. Sbitnev and S. Yoon, "A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science. Part I: Threshold of Complexity," *Int. J. Bifur. Chaos*, vol. 12, pp. 2655-2766, 2002.

[7] L. O. Chua, G. E. Pazienza and J. Shin, "A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science. Part X: Period-1 Rules," *Int. J. Bifur. Chaos*, vol. 19, pp. 1425-1654, 2009.

[8] L. O. Chua, G. E. Pazienza and J. Shin, "A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science. Part XI: Period-2 Rules," *Int. J. Bifur. Chaos*, vol. 19, pp. 1751-1930, 2009.

[9] L. O. Chua and G. E. Pazienza, "A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science. Part XII: Period-3, Period-6, and Permutive Rules," *Int. J. Bifur. Chaos*, vol. 19, pp. 3887-4038, 2009.

[10] L. O. Chua and G. E. Pazienza, "A Nonlinear Dynamics Perspective of Wolfram's New Kind of Science. Part XIII: Bernoulli σ_τ-Shift Rules," *Int. J. Bifur. Chaos*, vol. 20, pp. 1859-2003, 2010.

[11] K. Cattell and J. C. Muzio, "Synthesis of One-Dimensional Linear Hybrid Cellular Automata," *IEEE Trans. Computers*, vol. 15, pp. 325-335, 1996.

[12] J. Binghama and B. Binghamb, "Hybrid One-Dimensional Reversible Cellular Automata are Regular," *Discrete Appl. Math.*, vol. 155, pp. 2555-2566, 2007.

[13] C. F. Rubio, L. H. Encinas, S. H. White, A. M. Rey and G. R. Sánchez, "The Use of Linear Hybrid Cellular Automata as Pseudorandom Bit Generators in Cryptography," *Neural Parallel Sci. Comput.*, vol. 12, pp. 175-192, 2004.

[14] D. Radu, "Hybrid Cellular Automata as Pseudo-Random Number Generators with Binary Synchronization Property," *in Proceedings of the Intl. Symposium on Signals Circuits and Systems*, Iasi Romania, pp. 389-392, 2009.

[15] D. Radu and D. Ioana, "Efficient and Cryptographically Secure Pseudorandom Number Generators based on Chains of Hybrid Cellular Automata Maps," *in Proceedings of COMM 2014 10th Intl. Conference on Communications*, 2014.

[16] J. B. Guan, S. W. Shen, C. B. Tang and F. Y. Chen. "Extending Chua's Global Equivalence Theorem on Wolfram's New Kind of Science," *Int. J. Bifur. Chaos*, vol. 17, pp. 4245-4259, 2007.

[17] Z. L. Zhou, *Symbolic dynamics*, Shanghai, Shanghai Scientific and technological Education Publishing House, 1997.

[18] F. Y. Chen, W. F. Jin ,G. R. Chen, F. F. Chen and L. Chen, "Chaos of Elementary Cellular Automata Rule 42 of Wolfram's Class II," *Chaos*, vol. 19, 013140, 2009.

[19] W. F. Jin, F. Y. Chen, G. R. Chen, L. Chen, and F. F. Chen, "Extending the Symbolic Dynamics of Chua's Bernoulli-Shift Rule 56," *J. Cell. Automata*, vol. 5, pp. 121-138, 2010.

[20] W. F. Jin and F. Y. Chen, "Topological Chaos of Universal Elementary Cellular Automata Rule," *Nonlinear Dyn.*, vol. 63, pp. 217-222, 2011.

Simplified cellular automata model of aluminum anodization

J. Stafiej[1], **Ł. Bartosik**[2]
[1]Department of Mathemathics and Natural Sciences, Cardinal Stefan Wyszynski University, Warsaw, Poland
[2]Institute of Physical Chemistry, Warsaw, Poland

Abstract— *Recently we have shown that the pore formation in anodized alumina passive layers can be modeled in a cellular automata approach taking into account the oxygen anion formed at the layer-solution interface and drifting towards metal-layer interface where it oxidizes the metal. Here we present an alternative, simpler anodization model. The simplification amounts to reducing the number of states by eliminating the state considered to represent the anion and its drift across the passive layer. The aim of this paper is to show that the simpler model can reproduce least qualitatively the results of the more complex model. The anion drift effect can be viewed as similar to hole conductivity where holes are layer states free of walkers that simulate the effect of the electric field. By the known symmetry argument the drift of walkers from high to low density region incurs an equivalent and opposite drift of holes. We also modify our approach to random walker generation and metal oxidation at the metal-oxide interface to conform better to electrostatic condition implying that the metal surface is equipotential. We compare the results obtained for both of these models. The essential feature of the original model, obtaining a hexagonally ordered, porous oxide layer is preserved in the simplified model.*

Keywords: Cellular automata, Parallel programming, Anodization, Corrosion

1. Introduction

Anodization is a process extensively used in corrosion protection and staining of metallic parts for about a century. It can be applied to valve metals in a simple way enhancing the protective and decorative functions of the passive layer obtained particularly on aluminum. A successful anodization of a given metallic surface requires only a specific electrolyte bath and external voltage to polarize the system. In such conditions anodic oxidation and related phenomena occur. The structure formation of anodic films on aluminum was first observed in 1953 by Keller *et. al.*[1] using transition electron microscopy. Unexpectedly, a well organized lattice of hexagonally arranged pores appeared to form in the layer. This discovery did not attract a particular attention at first. Recent interest in anodic aluminum oxide has been renewed due to its potential applications in nanotechnology as scaffolds for other nanostructure synthesis[2]. The experimental procedures for preparation of the hexagonal alumina layers are described by Masuda *et. al.*[3] and further developed by Li[4] and Jessensky[5].

Theoretical work on anodization lags behind the experimental development and cannot describe all aspects of the process to help design and rationalize the experimental procedures. There are two most often considered theoretical approaches to the process, namely Field Assisted Dissolution(FAD)[6] and Field Assisted Flow(FAF)[7]. The approaches have some basic features in common but differ widely in defining the fundamental cause of the emergence of an organized porous structure. In the case of the Field Assisted Flow approach the driving force of the process is assumed to be mechanical stress generated due to the Peeling-Bedworth factor associated with forming oxide on the metal oxide interface. This mechanical stress causes movement of the quasi fluid oxide layer and as consequence repulsive interactions between individual pores. The result of these repulsive interactions is a hexagonally ordered layer of pores. However the stability of the asymptotic solutions of the Field Assisted Flow model proposed by Singh, *et. al*[8] is put to question by the work of Gomez and Paris[9]. This model and our previous model assume the Field Assisted Dissolution approach. The main principle of the Field Assisted Dissolution is that reaction rates during anodization depend on the electric field across the layer and particularly at the metal oxide and oxide solution interface. The electric field is stronger at the bottom of a pore than at its top due to the difference in oxide thickness while the potential difference is constant. Hence a dynamic equilibrium between metal oxidation and oxide dissolution can be achieved.

2. Methods

To model the process of anodization we employ a probabilistic, three dimensional, asynchronous cellular automaton approach with periodic boundary conditions along two axes and fixed boundary conditions along the third axis. We use the 3D Moore neighborhood for a given cell. For the cell update however we select at random a pair of neighbors and update them according to a formal reaction scheme depending on their state. The basic implementation of this model is the same as our former model[10]. We employ 4 cell states: *M* the state corresponding to the metal, *OX* - to the oxide, *EF* - to the walker representation for electric field in the oxide that will be detailed further, *S* - to solvent. The

rules we use are as follows:

$$M + S = OX + OX \quad (1)$$
$$EF + S = S + S \quad (2)$$
$$EF + S = OX + S \quad (3)$$
$$M + OX = EF + EF \quad (4)$$
$$M + OX = M + EF \quad (5)$$
$$EF + OX = OX + EF \quad (6)$$
$$S + OX = OX + S \quad (7)$$
$$S + EF = EF + S \quad (8)$$
$$(9)$$

The differences between the two models can be described as the anion state *A* is discarded and unified with the *OX* state. The functionality of the discarded state is emulated by rule 4 and the drift of *A* is modeled by the drift of *OX*. The rules 1- 8 reflect chemical reaction-like behavior (1 to 5), model the electric field (6) and perform surface reorganization (7 and 8). The coding of the rules is the same as in [10] and explained in a greater detail here.

Reaction-like rules can be viewed as descriptions of the well known electrochemical processes taking place during anodization. These are:

1) Passivation of the active metal (rule 1)
2) Dissolution of the oxide in high electric field (rule 2)
3) The potential vanishes in the solution thus electric field walkers disappear when stepping into the solution (rule 3)This can be viewed as hole regeneration equivalent in a sense to anion incorporation in our previous model [10].
4) Oxidation of the active metal in contact with oxide side (rule 4).

The final reaction-like rule creates electric field walkers that we use to model the presence of electric field in the oxide layer without oxidizing the metal itself. All of the reaction-like rules have a fixed probability during simulations. Additionally the probabilities of rules 2 and 3 sum to 1 as do the probabilities of reactions 4 and 5.

We base our simulations on the Field Assisted Dissolution model presented by Parkhutik and Shershulsky[11]. Assuming the same boundary conditions apply the Poisson equation which describes the electric field is reduced to a Laplace equation. The steady state diffusion equation with similar boundary conditions is also reduced to the Laplace equation. The diffusion equation can be solved by the random walk of particles. Hence we use the random walk approach of electric field particles to model the influence of the electric field in the oxide layer. A detailed explanation of how we account for the electric field presence is given in detail in our previous work[10]. The diffusion-like rule 6 governs the random walk of electric field walkers and therefore the effective displacement of holes inside the oxide layer. This rule differs from reaction-like rules in the sense that it conserves the species involved leading to their redistribution. Its probability is set to 1 if the correct particles encounter one another. Let us emphasize again that the hole movement viewed as a free oxide transport in walker medium is a dual phenomenon to walker movement. The electric field represented by walker concentration gradient is both related by 1st Fick equation to walker flux and opposite hole flux. During anodization the electric field arises from the onset of the voltage between the metal and the solution. It is strongest at the thinnest metal-solution separation by the oxide layer and becomes weaker as the oxide layers gets thicker. The concentration gradient of the electric field walker is balanced in quasistationary conditions by the electric walker flux and then both can be thought to mimic the presence of electric field.

Rules 7 and 8 describe surface reorganization. They introduce surface tension and prevent excessive branching of the developing pores. Oxide particles are kept in place by bonds to other oxide particles. Let us recall that "oxide particles" means both *oxide* (hole) and *electric field* (walker) states. These rules remind diffusion but there is a difference with the diffusion rules mentioned previously. The decision if a swap of particles occurs is based on two factors: the neighborhoods of the locations chosen for the swap and a predefined probability,P_{BOND}, called "unbounding" probability. If the *solvent* site has more oxide neighbors then the *oxide* or *electric field* site a swap occurs. Conversely a swap may still occur but its probability is given by the power law:

$$P = P_{BOND}^{N} \quad (10)$$

where P_{BOND} is the "unbounding" probability and N is the difference of the number of neighbors between the sites. If an oxide site has no oxide neighbors it dissolves into a solvent site. The relative frequency of reaction-like, diffusion-like and surface reorganization events can be set at the beginning of the simulations.

In our simulations we use Nvidia Tesla GPGPU to decrease the time needed to complete them. The parallel CUDA architecture is a perfect match to the cellular automaton formalism. Individual threads can be assigned to portions of the grid of cells greatly increasing efficiency. The application of parallel techniques came with its own set of challenges and problems that are discussed in our previous paper[10]. Without going into much detail we can assure that there are no spurious correlations are aphysical behaviors introduced into the simulated system compared to a sequential code.

3. Results and Discussion

We present our results as a comparison between our previous model and the simpler one currently employed. We must stressed that the presented results were not selected to give layers of the same qualities. The two models do

not translate directly into one another due to the different treatment of *EF* particle generation. This comparison serves only to compare the two model results on a qualitative level. First let us analyze the profiles of *oxide-like* particles in both models presented in figure 1. The simulations were conducted in systems of size 100 by 100 by 200 in case of both models. A difference in the method of presenting the results is also made. Previous model side views and cross sections present only the interfacial *oxide-like* particles while the new model results portrait all *oxide-like* particles in the specified ranges. Both distributions of *oxide-like* particles

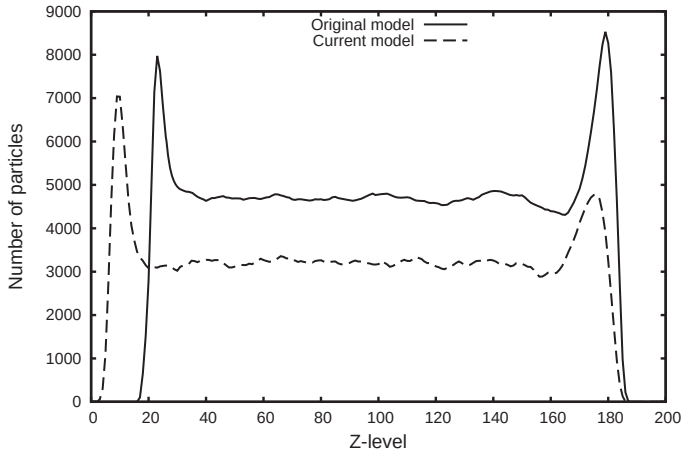


Fig. 1: Comparison of profiles of particles in respective simulations

have similar shapes with peaks appearing at both ends of the oxide layer and a large uniform area between the peaks. We have previously established that this shape of the *oxide-like* particle profile corresponds well to the shape of pores in a layer. The significant difference between the two profiles is the absolute amount of *oxide-like* particles present which indicates the two layers differ in porosity.

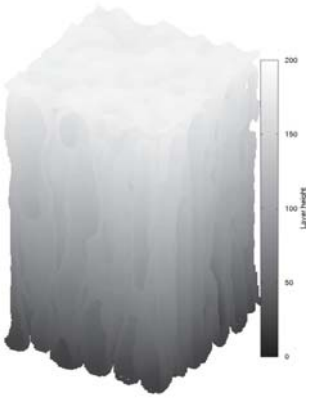


Fig. 2: Side view of the previous model simulation.

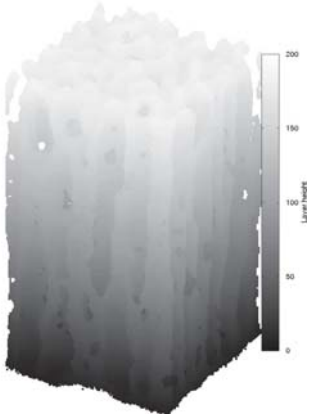


Fig. 3: Side view of the current model simulation.

In figures 2 and 3 side views of the simulated layers are presented. The layers are similar in their general appearance both having features of an organized porous structure. As expected from the particle profiles in figure 1 the current model exhibits a larger porosity manifesting in larger empty areas and a slightly more rugged structure compared to the previous simulations. The differences in structure are further examined in figures 4 and 5.

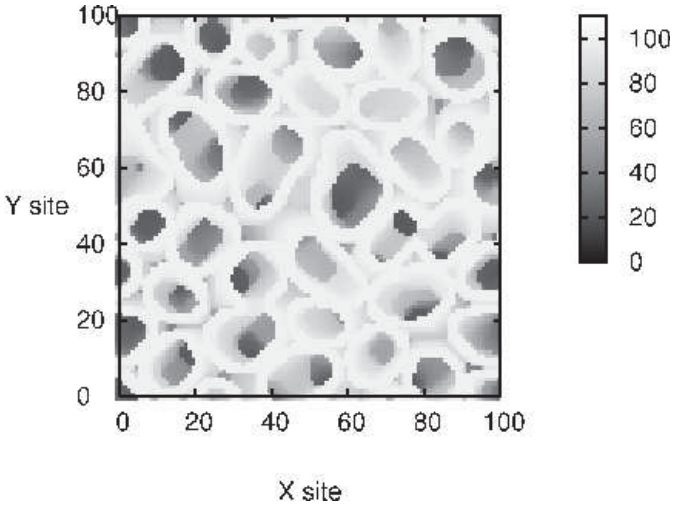


Fig. 4: Cross section of previous model simulated layer at half of the simulation box

A porous structure is clearly visible in both cross sections. The difference in the structures can be attributed to two factors. The first factor is lesser porosity of the layer obtained via the new model. This means that more space is filled with the solvent in comparison to the previous model layer. The second factor is the way the layers are presented. The tube-like appearance of pores in figure 4 is caused by picturing only interfacial *oxide-like* sites. In reality the "empty" spaces between individual pores also contain oxide material. To

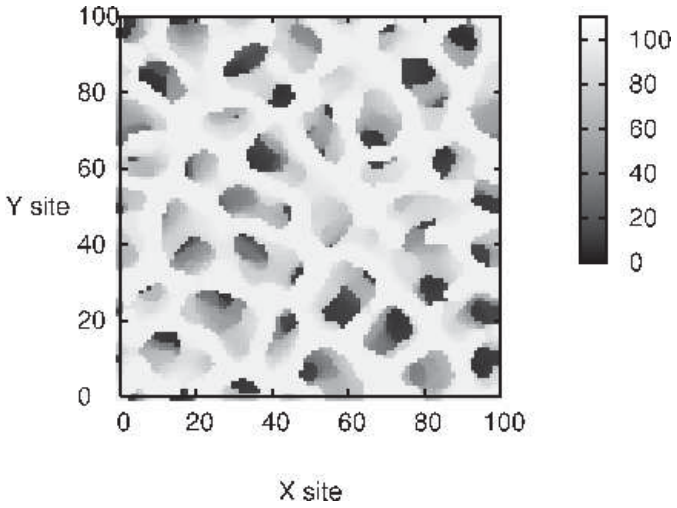


Fig. 5: Cross section of current model simulated layer at half of the simulation box

better analyze the these layers we made Fourier transforms of the cross sections to find what, if any, symmetry these structures poses.

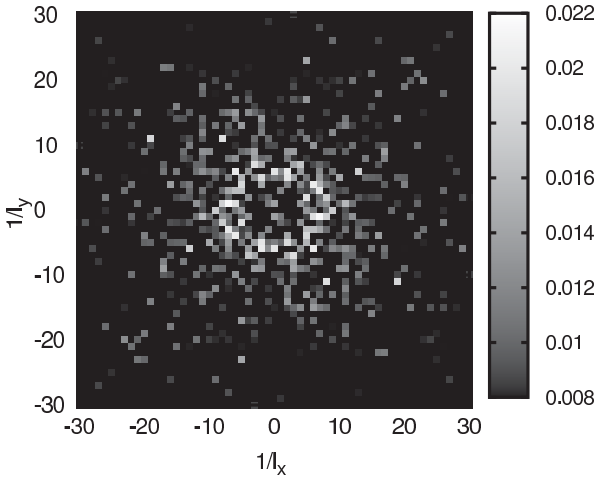


Fig. 6: 2D Fourier transform of the previous model cross section.

Both Fourier transforms show evidence of an organized structure. However the hexagonal pattern is much more clearly visible in the case of our previous work pictured in the center of figure 6. The transform of our current model also shows a organized structure but the symmetry is less pronounced. This behavior may be caused by the relatively little amount of simulation work done up to date. The regime for organized porous growth of the oxide layer may still not be properly explored. As a final way to analyze the structure we calculate the Fourier transform modulus with respect to the wave vector modulus. The results for

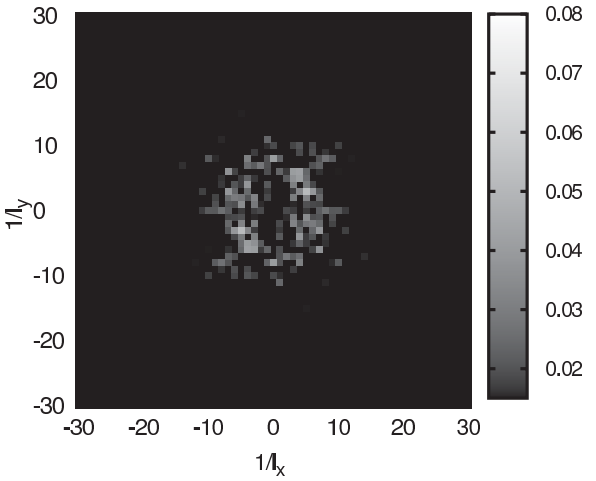


Fig. 7: 2D Fourier transform of the current model cross section.

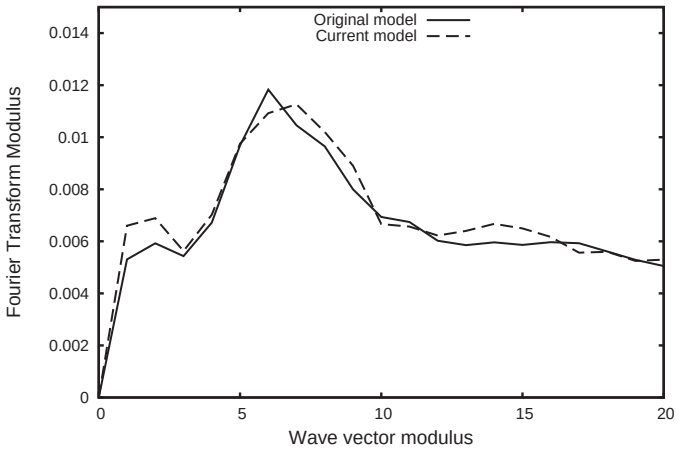


Fig. 8: Comparison of wave vector moduli in the respective simulations.

both simulations visible in figure 8 are similar with a broad maximum of the Fourier transform modulus. The maximum is slightly shifted to the right for the new model, which signifies the distance between pores is smaller than in the case of simulations of the previous model. This result is in line with previous result as increased porosity requires a larger pore diameter, less distance between pores or a combination of the aforementioned factors.

4. Conclusions

We conclude that abandoning the *Anion* state of our previous model did not significantly alter the qualitative properties of layers obtained in simulations. This study finds that hole diffusion is a viable modeling alternative to actual species diffusion. The currently used model is less computationally costly than its predecessor and offers the

same capabilities in terms of obtained layer morphology. We acknowledge the slightly lower quality of our current results in comparison to our previously reported work. We believe this lowered quality is the result of not fully exploring model parameters and that quality will improve along with the works progress. We are aware however of the limitations of this approach. Its implementation was only possible because of the relatively simple nature of *Anion* interactions. The usage if such simplifications may be impossible in case of more complex interactions reflecting more complex chemical interactions such as reactions between anions and oxide in the layers that were discarded in both of the analyzed models. Many of the qualitative features of the layers show a dependency on only a handful of model parameters. This makes analysis and relating the model parameters to real life parameters difficult and potentially limits the number of parameter combinations that may be simulated.

References

[1] Keller, F. and Hunter, M.S. and Robinson, D.L.: Journal of the Electrochemical Society **100** (1953) 411–419
[2] Mutalib Jani A., Losic D., Voelcker N., Progress. Mater. Sci. **58** (2013) 636–704
[3] Masuda, H. and Fukuda, K.: Science **268** (1995) 1466–1468
[4] Li, A. P., et al.: Journal of Applied Physics **84** (1998) 6023–6026
[5] Jessensky, O. and Muller, F. and Gosele, U.: Journal of the Electrochemical Society **145** (1998) 3735–3740
[6] Hoar, T. P. and Mott, N. F. : Journal of Physics and Chemistry of Solids **9** (1959) 97–99
[7] Garcia-Vergara, S.J. and Iglesias-Rubianes, L. and Blanco-Pinzon, C.E. and Skeldon, P. and Thompson, G.E. and Campestrini, P. : Proc. R. Soc. A **462** (2006) 2345–2358
[8] Singh, G. K. and Golovin, A. A. and Aranson, I. S. and et al. : Europhys. Lett. **70** (2005) 836-842
[9] Gomez, H. and Paris, J. : Phys. Rev. E. **83** (20011) 046702-1;046702-11
[10] Bartosik, L. and Stafiej, J. and Di Caprio, D. : Cellular Automata Lecture Notes in Computer Science **8751** (2014) 176-186
[11] Parkhutik, V. P. and Shershulsky, V. I. : J.Phys. D: Apply. Phys. **25** (1992) 1258-1263

Glider Collisions in Hybrid Cellular Automata Composed of Rule 9 and 74

Fangyue Chen[1], **Bo Chen**[1], **Junbiao Guan**[1] **and Weifeng Jin**[2]
[1]Department of Mathematics, School of Science, Hangzhou Dianzi University, Hangzhou, Zhejiang, China
[2]School of Science, Shanghai University, Shanghai, China

Abstract— *Elementary cellular automata (ECA) rule 9 and rule 74, members of Wolfram's class II, could generate a host of gliders and complicated glider collisions by introducing the hybrid mechanism, which are much richer than those generated by ECA rule 110. A systematic analysis is carried out to show the simulation results.*

Keywords: Hybrid cellular automata, glider, collision, long reaction progress, swerve

1. Introduction

Among infinitely many cellular automata (CA), the ones exhibiting plentiful gliders and glider guns have received special attention. They display complex behaviors via the interactions of gliders, and have the potential to emulate universal Turing machines. Gliders are one of important features inherited in complex CA rules. In 1991, Boccara et al produced a list of gliders and discussed the existence of glider gun [1]. In 1997, Hanson and Crutchfield applied finite state machine language representation to study defect dynamics in one-dimensional CA and derived motion equations of filtered gliders [2]. Basing on previous work, Wolfram investigated glider collisions with long period and derived several filters for detecting gliders and defects [3]. Martin designed an algebraic group to represent collisions between basic gliders [4]. There are many more research on gliders, see [5-10] and references therein.

Notably, Elementary cellular automata (ECA) rule 110 has received special attention due to the existence of a great variety of gliders in its evolution space. It is worth mentioning that Cook proved ECA rule 110 is universal via simulating a cyclic tag system [8]. As the study of unconventional computation, Martínez et al highlighted the dynamical characteristics of gliders in ECA rule 110 and rule 54 [11-14]. However, as gliders in ECA rule 54 are less complicated than those in ECA rule 110, so far no literature has proven that rule 54 is universal.

For an one-dimensional CA, when the evolution of all its cells is hinging on the one and only global function, it is called uniform, otherwise it will be called hybrid, i.e. hybrid cellular automata (HCA). Denoted by HCA(N,M), HCA rule composed of ECA rule N and ECA rule M, is specified to obey the rule of ECA N at odd sites of the cell array and obey the rule M at even sites of the cell array [15,16].

In spite of the HCA are endowed with simple hybrid rules and evolve on the same square tile structures, the evolution of HCA may exhibit rich dynamical behavior through local interactions. On the basis of a great amount of computer simulations and empirical observations, we found that the HCA(9,74) could generate plentiful gliders which are more complicated than those in ECA rule 110 or ECA rule 54. In order to gain further insights into rich dynamics generated by HCA(9,74), we present a systematic analysis of computational glider behaviors in present paper. By designing a single filter, the gliders are easy to be distinguished from each other. Thereafter, the gliders are classified and coded according to diverse configurations clarified by shift speed and volume. In addition, many simulation results are obtained via detailed analysis of collisions generated by two gliders.

The set of bi-infinite configurations is denoted by $S^Z = \cdots S\times S\times S\cdots$ and a metric d on S^Z is defined as$d(x,\overline{x}) = \sum_{i=-\infty}^{+\infty} \frac{1}{2^{|i|}} \frac{\widetilde{d}(x_i,\overline{x}_i)}{1+\widetilde{d}(x_i,\overline{x}_i)}$, where $S = \{0,1,\cdots,k-1\}$, $x,\overline{x} \in S^Z$ and $\widetilde{d}(\cdot,\cdot)$ is the metric on S defined as $\widetilde{d}(x_i,\overline{x}_i) = 0$, if $x_i = \overline{x}_i$; otherwise, $\widetilde{d}(x_i,\overline{x}_i) = 1$. Each ECA local rule can be endowed with a Boolean function [17]. For example, the Boolean function of ECA rule 9 is expressed as $N_9(x_{[i-1,i+1]}) = \overline{x}_{i-1}\overline{x}_i\overline{x}_{i+1} \oplus \overline{x}_{i-1}x_ix_{i+1}, \forall i \in Z$, where $x_i \in S$, "$\cdot$", "$\oplus$" and "$-$" denote "AND", "XOR" and "NOT" logical operations, respectively. The Boolean function of ECA rule 74 is expressed as $N_{74}(x_{[i-1,i+1]}) = x_{i-1}x_i\overline{x}_{i+1} \oplus \overline{x}_{i-1}x_{i+1}, \forall i \in Z$. For clarity, the truth table of two Boolean functions is displayed in Table 1.

x_{i-1}	x_i	x_{i+1}	N_9	N_{74}
0	0	0	1	0
0	0	1	0	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	0	0

Table 1: The truth table of two Boolean functions.

Consequently, the Boolean function of HCA(9,74) is intro-

duced as $N(x_{[i-1,i+1]}) = \begin{cases} N_9(x_{[i-1,i+1]}) & \text{if i is odd,} \\ N_{74}(x_{[i-1,i+1]}) & \text{if i is even} \end{cases}$. An example of spatio-temporal pattern generated by HCA(9,74) with random initial configurations is illustrated in Fig.1.

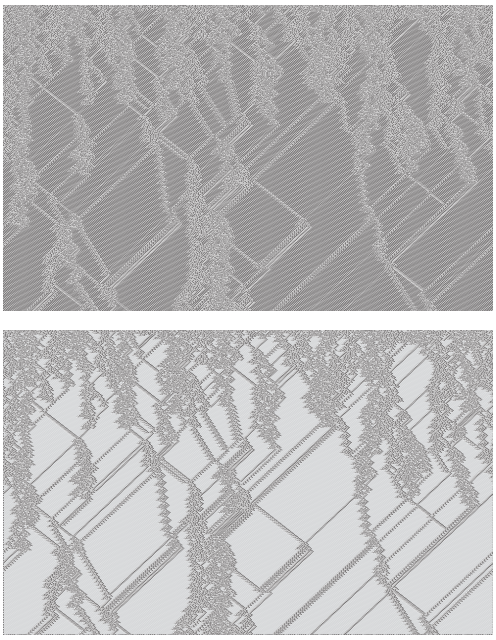


Fig. 1: (a) Spatio-temporal pattern of HCA(9,74), where white pixels are cells with state 0, and black pixels are cells with state 1. (b) The filter is applied to (a).

2. Classification and codings of gliders

Two types of gliders—original glider and composite glider—are specified in this section. As different characteristics of shift configuration, 22 original gliders (OGs) in HCA(9,74) are enumerated and coded. Furthermore, through analyzing the phenomenology of two different gliders collision, a great variety of composite gliders are obtained, which actually are combination of OGs. Each gliders is evolved under the uniform background of ether. We call the minimum component element of ether an ether unit. In particular, the background of ether in HCA(9,74) is composed of ether unit $\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & & \end{pmatrix}$. For convenience, the feature of ether unit is displayed in Table 2.

2.1 The catalogue of original gliders

Besides the ether unit, different shift configurations without any compositions of gliders are called OGs. By observing the dynamic behaviors illustrated in Fig. 1, 22 OGs are discovered. Their properties are listed in Table 2 where the fist column shows the labels of gliders, the second column shows velocity, and the third column indicates maximal and minimal sizes of gliders. As for a certain glider, the velocity is calculated from its shift number divided by its period. The plus sign denotes that the glider shifts to right and the minus sign denotes that the glider shifts to left. Figure 2 illustrates spatio-temporal patterns of OGs.

Original Gliders	Velocity	With
ether unit	-4/6	2-4
a	-2/6	6
b	-2/6	8
c	-4/12	6-10
d	-4/12	8-14
e	-4/12	8-14
f	4/10	10-16
g	2/5	10-12
g_2	2/5	12-14
g_3	2/5	8-10
h	-4/12	14-18
i	4/21	10-20
i_2	4/21	10-20
j	10/47	16-30
k	2/16	18-24
k_2	2/16	24-32
l	2/49	8-22
m	10/102	18-42
n	0/55	20-40
o	-2/17	14-22
p	2/27	16-24
p_2	6/81	36-52
q	4/46	36-46

Table 2: Characterizations of 22 OGs.

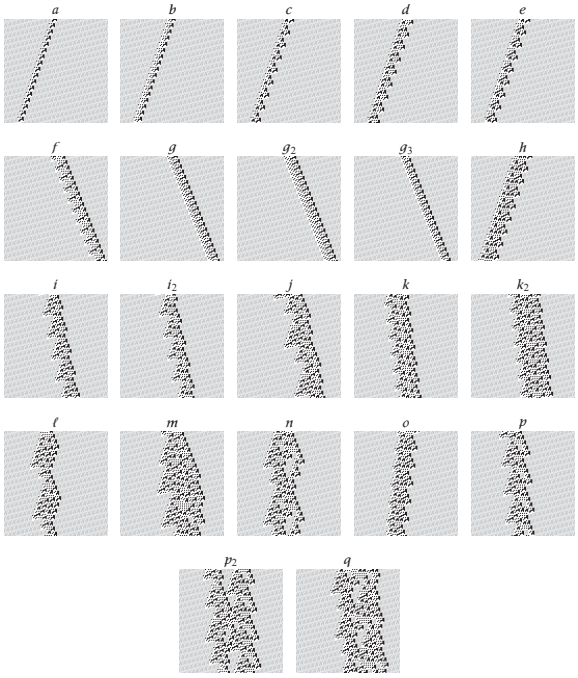


Fig. 2: The spatio-temporal patterns of 22 OGs

2.2 The catalogue of composite gliders

If two different OGs have the same shift velocity, they will be combined together without being divided by either

unit. We treat these shift configurations as new gliders, which are named as double composite gliders (DCGs). A series of DCGs are discovered by observing the collision phenomena between two different OGs. The properties of 22 DCGs are coded and listed in Table 3. The Figure 3 illustrates spatio-temporal patterns of DCGs. However, if a glider is composed of two or more same OGs, it is not regarded as a new composite glider. For example, the gliders $aa, bbb, cc, gg, ff, jj, \cdots$ are recognized as simple changes of OGs $a, b, c, g, f, j, \cdots$

Double composite gliders	Velocity	With
ab	-4/12	12-14
ac	-4/12	12-16
ad	-4/12	14-18
ae	-4/12	14-18
ba	-4/12	12-14
bd	-4/12	16-20
be	-4/12	16-20
ca	-4/12	14-16
cb	-4/12	16-18
da	-4/12	18-20
db	-4/12	20-22
dc	-4/12	18-24
de	-4/12	20-24
ea	-4/12	16-18
eb	-4/12	18-20
ec	-4/12	18-22
ed	-4/12	20-26
fg	4/10	22-26
g_2g	2/5	24
gf	4/10	24-26
hb	-4/12	26
hc	-4/12	24-30

Table 3: Characterizations of 22 DCGs

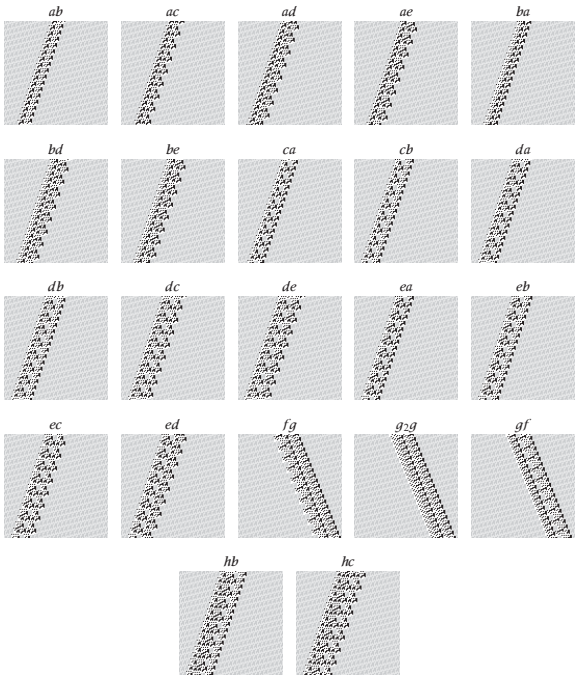


Fig. 3: The spatio-temporal patterns of DCGs

Likewise, if OGs and DCGs have the same shift speed, they also may be combined together without being divided by either unit. We also call these shift configurations the new gliders, which are named as multiple composite gliders (MCGs). More specifically, the properties of several MCGs are coded and listed in Table 4, where the fist column shows the label of gliders and the second column shows their composition formulae of different gliders.

Multiple composite gliders	Composition formule
abb	$ab + bb$
aeb	$ae + eb$
$aebbb$	$aeb + bbb$
bae	$ba + ae$
bad	$ba + ad$
$badb$	$bad + db$
beb	$be + eb$
$bhbb$	$bh + hbb$
cae	$ca + ae$
daa	$da + aa$
eaa	$ea + aa$
eab	$ea + ab$
$eaac$	$eaa + ac$
hbb	$hb + bb$

Table 4: Characterizations of several MCGs

3. Collisions between gliders

In order to explore the mathematical definition of collisions between two gliders, ether factor $E = (0,0,1,1,0,1,1,1,0,1,0,1,0,0,0,0,0,0,1,0,1,1)$ is introduced, which refers to arranging cell states of all rows of ether unit orderly. Obviously, E is determined by the shift characteristic and width of ether unit. For a pair of gliders with different velocity, the collision results may be diverse if there are different numbers of ether factors E between the gliders. However, the collision results are changing periodically according to the number of E between two gliders. Let Q indicate the period value. The form of each collision formula is set as $glider\ 1 \bigcup (QN + I)E \bigcup glider\ 2 \rightarrow \{result\}$, where $QN + I$ is the number of ether factor E between two gliders, N is natural number and $I = 1, 2, ...Q$. Then the following five observations can be obtained via the simulation results.

Observation 1: When the gliders have sizable periods and widths, their collision results may be very complicated, i.e., OGs j, l, m, n, p_2, q.

For example, the collision formulae between glider j and other OGs are quite diverse. Collision $j \leftrightarrow a$ has 7 cases. Collision $j \leftrightarrow b$ has 7 cases. Collision $j \leftrightarrow c$ has 1 case. Collision $j \leftrightarrow d$ has 10 cases. Collision $j \leftrightarrow e$ has 8 cases. Collision $j \leftrightarrow h$ has 14 cases. Collision $j \leftrightarrow i$ has 1 case. Collision $j \leftrightarrow i_2$ has 1 case. Collision $j \leftrightarrow k$ has 3 cases. Collision $j \leftrightarrow l$ has more than 15 cases. Collision $j \leftrightarrow m$ has more than 11 cases. Collision $j \leftrightarrow n$ has more than 11 cases. Collision $j \leftrightarrow o$ has 10 cases. Collision $j \leftrightarrow p$ has 8 cases. Collision $j \leftrightarrow q$ has 5 cases.

Observation 2: All OGs can be generated by collisions.

Since the DCGs and MCGs can be compounded from OGs, all gliders in the evolution space of HCA(9,74)

constitute a closed set in the sense of mutual collisions. In fact, different collisions would produce same or similar results, only one case is illustrated in Table 5 through the analysis of collision formulae of OGs.

original gliders	collision formulae
a	$g_2 \bigcup (2N+2)E \bigcup d \rightarrow \{a\}$
b	$k \bigcup (N+1)E \bigcup e \rightarrow \{b\}$
c	$g \bigcup (2N+2)E \bigcup e \rightarrow \{c\}$
d	$i_2 \bigcup (N+1)E \bigcup l \rightarrow \{d\}$
e	$i_2 \bigcup (5N+2)E \bigcup o \rightarrow \{e\}$
f	$g_2 \bigcup (2N+1)E \bigcup p \rightarrow \{f\}$
g	$m \bigcup (4N+4)E \bigcup d \rightarrow \{a,l,m,g\}$
g_2	$m \bigcup (4N+3)E \bigcup c \rightarrow \{e,a,p,g_2\}$
g_3	$k \bigcup (N+1)E \bigcup a \rightarrow \{a,a,d,g_3\}$
h	$g \bigcup (2N+2)E \bigcup p \rightarrow \{e,h,l,i\}$
i	$g_2 \bigcup (N+1)E \bigcup c \rightarrow \{i\}$
i_2	$i2 \bigcup (N+1)E \bigcup e \rightarrow \{k\}$
j	$f \bigcup (N+1)E \bigcup c \rightarrow \{j\}$
k	$p \bigcup (2N+2)E \bigcup e \rightarrow \{k\}$
k_2	$g_3 \bigcup (N+1)E \bigcup i \rightarrow \{k_2\}$
l	$g_3 \bigcup (N+1)E \bigcup b \rightarrow \{l\}$
m	$l \bigcup (5N+1)E \bigcup b \rightarrow \{m\}$
n	$q \bigcup (5N+1)E \bigcup a \rightarrow \{n,f\}$
o	$m \bigcup (2N+1)E \bigcup b \rightarrow \{a,o,g_3\}$
p	$i \bigcup (2N+2)E \bigcup c \rightarrow \{p\}$
p_2	$j \bigcup (8N+2)E \bigcup p \rightarrow \{p_2\}$
q	$n \bigcup (5N+3)E \bigcup b \rightarrow \{q\}$

Table 5: All OGs can be generated by collisions.

Observation 3: For collisions of two OGs, the phenomenon of long reaction process is discovered. Usually a lot of gliders can be generated after long reaction process.

By systematically analyzing the spatio-temporal patterns of collisions generated from OGs, a majority of collisions have long reaction progress. However, the reaction leads to regular bifurcation rather than evolving to an unordered state. For example, $f \bigcup (7N+3)E \bigcup m \rightarrow \{e,b,jj,f,g,f,g_2\}$ and $f \bigcup (7N+4)E \bigcup m \rightarrow \{d,a,b,b,b,b,f,f\}$ are described in Fig.4.

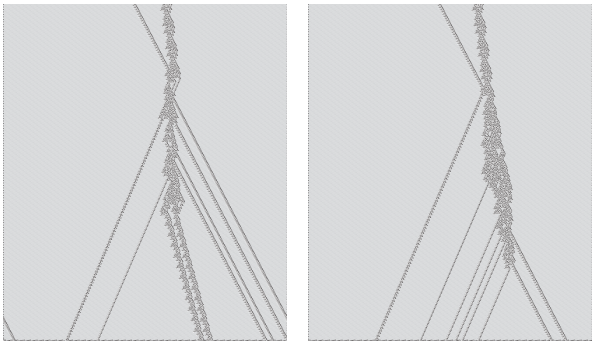


Fig. 4: The phenomenon of long reaction process. (a) Spatio-temporal pattern of collision j ↔ m with 3E distance, (b) Spatio-temporal pattern of collision j ↔ m with 4E distance.

Observation 4: For certain processes of collision, the phenomenon of "swerve" can be discovered.

The "swerve" is meaning that the reaction configuration suddenly converges toward left and forms the corresponding gliders. Loosely speaking, we think that the emerging reason of this phenomenon might be a result of shift characteristic of ether and hybrid evolutionary rule. Due to the visualization of "swerve", $g \bigcup (N+1)E \bigcup a \rightarrow \{a,e,hbb\}$ and $g \bigcup (N+1)E \bigcup b \rightarrow \{a,a,d\}$ are described in Fig.5.

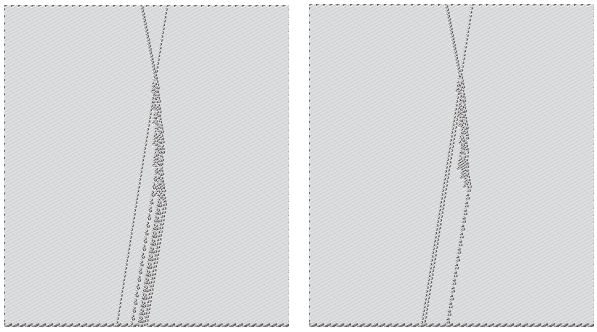


Fig. 5: The phenomenon of "swerve". (a) Spatio-temporal pattern of collision g ↔ a; (b) Spatio-temporal pattern of collision g ↔ b.

Observation 5: One gun and a series of solitons are discovered by collisions. as illustrated in Fig.6.

For example, the gun can be obtained by collisions: $g_3 \bigcup (4N+4)E \bigcup l \rightarrow \{gun\}$, $i \bigcup (3N+1)E \bigcup p_2 \rightarrow \{gun\}$ and so on.

A series of solitons are obtained as follows:
$f \bigcup (2N+1)E \bigcup e \rightarrow \{e,f\}$. $f \bigcup (2N+1)E \bigcup i \rightarrow \{i,f\}$.
$l \bigcup (10N+4)E \bigcup e \rightarrow \{e,l\}$. $l \bigcup (10N+2)E \bigcup d \rightarrow \{d,l\}$.
$l \bigcup (10N+7)E \bigcup d \rightarrow \{d,l\}$. $p \bigcup (2N+1)E \bigcup e \rightarrow \{e,p\}$.

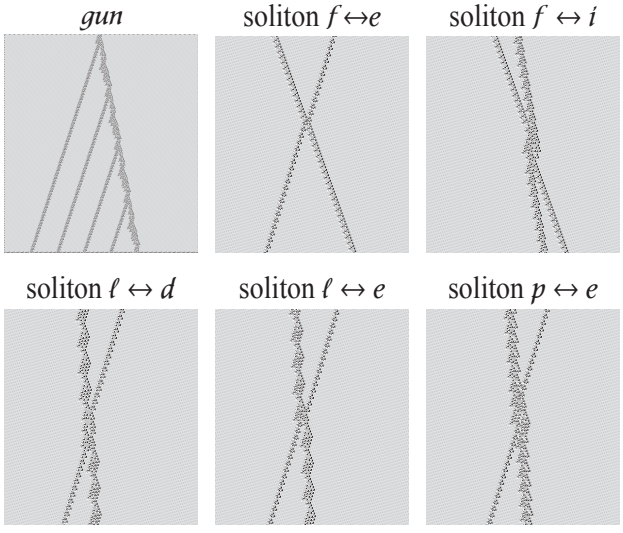


Fig. 6: The spatio-temporal patterns of solitons

Acknowledgments

This research was supported by the Natural Science Foundation of Zhejiang Province (Grant No. LY13F030014) and NSFC (Grants no. 11171084 and 60872093).

References

[1] N. Boccara, J. Nasser and M. Roger, "Particle Like Structures and Their Interactions in Spatio-temporal Patterns Generated by One-Dimensional Deterministic Cellular Automaton Rules," *Phys Rev A*, vol. 44, pp. 866-875, 1991.

[2] J. E. Hanson and J. P. Crutchfield, "Computational Mechanics of Cellular Automata: An Example," *Physics D*, vol. 103, pp. 169-189, 1997.

[3] S. Wolfram, *A New Kind of Science*, Champaign, Wolfram Media, 2002.

[4] B. Martin, "A Group Interpretation of Particles Generated by One-Dimensional Cellular Automaton, Wolfram's Rule 54," *Int. J. Mod. Phys. C*, vol. 11, pp. 101-123, 2000.

[5] M. Redeker, "Gliders and Ether in Rule 54," *Automata 2010: 16th International Workshop on Cellular Automata and Discrete Systems*, pp. 299-308, 2010.

[6] A. P. Goucher, "Gliders in Cellular Automata on Penrose Tilings," *J. Cell. Autom.*, vol. 7, pp. 385-392, 2012.

[7] I. Georgilas, A. Adamatzky and C. Melhuish, "Manipulating Objects with Gliders in Cellular Automata," *8th IEEE International Conference on Automation Science and Engineering*, PP. 936-941, 2012.

[8] M. Cook, "Universality in Elementary Cellular Automata," *Complex System*, vol. 15, pp. 1-40, 2004.

[9] J. B. Guan and F. Y. Chen, "Replication and Shift Representation of One-Dimensional Prototype Universal Cellular Automaton," *J. Cell. Autom.*, vol. 8, pp. 299-310, 2013.

[10] W. F. Jin, F. Y. Chen and G. R. Chen, "Glider Implies Li-Yorke Chaos for One-dimensional Cellular Automata," *J. Cell. Autom.* , vol. 9, pp. 315-329, 2014.

[11] G. J. Martínez, A. Andrew and H. V. McIntosh, "Phenomenology of Glider Collisions in Cellular Automaton Rule 54 and Associated Logical Gates," *Chaos, Solitons & Fractals*, vol. 28, pp. 100-111, 2006.

[12] G. J. Martínez, H. V. Mclntosh, J. C. S. T. Mora and S. V. C. Vergara, "Rule 110 Objects and Other Collision-based Constructions," *J. Cell. Autom.*, vol. 2, pp. 219-242, 2007.

[13] G. J. Martínez, A. Andrew, F. Y. Chen and L. O. Chua, "On Soliton Collisions Between Localizations in Complex ECA: Rules 54 and 110 and Beyond," *Complex Systems*, vol. 21, pp. 117-42, 2012.

[14] G. J. Martínez, A. Andrew and H. V. Mclntosh, "Complete Characterization of Structure of Rule 54," *Complex Systems*, vol. 23, 2014.

[15] K. Cattell and J. C. Muzio, "Synthesis of One-dimensional Linear Hybrid Cellular Automata," *IEEE Trans. Computers*, vol. 15, pp. 325-335, 1996.

[16] J. Binghama and B. Binghamb, "Hybrid One-Dimensional Reversible Cellular Automata are Regular," *Discrete Appl. Math.*, vol.155, pp. 2555-2566, 2007.

[17] J. B. Guan, S. W. Shen, C. B. Tang and F. Y. Chen, "Extending Chua's Global Equivalence Theorem on Wolfram's New Kind of Science," *Int. J. Bifur. Chaos*, vol. 17, pp. 4245-4259, 2007.

SESSION

SCIENTIFIC COMPUTING, ALGORITHMS AND APPLICATIONS + STATISTICAL METHODS

Chair(s)

TBA

Slow MHD Waves in the Turbulent Magnetized Plasma

G. Jandieri
Special Department/Physics Department, Georgian Technical University, Tbilisi, Georgia

Abstract - *Peculiarities of the magnetosonic waves in weakly-ionized ionospheric E-region with randomly varying spatial-temporal turbulent plasma parameters are considered. Statistical characteristics: variances of both the directional fluctuations causing curvature of the phase surface and frequency fluctuations leading to the broadening of the temporal power spectrum of scattered magnetosonic waves are investigated analytically and numerically. Energy exchange between fast and slow magnetosonic waves, and turbulent plasma is analyzed on the bases of the stochastic transport equation using the ray (- optics) approximation. Experimental data have been used in numerical calculations for the anisotropic Gaussian correlation function of the density fluctuations. It is shown that the energy balance between magnetosonic wave-nonstationary medium is different in the direction of the wave propagation and perpendicular plane leading to the compression and stretching of the ray tubes.*

Keywords: ionosphere, magnetosonic waves, fluctuations, statistical characteristics, energy exchange

1 Introduction

The wavy processes in the upper atmosphere have both, hydrodynamic and electromagnetic nature. In the first class of waves belong the acoustic (sonic), gravitational and MHD (Alfven and magnetosonic) waves, while the second class of waves contains planetary Rossby waves and magnetogradient waves. The general dispersion equation was derived for the magneto-acoustic, magneto-gravity and electromagnetic planetary waves in the ionospheric E and F- regions [1]. In the ionosphere geomagnetic field generates small and medium-scale MHD waves.

The ionospheric observations reveal the electromagnetic perturbations in the ionospheric E - region known as the slow MHD waves [2]. These waves are insensitive to the spatial inhomogeneities of the Coriolis and Ampere forces and are propagated in the ionospheric medium more slowly than the ordinary MHD waves. Observations show [3] that during earthquakes, man-made explosions, magnetic storms, launching of space crafts, worldwide networks of ionospheric and magnetic observatories (located approximately along one latitude) in the E region ($70-150\,\text{km}$) of ionosphere besides the well-known wave modes the large-scale ($\lambda \sim 10^3 - 10^4$ km) ionospheric wave disturbances of electromagnetic nature are clearly registered propagating along the parallel around the Earth with high (supersonic) speeds (higher than $1\,\text{km}\cdot\text{s}^{-1}$) and having periods from several minutes to several hours. For the E-region the plasma component behaves like a passive impurity. The neutrals completely drag ions and the "ionospheric" friction between neutrals and ions can be neglect [1]. Therefore velocity of the neutral component $H_0/\sqrt{4\pi M N_n}$ is much lower than velocity of the plasma component $H_0/\sqrt{4\pi M N}$, where M is mass of ion (or molecule), N_n and N denote concentrations of the neutral particles and charged particles of the ionospheric plasma, respectively, H_0 is the geomagnetic field.

The features of low-frequency waves in homogeneous magnetized plasma are well studied now, however little attention is devoted to the investigation of statistical characteristics of MHD waves in the turbulent plasma flows. It was established that statistical moments of these waves substantially depend on a type of waves [4]. Therefore propagation of MHD waves in the turbulent plasma streams is of practical interest. Some peculiarities of statistical characteristics of MHD waves in randomly inhomogeneous plasma using the "freezing-in" turbulence approximation have been investigated [5].

Stochastic wave equation for the dynamo field is used for investigation of statistical characteristics of slow magnetosonic waves in a weakly-ionized ionospheric E-region. The influence of the spatial-temporal fluctuations of plasma density on the second-order moments is investigated in the ray (optics) approximation using the perturbation method. The mean energy flux densities are calculated for both "fast" magnetosonic (FMS) and "slow" magnetosonic (SMS) waves growing or decreasing due to parametric energy exchange with plasma flow.

2 Small oscillations of the Earth's ionosphere

Neglecting Rossby and acoustic-gravity waves we will interested in only perturbations having electromagnetic nature. It is well-know that these wave modes disappear if quasi-static and quasi-neutral conditions are fulfilled. Linearized equation of motion taking into account Halls' effect can be presented in the following form $\partial \mathbf{V}/\partial t = [\mathbf{j}\mathbf{H_0}]/\rho c$ [1], where $\boldsymbol{V}$ and

$\mathbf{H_0}$ are vectors of the fluid velocity and magnetic field, $\mathbf{j}$ is the current density, $\rho = \rho_n + \rho_{pl} \approx \rho_n = M\,N_n$, c is the speed of light. Limiting ourselves to the moderate and high latitudes, $\mathbf{H_0} = H_{0z}\,\mathbf{e_z}$, generalized Ohm's law beyond 80 km is [1]

$$\mathbf{E} = -\mathbf{w} + \frac{1}{\Omega_i}\left[\frac{\partial \mathbf{w}}{\partial t}\,\boldsymbol{\tau}\right] - \frac{1}{\nu_i}\frac{\partial \mathbf{w}}{\partial t}\ , \qquad (1)$$

where: $\mathbf{w} = [\mathbf{V}\cdot\mathbf{H_0}]/c$ is the dynamo field; $\boldsymbol{\tau} = \mathbf{H_0}/H_0$ is the unite vector along the geomagnetic field $\Omega_i = \eta\,\omega_t$ is modified by the ionization degree cyclotron frequency of ions, in the E (70–150km) and F (150–600km) regions $\eta = N/N_n \sim (10^{-8} \div 10^{-4}) << 1$ is the ionization degree of the ionospheric medium, $\omega_t = e\,H_{0z}/M\,c$ is the cyclotron frequency of ions, e denote electron, $\nu_i = \eta\,\nu_{in}$ is modified by the ionization degree collision frequency of ions with neutrals ν_{in}. Consequently, we naturally come to the consideration of slow (in the electrodynamics sense) long-period MHD waves in the ionosphere. Using $\mathbf{D} = \hat{\varepsilon}\,\mathbf{E} = 4\pi i\,\mathbf{j}/\omega$, neglecting displacement current, in the MHD approximation for the low-frequency wave processes we obtain:

$$\left\{\hat{\varepsilon}\,(1 - i\,s) - \frac{c^2}{V_a^2}\right\}\mathbf{w} = -\,i\,R\,\hat{\varepsilon}\left[\mathbf{w}\cdot\boldsymbol{\tau}\right]\ ,$$

$$\hat{\varepsilon} = \frac{c^2}{V_a^2}\,(1 \pm R - i\,s)^{-1}\ .$$

Generally slow MHD waves in the ionosphere will suffer dispersion due to Hall's effect $(\Omega_i \neq 0)$ and absorption due to ionic damping $\nu_i \neq 0$.

In the ionospheric E-region the plasma component behaves like passive impurity. The neutrals completely drag ions and the "ionospheric" friction between neutrals and ions can be neglected. Using Maxwell's equation $rot\,rot\,\mathbf{E} = -\,(4\pi/c^2)\cdot\partial\,\mathbf{j}/\partial t$, at $\mathbf{w} \sim \exp(ik_x x + ik_z z - i\omega t)$ in the frequency band $\omega \le \omega_i$ we get the wave equation

$$\frac{\partial^2 \mathbf{w}}{\partial t^2} + V_a^2\,rot\,rot\,\mathbf{w} = i\,R\,V_a^2\,rot\,rot[\mathbf{w}\cdot\boldsymbol{\tau}]\ , \qquad (2)$$

where: $R = \omega/\Omega_i$, $V_a = \sqrt{\eta}\,V_A$ and $V_A = H_0/\sqrt{4\pi MN}$ are the velocities of the Alfven wave in the neutral and plasma components of the ionosphere, respectively. The last term takes into account the Hall's effect. For small-scale and medium-scale processes neglecting latitudinal variations of the geomagnetic field we obtain the dispersion equation [1]

$$(\omega^2 - V_a^2\,k_z^2)\,(\omega^2 - V_a^2\,k^2) = \omega^2\,\frac{V_a^4}{\Omega_i^2}\,k_z^2\,k^2\ , \qquad (3)$$

where $k^2 = k_x^2 + k_z^2$ describes very slow, long-period MHD waves in the ionospheric E-region. If $k_z^2 << k_x^2$ and $\omega >> V_a\,k_z$ for the magnetosonic waves we get:

$$\omega = V_a\,k_x\left(1 + \frac{V_a^2\,k_z^2}{\Omega_i^2}\right)^{1/2}\ . \qquad (4)$$

From Eq. (4) follows that the characteristic horizontal wavelength $\lambda_0 = (2\pi V_a)/\Omega_i$ exists, determining the characteristic "length of dispersion" caused by the Halls' effect. At $\lambda_x > \lambda_0$ magnetosonic wave undergoes weak dispersion, and at $\lambda_x < \lambda_0$ the dispersion is strong. From Eq. (4) also follows that at small k_x frequency of the magnetosonic wave $\omega_{MS} = V_a\,k_x$ increases linearly with k_x.

Moreover the features of the fast and slow weakly damping electromagnetic planetary-scale electromagnetic waves (with a wavelength of 10^3 or more), generating in both the E- and F- layers of the ionosphere by the permanently acting factor – latitude variation of the geomagnetic field have been considered in [6]. It was shown that four normal modes: small-scale inertial waves, atmospheric whistles (helicons), fast large-scale electromagnetic planetary waves and slow Rossby-type waves are exist in the E– region of the ionosphere. Modified small-scale slow Alfven waves, fast large-scale electromagnetic planetary waves and ordinary slow planetary Rossby waves must be generated in the F– region of the ionosphere.

These waves having a weather forming nature propagate eastward and westward along the parallels cause substantial disturbances of the geomagnetic field (up to ten nanoTesla). The fast waves have phase velocities (1–5) $\mathrm{km\cdot s^{-1}}$ and frequencies $(10^{-1} - 10^{-4})\ \mathrm{s^{-1}}$, and the slow waves propagate with velocities of the local winds with frequencies $(10^{-4} - 10^{-6})\ \mathrm{s^{-1}}$ and are generated in the E- region of the ionosphere. In the E– region of the ionosphere $\Omega_0 << \omega_0$, for large-scale processes $(L \sim 10^3 - 10^4\ \mathrm{km})$, when latitude variation of the geomagnetic field $\mathbf{H_0}$ is not negligible, electromagnetic analogy of slow planetary Rossby waves are [7]:

$$\omega = \frac{c\,\beta_I}{4\pi e N}\,k_x\ , \quad \omega_H = \frac{c\,H_p}{4\pi e N}\,\frac{\sqrt{1 + 3\sin^2\theta}}{r_0}\,k_x\ , \qquad (5)$$

where: $2\Omega_0 = (e\eta/M\,c)\,H_0$, $\omega_0 = 7.3\cdot 10^{-5}\ \mathrm{s^{-1}}$, r_0 is the Earth's radius, $\beta_I = -\,2H_p\sin\theta_0/r_0$, $H_p = 3.2\cdot 10^{-5}\ \mathrm{T}$ is the value of geomagnetic field strength on the equator, θ_0 is the colatitude, $\theta = \pi/2 - \varphi$, φ – geographical latitude. Numerical calculations show that at $\theta = 45^0$ in the interval of heights (90 –150) km phase velocity of waves $C_H = \omega_H/k_x$ vary from 4

to $1.4\,\mathrm{km\cdot s^{-1}}$ at night, and from 400 to $800\,\mathrm{m\cdot s^{-1}}$ in the daytime. Periods are in the interval of (1.5–6) h in the daytime and (4–12) min at night. Perturbation of the geomagnetic field of these waves is 8 and 80 nT. The ground-based and satellite observations verify [8,9] the presence of slow (with phase velocities equal to local winds velocities), long period (a few days and more) and large-scale waves (with wavelength $\lambda \sim 10^3 - 10^4$ km) in the E- layer of the ionosphere at any seasons of the year. Hence, in the dispersion equations (4) and (5) the dependence of frequency ω on the wave vector k_x is the same.

3 Second order statistical moments of the magnetosonic waves in the turbulent plasma flow

Using the dispersion equations (4) and (5) and expressions of the instant frequency $\omega(\mathbf{r},t) = \partial\varphi/\partial t$ and the wave vector $\mathbf{k}(\mathbf{r},t) = |k|\,\mathbf{s} = -\nabla\varphi$, eikonal equation for low-frequency $\omega_0 << \omega_i$ magnetosonic waves in the ray (–optics) approximation [4] has the following form [5]:

$$\omega - (\mathbf{k}\,\mathbf{V_0}) = \pm V_a\, k\,. \qquad (6)$$

Here upper and lower signs correspond to the FSM wave (phase velocity $\omega/k > V_0$) and SMS wave (phase velocity $\omega/k < V_0$); the group velocity is $\mathbf{V_{gr}} = \mathbf{V_0} \pm V_a\,\mathbf{s}$, the mean constant macroscopic velocity of a turbulent plasma flow $\mathbf{V_0}$ substantially exceeds the root-mean-square velocity of turbulent mixing $V_0 >> \sqrt{<V_1^2>}$; $V_1(\mathbf{r}, t)$ represents small turbulent pulsations of the macroscopic velocity. Vector $\mathbf{V_0}$ is directed along the external magnetic field $\mathbf{B_0}$ locating in the *XZ* plane (principle plane) of the Cartesian coordinate system having angle of inclination θ with respect to the *Z* - axis, We suppose that density fluctuations of the neutral particles exceed velocity pulsations $V_1/V_a << N_1/N_n << 1$. Frequency and wave number of the magnetosonic waves with smooth fluctuating plasma parameters satisfy the conditions [10]: $k_0\, l >> 1$, $\omega_0\, T >> 1$ and $\omega_0\, l/V_0 >> 1$ (l and T are characteristic spatial-temporal scales of plasma irregularities). Velocity and density of the neutral particles can be expressed as sum of the regular and fluctuating components which are slowly varying random functions of the spatial coordinates and time $\mathbf{V}(\mathbf{r},t) = \mathbf{V}_0 + \mathbf{V}_1(\mathbf{r},t)$, $N_n(\mathbf{r},t) = N_0 + N_1(\mathbf{r},t)$. Should be emphases that in the turbulent plasma flow, contrary to the Alfven wave, directional fluctuations of the group velocity of magnetosonic wave and, hence, the unit vector $\mathbf{s}$ lead to the compression and stretching of the ray tubes caused by the amplitude fluctuations [10]. Substituting phase $\varphi(\mathbf{r},t) = \varphi_0(\mathbf{r},t) + \varphi_1(\mathbf{r},t)$ ($\varphi_1 << \varphi_0$, $\varphi_0 = \omega_0 t - k_0 z$ is the regular phase) into Eq. (1), we obtain stochastic transport equation for the phase fluctuation:

$$\frac{\partial\varphi_1}{\partial t} + (\mathbf{V_{gr}}\cdot\nabla\varphi_1) = \mp\frac{1}{2}\,k_0\,V_{a\,0}\cos\theta\,\frac{N_1}{N_0}\,, \qquad (7)$$

where: $k_0 = \omega_0/V_2$, $V_2 \equiv V_{gr\,z} = V_0\cos\theta \pm V_{a\,0}$, $V_{a\,0} = B_0/\sqrt{4\pi N_0 M}$. This equation easily solved using method of characteristics:

$$\varphi_1(\mathbf{r},t) = m\int_0^L dz'\,N_1(x', y, z', t')\,, \qquad (8)$$

where: $x' = x - (z - z')\,V_0\sin\theta/V_2$, $t' = t - (z - z')/V_2$, $m = \mp\,k_0/2\,N_0\,\alpha$, $\alpha = V_2/V_{a0}$, L is the distance covered by magnetosonic wave in the turbulent plasma flow.

Temporal spectrum of scattered waves is applied in diagnostics of the ionospheric plasma. The variance of an instant frequency $<\omega_1^2>$ determines the broadening of the temporal power spectrum easily measuring by experiment. It can be obtained from the correlation function of the phase fluctuations multiplying integrand on the factor ω^2. The obtained statistical characteristics of low-frequency slow magnetosonic waves in the E-region of ionosphere are valid for arbitrary correlation function of the density fluctuations of the neutral components.

The most important problem of wave propagation in a nonstationary medium is the energy exchange between wave and medium [4] that the transfer equation for the wave amplitude E or log-amplitude $\chi = \ln E/E$ is derived from the compatibility condition of set of equations in the ray (optics) approximation. In weakly collision plasma, neglecting ionization processes, we can proceed from the condition of the existence of adiabatic invariant – the ration of the energy wave packet to its frequency. In the quasi-static case compression and stretching of ray tubes have main influence on the log-amplitude fluctuations Specific features arise at propagation of the magnetosonic waves in the turbulent nonstationary plasma with chaotically varing parameters. The solution of this problem is based on the calculation of the mean energy flux density $\mathbf{S} = \eta\,E^2\,\mathbf{V_{gr}}$ [11] Neglecting dissipation processes in the ray (-optics) approximation amplitude E satisfies the transport equation [4]:

$$\frac{\partial}{\partial t}(\eta\,E^2) + div(\mathbf{V_{gr}}\cdot\eta\,E^2) = -\frac{\partial\varepsilon}{\partial t}\,E^2\,. \qquad (9)$$

Here: $\eta = \frac{1}{\omega}\frac{\partial}{\partial\omega}(\omega^2\varepsilon_{xx})$ is the coefficient between the energy density and E^2 [4], $\varepsilon_{xx} = \frac{c^2}{V_a^2}\frac{(\omega - \mathbf{k}\,\mathbf{V_0})}{\omega^2}$ is the component of dielectric permittivity obtaining for a moving plasma applying method submitting in [11]. Fluctuating

parameters in the Eq. (9) for magnetosonic waves have the following form:

$$\eta_1(\mathbf{r},t)=\frac{c^2\,(1\pm\alpha)}{V_2^2\,N_0}N_1(\mathbf{r},t)\,,\quad \frac{\partial\,\varepsilon_1(\mathbf{r},t)}{\partial t}=\frac{c^2}{N_0\,V_2^2}\frac{\partial\,N_1(\mathbf{r},t)}{\partial t}$$

At first we consider energy exchange between the magnetosonic wave and turbulent plasma flow in the direction of unperturbed wave propagation (along the Z axis); vector $\mathbf{S}$ is located in the XZ principle plane. Unit vector and energy flux density we will present as: $<\mathbf{S}>=<\eta\,E^2\,\mathbf{V_{gr}}>=\mathbf{S_0}+$ $+<\mathbf{S}>_1+<\mathbf{S}>_2$. Substituting the last expression into transfer Eq. (9) we obtain first-order stochastic differential equation for the log-amplitude of the magnetosonic wave. The solution is:

$$\chi_1(\mathbf{r},t)=\frac{b_1}{V_2\,V_0\sin\theta}\int_0^L d\,z'\frac{\partial}{\partial\,t'}N_1(x',y,\,z',t')\,,\qquad(10)$$

where: $b=(2\pm\alpha)\,(1\pm2+3\alpha)$. It's evident $<S_z>_1=0$. Hence the mean energy flux density contains two terms $<S_z>=S_{z0}+<S_z>_2$:

$$S_{z0}=\pm\,E_0^2\,\frac{2c^2}{V_{a0}}\,,$$

$$<S_z>_2=E_0^2\frac{c^2L}{V_2^3\,N_0^2}\frac{b}{2\alpha}\int_{-\infty}^{\infty}d\rho_z\frac{\partial^2}{\partial\,\tau^2}\,W_N(\rho_x,\rho_y,\rho_z,\tau)\,,(11)$$

where: $\rho_x=(V_0\sin\theta/V_2)\,\rho_z$, $\rho_y=0$, ρ_x and ρ_y distances between observation points in the XY plane; density fluctuations are statistically homogeneous and stationary with the zero mean value $\tau=\rho_z/V_2$. Comparing Eq. (11) with the variance of the frequency fluctuations:

$$<\omega_1^2>=-m^2\,z\int_{-\infty}^{\infty}d\rho_z\frac{\partial^2}{\partial\,\tau^2}W_N(\rho_x,0,\rho_z,\tau)\,,\quad(12)$$

where $m=\mp\,\omega_0/2\alpha N_0\,V_2$, finally we have

$$<S_z>=\,E_0^2\,\frac{2\,c^2}{V_{a0}}\left(\pm1-b\,\frac{<\omega_1^2>}{\omega_0^2}\right).\qquad(13)$$

FMS wave has positive energy flow, SMS – negative anergy flow. Negative sign of the energy is connected with the thermodynamic equilibrium of plasma caused by both anisotropy of the task and nonstationarity of turbulent plasma parameters. Growth of the energy flow along the Z- axis means the energy transfer from medium to the wave and vice versa. The second-order statistical moments calculating in the ray (-optics) approximation not include diffraction effects. Meanwhile effect of the fluctuations accumulation for the wave parameters reveals more brightly at great distances from source.

Similar expression can be found for the energy flus density in the perpendicular direction, XOY plane. Calculations show that this second order moment would be more or less than in the direction of an incident wave propagation leading to the compression and stretching of ray tubes. Analyses show that the parametric energy exchange of magnetosonic waves in the turbulent plasma substantially depend on: the regular velocity of plasma flow, ionization degree and Alfven velocity in the ionospheric E- region, characteristic spatial-temporal scales and anisotropy factor of plasma irregularities, angles of magnetosonic waves propagation and the angle of inclination of plasma inhomogeneities with respect to the external magnetic field. Energy of the FMS wave decreases while for SMS wave is increases. This is well-known effect that can be explained by negative energy density of slow wave, energy flux is directed opposite to the Z - axis.

4 Numerical calculations

Observations of ionospheric irregularities detected by radio wave sounding of the lower *E*-region (altitudes near 100 km) have shown [12] that the average values of the speeds and horizontal spatial scales are being near 80 $\mathrm{m\cdot s^{-1}}$ and 30 km. The mean drift speed in the *E*- region of ionosphere is of an order 100-150 $\mathrm{m\cdot s^{-1}}$ depending on geomagnetic activity.

Large-scale anisotropic irregularities have been observed in the E - region of ionosphere. They generated due to wavy movements of an internal waves. Anisotropic coefficient of irregularities at $\chi<5$ is not connected with the geomagnetic field, but substantial elongation $\chi\geq10$ is defined by it. Velocities of irregularities movement is in the range of $40\div160$ $\mathrm{m\cdot s^{-1}}$; the most probable drift speed is ~100 $\mathrm{m\cdot s^{-1}}$ that is an agreement with other experimental data. The variance of neutral particles density fluctuations $\sigma_N^2=<N_1^2>/N_0^2$ was measured using pulse and radio-astronomical methods. Observations of the *E*-region have shown that characteristic linear scale of irregularities is about 1-2 km and $\sigma_N^2\sim10^{-4}$. Analytical and numerical calculations will be carried out for anisotropic Gaussian correlation function of density of the neutral components having in the principle plane following form [13]:

$$W_N(\mathbf{k},\omega)=\sigma_N^2\,\frac{l_\perp^2\,l_\parallel\,T}{16\pi^2}$$

$$\exp\left(-p_1\frac{k_x^2\,l_\parallel^2}{4}-\frac{k_y^2\,l_\perp^2}{4}-p_2\frac{k_z^2\,l_\parallel^2}{4}-p_3\,k_x\,k_z\,l_\parallel^2-\frac{\omega^2\,T^2}{4}\right),(14)$$

here:

$$p_1=(\sin^2\gamma_0+\chi^2\cos^2\gamma_0)^{-1}\left[1+(1-\chi^2)^2\sin^2\gamma_0\cos^2\gamma_0\right.$$

$\chi^{-2}\Big]$, $p_2 = (\sin^2\gamma_0 + \chi^2\cos^2\gamma_0)/\chi^2$, $p_3 = (\chi^2 - 1)\sin\gamma_0 \cdot \cos\gamma_0 / 2\chi^2$. This function contains anisotropy factor of irregularities $\chi = l_\square / l_\perp$ (ratio of longitudinal and transverse linear scales of plasma irregularities) and inclination angle γ_0 of prolate irregularities with respect to the external magnetic field, T is characteristic temporal scale of the turbulent plasma parameters fluctuations.

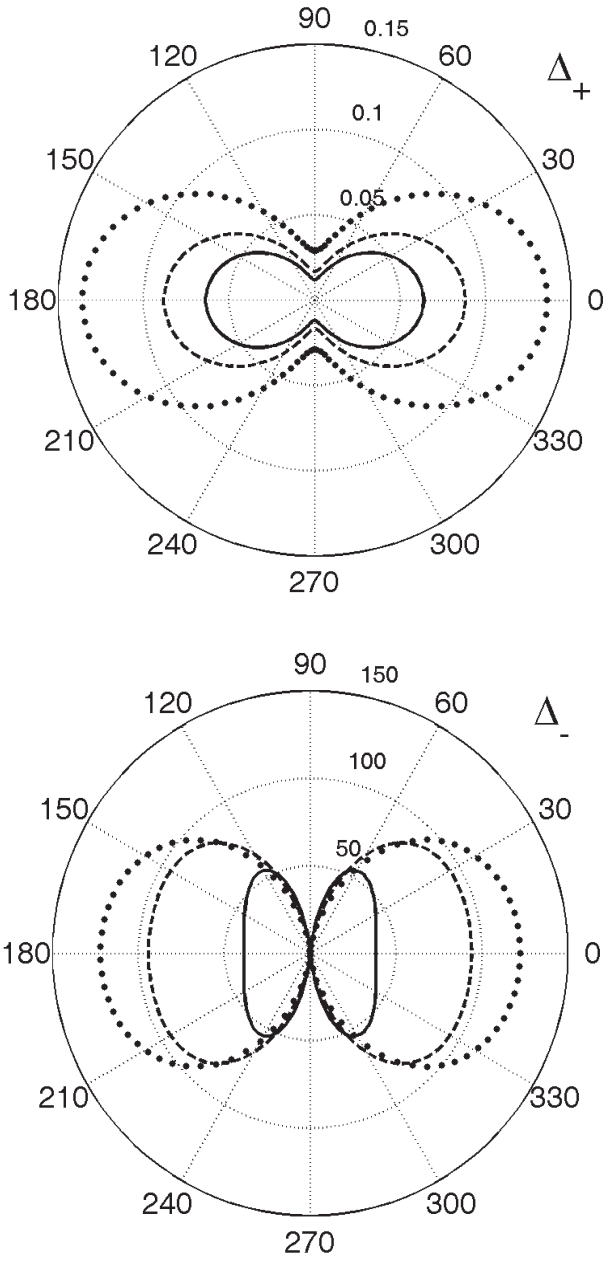


Fig. 1: represents the dependence of the normalized variance of the frequency fluctuations in the polar coordinate system for FMS wave (Δ_+ upper figure) and SMS wave (Δ_- lower figure) for different parameter G.

Numerical calculations have been carried out for scattered FMS wave ($V_0 = 30\ \mathrm{m\cdot s^{-1}}$, $V_{a0} = 50\ \mathrm{m\cdot s^{-1}}$, $c_0 = V_0 / V_{a0} = 0.6$) and SMS wave ($V_0 = 100\ \mathrm{m\cdot s^{-1}}$, $V_{a0} = 50\ \mathrm{m\cdot s^{-1}}$, $c_0 = 2$) in the turbulent plasma flow. Substituting (17) into (9) and using (8), for the variance of the phase fluctuations and the normalized broadening of the temporal spectrum ($\Delta = <\omega_1^2>/\omega_0^2$) in the polar coordinate system ($k_x = k\cos\varphi$, $k_y = k\sin\varphi$) we obtain:

$$<\varphi_1^2>_\pm = \sigma_N^2 \frac{L}{l_\parallel} \frac{(\omega_0 T)^2}{8\sqrt{\pi}} \frac{G^2 \tilde{M}_S^2}{\alpha^2 \eta_0 \eta_5^2 \chi^2},$$

$$\Delta_\pm = \sigma_N^2 \frac{L}{l_\parallel} \frac{1}{4\sqrt{\pi}} \frac{G^2 \tilde{M}_S^2}{\alpha^2 \eta_0^3 \eta_5^2 \chi^2} \left(1 + \frac{8\,\eta_3\, G^2 \tilde{M}_S^2}{\eta_0^2\, \eta_5^2} \cos^2\varphi\right), \quad (15)$$

where: $\eta_0 = (1 + p_2\, G^2 \tilde{M}_S^2)^{1/2}$, $\tilde{M}_S = (c_0 \cos\theta \pm 1)^{-1}$,

$\eta_2 = p_1 + p_2\, \Upsilon^2 \sin^2\theta - 4\, p_3\, \Upsilon \sin\theta$,

$\eta_3 = \left(\frac{1}{2}\, p_2\, \Upsilon \sin\theta - p_3\right)^2$, $\eta_4 = \left(\eta_2 - 4\frac{\eta_3}{\eta_0^2} G^2 \tilde{M}_S^2\right)^{1/2}$,

$\eta_5^2 = \eta_4^2 \cos^2\varphi + \frac{\sin^2\varphi}{\chi^2}$, $G = l_\square / V_{a0}\, T$ is the parameter of

turbulence. Increasing the ripple frequency band ($\nu \sim 1/T$) temporal power spectrum of scattered FMS and SMS waves in the ionospheric E- region broadens. Substituting Eq. (14) into Eq. (12) for the normalized variance of the frequency fluctuations we obtain

$$\Delta_\pm = \frac{\sqrt{\pi}}{2} \sigma_N^2 \frac{G^2 \tilde{M}_S^2}{\alpha^2 a_1 g^3 \chi} \frac{L}{l_\square} \left(1 + \frac{a_2^2 G^2 \tilde{M}_S^2}{2 a_1^2 g^2} X^2\right)$$

$$\exp\left[-\frac{X^2}{a_1^2}\left(1 + \frac{a_2^2 G^2 \tilde{M}_S^2}{4 g^2}\right) - \chi^2 Y^2\right], \quad (16)$$

where: $X = \rho_x / l_\square$, $Y = \rho_y / l_\square$ are the normalized distances between observation points, $\Delta \equiv <\omega_1^2>/\omega_0^2$,

$a_2 = 4\, p_3 - 2\, p_2 V_0 \sin\theta / V_2$, $a_3 = 1 + p_2\, G^2 \tilde{M}_S^2$,

$a_1 = \left(p_1 + p_2 V_0^2 \sin^2\theta / V_2^2 - 4\, p_3 V_0 \sin\theta / V_2\right)^{1/2}$,

$\tilde{M}_S = (c_0 \cos\theta \pm 1)^{-1}$.

Figure 2 represents the phase portraits of the normalized variance $\Psi_\pm = G^2 \tilde{M}_S^2 / \alpha^2 \eta_0 \eta_5^2 \chi^2$. At fixed anisotropy factor $\chi = 10$, $\gamma_0 = 10^0$; $\theta = 25^0$, varying parameter G = 8, 14 and 20 the phase portrait for the FMS wave represents concentrated circles, while for SMS wave it has the oval form.

This means that velocity of plasma flow has a substantial influence on the SMS wave. Estimations show that increasing parameter of turbulence G, FMS waves are dumping strongly than SMS wave due to transformation of the mean energy into scattered one. Taking into account that $< E >_+ / < E >_- =$, $\Gamma_\pm = 10^{-2} \int_0^{2\pi} d\varphi \ \Psi_\pm$, the ratio Γ_+ / Γ_- = 1.5; 4.6 and 14 at G = 10, 30 and 50, respectively.

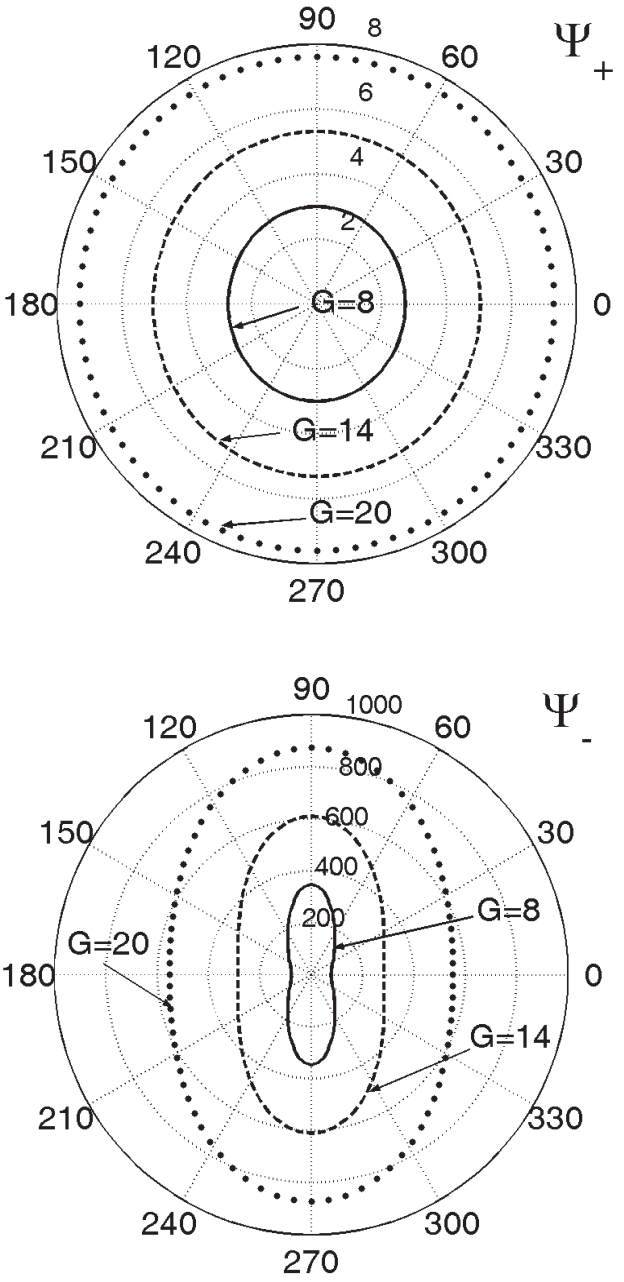


Fig. 2: Phase portraits of the normalized variance of the phase fluctuations for FMS wave (Ψ_+) and SMS wave (Ψ_-).

Quantitative description of the energy exchange of magnetosonic waves with plasma flow is analyzed at fixed anisotropy factor $\chi = 2$ and $\gamma_0 = 10^0$. Numerical calculations show that Increasing parameter of turbulence G = 0-2, the ration $Q_+ = S_{Z+} / S_{X+}$ of the energy flows of scattered FMS waves along and transversal directions (XOY plane) with respect to the incident wave propagation sharply increases reaching at $G \approx 2$ maxima: $Q_{+\,max}(\theta = 10^0) = 18$, $Q_{+\,max}(\theta = 30^0) = 5.9$, $Q_{+\,max}(\theta = 50^0) = 3.45$ Therefore in the ray (-optics) approximation at small angles θ energy flow of scattered FMS waves is directed forward (along Z-axis) due to the energy exchange with the turbulent plasma flow meaning the tension of the ray tubes along Z- axis. Increasing G = 2-30 parameter Q_+ decreases along Z- axis and the energy transfer in a transverse direction corresponding tension of the ray tubes in the XOY plane. Hence for FMS waves tension-compression process of the ray tubes occurs with respect to the Z- axis.

In the interval $\theta = 10^0 - 30^0$ increasing parameter of turbulence G = 0-2 the ration $Q_- = S_{Z-} / S_{X-}$ of the energy flow of scattered SMS waves slightly increases reaching at $G \approx 2$ maxima $Q_{-\,max}(\theta = 10^0) = 2.8$ and $Q_{-\,max}(\theta = 30^0) = 0.7$ (amplitude of Q_- approximately six times less than amplitude of Q_+). At $\theta = 40^0 - 50^0$ parameter Q_- decreases reaching minimum $Q_{-\,\min}(\theta = 50^0) \approx 0.2$. Increasing G = 2-40 in the interval $\theta = 10^0 - 30^0$ parameter Q_- at the initial stage slightly decreases and at $\theta = 40^0 - 50^0$ - increases. Therefore at small angles θ the flow of scattered SMS waves is directed forward. At big angles $\theta = 50^0$ flow is directed along X-axis (plane XOY) and then due to the parametric energy exchange energy flow of scattered SMS waves in the XOY plane decreases and increases along the Z-axis. In the ray (-optics) approximation this effect corresponds to the compression of the ray tubes. Hence for SMS wave compression-tension process of the ray tubes occurs with respect to the Z- axis.

5 Conclusions

Using the stochastic eikonal equation second order statistical moments are calculated for scattered fast and slow magnetosonic waves in the turbulent plasma for the arbitrary correlation function of electron density fluctuations. Variances of both the directional fluctuations of a unit vector describing curvature of the wave front and frequency fluctuations determining the broadening of the temporal power spectrum of scattered magnetosonic waves exceed corresponding statistical characteristics of Alfven wave. The reason is that these second

order moments of the magnetosonic waves are defined by regular velocity of plasma flow while for Alfven wave - by the turbulent pulsations of macroscopic velocity taking also into account that the group velocity is directed along the external magnetic field. All fluctuating characteristics of slow magnetosonic wave are increased approaching the mean regular velocity of plasma flow to the Alfven velocity. Therefore the condition $V_2 << V_{a0}$ should be fulfilled as the wavelength becomes very small and therefore application of the single liquid MHD approximation violates.

Energy balance of the magnetosonic waves and turbulent plasma flow has been analyzed on the bases of stochastic transport equation. Analytical expressions are obtained for the mean energy flux densities containing anisotropy caused by the flow motion, spatial-temporal parameters of plasma irregularities. It was shown that the energy flow in the perpendicular direction contains the energy flow in the forward direction (along propagation of magnetosonic wave) leading to the compression and stretching of the ray tubes in the ray (-optics) approximation. Parametric energy exchange of the magnetosonic waves and the turbulent plasma in the ionospheric E- region substantially depend on: regular velocity of plasma flow, ionization degree and Alfven velocity, characteristic spatial-temporal scales and anisotropy factors of plasma irregularities, angle of magnetosonic wave propagation with respect to the external magnetic field. Energy balance leads to the energy redistribution of scattered FMS and SMS waves in the forward and perpendicular directions, and hence compression-tension of the ray tubes in the ray (-optics) approximation.

All above derived formulae are valid for the angles θ not taking into account transformation of the magnetosonic wave into the Alfven wave [4,14] and imposes well-known restriction on the distance covered by wave in random medium $L / k_0 l^2 << 1$ [10,15]. However numerous investigations have shown that in this approximation phase statistical characteristics are a good approximation to reality at great distances. From the obtain formulas follow that in the quasi-static case log-amplitude wave fluctuations are determined by spatial structure of a turbulent flow, while the frequency fluctuations and energy exchange are caused by medium nonstationarity.

6 References

[1] G.V. Jandieri, A.G. Khantadze, A. Ishimaru, Zh.M. Diasamidze. "Electromagnetic oscillations of the Earth's upper atmosphere (review)," Ann. Geophys, Vol. 28, 1387-1399, 2010.

[2] V.M. Sorokin, G. V. Fedorovich. Physics of slow MHD waves in the ionospheric plasma, Nauka, Moscow, 1982 (in Russian).

[3] P.R. Fagundes, V.G. Pillat, M. J. A. Bolzan. "Observations of F layer electron density profiles modulated by planetary wave type oscillations in the equatorial ionospheric anomaly region," J. Geophys. Res, Vol. 110, 1302-1312, 2005.

[4] Yu.A. Kravtsov, L.A. Ostrovsky, N.S. Stepanov. "Geometrical optics of inhomogeneous and nonstationary dispersive media," Proc. IEEE, Vol. 62, 1492-1510, November 1974.

[5] G.V. Jandieri, V.G. Gavrilenko, A.A. Semerikov. "To the theory of magnetohydrodynamic waves propagation in turbulent plasma stream," Plasma Physics Reports, Vol. 11, 1193-1198, October 1985.

[6] G.D. Aburjania, K.Z. Chargazia, G.V. Jandieri, A.G. Khantadze, O.A. Kharshiladze. "On the new modes of planetary-scale electromagnetic waves in the ionosphere," Ann. Geophys., Vol. 22, 1203-1211, 2004.

[7] A.G. Khantadze. "Electromagnetic planetary waves in the Earth ionosphere," Geomag. Aeron., Vol. 333-335, 2002.

[8] Z.S. Sharadze, N.B. Mosashvili, G.N. Pushkova, L.A. Judovich. "Long-period wave disturbances in E-region of the ionosphere," Geomagn. Aeron., Vol. 29, 1032-1035, 1989.

[9] D.J. Cavalieri, R.J. Deland, R.F. Gavin. "The correlation of VLF propagation variations with atmospheric planetary scale waves," J. Atmosph. Terr. Phys., Vol. 36, 561-574, 1974.

[10] A. Ishimaru. Wave Propagation and Scattering in Random Media, Vol. 2, Multiple Scattering, Turbulence, Rough Surfaces and Remote Sensing, IEEE Press, Piscataway, New Jersey, USA, 1997.

[11] A.I. Akhiezer, I.A. Akhiezer, R.V. Polovin, A.G. Sitenko, K.N. Stepanov. *Electrodynamics of Plasma*, Nauka Moscow, 1974, (in Russian).

[12] R.A. Vincent. "Ionospheric irregularities in the E-region," J. Atm. Solar Terr. Phys., Vol. 34, 1881-1898, 1972.

[13] G.V. Jandieri, A.Ishimaru, V.G. Jandieri, A.G. Khantadze, Zh.M. Diasamidze. "Model computations of angular power spectra for anisotropic absorptive turbulent magnetized plasma," Progress In Electromagnetics Research, PIER, Vol. 70, 307-328, 2007.

[14] Yu.A Kravtsov, Yu.I. Orlov. Geometrical optics of inhomogeneous media, Moscow, Nauka, 1980 (in Russian).

[15] S.M. Rytov, Yu.A. Kravtsov, V.I. Tatarskii. Principles of Statistical Radiophysics. vol.4. Waves Propagation Through Random Media. Berlin, New York, Springer, 1989.

Nonsingular Robust Covariance Estimation in Multivariate Outlier Detection

Maximilian Wang[1], Rebecca Martin[1], and Guifen Mao[2]
[1]Lowndes High School, Valdosta, Georgia 31601, USA
[2]Department of Mathematics and Computer Science
Valdosta State University, Valdosta, Georgia 31698, USA

Abstract - *Rousseeuw's minimum covariance determinant (MCD) method is a highly robust estimator of multivariate mean and covariance. In practice, the MCD covariance estimator may be singular. However, a nonsingular covariance estimator is required to calculate the Mahalanobis distance. In order to fix this singular problem, we propose an improved version of the MCD estimator, which is a combination of the maximum likelihood estimator and the classical unbiased estimator. This estimator is nonsingular, robust, and as good as the MCD estimator with the same computational complexity.*

Keywords: Outlier Detection; Robustness; MCD Estimator; Mahalanobis Distance

1. Introduction

An outlier, by definition (Hawkins 1980), is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism. Outlier Detection is a classic topic in both the computer science and statistics communities (Aggarwal 2013, Hawkins 1980, and Chandola, V., Banerjee, A., and Kumar, V. 2009).

In recent years, the real-world has shown a great interest in outlier detection. Mining outliers has numerous applications, including credit card fraud detection, network intrusion, computer virus attack, clinical trials, severe weather prediction, voting irregularity analysis, athlete performance analysis, military surveillance for enemy activities, search for terrorism, Stock Selection, and many other data mining tasks.

In today's big data age, the real-world generates quintillions of bytes of data every single day. This steadily increasing size of dataset makes it more difficult to detect outliers. Especially for multivariate data, it is often impossible to see data structure by visualization (Filzmoser 2005). Outliers can influence the fit of a statistical model. The mean and covariance estimators are biased by the outliers. Due to this masking effect, the Mahalanobis distance no longer suffices. The multivariate outliers do not necessarily have the large Mahalanobis distances (Rousseeuw and Driessen 1999). This problem can be avoided by using distance based on robust estimators of multivariate mean and covariance.

Rousseeuw and Driessen (1999) proposed a fast algorithm to derive the robust MCD estimator. It is very popular and available in standard statistical software packages such as SAS and R.

However, this MCD estimator is not always invertible. In practice, we cannot calculate the Mahalanobis distance. Therefore the outlier detection procedure is infeasible.

We propose a new robust covariance estimator to modify the MCD method. This new estimator is positive definite (nonsingular) and robust.

2. Related Work

Mining outliers in high dimensions is very challenging. It is totally different from one-dimensional methods. In one dimension, the Euclidean distance is the common measurement for detecting outliers. For a multivariate dataset, the Euclidean distance is not a suitable distance for detecting outliers. Variables from a multivariate dataset are usually correlated in general. The Euclidean distance cannot catch the variation and correlations among different dimensions. The standard methods for multivariate outlier detection are based on the well-known Mahalanobis distance that takes into account the covariance matrix:

$$\boldsymbol{MD}(\boldsymbol{x_i}) = \sqrt{(\boldsymbol{x_i} - \boldsymbol{T})'\boldsymbol{S}^{-1}(\boldsymbol{x_i} - \boldsymbol{T})} \qquad (1)$$

for a p-dimensional observation $\boldsymbol{x_i}$ and $i = 1,2,\ldots, n$, where $\boldsymbol{T}$ is the mean (location) and $\boldsymbol{S}$ is the covariance (scatter) of the multivariate normal distribution. In theory, Euclidean distance is a special case of Mahalanobis distance, when the covariance matrix is the identity matrix.

In practice, $\boldsymbol{T}$ and $\boldsymbol{S}$ are unknown and need to be estimated in order to calculate the Mahalanobis distance. If $\boldsymbol{T}$ and $\boldsymbol{S}$ are estimated by their robust estimators respectively in (1), then the distance of (1) is called the robust Mahalanobis distance. For the multivariate normal model setting, this Mahalanobis distance approximately follows chi-square distribution with p degrees of freedom. The existence of the Mahalanobis distance depends on the inverse of the estimated covariance matrix. If the estimated covariance matrix is not invertible (singular), then the Mahalanobis distance does not exist.

In calculating the Mahalanobis distance, both estimators of mean and covariance are extremely sensitive to outliers. Outliers can skew the mean and covariance. Even the Mahalanobis distance itself can be affected by outliers. A robust procedure is necessary to make the statistical estimator less sensitive to outliers (Chawla and Gionis 2013).

Rousseeuw proposed a MCD robust estimator of multivariate location and scatter (Rousseeuw and Driessen 1999). The following is the main result of Rousseeuw's MCD method:

Theorem 1: (Rousseeuw and Driessen 1999) Consider a dataset $\boldsymbol{X_n} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\}$ of p-variate observations. Let $H_1 \subset \{1,2, \dots, n\}$ with $|H_1| = h$, and put

$$\boldsymbol{T_1} := \frac{\mathbf{1}}{\boldsymbol{h}} \sum_{i \in H_1} \boldsymbol{x_i} \text{ and } \boldsymbol{S_1} := \frac{\mathbf{1}}{\boldsymbol{h}} \sum_{i \in H_1} (\boldsymbol{x_i} - \boldsymbol{T_1})(\boldsymbol{x_i} - \boldsymbol{T_1})'$$

If $\det(\boldsymbol{S_1}) \neq 0$, define the relative distances

$$d_1(i) := \sqrt{(\boldsymbol{x_i} - \boldsymbol{T_1})' \boldsymbol{S_1}^{-1} (\boldsymbol{x_i} - \boldsymbol{T_1})} \text{ for } i = 1, \dots, n.$$

Now take $\boldsymbol{H_2}$ such that $\{d_1(i);\ i \in H_2\} := \{(d_1)_{1:n}, \dots, (d_1)_{n:n}\}$, where $(d_1)_{1:n} \leq (d_1)_{2:n} \leq \cdots \leq (d_1)_{n:n}$ are the ordered distances, and compute $\boldsymbol{T_2}$ and $\boldsymbol{S_2}$ based on H_2.
Then

$$\det(\boldsymbol{S_2}) \leq \det(\boldsymbol{S_1})$$

with equality if and only if $\boldsymbol{T_2} = \boldsymbol{T_1}$ and $\boldsymbol{S_2} = \boldsymbol{S_1}$.

Repeating the procedure of Theorem 1 yields an iteration process. The sequence

$$\det(\boldsymbol{S_1}) \geq \det(\boldsymbol{S_2}) \geq \det(\boldsymbol{S_3}) \geq \cdots$$

is nonnegative and hence must converge. If for some integer *m*,

$$\det(\boldsymbol{S_m}) = 0, \quad \text{or} \quad \det(\boldsymbol{S_m}) = \det(\boldsymbol{S_{m-1}}),$$

we stop.
For the implementation, select

$$h = \left[\frac{n+p+1}{2}\right].$$

The key idea here is to use about half of the dataset to estimate the mean and covariance matrix. This half dataset almost has no outliers. This is why the estimator of multivariate mean and covariance is robust and less sensitive to outliers.

In practice, if for some integer *m*, $\det(\boldsymbol{S_m}) = 0$, we stop. Here $\boldsymbol{S_m}$ is singular and thus it is not invertible. We cannot perform the outlier detection via the Mahalanobis distance. An improved nonsingular robust covariance estimator is needed in order to calculate the Mahalanobis distance.

3. A Nonsingular Robust Covariance Estimator

In this section, we propose a nonsingular covariance estimator. In theory, the covariance matrix is positive semi-definite, if it exists. However it is usually unknown and has to be estimated from the existing dataset. The estimated covariance matrix can be positive definite, or positive semi-definite, or indefinite due to numerical or estimation errors. However, we can prove that the maximum likelihood estimator and the unbiased estimator are always positive semi-definite in theory.

Let $\boldsymbol{x} = (\boldsymbol{x_1}, \dots, \boldsymbol{x_p})'$ be a multivariate random variable. Its variance is defined as

$$\boldsymbol{S} = \mathbf{Var}(\boldsymbol{x}) = \mathbf{E}(\boldsymbol{x} - \boldsymbol{T})(\boldsymbol{x} - \boldsymbol{T})'$$

Theorem 2: If the covariance matrix $\boldsymbol{S}$ exists, then it must be positive semi-definite.

Proof: For any constant vector $c = (c_1, \dots, c_p)'$, we have

$$\mathbf{Var}\,(c'\boldsymbol{x}) \geq 0.$$

$$0 \leq \mathbf{Var}(c'\boldsymbol{x}) = \mathbf{Cov}(c'\boldsymbol{x}, c'\boldsymbol{x}) = \mathbf{E}[(c'\boldsymbol{x} - \boldsymbol{E}(c'\boldsymbol{x})]^2$$

$$= \mathbf{E}\Big[c'\big(\boldsymbol{x} - \mathbf{E}(\boldsymbol{x})\big)\big(\boldsymbol{x} - \boldsymbol{E}(\boldsymbol{x})\big)'c\Big]$$

$$= c'\boldsymbol{E}\Big[\big(\boldsymbol{x} - \boldsymbol{E}(\boldsymbol{x})\big)\big(\boldsymbol{x} - \boldsymbol{E}(\boldsymbol{x})\big)'\Big]c = c'\boldsymbol{S}\,c.$$

Therefore $\boldsymbol{S}$ is positive semi-definite.

Following the same notation of Theorem 1, we define the sample mean:

$$\boldsymbol{T} := \frac{\mathbf{1}}{\boldsymbol{h}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{h}} \boldsymbol{x_i}$$

The maximum likelihood covariance estimator:

$$\boldsymbol{S^1} := \frac{\mathbf{1}}{\boldsymbol{h}} \sum_{i=1}^{h} (\boldsymbol{x_i} - \boldsymbol{T})(\boldsymbol{x_i} - \boldsymbol{T})',$$

and the unbiased covariance estimator:

$$\boldsymbol{S^2} := \frac{\mathbf{1}}{\boldsymbol{h-1}} \sum_{i=1}^{h} (\boldsymbol{x_i} - \boldsymbol{T})(\boldsymbol{x_i} - \boldsymbol{T})'.$$

There is no big difference in between $\boldsymbol{S^1}$ and $\boldsymbol{S^2}$ if the sample size *h* is large. In addition, they have the following nice properties.

Theorem 3: Both $\boldsymbol{S^1}$ and $\boldsymbol{S^2}$ are positive semi-definite.

Proof: The only difference between $\boldsymbol{S^1}$ and $\boldsymbol{S^2}$ is the front constants. For any constant vector $c = (c_1, \dots, c_p)'$, we have,

$$c' \sum_{i=1}^{h} (\boldsymbol{x_i} - \boldsymbol{T})(\boldsymbol{x_i} - \boldsymbol{T})'\, c = \sum_{i=1}^{h} c'(\boldsymbol{x_i} - \boldsymbol{T})(\boldsymbol{x_i} - \boldsymbol{T})'c$$

$$= \sum_{i=1}^{h} c'(\boldsymbol{x_i} - \boldsymbol{T})[c'(\boldsymbol{x_i} - \boldsymbol{T})]' = \sum_{i=1}^{h} [c'(\boldsymbol{x_i} - \boldsymbol{T})]^2 \geq 0.$$

This proves both $\boldsymbol{S^1}$ and $\boldsymbol{S^2}$ are positive semi-definite.

Now we combine $\boldsymbol{S^1}$and $\boldsymbol{S^2}$ to form a new positive definite covariance estimator: replacing the main diagonal of $\boldsymbol{S^1}$ by the main diagonal of $\boldsymbol{S^2}$. Let $\boldsymbol{S^*}$ be the new covariance estimator, we have the following result:

Theorem 4: The combined new covariance estimator $\boldsymbol{S^*}$ is positive definite.

Proof: We can view $\boldsymbol{S^*}$ is sum of two matrices:

$$\boldsymbol{S^* = S^1 + E}$$

where $\boldsymbol{E}$ is a diagonal matrix with elements:

$$e_{ii} = \frac{1}{h(h-1)} = \sum_{k=1}^{h} (x_{ik} - \bar{x}_i)^2$$

here $\bar{x}_i = \frac{1}{h}\sum_{k=1}^{h} x_{ik},\ i = 1, \dots, p.$

For any non-zero constant vector $c = (c_1, \dots, c_p)'$, we have,

$$c'\boldsymbol{S^*}c = c'\boldsymbol{S^1}c + c'\boldsymbol{E}c \geq c'\boldsymbol{E}c$$

$$= \frac{1}{h(h-1)}\sum_{k=1}^{h} (x_{ik} - \bar{x}_i)^2 c_i^2 > 0.$$

We conclude that $\boldsymbol{S^*}$ is positive definite. This implies that $\boldsymbol{S^*}$ is nonsingular (invertible).

Let's compare our new $\boldsymbol{S^*}$ estimator with the MCD estimator $\boldsymbol{S^1}$. Recall the Mahalanobis distance

$$d_{\boldsymbol{S}}(i) := \sqrt{(\boldsymbol{x_i} - \boldsymbol{T})'\boldsymbol{S}^{-1}(\boldsymbol{x_i} - \boldsymbol{T})} \text{ for } i = 1, \dots, n.$$

We have the following result.

Theorem 5: If $d_{\boldsymbol{s^1}}(i) \leq d_{\boldsymbol{s^1}}(j)$ for the MCD estimator $\boldsymbol{S^1}$, then we have the same order relationship under estimator $\boldsymbol{S^*}$,

$$d_{\boldsymbol{s^*}}(i) \leq d_{\boldsymbol{s^*}}(j)$$

for the new combined covariance estimator $\boldsymbol{S^*}$.

Proof: From the construction of the new combined covariance estimator S^*, there is no significant difference with the MCD estimator $\boldsymbol{S^1}$**.** The only difference is the constant factor on the main diagonal:

$$\frac{1}{h} \text{ and } \frac{1}{h-1}.$$

For large sample size, they are no difference (very close). Both matrices $\boldsymbol{S^1}$ and $\boldsymbol{S^*}$ converge to the true scatter matrix almost surely. Therefore, their inverse matrices $\boldsymbol{S^{1^{-1}}}$ and $\boldsymbol{S^{*^{-1}}}$ should be very close.

If $d_{\boldsymbol{s^1}}(i) \leq d_{\boldsymbol{s^1}}(j)$, then we have,

$$\sqrt{(\boldsymbol{x_i} - \boldsymbol{T})'\boldsymbol{S^{1^{-1}}}(\boldsymbol{x_i} - \boldsymbol{T})} \leq \sqrt{(\boldsymbol{x_j} - \boldsymbol{T})'\boldsymbol{S^{1^{-1}}}(\boldsymbol{x_j} - \boldsymbol{T})}.$$

Therefore, we have the same relationship if the sample size is large enough,

$$\sqrt{(\boldsymbol{x_i} - \boldsymbol{T})'\boldsymbol{S^{*^{-1}}}(\boldsymbol{x_i} - \boldsymbol{T})} \leq \sqrt{(\boldsymbol{x_j} - \boldsymbol{T})'\boldsymbol{S^{*^{-1}}}(\boldsymbol{x_j} - \boldsymbol{T})}.$$

Thus, we have

$$d_{\boldsymbol{s^*}}(i) \leq d_{\boldsymbol{s^*}}(j).$$

Theorem 5 indicates that the ordered statistics of Mahalanobis distance measurements have the same order under the MCD estimator $\boldsymbol{S^1}$ and the new combined covariance estimator $\boldsymbol{S^*}$. We summarize our main results into the following theorem.

Theorem 6: If we replace $\boldsymbol{S^1}$ in the MCD algorithm by the new combined covariance estimator $\boldsymbol{S^*}$, for this modified MCD algorithm, we have,

1. The outlier set is the same;
2. The computational complexity stays the same;
3. $\boldsymbol{S^*}$ is nonsingular (invertible);
4. $\boldsymbol{S^*}$ is robust;
5. The Mahalanobis distance is always well defined.

4. Future Work

Filzmoser (2005) has performed a simulation study to compare Rousseeuw's MCD method with other two methods for outlier detection. The methods of comparison are based on the Mahalanobis distance. The simulation results with high dimensional dataset reflect the limitation of the MCD to identify higher percentages of outliers. As a way out, Filzmoser (2005) suggests to use other estimators of multivariate location and scatter.

Rousseeuw's MCD method only uses about half of the dataset *[(n+p+1)/2]*. Almost all outlier candidates are excluded for estimating the covariance matrix. No doubt, the MCD is not sensitive to all outliers Therefore it is robust. There is tradeoff. The information of the whole dataset is not fully used. It may have a significant difference from the true scatter parameter. Therefore, the MCD estimator, as an estimator itself, may not be a good estimator in general (not only for the robustness).

In order to overcome this shortcoming, we suggest to increase the size *h* in the MCD method. The size *h* should be related to the size of the outlier set. For example, if there are about 5% outliers, then we should use about 95% dataset to perform the MCD method. In reality, the size of outlier is unknown. However it can be estimated in many ways. In theory, this size-change should be an improvement of the MCD method. This new estimator should be robust too with better qualities.

In the future study, we will run simulation studies to compare the performance in two new things:

1. Change the size *h* in the MCD method to $n - t$the size of outlier set;
2. Replace the MCD estimator $\boldsymbol{S^1}$ by the new combined covariance estimator $\boldsymbol{S^*}$**.**

5. Conclusions

The proposed new covariance estimator provides the same quality performance as the MCD estimator. In addition, it is positive definite. Its inverse matrix always exists. Furthermore, the Mahalanobis distance is well defined, and the determinant of the new proposed covariance estimator has increased very little. In theory, it does not affect the outlier detection, since the modified MCD algorithm will find the same outlier set.

6. REFERENCES

[1] Aggarwal, C. C. 2013. *Outlier Analysis.* New York: Springer.

[2] Chandola, V., Banerjee, A., and Kumar, V. 2009. Anomaly detection: A survey, *ACM Computing Surveys*, Vol. 41(3), Article 15.

[3] Chawla, S. and Gionis, A. 2013. *k*-means--: A unified approach to clustering and outlier detection, *SDM,* SIAM, 189-197.

[4] Filzmoser, P. 2005. Identification of Multivariate Outliers: A Performance Study, *Austrian Journal of Statistics*, Vol. 34 (2), 127-138.

[5] Hawkins, D. M. 1980. *Identification of Outliers.* New York: Chapman and Hall.

[6] Rousseeuw, P. J. and Driessen, K. V. 1999. A fast algorithm for the minimum covariance determinant estimator, *Technometrics,* Vol. 41(3), 212-223.

Solutions of a coupled four-parameter family of Boussinesq system

Chaudry Masood Khalique
International Institute for Symmetry Analysis and Mathematical Modelling,
Department of Mathematical Sciences, North-West University, Mafikeng Campus,
Private Bag X 2046, Mmabatho 2735, Republic of South Africa
Email: Masood.Khalique@nwu.ac.za

Abstract— *In this paper we study a four-parameter family of Boussinesq systems which describe the motion of small amplitude long waves on the surface of an ideal fluid under the gravity force and in situations where the motion is sensibly two dimensional. Exact travelling wave solutions of the system are obtained by using the $(G'/G)-$expansion method. The solutions obtained are expressed in the form of hyperbolic functions, trigonometric functions and rational solutions.*

Keywords: Boussinesq system, $(G'/G)-$expansion method, travelling wave solutions

1. Introduction

In [1] the authors considered different variants of the classical Boussinesq system and their higher-order generalizations. The classical Boussinesq system was first derived by Boussinesq to describe the two-way propagation of small-amplitude, long wavelength, gravity waves on the surface of water in a canal. Such systems and their higher-order generalizations arise also when modeling the propagation of long-crested waves on large lakes or on the ocean and in other situations.

In [1] the authors derived a four-parameter family of Boussinesq systems of the form

$$u_t + v_x + (uv)_x + av_{xxx} - bv_{xxt} = 0, \quad (1a)$$
$$v_t + u_x + vv_x + cu_{xxx} - dv_{xxt} = 0, \quad (1b)$$

where the parameters a, b, c, d are constants. The systems (1) describe the motion of small amplitude long waves on the surface of an ideal fluid under the gravity force and in situations where the motion is sensibly two dimensional. The systems (1) are all approximations to the same order of the Euler equations [1]. In (1), η is the elevation from the equilibrium position, and $w = w_\theta$ is the horizontal velocity in the flow at height θh, where h is the undisturbed depth of the liquid.

In this paper we employ the $(G'/G)-$expansion method to find exact solutions of the system (1). The $(G'/G)-$expansion method was introduced by Wang et al. [2], by which the travelling wave solutions can be obtained. The travelling wave solutions can be expressed by the hyperbolic functions, the trigonometric functions and the rational functions. This method is concise and effective, and can be used for many other nonlinear partial differential equations (NLPDEs).

The NLPDEs are extensively used as models to describe physical phenomena in applied sciences and engineering. It is therefore important to obtain exact solutions of NLPDEs. Unfortunately, it is almost impossible to find all the solutions of a NLPDE. Finding solutions of such an equation is a huge undertaking and only in certain special cases one can write down the solutions explicitly.

However, in the recent past many effective approaches for obtaining exact solutions of NLPDEs have been discovered and used to find diverse types of solutions of NLPDEs [3]–[26]. Some of the methods found in the literature include homogeneous balance method [3], the ansatz method [4], [5], variable separation approach [6], inverse scattering transform method [7], Bäcklund transformation [8], Darboux transformation [9], Hirota's bilinear method [10], the (G'/G)-expansion method [2], [11], the reduction mKdV equation method [12], the tri-function method [13], [14], the projective Riccati equation method [15], the sine-cosine method [16], the Jacobi elliptic function expansion method [17], [18], the $F-$expansion method [19], the exp-function expansion method [20], [21] and the Lie symmetry method [22]–[26].

2. Exact solutions of (1)

As a first step we transform (1) to a nonlinear ordinary differential equation (ODE) system using the traveling wave variable

$$u(x,t) = F(z), \ \ v(x,t) = H(z), \ \text{ where } \ z = x - \nu t. \quad (2)$$

Using the above transformations, (1) transforms to the nonlinear ODEs

$$aH'''(z) + b\nu H^{(3)}(z) + H(z)F'(z) - \nu F'(z) + F(z)H'(z) + H'(z) = 0, \quad (3a)$$
$$cF'''(z) + d\nu H'''(z) + F'(z) - \nu H'(z) + H(z)H'(z) = 0, \quad (3b)$$

where the primes denotes the derivative with respect to z.

The $(G'/G)-$expansion method assumes the solutions of (3) to be of the form

$$F(z) = \sum_{i=0}^{M} A_i (G'/G)^i \quad \text{and} \quad H(z) = \sum_{i=0}^{M} B_i (G'/G)^i, \quad (4)$$

where A_i, B_i, $i = 0, 1, \cdots, M$ are parameters to be determined and $G(z)$ satisfies the second-order linear ODE with constant coefficients, viz.,

$$G'' + \lambda G' + \mu G = 0, \quad (5)$$

where λ and μ are constants.

The balancing procedure yields $M = 2$, so the solutions of the ODEs (3) are of the form

$$F(z) = A_0 + A_1(G'/G) + A_2(G'/G)^2, \quad (6a)$$
$$H(z) = B_0 + +B_1(G'/G) + B_2(G'/G)^2. \quad (6b)$$

Substituting (6) into (3) and making use of (5), and then collecting all terms with same powers of (G'/G) and equating each coefficient to zero, yields a system of algebraic equations. Solving this system of algebraic equations, using Mathematica, we obtain the following:

$$\begin{aligned}
&-24\,b\nu\,A_2 - 24\,aB_2 - 4\,A_2B_2 = 0,\\
&-24\,d\nu\,B_2 - 24\,cA_2 - 2\,{B_2}^2 = 0,\\
&-54\,d\lambda\,\nu\,B_2 - 54\,c\lambda\,A_2 - 6\,d\nu\,B_1 - 2\,\lambda\,{B_2}^2\\
&-6\,cA_1 - 3\,B_1B_2 = 0,\\
&-54\,b\lambda\,\nu\,A_2 - 54\,a\lambda\,B_2 - 6\,b\nu\,A_1 - 4\,\lambda\,A_2B_2\\
&-6\,aB_1 - 3\,A_1B_2 - 3\,A_2B_1 = 0,\\
&-d\lambda^2\mu\,\nu\,B_1 - 6\,d\lambda\,\mu^2\nu\,B_2 - c\lambda^2\mu\,A_1 - 6\,c\lambda\,\mu^2A_2\\
&-2\,d\mu^2\nu\,B_1 - 2\,c\mu^2A_1 + \mu\,\nu\,B_1 - \mu\,B_0B_1\\
&-\mu\,A_1 = 0,\\
&-b\lambda^2\mu\,\nu\,A_1 - 6\,b\lambda\,\mu^2\nu\,A_2 - a\lambda^2\mu\,B_1 - 6\,a\lambda\,\mu^2B_2\\
&-2\,b\mu^2\nu\,A_1 - 2\,a\mu^2B_1 + \mu\,\nu\,A_1 - \mu\,A_0B_1\\
&-\mu\,A_1B_0 - \mu\,B_1 = 0,\\
&-38\,d\lambda^2\nu\,B_2 - 38\,c\lambda^2A_2 - 12\,d\lambda\,\nu\,B_1 - 40\,d\mu\,\nu\,B_2\\
&-12\,c\lambda\,A_1 - 40\,c\mu\,A_2 - 3\,\lambda\,B_1B_2 - 2\,\mu\,{B_2}^2 + 2\,\nu\,B_2\\
&-2\,B_0B_2 - {B_1}^2 - 2\,A_2 = 0,\\
&-38\,b\lambda^2\nu\,A_2 - 38\,a\lambda^2B_2 - 12\,b\lambda\,\nu\,A_1 - 40\,b\mu\,\nu\,A_2\\
&-12\,a\lambda\,B_1 - 40\,a\mu\,B_2 - 3\,\lambda\,A_1B_2 - 3\,\lambda\,A_2B_1\\
&-4\,\mu\,A_2B_2 + 2\,\nu\,A_2 - 2\,A_0B_2 - 2\,A_1B_1 - 2\,A_2B_0\\
&-2\,B_2 = 0,\\
&-d\lambda^3\nu\,B_1 - 14\,d\lambda^2\mu\,\nu\,B_2 - c\lambda^3A_1 - 14\,c\lambda^2\mu\,A_2\\
&-8\,d\lambda\,\mu\,\nu\,B_1 - 16\,d\mu^2\nu\,B_2 - 8\,c\lambda\,\mu\,A_1 - 16\,c\mu^2A_2\\
&+\lambda\,\nu\,B_1 - \lambda\,B_0B_1 + 2\,\mu\,\nu\,B_2 - 2\,\mu\,B_0B_2 - \mu\,{B_1}^2\\
&-\lambda\,A_1 - 2\,\mu\,A_2 = 0,\\
&-8\,d\lambda^3\nu\,B_2 - 8\,c\lambda^3A_2 - 7\,d\lambda^2\nu\,B_1 - 52\,d\lambda\,\mu\,\nu\,B_2\\
&-7\,c\lambda^2A_1 - 52\,c\lambda\,\mu\,A_2 - 8\,d\mu\,\nu\,B_1 - 8\,c\mu\,A_1\\
&+2\,\lambda\,\nu\,B_2 - 2\,\lambda\,B_0B_2 - \lambda\,{B_1}^2 - 3\,\mu\,B_1B_2 - 2\,\lambda\,A_2\\
&+\nu\,B_1 - B_0B_1 - A_1 = 0,\\
&-b\lambda^3\nu\,A_1 - 14\,b\lambda^2\mu\,\nu\,A_2 - a\lambda^3B_1 - 14\,a\lambda^2\mu\,B_2\\
&-8\,b\lambda\,\mu\,\nu\,A_1 - 16\,b\mu^2\nu\,A_2 - 8\,a\lambda\,\mu\,B_1 - 16\,a\mu^2B_2\\
&+\lambda\,\nu\,A_1 - \lambda\,A_0B_1 - \lambda\,A_1B_0 + 2\,\mu\,\nu\,A_2 - 2\,\mu\,A_0B_2\\
&-2\,\mu\,A_1B_1 - 2\,\mu\,A_2B_0 - \lambda\,B_1 - 2\,\mu\,B_2 = 0,\\
&-8\,b\lambda^3\nu\,A_2 - 8\,a\lambda^3B_2 - 7\,b\lambda^2\nu\,A_1 - 52\,b\lambda\,\mu\,\nu\,A_2\\
&-7\,a\lambda^2B_1 - 52\,a\lambda\,\mu\,B_2 - 8\,b\mu\,\nu\,A_1 - 8\,a\mu\,B_1\\
&+2\,\lambda\,\nu\,A_2 - 2\,\lambda\,A_0B_2 - 2\,\lambda\,A_1B_1 - 2\,\lambda\,A_2B_0\\
&-3\,\mu\,A_1B_2 - 3\,\mu\,A_2B_1 - 2\,\lambda\,B_2 + \nu\,A_1\\
&-A_0B_1 - A_1B_0 - B_1 = 0.
\end{aligned}$$

Solving the above system of algebraic equations, with the aid of Mathematica, we obtain

$$\begin{aligned}
&\beta = (-6\,b^2\lambda\,\nu^2 - 8\,bd\lambda\,\nu^2 - 24\,d^2\lambda\,\nu^2 - 16\,ac\lambda)\alpha\\
&-72\,b^2d\lambda\,\nu^3 - 144\,bd^2\lambda\,\nu^3 + 72\,abc\lambda\,\nu + 144\,acd\lambda\,\nu,\\
&\gamma = (-b\nu - 2\,d\nu)\,\alpha - 16\,bd\nu^2 + 16\,ac],\\
&m_1 = -\frac{(2\,d\nu + \alpha)\,\beta}{\gamma} + 5\,\lambda\,\nu\,(b + 2\,d)\,\alpha\\
&+60\,bd\lambda\,\nu^2 - 60\,ac\lambda,\\
&m_2 = 19\,c\lambda^2 + 20\,c\mu + 1,\\
&m_3 = -12\,bd\nu^2 - \alpha\,b\nu + 12\,ac,\\
&m_4 = -19\,d\lambda^2 + 6\,b\mu - 8\,d\mu + 1,\\
&m_5 = 38\,b\lambda^2\nu + 40\,b\mu\,\nu - 3\,\frac{\lambda\,\beta}{\gamma} + 4\,\mu\,\alpha - 2\,\nu,\\
&m_6 = 12\,d\lambda\,\nu + 3\,\alpha\,\lambda,\\
&m_7 = 12\,b\lambda\,\nu + 3\,\alpha\,\lambda - 2\,\frac{\beta}{\gamma},\\
&m_8 = 19\,a\lambda^2 + 20\,a\mu + 1,\\
&A_0 = \frac{3m_3m_1\lambda}{2\alpha^2c} + \frac{{m_3}^2m_2}{4\alpha^2c^2} - \frac{m_3\beta^2}{4\alpha^2c\gamma^2}\\
&+\frac{m_3m_6\beta}{4\alpha^2c\gamma} + \frac{m_3\nu\,m_4}{2\alpha\,c} + \frac{36m_3bd\mu\,\nu^2}{\alpha^2c}\\
&-\frac{36m_3a\mu}{\alpha^2} + \frac{m_7m_1}{4\alpha\,c} + \frac{m_5m_3}{4\alpha\,c} + 6\,\frac{\beta\,a\lambda}{\alpha\,\gamma}\\
&-m_8,\\
&A_1 = \frac{1}{c}\Big(\frac{(2\,d\nu + \alpha)\beta}{2\gamma} - \frac{5\lambda\,\nu\,(b + 2\,d}{2})\alpha\\
&-30\,bd\lambda\,\nu^2 + 30\,ac\lambda\Big),\\
&A_2 = \frac{12\,bd\nu^2\alpha\,b\nu - 12\,ac}{2c},
\end{aligned}$$

$$B_0 = \frac{3m_1\lambda}{\alpha} + \frac{m_2m_3}{2\alpha\, c} - \frac{\beta^2}{2\alpha\,\gamma^2} + \frac{6\beta\, d\lambda\,\nu}{\alpha\,\gamma}$$
$$+\frac{3\beta\,\lambda}{2\gamma} + \nu\, m_4 + \frac{72bd\mu\,\nu^2}{\alpha} - \frac{72ac\mu}{\alpha},$$
$$B_1 = -\frac{\beta}{\gamma},$$
$$B_2 = \alpha,$$

where α is any root of

$$\alpha^2 + (6\,b\nu + 12\,d\nu)\,\alpha + 72\,bd\nu^2 - 72\,ac = 0.$$

Thus, we have the following three types of travelling wave solutions of (1):

When $\lambda^2 - 4\mu > 0$, we obtain the hyperbolic function solutions

$$\begin{aligned} u_1(x,t) &= \mathcal{A}_0 + \mathcal{A}_1\Big(-\frac{\lambda}{2}\Big) \\ &\quad +\mathcal{A}_1\Big(\delta_1\frac{C_1\sinh(\delta_1 z) + C_2\cosh(\delta_1 z)}{C_1\cosh(\delta_1 z) + C_2\sinh(\delta_1 z)}\Big) \\ &\quad +\mathcal{A}_2\Big(-\frac{\lambda}{2} \\ &\quad +\delta_1\frac{C_1\sinh(\delta_1 z) + C_2\cosh(\delta_1 z)}{C_1\cosh(\delta_1 z) + C_2\sinh(\delta_1 z)}\Big)^2, \quad (7a) \\ v_1(x,t) &= \mathcal{B}_0 + \mathcal{B}_1\Big(-\frac{\lambda}{2}\Big) \\ &\quad +\mathcal{B}_1\Big(\delta_1\frac{C_1\sinh(\delta_1 z) + C_2\cosh(\delta_1 z)}{C_1\cosh(\delta_1 z) + C_2\sinh(\delta_1 z)}\Big) \\ &\quad +\mathcal{B}_2\Big(-\frac{\lambda}{2} \\ &\quad +\delta_1\frac{C_1\sinh(\delta_1 z) + C_2\cosh(\delta_1 z)}{C_1\cosh(\delta_1 z) + C_2\sinh(\delta_1 z)}\Big)^2, \quad (7b) \end{aligned}$$

where $z = x - \nu t$, $\delta_1 = \frac{1}{2}\sqrt{\lambda^2 - 4\mu}$, C_1 and C_2 are arbitrary constants.

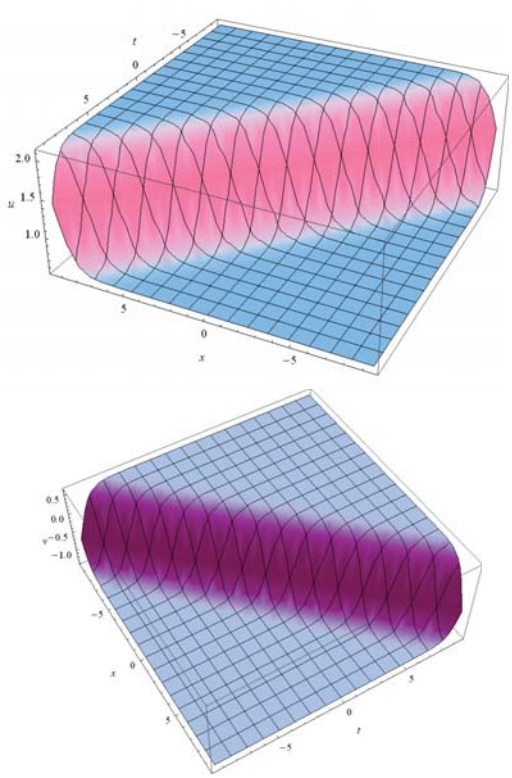


Fig. 1: Profile of (7)

When $\lambda^2 - 4\mu < 0$, we obtain the trigonometric function solutions

$$\begin{aligned} u_2(x,t) &= \mathcal{A}_0 + \mathcal{A}_1\Big(-\frac{\lambda}{2}\Big) \\ &\quad +\mathcal{A}_1\Big(\delta_2\frac{-C_1\sin(\delta_2 z) + C_2\cos(\delta_2 z)}{C_1\cos(\delta_2 z) + C_2\sin(\delta_2 z)}\Big) \\ &\quad +\mathcal{A}_2\Big(-\frac{\lambda}{2} \\ &\quad +\delta_2\frac{-C_1\sin(\delta_2 z) + C_2\cos(\delta_2 z)}{C_1\cos(\delta_2 z) + C_2\sin(\delta_2 z)}\Big)^2, \quad (8a) \\ v_2(x,t) &= \mathcal{B}_0 + \mathcal{B}_1\Big(-\frac{\lambda}{2}\Big) \\ &\quad +\mathcal{B}_1\Big(\delta_2\frac{-C_1\sin(\delta_2 z) + C_2\cos(\delta_2 z)}{C_1\cos(\delta_2 z) + C_2\sin(\delta_2 z)}\Big) \\ &\quad +\mathcal{B}_2\Big(-\frac{\lambda}{2} \\ &\quad +\delta_2\frac{-C_1\sin(\delta_2 z) + C_2\cos(\delta_2 z)}{C_1\cos(\delta_2 z) + C_2\sin(\delta_2 z)}\Big)^2, \quad (8b) \end{aligned}$$

where $z = x - \nu t$, $\delta_2 = \frac{1}{2}\sqrt{4\mu - \lambda^2}$, C_1 and C_2 are arbitrary constants.

When $\lambda^2 - 4\mu = 0$, we obtain the rational function

solutions

$$\begin{aligned} u_3(x,t) &= \mathcal{A}_0 + \mathcal{A}_1\Big(-\frac{\lambda}{2} + \frac{C_2}{C_1 + C_2 z}\Big) \\ &\quad + \mathcal{A}_2\Big(-\frac{\lambda}{2} + \frac{C_2}{C_1 + C_2 z}\Big)^2, \\ v_3(x,t) &= \mathcal{B}_0 + \mathcal{B}_1\Big(-\frac{\lambda}{2} + \frac{C_2}{C_1 + C_2 z}\Big) \\ &\quad + \mathcal{B}_2\Big(-\frac{\lambda}{2} + \frac{C_2}{C_1 + C_2 z}\Big)^2, \end{aligned}$$

where $z = x - \nu t$, C_1 and C_2 are arbitrary constants.

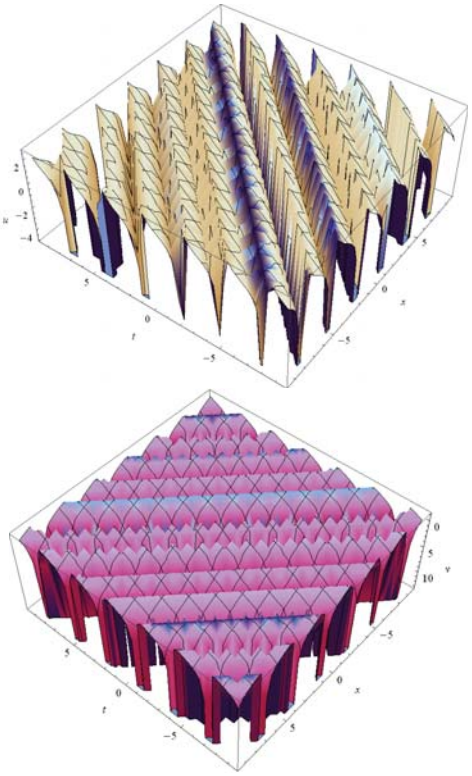


Fig. 2: Profile of (8)

3. Conclusion

In this paper, we obtained exact travelling wave solutions of (1) by employing the $(G'/G)-$expansion method. The solutions obtained were expressed in the form of hyperbolic functions, trigonometric functions and rational solutions.

References

[1] J.L. Bona, M. Chen, J.-C. Saut, Boussinesq equations and other systems for small-amplitude long waves in nonlinear dispersive media. I. Derivation and linear theory, J. Nonlinear Sci. 12 (4) (2002) 283–318.

[2] M.L. Wang, X.Z. Li, and J.L. Zhang, The $(G'/G)-$expansion method and travelling wave solutions of nonlinear evolution equations in mathematical physics, Phys. Lett. A 372 (2008), 417–423.

[3] M. Wang, Y. Zhou, Z. Li, Application of a homogeneous balance method to exact solutions of nonlinear equations in mathematical physics, Phys. Lett. A. 216 (1996) 67-75.

[4] J.L. Hu, Explicit Solutions to Three Nonlinear Physical Models, Phys. Lett. A. 287 (2001) 81-89.

[5] J.L. Hu, A New Method for Finding Exact Traveling Wave Solutions to Nonlinear Partial Differential Equations, Phys. Lett. A. 286 (2001) 175-179.

[6] S.Y. Lou, J. Z. Lu, Special Solutions from Variable Separation Approach: Davey-Stewartson Equation, J Phys A-Math Gen . 29 (1996) 4209-4215.

[7] M.J. Ablowitz, P. A. Clarkson, Soliton, Nonlinear Evolution Equations and Inverse Scattering, Cambridge University Press, Cambridge, 1991.

[8] C.H. Gu, Soliton Theory and Its Application, Zhejiang Science and Technology Press, Zhejiang, 1990.

[9] V.B. Matveev, M. A. Salle, Darboux Transformation and Soliton, Springer, Berlin, 1991.

[10] R. Hirota, The Direct Method in Soliton Theory, Cambridge University Press, Cambridge, 2004.

[11] D.M. Mothibi, C.M. Khalique, On the exact solutions of a modified Kortweg de Vries type equation and higher-order modified Boussinesq equation with damping term, Advances in Difference Equations, 2013, 2013:166

[12] Z.Y. Yan, A Reduction mKdV Method with Symbolic Computation to Construct New Doubly-Periodic Solutions for Nonlinear Wave Equations, Int J Mod Phys C. 14 (2003) 661-672.

[13] Z.Y. Yan, The New Tri-Function Method to Multiple Exact Solutions of Nonlinear Wave Equations, Phys Scripta. 78 (2008) Article ID: 035001.

[14] Z.Y. Yan, Periodic, Solitary and Rational Wave Solutions of the 3D Extended Quantum Zakharov-Kuznetsov Equation in Dense Quantum Plasmas, Phys. Lett. A. 373 (2009) 2432-2437.

[15] D.C. Lu, B.J. Hong, New Exact Solutions for the (2+1)-Dimensional Generalized Broer-Kaup System, Appl. Math. Comput. 199 (2008) 572-580.

[16] M. Wazwaz, The Tanh and Sine-Cosine Method for Compact and Noncompact Solutions of Nonlinear Klein Gordon Equation, Appl. Math. Comput. 167 (2005) 1179-1195.

[17] D.C. Lu, Jacobi Elliptic Functions Solutions for Two Variant Boussinesq Equations, Chaos Soliton Fract. 24 (2005) 1373-1385.

[18] Z.Y. Yan, Abundant Families of Jacobi Elliptic Functions of the (2+1) Dimensional Integrable Davey-Stewartson-Type Equation via a New Method, Chaos Soliton Fract. 18 (2003) 299-309.

[19] M. Wang, X. Li, Extended F-Expansion and Periodic Wave Solutions for the Generalized Zakharov Equations, Phys. Lett. A. 343 (2005) 48- 54.

[20] J.H. He, X. H. Wu, Exp-Function Method for Nonlinear Wave Equations, Chaos Soliton Fract. 30 (2006) 70.

[21] G. Magalakwe, CM Khalique, New Exact Solutions for a Generalized Double Sinh-Gordon Equation, Abstract and Applied Analysis, Volume 2013, Article ID 268902, 5 pages

[22] G.W. Bluman, S. Kumei, Symmetries and Differential Equations, Applied Mathematical Sciences, 81, Springer-Verlag, New York, 1989.

[23] P.J. Olver, Applications of Lie Groups to Differential Equations, Graduate Texts in Mathematics, 107, 2nd edition, Springer-Verlag, Berlin, 1993.

[24] N.H. Ibragimov, CRC Handbook of Lie Group Analysis of Differential Equations, Vol 1-3, CRC Press, Boca Raton, Florida, 1994-1996.

[25] L.V. Ovsiannikov, Group Analysis of Differential Equations, Academic Press, New York, (English translation by W.F. Ames) 1982.

[26] A.R. Adem, C.M. Khalique, Exact Solutions and Conservation Laws of a Two-Dimensional Integrable Generalization of the Kaup-Kupershmidt Equation, Journal of Applied Mathematics, Volume 2013, Article ID 647313, 6 pages,

On Initial Effects of the k-Means Clustering

Sherri Burks, Greg Harrell, and Jin Wang
Department of Mathematics and Computer Science
Valdosta State University, Valdosta, Georgia 31698 USA

Abstract - *There are many research studies conducted in order to find a more optimal way to initialize the k-means algorithm, also referred to as Lloyd's algorithm. Despite the appreciated efficiency of the k-means process, occasionally it may return a less than optimal clustering solution. It is widely believed that modifications to the initialization process will improve results. Here, the choice of initial centroids for the k-means clustering technique is reviewed with respect to efficiency in stabilizing or convergence with different initialization methods. Several proposed initialization techniques are evaluated on a two dimensional model in an attempt to verify or reproduce results similar to those of the studies chosen.*

Keywords: Algorithm; Cluster; Data Mining; K-Means

1 Introduction

1.1 Definition

The k-means clustering algorithm is a procedure that is widely applied in data mining, biometrics, and signals processing to aid in aggregating or visualizing related data. It is used to establish a set number (k) partitions or clusters of a dataset in which each element of the cluster is most like, or within the closest proximity of that cluster's mean. This is determined by a series of recursive calls that begins with assigning a data point to one of the k arbitrarily chosen initial cluster centroids based upon its closest Euclidian proximity, defined below in Figure 1.

$$d_{xy} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad (1)$$

Once all the data points have been assigned to a cluster, the cluster means are computed and the data points are re-assigned to the new centroids. The process is repeated until it converges and no changes in clusters or centroids are observed. Section 2 explains the k-means algorithm more simply.

The effects of the initialization of the k-means process is claimed to have a significant impact upon the stability and optimization of the results. The deliberations surrounding initializing the event consists of determining the quantity of clusters (k) and thus the number of centroids, which at times may be determined by the user's application, and also consists of the method used in choosing the initial centroids for the first iteration. Although some of the methods reviewed address determining the proper quantity of clusters, the scope of this study will focus on the latter. Several methods of initializing the k-means algorithm will be reviewed and the effects of the initialization compared using the same data sets and number of clusters.

1.2 Application

As the amount of large scale data collection continues to increase, the need for efficient and reliable methods to process big data becomes essential. The utilization of the k-means algorithm as a solution to clustering problems can be found in applications from data mining to fuzzy vectors and visual graphics. Implementations of the k-means algorithm are extensively found in data visualization for purposes of data mining and recommender systems. These recommender systems may be used to find venues or items for individuals of like mindedness, or location based similarity such as a nearest neighbor. Additionally, variations of the k-means are found in decision engines, medical imaging, routing, and in signal cleaning, removing noise and detecting outliers. Determining a better initialization method may result in a faster convergence or may avoid a less than optimal solution. More reliable clustering results would help facilitate trust and reliance upon machine learning based systems, particularly in those of a critical nature such as in medical applications.

Traditional k-means is a greedy algorithm that can present issues with stability, particularly with the selection of outliers and low density areas as cluster centers. For example, a situation may be observed where an element may be equally distant between two centroids and may oscillate between two clusters in assignment. Additionally, k-means may exhibit a tendency of convergence to a local minima. It does not ensure a global minima, nor does it guarantee unique clustering [1]. It has been reported that noise or isolated outliers may greatly affect average values [4] and skew cluster regionalization. Finding a more advantageous initialization method for k-means should produce more optimal results, faster convergence and stability, and may improve the user's trust and reliability of the results in data analysis applications.

2 K-means algorithm

2.1 Algorithm pseudo-code

The basic algorithm used for k-means is as follows:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters

Output:

C: set of k clusters, indicated by Ci …Ck

Process:

1. Choose k points randomly from S as the initial centroids (means or center of mass)
2. Calculate the Euclidian distance from each point to each centroid. Assign the point to the set of the nearest cluster C_i.
3. Recalculate the means for each cluster and shift the centroid as necessary.
4. Repeat step 2 and 3 until there are no changes.

2.2 Computer implementation

A java program was executed on a 64-bit Intel Core i5-3337U CPU @ 1.80GHz with 4 GB memory running the Windows 8.1 Operating System. The original k-means clustering was implemented by generating random data sets of two dimensional Cartesian coordinate values. As each coordinate was created it was checked to make sure it was not a duplicate before being added to the data set. This same data set is used in later comparisons for all initialization process types. A pre-determined number of clusters was established to be 3. (Future comparison studies may involve 4 and 5 clusters on larger data sets). A random centroid was chosen to be each cluster's initial center, and the k-means algorithm proceeded by assigning points to nearby cluster centers and then adjusting the location of the centroid based upon the mean value of each cluster. This execution continued until changes in cluster assignment or movements of the centroids were no longer observed and the process stabilized. In order to accomplish this, the sets of coordinates and centroids were stored as linked lists of point objects and Boolean variables were used to identify when convergence was reached. New datasets are generated for subsequent comparisons, and results are averaged.

3 K-means initial cluster center

3.1 Initialization Effects

Although deciding upon the best manner to initialize the cluster center is of some contention, there seems to be much agreement on the effects of poor initialization and the importance of choosing a proper method of seeding the process. There is concurrence that the choice of the initial cluster centers may result in different clustering results of the same data set. In addition, the number of iterations that the process requires before stabilizing may be affected by a poor initialization. Concerns of becoming stuck in a local minimum and instability issues as defined by [4] as the random sampled mean minimal matching distance between two clusters on two sets of data points. According to [6], the initialization effects are more appreciated when there are large numbers of clusters.

3.2 K-means sensitivity (stability) analysis

Different choices of initialized cluster centers may lead to unstable clustering results as it may likely produce different partitions. [5] Such instability is described by [4] as the random sampled mean minimal matching distance between two clusters on two sets of data points. Noise, outlying data points, and isolated data points affect the mean values of dense clusters, and tend to bias the clustering into less than optimal results. In one instance that was observed on a sample run, the partitions were not the best choice, as the cluster that had two extreme outliers stretched over a greater margin on the y-axis, rather than having some of those points assigned to a more logical cluster. In this case, the centers were closer together and less than optimal to start. Another observation was made where the partitioning took place in a more horizontal fashion when a slightly more vertical partitioning appeared more likely.

Numerous studies have been published proposing solutions to these issues, recommending various methods of the initialization process of selecting the first set of cluster centroids. Some of the less involved approaches that use comparable methods will be addressed in section 4.

4 K-means initial selection algorithms

There are a number of sources of initial selection algorithms to choose from. Some of the studies reviewed focused on discovering the proper number of clusters prior to running the clustering process. One publication additionally noted the importance of the number of centroids with respect to the number of clusters. Because of this, techniques with a common, pre-determined number of clusters (k) were used in this study. However all of the techniques reviewed emphasized the importance of finding the appropriate centroids to begin the clustering process. The options reviewed ranged as follows: using weights based upon probability, choosing a centroid location based upon density, subdividing clusters into smaller subsections prior to choosing a centroid, using a graph based method, and combinations of these techniques.

4.1 Survey of algorithms

4.1.1 K-means++

The k-means++ algorithm uses an initialization method of choosing the random starting centroids based upon the probability of that point being a center in an attempt to produce clusters that are more diffuse. The first center is chosen at random. Each additional k-1 centers are chosen based upon a weighted probability such that once the first random centroid is set the selection of each successive center is affected by the other centers that have been chosen prior. Once all of the

centers are chosen, the process proceeds based upon the original k-means algorithm, assigning points to the clusters until a convergence is reached. The calculation of the probability for the k-1 centers is a ratio of a candidate center point's distance to the other centers. It is measured as the squared shortest distance of the point to the closest centroid divided by the sum of the squared distance of all other candidate points to the center. One criticism of this technique is that it is not scalable for big data. [3]

4.1.2 Density

A density based approach to cluster initialization relies heavily upon accurately determining those areas of high density. Density based methods do not rely upon random selection, but instead determine the density of an area within a radius and then find other high density areas that are further apart. One approach selects the initial centroid from the highest density area. That centroid and its assigned cluster are removed from the data pool and the process repeats to find the next centroid and its corresponding cluster, continuing until all k clusters have been determined. Here a strong emphasis must be placed upon establishing a proper radius. Incorporating distance into the decision is more harmonious with the source distribution [5] and in the absence of outliers it yields better clustering results by having the centroids in those high density areas farther from each other. *It is of great importance to note that this procedure assumes that the low density areas may be outliers.* Any density based approach levies great importance upon deciding what constitutes an area of high density.

4.1.3 Reverse nearest neighbor and coupling

[1] This concept combines the method of conducting a proximity search for a reversed nearest neighbor and the method of determining and eliminating candidate centroids of similar likeness. A reverse nearest neighbor is the point(s) to which the current point is closest to, rather than the closest neighbor to that point. This approach transpires by creating three sets. To begin, all points are included in the candidate set. Representative points are selected by finding and counting the reversed nearest neighbor for each point in the candidate set, and adding the point with the highest count to the representative set. The point and its corresponding points are removed from the candidate set. This repeats until there are only points with a count of one remaining. Choosing the centroids are selected from this representative set using a coupling technique. The average distance between all points is calculated and the coupling degree between all pairs of points is calculated. Points with the largest count of neighbors and coupling degree are selected. A mean point is chosen as calculated for that point and its neighbors, and added to the centroid set. The point and its corresponding neighbors are removed from the representative set and the process repeats until all k initial cluster centroids have been selected. Then the k-means process continues as usual.

4.1.4 Mean-shift and k-means++

A multi-faceted approach from [6] involves combining methods used in mean shift and k-means++. As a premise, it requires reducing the number of dimensions down to two dimensional data consisting of the two most significant, independent variables to represent the overall data. The first is chosen from the highest calculated coefficient of variation and the second variable is the one with the lowest calculated correlation coefficient. (Test data for this study is already two dimensional. Therefore this process is eliminated in test comparisons in order to both keep the evaluation on a more even standing and tighten the focus on cluster initialization centroid selection effects.) This is followed by determining an appropriate radius for the likeness neighborhood, essentially determining the cluster boundaries or areas of most data similarity. Finally the cluster centers are chosen. The first is an arbitrarily chosen centroid. The mean of the data points within that centroid's radius is calculated and the centroid is shifted to the nearest data point. A probability ratio is calculated similar to that of k-means++ where the shortest distance of each data point to its corresponding centroid is divided by the maximum distance to from all points to the centroids. The calculated ratio is the likeliness of that point being a candidate centroid and is used in selecting the remaining centroids. Each new centroid that is selected is shifted towards the cluster mean, and the process continues until all centroids and clusters have been established.

4.2 K-means algorithm with different initial selections pseudo-code

4.2.1 K-means++

The k-means++ algorithm is as follows:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters

Output:

C: set of k clusters, indicated by Ci …Ck

Process:

1. Choose a point randomly from S as the first initial centroid C1.

2. Choose the following k-1 centroids C2… Ck by determining probability in the following manner: For all remaining points, find $D(x)^2$: Square the Euclidian distance from a candidate point to C1. Add these values to an accumulator that holds the sum of $D(xi…k-1)^2$ distances for all candidate points. Divide the $D(x)^2$ for each point by that accumulated sum to obtain that point's probability. Finally choose the centroid randomly from the candidates based upon the weighted probabilities.

3. Repeat step 2 until all k centroids have been chosen.

4. Once all initial centroids have been established, continue by using the original k-means process of assigning the data points to cluster centers and re-calculating the centroids until the process stabilizes.

4.2.2 Density

The density based algorithm discussed is as follows:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters
r: the radius used for calculating density

Output:

C: set of k clusters, indicated by Ci …Ck

Process:

1. For each point, count the number of nearest neighbors within a certain radius.

2. For the first centroid, select the point with the highest neighbor count and add it to set C.

3. Remove that point and its nearest neighbors from the set of available points to pick from.

4. Repeat step 2 and 3 until all k centroids have been chosen.

5. Proceed with clustering using the original k-means process of assigning the data points to cluster centers and re-calculating the centroids until the process stabilizes.

4.2.3 Reverse nearest neighbor and coupling

The reverse nearest neighbor and coupling algorithm follows:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters
r: the radius used for calculating density

Output:

C: set of k clusters, indicated by Ci …Ck

Additional structures used:

CS: set of candidate points
RNN: a list of each point's RNN count
RS: representative set

Process:

Phase 1:

1. All points in S are included in CS.
2. For each point, its reverse nearest neighbors (RNN) are counted and the number is stored in the RNN list.
3. The point with the max RNN count is added to the RS set and deleted from the CS set. Its corresponding RNN points are also deleted.
4. Repeat step 3 until the only values remaining in RNN equals one.

Phase 2:

5. Calculate the average distance between all points.
6. Calculate the coupling degree between all possible pairs of points in the RS set.
7. Select the point with the greatest number of neighbors and coupling degree and calculate the mean for that point and its neighbors.
8. Add that point to the set of cluster centroids, C. Delete that point and its neighbors from RS.
9. Repeat steps 7 – 8 until k centroids have been selected.

4.2.4 Mean-shift and k-means++

This method proceeds in the following manner:

Input:

S: set of data points, ex.: (x, y)
k: the number of clusters

Output:

C: set of k clusters, indicated by Ci …Ck

Process:

Phase 1:

1. Reduce the number of dimensions down to a two dimensional representative subspace by selecting the two most significant, independent variables. The first is chosen from the highest calculated coefficient of variation and the second variable is the one with the lowest calculated correlation coefficient. (Assume this has been completed for the study comparison).

Phase 2:

2. Determine the radius:
   a) Randomly select 100 points from S.
   b) Compute each point's distance to its nearest neighbor.
   c) The radius equals 4 times the max of the distances found in step b.

Phase 3:

2. Randomly choose a point from S and find the mean of its neighbors within the radius from 2.c.

3. Shift the centroid to the point nearest the calculated mean. Repeat this until the centroid location stabilizes.

4. Use a probability ratio to find the next candidate centroid:
   a) Find the minimum distance from each data point in S to the centers.
   b) Divide that value by maximum distance from all data to the centers.
   c) Use that ratio as the probability for the point to be a centroid.
   d) Choose the most probable point as the next centroid.

6. Shift the cluster center as in step 4 until convergence is reached.

7. Repeat steps 5 -6 until all k cluster centroids are found.

4.3 Computer implementation – programming

A java program was implemented on a 64-bit Intel Core i5-3337U CPU @ 1.80GHz with 4 GB memory running the Windows 8.1 Operating System. Random sets of two dimensional Cartesian coordinates were generated. For simplicity and scientific fidelity for comparisons the same data sets and the same number of clusters were used for all initializing algorithms. Array lists were used to store coordinates (as point objects) and centroids (also point objects). A table was used to keep track of the assignment of points to centers. Also, a timer was placed to keep track of execution time of the entire process from initialization to stabilization so that the expense of a complex seeding can be measured against a simpler initialization process. Additionally, a counter was used to keep track of the number of iterations for the k-means process to stabilize (post-initialization) so that the effectiveness and efficiency of the particular methods may be evaluated. This is in line with experiments conducted in study [1], asserting that lower iterations to achieve convergence indicates a more accurate initial centroid selection and the execution time of the algorithm can be used as a measure of performance. For the initialization procedures that required a radius, the same method was used in determining the cluster radius as outlined in [6]. The reasoning for this is twofold. It asserts that the techniques are compared in a more even manner and compensates for those studies using radius with an unspecified definition. Finally, distances from cluster points to their centroid were stored for comparing the overall partitioning outcome for each technique.

5 Comparisons

In the tables below, each of the methods were performed concurrently on the same data set. Often each process returned different resulting centroids, but in general, they were in fairly close proximity to the other methods centroids. The graphical placement was not studied in detail due to time constraints. However, the number of iterations and elapsed time indicate the efficiency of the initialization technique as the measurements were taken after the respective initialization process was conducted. It measures how much elapsed time the k-means process required to reach a convergence based upon how it was initialized and how many iterations were completed. In order to calculate the time, the system clock was started at the beginning of the k-means clustering and stopped at the point of convergence. For the comparisons, the test was repeated on newly generated datasets and the iterations and elapsed time was averaged for each process. As the datasets size increased, the initialization time required increased dramatically while the time for the k-means to converge remained relatively the same. The time expense of the density based approach and the combined approaches were quite obvious on large amounts of data. Future comparisons will include a clocked estimate for the overall process in order to compare the overall expense of a complex, highly iterative initialization.

Table 1. Results of Dataset 1 of 5000 points with k =3

Method	Iterations	Elapsed Time (ms)
K-means	4.2	21.8
K-means++	5.2	21.2
Density	4.3	13.9
MeanShift & K-means++	4.3	19

Table 2. Results of Dataset 2 of 10000 points with k =3

Method	Iterations	Elapsed Time (ms)
K-means	4.4	35.9
K-means++	5.1	35.9
Density	5.9	43.8
MeanShift & K-means++	4.4	28.2

Table 3. Results of Dataset 3 of 15000 points with k =3

Method	Iterations	Elapsed Time (ms)
K-means	7.5	81.2
K-means++	6.7	69
Density	6.9	70
MeanShift & K-means++	6.2	61.3

Table 4. Results of Dataset 4 of 20000 points with k =4

Method	Iterations	Elapsed Time (ms)
K-means	6.4	115.7
K-means++	7.9	142.2
Density	6.7	118.9
MeanShift & K-means++	5.6	101.5

6 Summary and conclusions

There appears to be universal agreement among the papers reviewed that the basic k-means process is greatly affected by the initialization of the centroids and number of clusters and may yield a less than optimal solution with a localized minima. [1,2,3,4,5,6] The number of clusters has significant importance but it was deemed important enough for a separate study independent from the centroid initialization, much like in study [4]. Many studies examined evaluated attempts at modifying both aspects. However, it was decided that examining the two factors separately will yield a better understanding of the impact of each component. The focus of this paper centralized on methods of seeding the cluster centers.

K-means++ initialization allows for a bias that an initial centroid will be further away from the other selected centers, resulting in improved clustering and less iterations to reach convergence [6][2]. In addition, there is also the possibility that the initial cluster chosen randomly, C1, could itself be an outlier, thus affecting the distance calculations and adding iterations prior to k-means stabilization.

Reverse nearest neighbor and coupling is expensive and requires several iterations and distance calculations though sets of points. The concept seems solid, however attempts to replicate the procedures used by [1] were difficult, as the method for selecting the point with the max neighbor count and coupling degree was not explicitly specified and therefore left to interpretation as to whether the values should be combined or a ratio, or which value took precedence. As a result experiments using this technique were incomplete and omitted from the results comparison charts.

Mean shift with k-means++ does not suffer the same vulnerability of the k-means++ and performed well in testing. Even though the initial center is randomly chosen, the focal point is shifted prior to determining the probability ratio and choosing the next centroid, thus reducing the effects of the first randomly chosen point occurring in a low density area. The algorithm suggested by [6] did not clearly indicate how to prevent the selection of a duplicate point as a candidate. Therefore, that was subject to programming discern. When the process continues according to the algorithm suggested, the k-means shuffles the centroids, eliminating the problem of the duplicate. However, the results of the study are not reliable because of the potential duplicates that incur despite the coding corrections. The mean shift did not facilitate for the overlap of radii between clusters.

With additional time on this study, more measurable and quantifiable calculations would be presented to determine the accuracy of each process in finding an optimal cluster center and additional initialization processes would be added for testing. In addition, more visual, graphical representations would be added. It did seem that some of the initialization processes were more costly than the amount of iterations saved on the k-means process. This can be determined with the addition of a timestamp prior to beginning each process and at the finish of the k-means.

Even the crude experimentation in this study made it apparent that the initialization of the k-means process unquestionably affects the resulting centroids and the assigned cluster of data points. Implementing comparisons of the distance between resulting centroids from each process and the size (count of points) of the clusters produced would likely reinforce this conclusion and further prove which initialization techniques produce more accurate clustering results.

In the future it would be interesting to perform a more exhaustive study on initialization effects by conducting a stripped k-means clustering comparison after hand-choosing the centers and comparing them with random on some pre-determined data sets such as the Iris data set used in study [1]. First by specifically choosing points that are extremities and comparing them to all of the different initializations mentioned here, then by specifically choosing centroids that are located in less dense areas, and finally choosing centroids in the 'optimal' tight, dense areas. Illustrating the worst case scenarios then comparing the outcomes may lead to new ideas or combinations of cluster seeding such as pre-processing via noise filtering prior to clustering.

7 References

[1] Ahmed, A. and Ashour, W. "An Initialization Method for the K-means Algorithm using RNN and Coupling Degree." International Journal of Computer Applications (0795-8887) Volume 25- No.1. (July 2011).

[2] Arthur, D. and Vassilvitskii, S. "K-Means++: The Advantages of Careful Seeding." Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics (Philadelphia, USA, 2007). 1027-1035.

[3] Bahman, B., Moseley, B., Kumar, R. Vattani, A., and Vassilivitskli, S. "Scalable K-Means++". Proceedings of the VLDB Endowment. Volume 5, Number 7. (Istanbul, Turkey, 2012), 622-633.

[4] Bubeck, S, Melia, M., and Luxburg, U. "How The Initialization Affects The Stability of the K-Means Algorithm". EDP Sciences. SMAI (2012). PS 16 436-452. http://www.esaun-ps.org.

[5] Joshi, K. and Nalwade, P. "Modified K-Means for Better Initial Cluster Centres". International Journal of Computer Science and Mobile Computing. Volume 2, Issue 7 (July 2013) 219-223.

[6] Qiao, J. and Lu, Y. "A New Algorithm for Choosing Cluster Centers for K-Means". Proceedings of 2nd International Conference on Computer Science and Electronics Engineering (Paris, France, 2013). Atlantic Press. 0527-0530.

Application LabVIEW® - based Gain Scheduling Programming to a 6-axis Articulated Robot considering Kinematic Analysis

A. Seong Bhin Kim[1], B. Won Jee Chung[1], and C. Jun Hyeong Bae[1]
[1]School of Mechatronics, Changwon National University, Changwon, The Republic of Korea

Abstract –

While industrial robots are coming into wide use, the control techniques of the robots are being developed as their performance is being enhanced. Specially, the dynamic performance of a 6-axis articulated industrial robot is greatly changed according to the position and orientation of the robot. This means that the PI parameter tuning of the robot's servo controllers should be tuned considering the dynamic characteristics of robot mechanism. The research performed in this paper is LabVIEW® programming to perform automatically parameter scheduling for various robot motions. Using forward and inverse kinematics of RS2, we can divide the working envelope of RS2 into 24 subspaces. Then we will perform gain tuning according to each subspace. Finally we do program the actual gain scheduling in which the optimized gain tuning for each subspace to be passed should be changed for various robot motions using LabVIEW®.

Keywords: Forward kinematics, Inverse kinematics, Subspace, Gain tuning, Gain scheduling, RS2, LabVIEW® Programming, Articulated robot.

1 Introduction

Nowadays the applications of industrial robots are spreading to a great extent so that various demands for industrial manipulators are increasing. While industrial robots are coming into wide use, the control techniques of the robots are being developed as their performance is being enhanced. [1] Specially, unlike Cartesian Robot and SCARA (Selective Compliance Assembly Robot Arm) which have wide application in assembling electronic parts, the dynamic performance of a 6-axis articulated industrial robot is greatly changed according to the position and orientation of the robot. This means that the PID (Proportional-Integral-Derivative) gain tuning of the robot's servo controllers should be tuned considering the dynamic characteristics of robot mechanism. It is well known that PID gain tuning can reduce the vibration phenomena of a robot so as to improve the performance of positional control. [2]

As one of previous studies, Kim *et al.* [3] has presented "Application of Gain Scheduling Technique to a 6-Axis Articulated Robot using LabVIEW®" in CSC 2014. In this paper, a *pseudo* gain scheduling using LabVIEW® has been performed for a 6-axis articulated *(lab-manufactured)* robot (*called as 'RS2', see Fig. 1*). In contrast to ref. [1], for accurate gain tuning of RS2 with less noise, Kim *et al*. [3] has utilized a program routine of DSA (Dynamic Signal Analyzer) [4] for frequency response method using LabVIEW®. Then robot transfer functions can be obtained experimentally using frequency response method with DSA program. Data resulted from the robot transfer functions has been transformed into Bode plots, based on which an optimal gain tuning has been executed. Also another contribution of the paper is that gain tuning can be performed according to the three positions (zero position, A position, B position) of robot's end-effector in workspace.

In this paper, we will incorporate kinematics analysis including forward and inverse kinematics of RS2 into a *psudo* gain scheduling. This leads to an actual gain scheduling based on LabVIEW® programming. Forward kinematics program calculates the position and orientation of end-effector corresponding to input angle of each joint. In the meanwhile, inverse kinematics program calculates joint angles corresponding to input values of 6 DOF($X, Y, Z, \alpha, \beta, \gamma$). Using forward and inverse kinematics of RS2, we can divide the working envelope of RS2 into 24 subspaces. Then we will perform gain tuning according to each subspace *with a great efforts*. Finally we do program the actual gain scheduling in which the optimized gain tuning for each subspace to be passed should be changed for various robot motions using LabVIEW®.

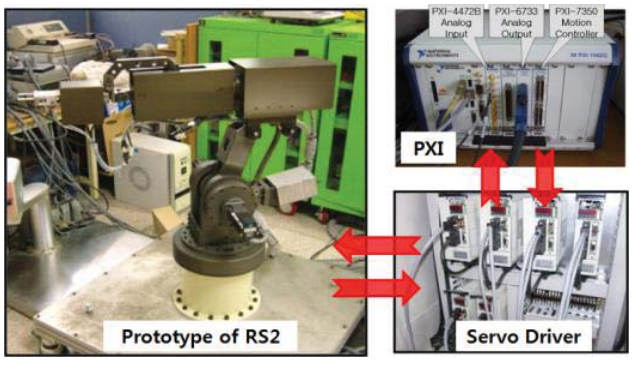


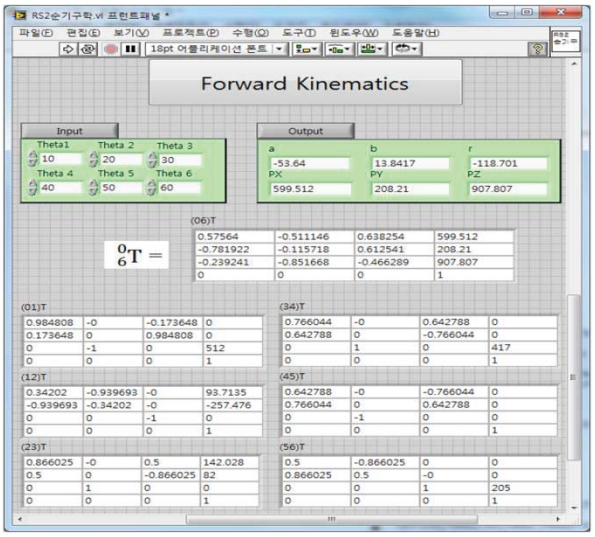
Fig. 1 RS2 System

2 Kinematic Analysis of RS2

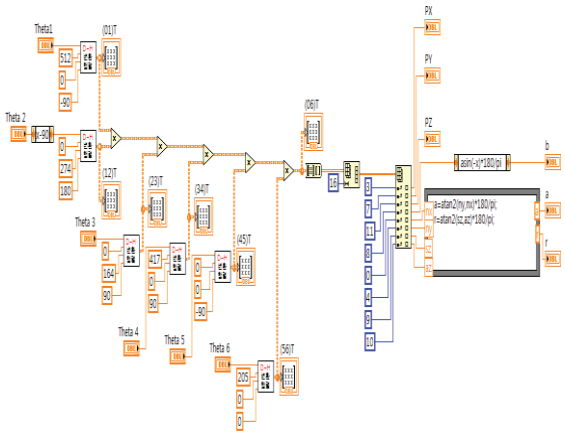
In forward kinematics, the length of each link and the angle of each joint are given and we have to calculate the position of any point in the robot. Specifically, forward kinematics is the computation of the position (X, Y, Z) and orientation (α, β, γ) of robot's end-effector. The orientation (α, β, γ) of robot denotes Euler angles [5]. In inverse kinematics, the length of each link and position of the point are given and we have to calculate the angle of each joint.

In the previous paper [6] of our lab, we had solved forward and inverse kinematics solution for RS2. In this paper, we just show the LabVIEW® graphical program, based on forward and inverse kinematics solutions for RS2. Forward kinematics program calculates the position and orientation of end-effector corresponding to input angle of each joint through the homogeneous transformation matrix ${}^0_6T$ as shown in Fig. 2 The advantage of developed program is that the homogeneous transformation matrix has been easily calculated only by modifying input angles. This forward kinematics routine of LabVIEW® is often called in the interpolation programs for RS2.

In the meanwhile, inverse kinematics program calculates joint angles corresponding to input values of 6 DOF$(X, Y, Z, \alpha, \beta, \gamma)$. Figure 3 shows a part of source routine of inverse kinematics for RS2, written in LabVIEW® graphical program. The inverse kinematics program is linked to interpolation programs as Sub VI type (in a format of subprogram LabVIEW®) for both dynamic simulation of interpolation and real implementation of interpolation on RS2. In the interpolation program, the inverse kinematics program calculates the angle of each joint every sampling time (a few milliseconds). Especially the results of inverse kinematics program play an important role in generating command values of joint angles for NI PXI motion controller of LabVIEW®.

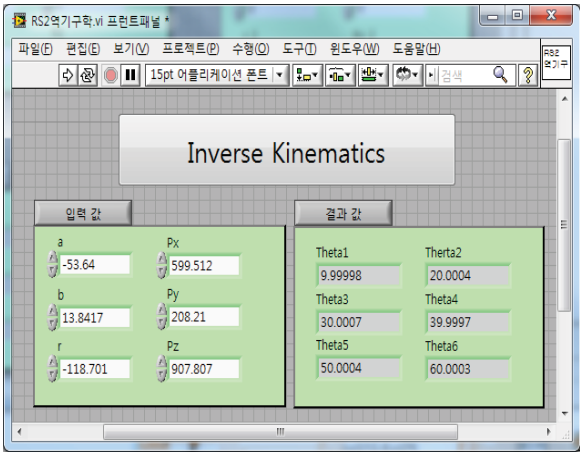


(a) Print Panel

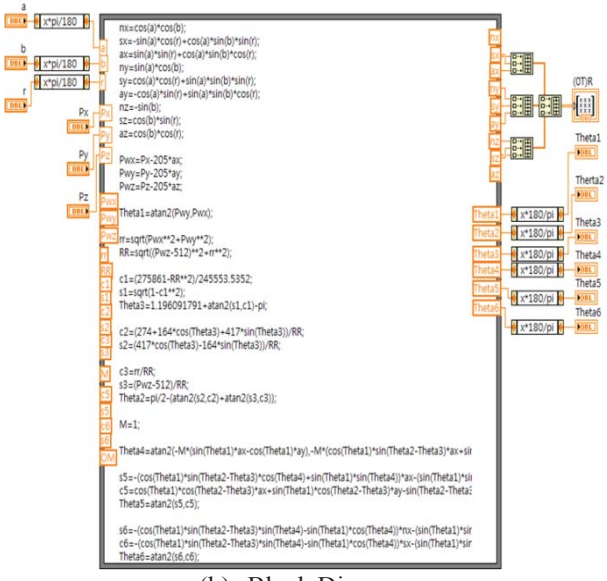


(b) Block Diagram

Fig. 2 Forward kinematics program



(a) Print Panel



(b) Block Diagram

Fig. 3 Inverse kinematics program

3 Kinematic Analysis of RS2

3.1 Proportional Gain (K_V) of Velocity Control Loop

For gain tuning of PID gains (K_v, K_i, and K_p), the LabVIEW® DAQ (Data AcQuisition) equipment is connected with the 6th (*i.e.*, the last) axis motor driver nearby the end effector of robot. First, an arbitrary value of proportional gain has been set for the motor driver. Then an appropriate value of the sine wave with amplitude *X* is selected according to ref. [7]. At this time, an integration effect has been eliminated by setting the integration time constant at 1000.[8] Finally frequency response test is conducted as follows: A sine wave of 0.5 V_{rms} (root mean square of voltage) from 2Hz through 500Hz is applied to the speed command pin of a servo driver as a source wave form from PXI-6733 of LabVIEW® DAQ; a Bode plot ($G_c(s)$) of a closed loop can be extracted using the programmed DSA.

We made use of the relationship between the closed loop transfer function $G_c(s)$ and the open loop transfer function $G_0(s)$ which is given by

$$G_0(s) = \frac{G_c(s)}{1 - G_c(s)} \tag{1}$$

Using Eq. (1), we can obtain the Bode plot of open loop transfer function. Using the Bode plot of open loop transfer function before gain tuning, gain margin (point where phase becomes 180 degrees) and phase margin (point where magnitude is 0) are obtained, as shown in Fig. 4.

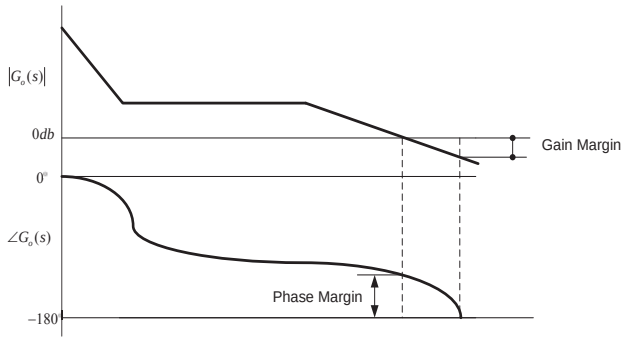


Fig. 4 Bode diagram of open loop transfer function

In general, according to Nyquist stability [9], gain margin should be -6dB ~ -20dB, while phase margin should be larger than 45 degree. Kim *et al*. [10] has suggested that the gain margin close to -6dB is better while it is between -6dB and -20dB. In this case, if we put the gain margin to be -6dB, we can obtain the new proportional gain of velocity loop K'_V, as follows :

$$\begin{aligned} &20\log x = (-6dB) - (-a) \\ &x = 10^{\frac{-6-(-a)}{20}} \\ &K'_v = xK_v \end{aligned} \tag{2}$$

where *a* denotes an arbitrary gain margin.

Use at least 2 centimeters (0.75 inch) for the left and right margins. Leave a 0.6 centimeters (0.25 inch) space between the two columns in the center of the page. Use font size (character size) 10 for text. The text should be prepared with single line spacing. *Do not use bold in the main text. If you want to emphasize specific parts of the main text, use italics.* Leave at least 2.0-2.5 centimeters margin at the page head (top of each page) for placing final page numbers and headers (final page numbers and running heads will be inserted by the publisher). Select a standard size paper such as A4 (210 X 297 mm) or letter (8.5 X 11 in) when preparing your manuscript.

3.2 Integral Gain (K_I) of Velocity Control Loop

The integral gain of velocity control loop, K_i, can be obtained from the integration time constant. Using the block diagram between input X (velocity command – velocity feedback) and output Y (current command), the transfer function G_{vo} can be obtained by

$$\begin{aligned} G_{vo} &= \frac{Y}{X} = \left(\frac{K_i}{S} + K_v\right) \\ &= K_v\left(\frac{\frac{K_v}{K_i}s + 1}{\frac{K_v}{K_i}s}\right) = K_v\left(\frac{T_iS + 1}{T_iS}\right) \end{aligned} \tag{3}$$

Figure 5 illustrates the Bode plot of Eq. (3). It can be noticed in Fig. 5 that phase margin becomes zero at the frequency of $10/T_i$ (so-called, *gain crossover frequency*). We need that the phase margin obtained from the proportional gain tuning of velocity control loop should not be changed even though an integrator would be used in the proportional gain tuning of velocity control loop. Thus we find out the integration time constant, *i.e.*, T_i, by making 1/10 times of the gain cross over frequency($10/T_i$) in Fig. 5 and then have its reciprocal.[11] Now, the integral gain K_i can be resulted from Eq. (4) as follows :

$$K_i = \frac{K_v}{T_i} \tag{4}$$

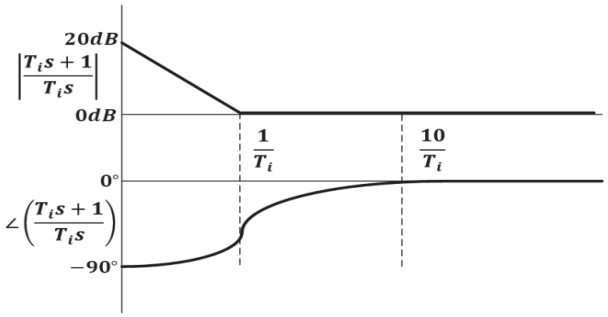


Fig. 5 Bode diagram of open loop transfer function

3.3 Proportional Gain (K_P) of Position Control Loop

Now it is ready to find out the proportional gain (K_p) of the position control loop shown in Fig. 6. By figuring out the closed loop frequency f_c as the -3dB frequency and putting the value of ζ (damping ratio) as 0.707 (usually given by experiments) in this paper, then K_p can be obtained by

$$K_p = \frac{\pi f_c}{2\zeta^2} \quad (5)$$

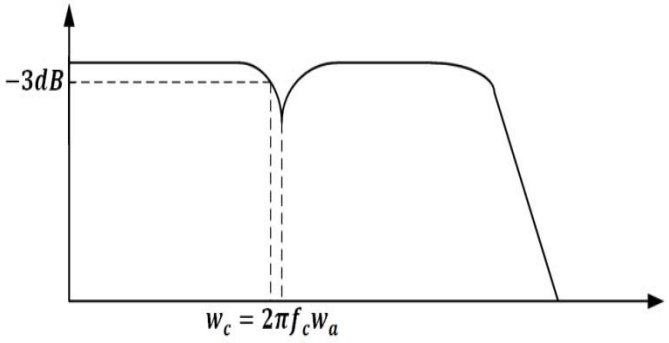


Fig. 6 Bode diagram of closed loop transfer function

4 Gain Scehduling Programming

4.1 Definition of Subspace

Workspace, or working envelope, refers to the area that a robot's end effector can reach. The workspace varies with robots' configurations, joints and link lengths. That is, each robot's configuration characterizes the shape of its working envelop, which can be created with the mathematical equation of motion defining robots' joints and links including each joint's range of motion. In this paper, partial ranges of a working envelope as shown in Table I are used for gain scheduling, by defining the ranges of motion in line with joint angles found in previous kinematic analyses of RS2.

Table I Partial Ranges of Joint Angles for Gain Scheduling

Joint	θ_1	θ_2	θ_3
Range	$-5° \sim 35°$	$-25° \sim 5°$	$-5° \sim 15°$
Joint	θ_4	θ_5	θ_6
range	$-5° \sim 5°$	$-5° \sim 5°$	$-5° \sim 5°$

Values of gain tuning depend on a robot's position and orientation. Thus, to implement gain scheduling, each subspace on the path of a robot's movement requires gain tuning. Therefore, the robot's working envelope assigned with reference to each rotational joint angle was sub-divided into 24 subspaces. With each joint's movable angle being split by 10°, θ_1, θ_2 and θ_3 are split into 4, 3 and 2, respectively, while θ_4, θ_5 and θ_6 are not split. These divisions lead to 24 subspaces.

Table II 24 Subspaces depending on joint angles

No.	θ_1	θ_2	θ_3
1	−5°~5°	−5°~5°	−5°~5°
2	−5°~5°	−5°~5°	5°~15°
3	−5°~5°	−15°~ − 5°	−5°~5°
4	−5°~5°	−15°~ − 5°	5°~15°
5	−5°~5°	−25°~ − 15°	−5°~5°
6	−5°~5°	−25°~ − 15°	5°~15°
7	5°~15°	−5°~5°	−5°~5°
8	5°~15°	−5°~5°	5°~15°
9	5°~15°	−15°~ − 5°	−5°~5°
10	5°~15°	−15°~ − 5°	5°~15°
11	5°~15°	−25°~ − 15°	−5°~5°
12	5°~15°	−25°~ − 15°	5°~15°
13	15°~25°	−5°~5°	−5°~5°
14	15°~25°	−5°~5°	5°~15°
15	15°~25°	−15°~ − 5°	−5°~5°
16	15°~25°	−15°~ − 5°	5°~15°
17	15°~25°	−25°~ − 15°	−5°~5°
18	15°~25°	−25°~ − 15°	5°~15°
19	25°~35°	−5°~5°	−5°~5°
20	25°~35°	−5°~5°	5°~15°
21	25°~35°	−15°~ − 5°	−5°~5°
22	25°~35°	−15°~ − 5°	5°~15°
23	25°~35°	−25°~ − 15°	−5°~5°
24	25°~35°	−25°~ − 15°	5°~15°

4.2 Gain Tuning in Subspace

Here, gain tuning is performed in each of 24 subspaces. The gain tuning performed in *subspace 1* in which the robot configuration (θ_1 = 0°, θ_2 = 0°, θ_3 = 0°, θ_4 = 0°, θ_5 = 0°, θ_6 = 0°) is selected for this purpose is described below. First, the bode plot of closed loop transfer function of the 6th axis (as shown in Fig. 7) is extracted and converted to the bode plot of open loop transfer function of the 6th axis (as shown in Fig. 8), using Eq. (1). Using the bode plot of the open loop transfer

function, the gain and phase margins are yielded. The gain and phase margins are -14.4dB and 52.3 deg., respectively, both of which come under safety ranges. Yet, to improve responsiveness, a new value (K_v') for the axis six is yielded with Eq. (2), so that the gain margin becomes -6 dB. Second, the *gain cross over frequency* yielded from the open-loop bode plot is 58Hz, whilst the time constant T_i can be found according Eq. (6).

$$T_i = \frac{10}{Gain\ cross\ over\ frequency} \quad (6)$$

Hereby T_i can be obtained as is 0.52ms. Then, using Eq. (4), the value of K_i for the 6th axis can be found. Third, from the bode plot of closed loop transfer function of the 6th axis, the frequency (f_c) of the spot with a resonance point -3dB is found to be 19Hz. Then, the value of K_p for the 6th axis can be found according to Eq. (5). Following the same method, the values of K_v, K_i, and K_p for the other axes can be yielded. Table III shows these finidngs. As aforementioned, gain tuning is applied to the remaining 23 subspaces.

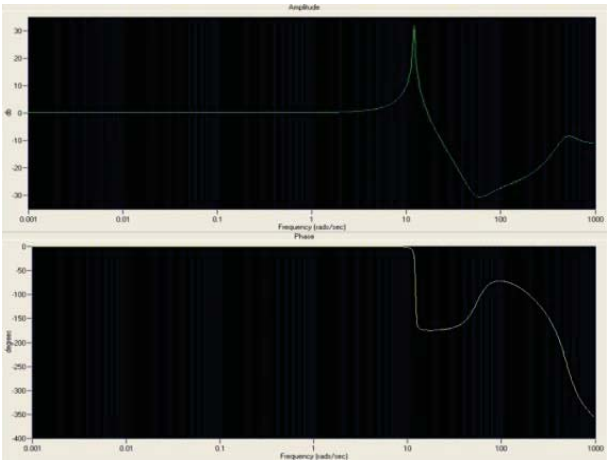


Fig. 7 Bode plot of closed loop transfer function of the 6th axis

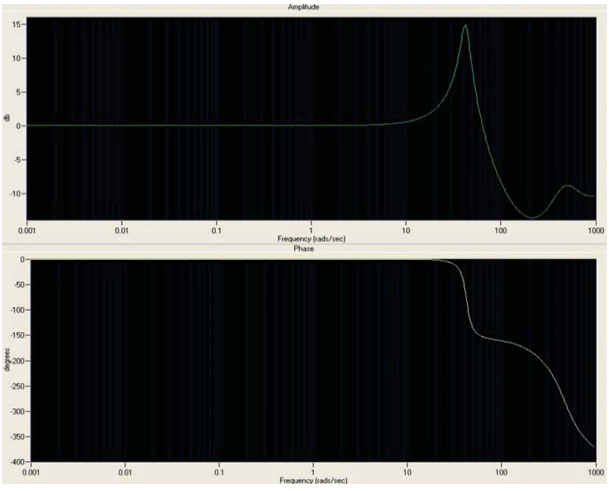


Fig. 8 Bode plot of open loop transfer function of the 6th axis

Table III. The Values of gain tuning for subspace 1

PID Gain tuning	Axis					
	1st	2nd	3rd	4th	5th	6th
K_v	64	82	141	77	89	132
K_i	256	713	1698	576	685	763
K_p	60	478	512	57	698	679

4.3 Gain Scheduling Programming

The gain scheduling program shown in Fig. 9 calculates each joint's angle upon an end-effector coordinates being placed by using inverse kinematics routine of Fig. 3 for RS2. Then RS2 moves to the end-effector coordinates given. While moving to the end-effector coordinates, the values of PID gains for each of the 24 subspaces which comes within the moving path can be applied.

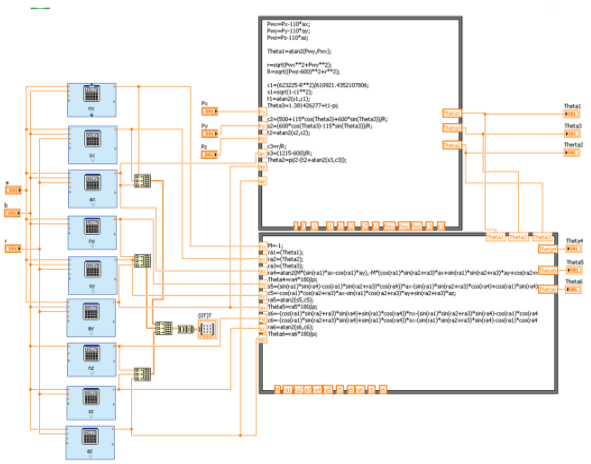


Fig. 9 Gain scheduling programming

4.4 Velocity Response

To verify the performance of the gain scheduling programming based on LabVIEW®, with the velocity commnad 0.5Vrms (root mean square Voltage), the velocity response of each axis is measured before and after the application of gain scheduling programming, so as to compare response performances. The responsiveness of a random spot C on the path of RS2 moving from A to B (so-called coordinated motion in robotics) as shown in Fig. 10 is compared. Figure 11 shows the stimulus command level (reference), the velocity response level with gain scheduling programming (red in "after" in Fig. 11) and the velocity response level without gain scheduling programming (red in "before" in Fig. 11). As a result of Fig. 11, the response level with gain scheduling programming is much closer to the

stimulus command level, compared with the response level without gain scheduling programming.

Fig. 10 Coordinated motion on RS2

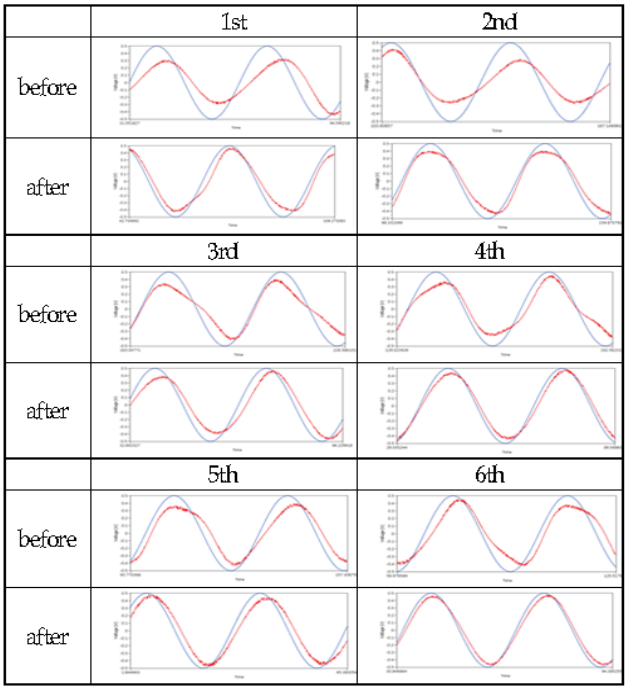


Fig. 11 Comparison of velocity response levels

5 Conclusion

The research performed in this paper is a LabVIEW® programming to perform automatically gain scheduling for various robot motion. We defined the ranges of motion in line with joint angles found in kinematic analyses of RS2 and then the robot's working envelope assigned with reference to each rotational joint angle was sub-divided into 24 subspaces. With each joint's movable angle being split by 10°, θ_1, θ_2 and θ_3 are split into 4, 3 and 2, respectively, while θ_4, θ_5 and θ_6 are not split. These divisions lead to 24 subspaces. To implement gain scheduling, each subspace on the path of a robot's movement requires gain tuning. Finally, the gain scheduling program calculates each joint's angle upon an end-effector coordinates being placed by using inverse kinematics routine for RS2. While moving to the end-effector coordinates, the values of PID gains for each of the 24 subspaces which come within the moving path can be applied. To verify the performance of the gain scheduling programming based on LabVIEW®, with the velocity commnad, the velocity response of each axis was measured before and after the application of gain scheduling programming, so as to compare response performances. As a result, the response level with gain scheduling programming is much closer to the stimulus command level, compared with the response level without gain scheduling programming. The findings of this study suggest optimal control of industrial robots performing

6 References

[1] Hyo Gon Kim, Won Jee Chung, Sung Gyu Park, Gyu Tak Kim, "Development of Experimental Gain Tuning Technique for 6-Axes Articulated Robot Manipulator Based on Bode Plot", WMSCI06, 2006

[2] Chang Doo Jung, Won Jee Chung, Dong Sun Lee, "Application of SolidWorks®, and Lab View®-based Simulation Technique to Gain Tuning of a 6-axis Articulated Robot", csc12, 2012

[3] Man Su Kim, Won Jee Chung, Seung Won Jeong, "Application of Gain Scheduling Technique to a 6-Axis Articulated Robot using LabVIEW®", csc14, 2014

[4] HEWLETT PACKARD, 1991, HP 35665A Dynamic Signal Analyzer Concepts Guide.

[5] MITSUBISHI, General-Purpose Interface MR-J2S- A Servo Amplifier Instruction Manual.

[6] Herbert Goldstein, Classical Mechanics (2nd ed.), Reading, MA: Addison-Wesley, ISBN 978-0-201-02918-5

[7] Jin Su Ahn, Won Jee Chung, "A Study on 6-Axis Articulated Robot Using a Quaternion Interpolation," KSMTE of Spring Conference 2010, pp 294~300, 2010Kuo, B. C., 1991, AUTOMATIC CONTROL SYSTEMS, Prentice-Hall, Inc. pp. 747~762.

[8] F. Haugen, "PID control of dynamics systems", int'1 specialized book service, 2004, pp.342-349

[9] B.C. Kuo, "Automatic control system", Prentice –Hall. 1991, pp 448-467

[10] Jung Hyun Kim, Won Jee Chung, Hyo Gon, Kim, "Prototyping and Visualization Techniques of 3-axis SCARA Robot Using DOE and LabVIEW®", MSV07, 2007

[11] Katusuhiko Ogata, Modern Control engineering, Prectice-Hall, Inc, 1990

The Interaction Between Ventilation and Natural Convection Flows in a Two-Dimension Enclosure

Tran Van Tran, Nguyen Thi Thuy
Department of Mechanics, Hanoi University of Science, 334 Nguyen Trai Road, Hanoi, Vietnam

Abstract. *Ventilation air flow in an enclosure is often unsteady (turbulent) at even very low Reynolds number (Re). Meantime natural convection in a box is stationary motion at large enough Rayleigh number (Ra). This paper deals with the interaction between two those flows in a two-dimensional room. The room has one inlet and two outlets. A linear heat source locates at the middle of the floor. The numerical simulation of the interaction is carried out at some values of Re and Ra. Some interesting characteristics of the resultant flow are discovered. The heat amount released by the source and that removed from the room by different types of this flow is also provided.*

Keywords: *Ventilation, convection, interaction, heat source, inlet, outlet.*

1. Introduction.

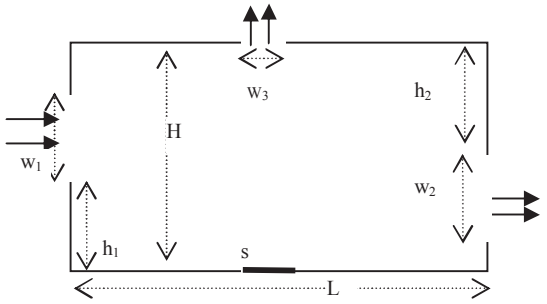
Ventilation of living or working enclosed spaces has a permanent practical interest. Removal of heat from a large industrial factory or small electronic equipment by ventilation has been a challenging problem for both theoretical investigation and practical application. An overview on studies related to this subject is done in the work of Qingyan Chen [1].

Natural convection in various enclosed spaces caused by heating or cooling boundary or inside heat sources has been investigated intensively since publishing Chandrasekhar's book [2]. Many of these studies have concerned the instability of such convective motion. Later researchers have paid more attention to applied aspect of the problem by considering natural or mixed convection in living or working enclosed spaces [3], [4], [5], [6]. The main concern of such investigations is to calculate the velocity field evoked by the convection as well as the temperature distribution in the enclosure. In paper [7] a warm air is supplied to a two-dimension room at moderate rates to find the better inlet-outlet location in order to create a good temperature distribution for living condition. Meantime the purpose of study [8] is defining the efficiency of contaminant removal from an enclosure by clean air flow.

The target of this paper is studying the interaction between ventilation and convection to determine what kind of the resultant flow (steady, periodic and quasi-periodic or turbulent) will occur in dependence on the relationship between the air supply rate (*Re*) and the intensity of the heat source (*Ra*) for a concrete enclosure.

2. Problem formulation.

Consider a two-dimensional room 6 m long (*L*) and 3 m height (*H*) as indicated in Fig. 1. A linear homogeneous isotemperature heat source (*s*) of length 0.6 m locates at the middle of the floor. Taking the inlet velocity *U*, the room height *H*, the ratio *H/U* and the difference between the temperature of the heat source and that at the inlet ΔT as characteristic values for velocity, length, time and temperature respectively one can obtain the following governing equations for laminar Navier-Stokes



$w_1 = w_2 = 1.2$, $w_3 = s = 0.6$, $h_1 = h_2 = 1.2$ (m)

Fig. 1 Configuration of the room inlet and outlet

system under Boussinesq approximation with constant physical properties:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \qquad (1)$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \qquad (2)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{Ra}{\mathrm{Pr}.\mathrm{Re}^2}T \qquad (3)$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \frac{1}{\mathrm{Re}\,\mathrm{Pr}}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right) \qquad (4)$$

where *u, v, p* and *T* are non-dimensional components of velocity, pressure and temperature respectively. The dimensionless values in equation (2), (3) and (4) are the Reynolds number, the Rayleigh number, the Grashof number and the Prandtl number respectively:

$$\mathrm{Re} = \frac{UH}{\nu},\ \mathrm{Ra} = \mathrm{Gr}/\mathrm{Pr},\ \mathrm{Gr} = \frac{g\beta\Delta T H^3}{\nu^2},\ \mathrm{Pr} = \frac{\nu}{\chi} \qquad (5)$$

where β, ν, χ are the coefficient of thermal expansion, the kinematical viscosity and the thermal diffusivity of liquid respectively. The boundary conditions are set as follows:

On the walls: $u = v = T = 0$ (6)

At the inlet: $u = 1, v = T = 0$ (7)

At the right outlet: $v = 0, u = w_{in} / \left(w_{out1} + w_{out2} \right)$ (8)

At the ceiling outlet: $u = 0, v = w_{in} / \left(w_{out1} + w_{out2} \right)$ (9)

At the heat source: $u = v = 0, T = 1$ (10)

where $w_{in}, w_{out1}, w_{out2}$ are the width of the inlet, the right outlet and ceiling outlet respectively. Conditions (8) and (9) are imposed to hold the mass conservation of the air flow in the Bousineqq approximation of the model.

Following [9] the condition for temperature at the outlets is taken as follows:

$$T_i^n = \overline{T}_i^{n-1} \quad (11)$$

where T_i^n is the temperature at the *n*-th time step of integration, *i*=1,2 in the accordance with the right and ceiling outlet, $\overline{T}_i^{n-1}$ is the average temperature at the *i*-th outlet on the previous time step. Note that condition (11) *at the outlets*, as indicated in [9], is more adequate than the heat zero-flux usually set by other researchers. Indeed, the condition that the first derivatives of all parameters of a flow vanish on a cross section is usualy used as a computational condition on a boundary far from an object past by the liquid, where the flow can be assumed uniform. So here it is hard to suppose the ventilating flow is uniform right at the outlets of the enclosure. On the other hand, condition (11) also helps to calculate the removal heat through the every outlet at any time step.

3. Numerical Method

In this paper the solution of governing equations (1)-(4) with boundary condition (6)-(11) and the initial condition

$$u = v = T\big|_{t=0} = 0 \quad (12)$$

is calculated by FEM. The crucial idea of the FEM procedure applied in this study is CBS (Characteristic Based Split) scheme first proposed by Zienkiewics and Cordina [10] and further developed by Zienkiewicz et al. [11] and by Nithiarasu [12]. This procedure is presented in a very good detail in [13]. To increase the accuracy and decrease the computational time a mesh with finer resolution in the vicinity of the inlet, outlets and the heat source is used.

4. Results and discussion

For recording the change in time of the flow parameters such as *u, v,* and *T* here seven points are chosen in the computational domain. They are numbered as follows:

$$P1\left(1.5, 0.5\right), P2\left(1.5, 0.3\right), P3\left(1.5, 0.8\right), P4\left(1, 0.8\right)$$

$P5\left(0.5, 0.8\right), P6\left(1, 0.3\right)$ and $P7(0.5, 0.3)$. So point *P*1 is the centre of the enclosure. Now firstly consider the ventilation flow without any heat source in the enclosure. Fig. 2 shows the velocity field at *Re*=20. Obviously, even at low Reynolds number the flow is not smooth. There are several vortices exist. These vortices make the flow unsteady as shown in Fig. 3 where *u*-component (Fig. 3a) and *v*-component of the velocity (Fig. 3b) at point *P*1 are presented. The velocity at all six remain points has the oscillating character similar to that in Fig. 3. It seems that for the room chosen here a steady ventilated flow without buoyancy can not exist at any rate of the air supply.

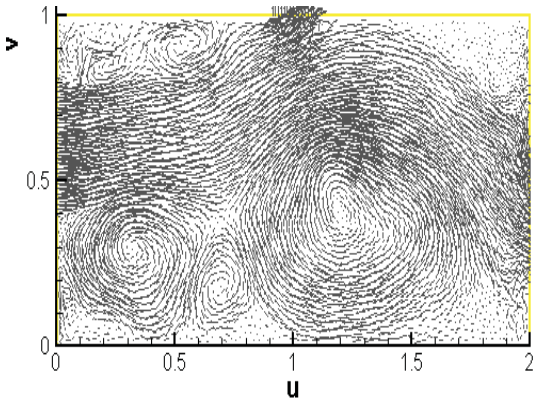
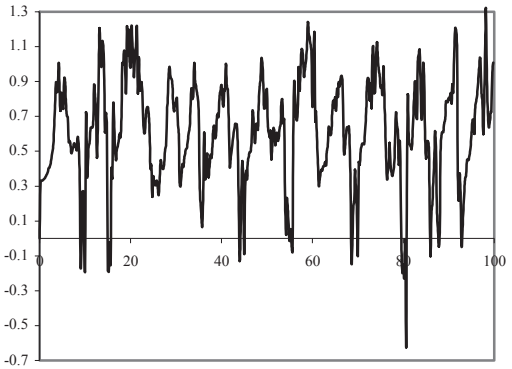
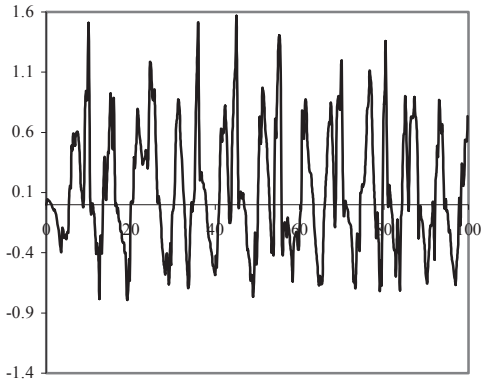


Fig.2. velocity field at *Re*=20



(a): u-velocity component at P1



(b): v-velocity component at P1

Fig.3. the change of velocity at point P1, *Re*=20

Now let consider the natural convection caused by only heat source (*s*) on the floor as indicated in Fig.1. In this case the room is an enclosure without any inlet and outlet. Fig. 4 shows the air velocity (*a*: *u* - component, *b*: *v* - component) at the seven chosen points for $Ra=10^6$. It is obvious that the convection becomes steady after a

short time interval after its onset. The properties of the convective motion at $Ra=10^7$ is fully different than those at $Ra=10^6$. As shown in Fig. 5, the natural convection for $Ra=10^7$ is actually unsteady motion. The difference between the steady convective motion at $Ra=10^6$ and unsteady convection at $Ra=10^7$ is more obvious in Fig. 6 where the temperature at point P1 (*a*) and at point P7 (*b*) is given for three values of *Ra*. This difference is also reflected clearly in Fig. 7 where the velocity field and the temperature distribution are presented for $Ra=10^6$ and $Ra=10^7$. Both the velocity and temperature fields are symmetrical at $Ra=10^6$ meantime this character is absent at $Ra=10^7$.

Now consider the air resultant motion induced simultaneously by ventilation and natural convection in the enclosure shown in Fig.1. As indicated above, the natural convection is steady at large enough *Ra* meantime the ventilated flow may be unsteady at any *Re*. So it is interesting to study the existence of the stationary resultant flow in the dependence on the relation between Reynolds number and Rayleigh one.

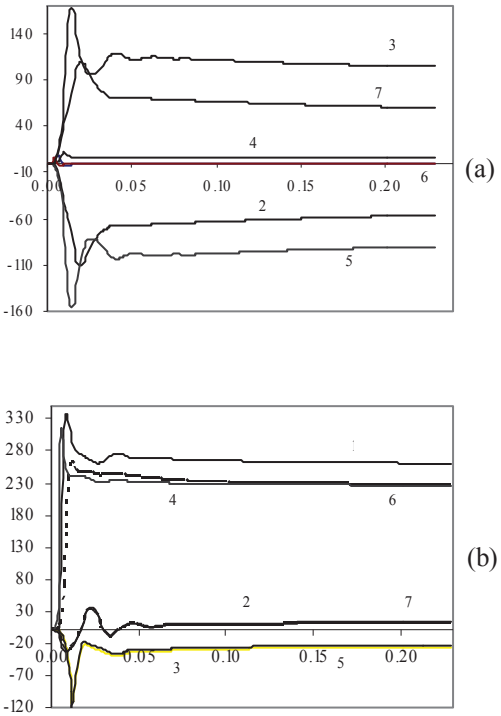


Fig.4. velocity at the seven chosen points for $Ra=10^6$

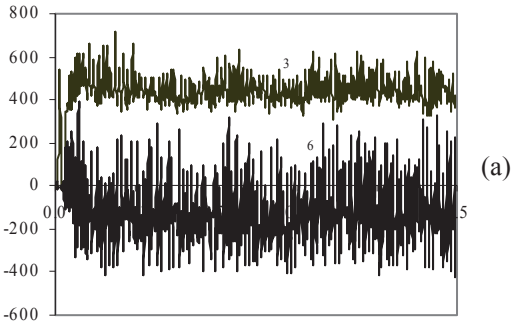


Fig.5a. u-velocity comp. at point P3 and P6 for $Ra=10^7$

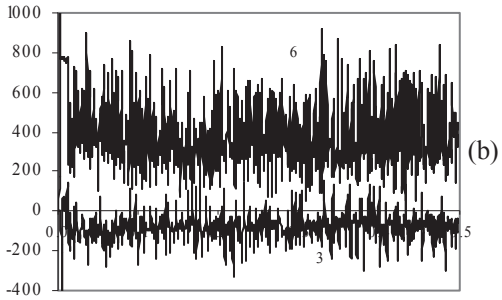


Fig.5b. v-velocity comp. at point P3 and P6 for $Ra=10^7$

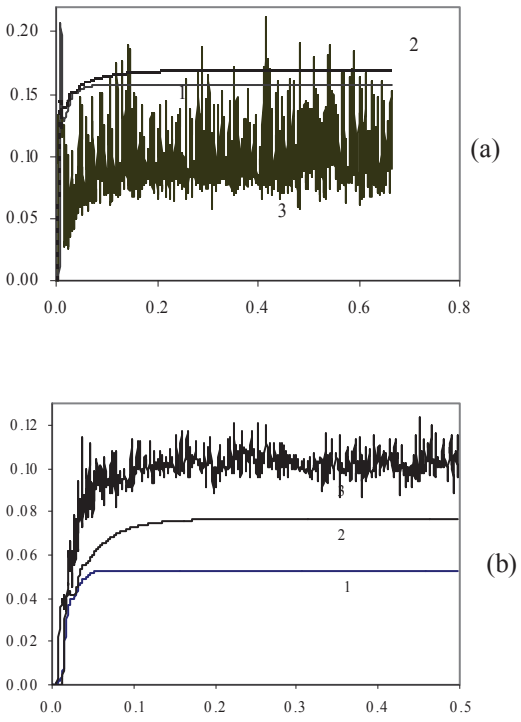


Fig.6. temperature at P1 (a), at P7 (b) (1:$Ra=10^5$, 2: $Ra=10^6$, 3: $Ra=10^7$)

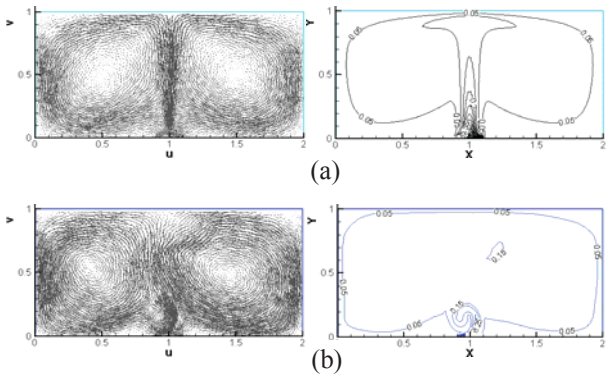


Fig.7. velocity field and temperature isolines: (a): $Ra=10^6$, (b): $Ra=10^7$

The simulation shows that (see Fig. 8) at $Re=10^3$ and $Ra=10^5$ the resultant air flow is strictly steady. The change in time of the velocity at point P1 for several values of *Ra* at $Re=10^3$ shown in Fig. 9 strongly proves

the existence of a stationary flow for $Ra=10^5$. It is very interesting to note that the solution corresponding to $Re=10^3$ and $Ra=10^7$ is exactly periodic. Some more properties of this solution are presented in Fig. 10. The variation in time of all the flow parameters at points P3, P4, P5 as well as the change of the heat flux from the source and that removed through the outlets shown in Fig.10 strongly indicate the periodic character of the solution at $Re=10^3$ and $Ra=10^7$. Note that in Fig 10.c, the temperature at point P5 is almost equal to zero at all the time so it is not shown there. Simulations also show that when Reynolds number Re exceeds 5.10^3 the resultant flow becomes unsteady at all values of Ra.

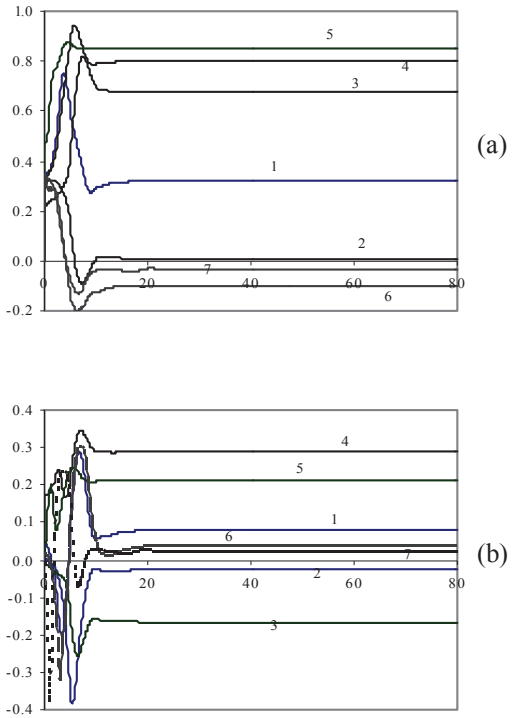


Fig. 8 (a): u-component, (b)$:$ v-component at seven points at $Re=10^3$, $Ra=10^5$

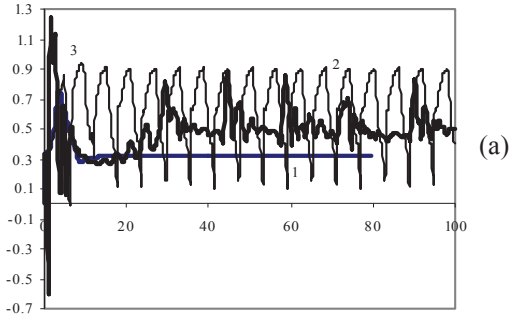


Fig. 9 (a): u-velocity component at P1 for $Re=10^3$, 1: $Ra=10^5$, 2: $Ra=10^7$, 3: $Ra=10^8$

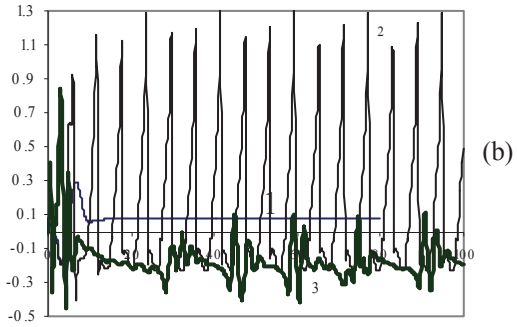


Fig. 9 (b): v- velocity component at P1 for $Re=10^3$, 1: $Ra=10^5$, 2: $Ra=10^7$, 3: $Ra=10^8$

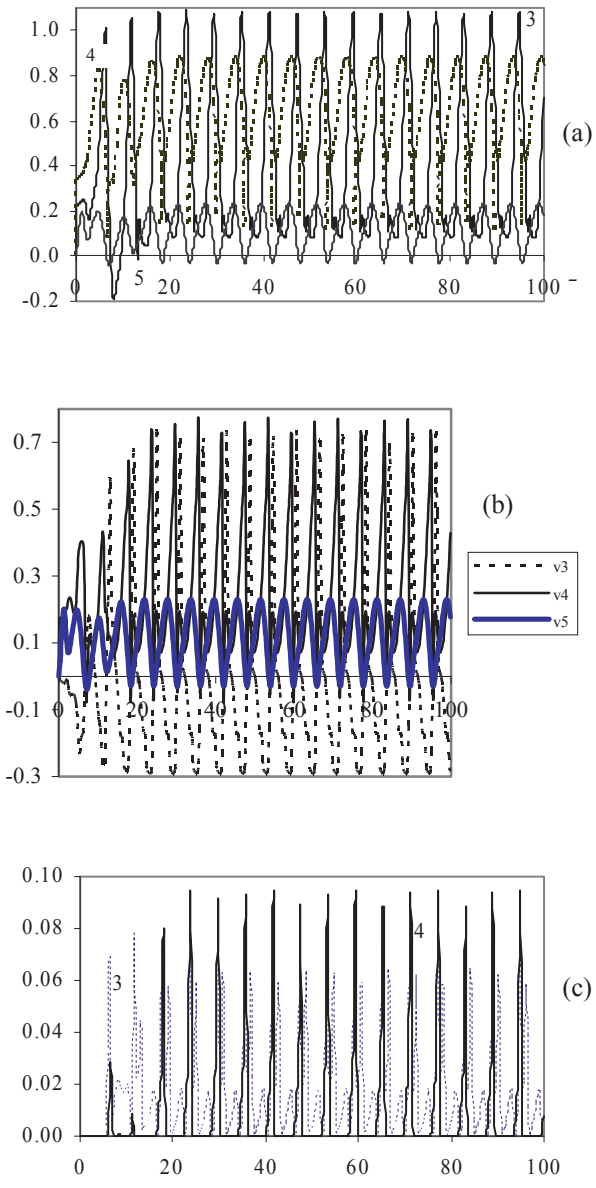


Fig. 10 (a): u-component, (b): v-component of velocity and (c): temperature in P3, P4, P5

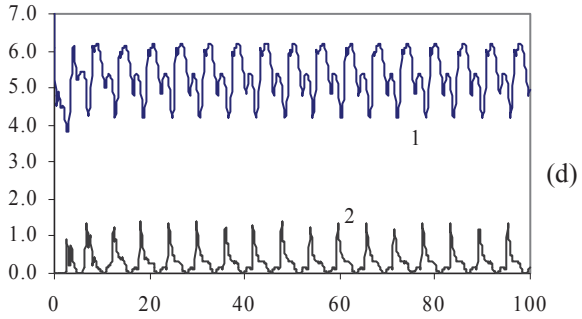


Fig. 10(d). 1: heat flux in and 2: heat flux out at $Re=10^3$, $Ra=10^7$

The 'efficiency' of the heat removal from the room in term of the ratio of the instant heat flux out through the outlets over the heat flux in released by the source is shown in Fig. 11. The general conclusion drawn from Fig. 11 is that the relative 'efficiency' decreases when Reynolds number based on the air supply increases. But this does not mean that the heat flux in and therefore the heat flux out is in inverse ratio to *Re*. On the contrary, both the heat flux in and heat flux out are directly proportional to *Re*. The depedence of the heat flux in on the Reynolds number and the Raleigh number is shown in Fig. 12. It is clearly from Fig. 12 that at the same *Re*, the larger *Ra* the bigger the heat flux becomes. Also, for a fixed *Ra* the heat flux increases with Reynolds number. This conclusion is in the accordance with the expectation in fact.

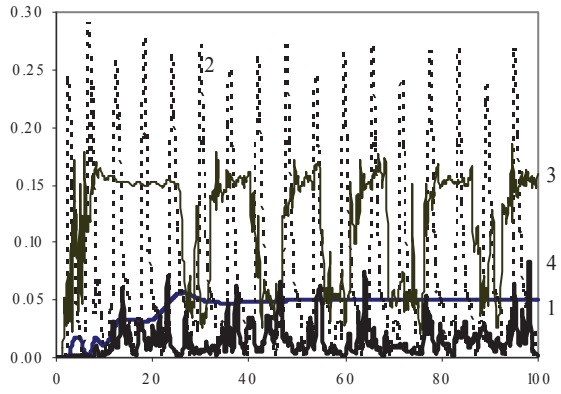


Fig.11. the ratio (heat flux in)/(heat flux out) 1: $Re=10^3$, $Ra=10^5$, 2: $Re=10^3$, $Ra=10^7$, 3: $Re=10^3$, $Ra=10^8$, 4: $Re=10^4$, $Ra=10^5$

The simulation shows that the natural convection (for which $Re=1$) is stationary until *Ra* approximately equal 6.10^6. For every *Re* not exceeding 5.10^3, there exists a limit value of *Ra* under which the resultant flow remains steady. This limit value decreases when the Reynolds number grows.

The region of the existence of steady resultant flows in (*Re*, *Ra*) plane as well as in their logarithm values is shown in Fig.13. This region in Fig. 13(a) of course does not include the left part from line $Re=1$ as well as the axis $Ra=0$. The calculation also shows that at *Re* larger than 5000, the flow is always unsteady.

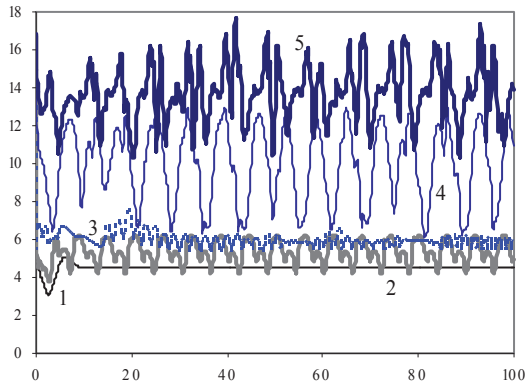
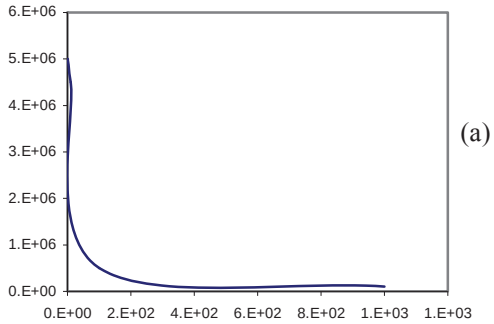


Fig.12 the released heat flux: 1- $Re=10^3$, $Ra=10^5$, 2- $Re=10^3$, $Ra=10^7$, 3- $Re=10^3$, $Ra=10^8$, 4- $Re=10^4$, $Ra=10^5$, 5- $Re=10^5$, $Ra=10^5$



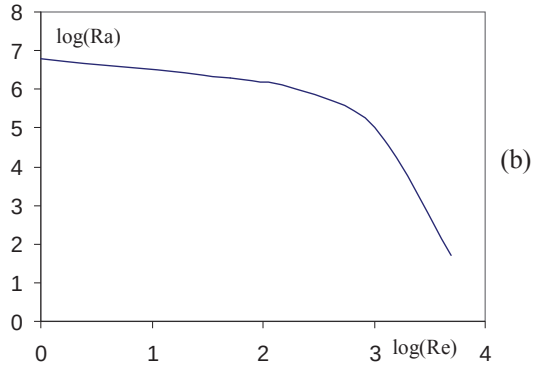


Fig. 13. The region of steady motion for the resultant air flow

5. Conclusion

In the framework of the Navier-Stokes equations with buoyancy in Boussinesq approximation, it is numerically shown that the air flow simultaneously caused by ventilation and natural convection in an enclosure may be steady, periodic or quasi-periodic. This depends on the relationship between the Reynlods number and the Rayleigh number as well as the location of the inlet and outlet of the enclosure. At a moderate *Re* there exists a limit of *Ra* under which the air flow has to be steady. When *Ra* exceeds this limit, the motion becomes periodic or quasi-periodic. At large *Re*, the flow as expected is fully unstationary or turbulent at any value of *Ra*. The region on plane (*Re*,*Ra*) where the air motion resulted from the interaction between the

ventilated flow and the natural convection caused by an inside heat source may be stationary is specific to every concrete enclosure with its determined size as well as the location of its inlet and outlet.

References

[1] Qingyan Chen. Ventilation performance prediction for buildings: A method overview and recent publications; Building and Environment, 44, 848-858, (2009).

[2] Chandrasekhar S. Hydrodynamic and Hydromagnetic stability; Oxford, Clarendon Press (1961).

[3] S. Ostrach. Natural convection in enclosure; Advances in Heat Transfer, vol. 8, Academic Press, NY (1972).

[4] T. Terai. Indoor Thermal Convection; Architectural Institution of Japan, Japan, p 63 (1959).

[5] Ratnakar Kulkarmi. Natural convection in enclosures with localized heating and cooling; PhD thesis, University Wollongong, Australia (1998).

[6] Goutam Saha et al. Natural convection in enclosure with discrete isothermal heating from below; J. Naval Architecture and Marine Engineering, 7, (2007).

[7] S.L. Sinha et al. Numerical simulation of two-dimensional room air flow with and without buoyancy; Energy and Buildings, 32, 121-129, (2000).

[8] J.L. Lage et al. Efficiency of transient contaminant removal from a slot ventilated enclosure; Int. J. Heat Mass Transfer, v32, 10, 2603-2615, (1991).

[9] T.V. Tran and N.T. Thuy. The effect of boundary conditions on the efficiency of heat or contaminant removal from a ventilated room; Vietnam J. Mechanics Vol. 37, 2,133-144, (2015).

[10] O. C. Zienkiewicz, R. Cordina. A general algorithm for compressible and incompressible flow, Part I, The split characteristic based scheme; Int. J. for Numerical Methods in Fluids, 20, 869-885 (1995).

[11] O. C. Zienkiewicz et al. An efficient and accurate algorithm for fluid mechanics problems. The characteristic based split (CBS) algorithm; Int. J. for Numerical Methods in Fluids, 31, 359-392, (1999).

12. Nithiarasu P. An efficient artificial compressibility (AC) scheme based on the characteristic based split (CBS) method for incompressible flows; Int. J. for Numerical Methods in Engineering, 56, 1815-1845, (2003).

[13] R. W. Lewis, P. Nithiarasu, K. N. Seetharamu. Fundamentals of the Finite Element Method for Heat and Fluid Flow. John Wiley & sons, Ltd (2004).

Understanding the K-Medians Problem

Christopher Whelan, Greg Harrell, and Jin Wang
Department of Mathematics and Computer Science
Valdosta State University, Valdosta, Georgia 31698, USA

Abstract *- In this study, the general ideas surrounding the k-medians problem are discussed. This involves a look into what k-medians attempts to solve and how it goes about doing so. We take a look at why k-medians is used as opposed to its k-means counterpart, specifically how its robustness enables it to be far more resistant to outliers. We then discuss the areas of study that are prevalent in the realm of the k-medians problem. Finally, we view an approach to the problem that has decreased its time complexity by instead performing the k-medians algorithm on small coresets representative of the data set.*

Keywords: K-medians; K-means; clustering

1. Introduction

The clustering problem is one well researched in the computer field due to its incredible variety of applications, be it unsupervised learning, geographic positioning, classifying, data mining, or other. K-medians and k-means are two widely popular approaches to performing this clustering task. Both involve finding k cluster centers for which the sum of the distance between a center and all points in that cluster is minimized. Where the two methods differ is in what they consider the "center" of the cluster to be. As one could infer by their names, k-means uses the mean (minimizing the 2-norm distances) while k-medians uses to median (minimizing the 1-norm distance). This paper will focus on the k-medians variation.

2. K-Medians

As mentioned above, the k-medians approach to clustering data attempts to minimize the 1-norm distances between each point and its closest cluster center. This minimization of distances is obtained by setting the center of each cluster to be the median of all points in that cluster. This section discusses why this is such a powerful method of clustering data, shows why it is a good alternative to the k-mean approach, and provides a brief overview of the k-medians algorithm to procure a better knowledge base concerning this topic.

2.1 Benefits over K-Means

The k-means problem was conceived far before the k-medians problem. In fact, k-medians is simply a variant of k-means as we know it. Why would k-medians be used, then, instead of a more studied and further refined method of locating k cluster centers? K-medians owes its use to robustness of the median as a statistic [1]. The mean is a measurement that is highly vulnerable to outliers. Even just one drastic outlier can pull the value of the mean away from the majority of the data set, which can be a high concern when operating on very large data sets. The median, on the other hand, is a statistic incredibly resistant to outliers, for in order to deter the median away from the bulk of the information, it requires at least 50% of the data to be contaminated [1].

Through its use of the median as the determining factor in placement of cluster centers, k-medians is able to assimilate the robustness that the median provides. Implementing variations of the k-medians method can further reduce the minimal shifts resulting from the presence of one of more outliers. Such variations include k-medians with outliers, in which points that exhibit attributes common with outliers are handle in a manner in which their distances will have a smaller effect on the positioning of the center, and robust k-medians with m outliers, which attempts to discard up to m points that the algorithm determines to be outliers.

2.2 K-Medians Algorithm

Given a set of points, the k-medians algorithm attempts to create k disjoint cluster that minimize the following equation. This means that the center of each cluster center minimizes this objective function [2].

$$Q\left(\{\pi_j\}_{j=1}^{K}\right) = \sum_{j=1}^{K} \sum_{x \in \pi_j} \|x - c_j\|_1$$

This minimization is defined by the geometric median.

$$\arg\min_{y \in R^n} \sum_{i=1}^{m} \|x_i - y\|_2$$

In order to begin this process, k initialization points must be selected as the cluster centers. The logic code below is then performed:

K-Medians Algorithm

```
Q = infinity
do
    for point in dataset
        min = infinity
        index = 0
        for i in k
            dist = distance(point, center[i])
            if dist < min
                min = dist
                index = i
        disjoint-sets.add(index, point)
    for i in k
        center[i] = median(disjoint-set.get(i))
    sum = 0
    for i in k
        for point in disjoint-set.get(i)
            sum = sum + distance(point, center[i])
    oldQ = Q
    Q = sum
while (oldQ - Q) > eps
```

The above code follows these steps

1. Assign each point in the data set to its closest center. The points assigned to the same center are then said to be in the same cluster, therefore they are added to the same disjoint-set. Because each point has been assigned to its closest center, the value of Q will not increase.
2. With the new disjoint-sets as the clusters, calculate their median to determine the updated value of that cluster's center. Because the center is a minimization of 1-norm distances, Q cannot increase as a result of this step.
3. Sum all distances between each point and its respective cluster center. This is the new value for Q.
4. If the improvements made by this iteration are less than a previously determined epsilon, quit. Otherwise, repeat the process.

Because both steps 1 and 2 can only decrease the overall value of Q, k-medians is a local optimization algorithm minimizing the sum of the 1-norm distances throughout the dataset [3].

3. Areas of Study

This section will cover a handful of topics that go into the k-medians algorithm that were abstracted away for simplicity. These topics include: median calculation, distance specifications, determining a value for k, and the initialization process for k cluster centers.

These topics have in no way been studied to completion, and many of the problems surrounding these topics have yet to be solved [3]. These subsections that follow will merely introduce these areas and current methods on how to approach them.

3.1 Finding the Median

K-medians uses the median as the statistic to determine the center of each cluster. It has been proven, however, that there exists no closed form that can determine the geometric median in every dimension. Because of this, methods of finding the median have turned to a more heuristic approach. We will now take a look at two of these methods, one that uses a simple simulated annealing algorithm, the other the more commonly implemented Weiszfeld's algorithm in order to understand the ideas surrounding median calculation [4].

3.1.1 Simulated Annealing

The simulated annealing approach is this general method

Simulated Annealing

```
step = arbitrary value
median = mean(points)
min = sum_distances(points, median)
improved = false
while step > eps
    for direction in directions
        temp_median = median, + (direction*step)
        d = sum_distances(points, temp_median)
        if d < min
            min = d
            median = temp_median
            improved = true
            break
    if !improved
    step = step / 2
```

In the above algorithm, "directions" is a list of all of the unit vectors of the dimension in the points are found in. The addition and multiplication shown are, by extension, vector addition and scalar multiplication respectively.

This method follows these steps

1. Initialize an arbitrarily large value for steps. Alternatively, this value could correspond to some statistical measurement of the points, be it variance, standard, deviation, median absolute deviation, or something else of the sort.

2. Initialize median to be the mean of the data set. This is just a starting approximation.
3. Check to see if moving the median in any direction by the value of step provides a better approximation for the median. If it does, update the median and continue.
4. If none of the movements improve our median, half the step size.
5. Stop once step has decreased below a predetermined value of epsilon

This method of approximation, while slow due to the constant calculation of the sum of distances (2d where d is the dimension in which the problem resides), will find a good median approximation with accuracy based on epsilon. The convex nature of the median guarantees that this method will not be trapped at a local minimum.

3.1.2 Weiszfeld's Algorithm

Weiszfeld's algorithm is as follows

Weiszfeld's Algorithm

```
median = mean(points)
for j, k
    m1 = {0}
    m2 = 0
    for point in points
        dist = distance(point, median)
        for i in m1.length
            m1[i] = m1[i] + point[i]
            m2 = m2 + (1/dist)
    median = m1/m2
```

1. Initialize median to mean. This is just an initial approximation.
2. Set up a variable to represent the sum of the points divided by the distance from that specific point to the approximated median (m1).
3. Initialize a variable to keep track of the sum of the inverted distances from each point to the approximated median (m2).
4. For each point, add the value of the point divided by the distance between it and the current median to m1. Add the inverse of the distance between it and the current median to m2.
5. The new approximation for the median is equal to the scalar division of m1 divided by m2

This method takes advantage of the alternate definition of the median [4]. We can see the similarities between this and this representation of Weiszeld's algorithm.

3.2 Different Forms of Distance

While Euclidean distance is the measure most commonly used when the k-medians algorithm is applied to a k-clusters problem, it is not always the appropriate choice to correctly model what the k-clustering is attempting to achieve. Many times, models of real world scenarios require certain restraints in distance measurement. One such deterrent may be the existence of obstacles prohibiting straight-line travel from one point to the next. In a situation such as this, Euclidean measurements are simply far too inaccurate, and other forms of distance, such as Manhattan distance, must be considered instead.

Manhattan distance is a measurement based on a grid system in which the points in question are placed. The concept is that in order to move from start to end point, one of four directions must be chosen for the point to advance: up, down, left, or right. Each decision will move the start point one unit in the chosen direction. The Manhattan distance is determined by the number of decisions required for the start point to reach the end point [5]. This method of distance can be extended to account for obstacles that may stand in between the two points. Luckily, this method is well suited for k-medians. The goal of k-medians is to minimizing the absolute deviations, which is in fact equivalent to the Manhattan distance.

These two measurements do not cover all scenarios, however. When using clustering to organize more categorical data, minimizing divergences, such as Bregman and Kullback-Leibler divergences [5], can be the desired outcome. Because of these different ways to determine the quantity of weight to be assigned to the edge between two points, the implementation of the k-medians algorithm must be shaped to appropriately handle the type of distance measurement best suited to model the problem at hand.

3.3 Selecting K

K-Medians clustering aims to separate a given data set into k clusters. The value of k, while important in the structure of the resulting k-cluster model, is given to the algorithm as input from the user. This value of k may be chosen based on assumptions of the data set, prior experience with like data set, or prior knowledge on the contents of the data set [6]. K, in this sense, is a fixed value that may or may not produce desired clustering, even if that clustering minimized the objective function. There do exist, however, approaches that can help approximate a better value for k. These approaches are discussed in this section.

One method takes advantage of the fact that in center based clustering, the clusters adhere to a unimodal distribution. In this case, that will be the Normal, or Gaussian, distribution [6]. The idea is that k will be initializing to an arbitrarily small value, for simplicity we will say 1. Are clusters are then analyzed for their distribution. If a cluster is shown to have a unimodal normal distribution, the cluster remains unchanged. If not, the current center is replaced with two

centers. After all clusters are analyzed, the clusters are recalculated using k-medians, and the process continues. The stopping point is reached when no cluster centers have to be broken into two [6].

Another approach find the value of k in a similar manner, but rather than splitting the centers until the desired distribution is found for each cluster, this method is supplied a range of potential k values and determines the best choice among the given options. It does so by creating a model scoring system based on Bayesian Information Criterion (BIC) [7]. This selection criterion provides a way to compare the distributions of the clusters resulting from a certain k selection to a different k selection. This method increases the value for k in a similar manner to that of the previous approach in that it splits a center into two distinct points. The center is chosen by comparing which split will most benefit the BIC score [7].

3.4 Initializing Centers

The k-medians approach to clustering locates a local optimal solution that minimizes the 1-norm distances from each point to its respective cluster center. It is important to note that this finds a local, rather than global, optimum. Because of this, k-medians is very sensitive to the initialization points of its k centers, each center having the tendency to remain roughly in the same cluster in which it is first placed [8]. Different ideas have therefore been proposed in order to find better initial placement for the k centers in hopes that they will converge to the global optimum. A handful of these propositions are as follows:

- **Random Initialization**- K points are generated at random. The k centers are initialized to these points.
- **Density Analysis**- K points are selected from viewing the distribution of the data set and isolating high density areas.
- **Single Dimension Subsets**- Column vectors are looked at independently in order to select the dimension with the largest variance. The points in this dimension are then divided into k subsets, and the centers are initialized by the median values of these subsets.
- **Diagonal Initialization**- The data set is divided in to k rows and k columns, creating a 2d grid. The cells on the diagonals of this grid are then weighted by their density. The k centers are then randomly selected from these cells, taking their weight into account for the selection.
- **Sampling**- Samples are taken from the data set. The k-medians algorithm is then applied to each sample using random initialization. The k centers are then randomly selected from the solutions generated [9].

These are only a few of the proposed concepts to generate better initialization points. While many of them show promise theoretically, however, their performance is often worse than or roughly equivalent to those result generated by random initialization. There is currently no method of initialization that will produce centers that will always converge to the global optimum.

4. References

[1] D. Feldman and L. J. Schulman (2012) Data reduction for weighted and outlier-resistant clustering Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms (pp. 1342-1354)

[2] K.Chen (2006) On K-Median Clustering in High Dimensions Proceedings of the seventeenth annual ACM-SIAM symposium on discrete algorithm (pp.1177-1185)

[3] B. Anderson, D. Gross, D. Musicant, A. Ritz, T. Smith, L. Steinberg (2006) Adapting K-Medians to Generate Normalized Cluster Centers Proceedings of the Sixth SIAM International Conference on Data Mining (pp.165-175)

[4] G. Hamerly and C. Elkan (2003) Learning the K in K-Means NIPS

[5] M. Ackermann, J. Blomer, C. Sohler (2010) Clustering for Metric and Non-Metric Distance Measures ACM Transactions on Algorithms 6:4

[6] G. Hamerly and C. Elkan (2003) Learning the K in K-Means NIPS

[7] D. Pelleg and A. Moore (2000) X-means: Extending K-means with Efficient Estimation of the Number of Clusters In Proceedings of the 17th International Conf. on Machine Learning (pp. 727-734)

[8] S. Bubeck, M. Meila, U. Von Luxembourg (2012) How the Initialization Affects the Stability of the K-Means Algorithm ESAIM: Probability and Statistics (pp.436-452)

[9] Mohammad F. Eltibi Wesam M. Ashour (2011) Initializing K-Means Clustering Algorithm using Statistical Information International Journal of Computer Applications (pp.51-55)

Response Variabilty Due to Randomness of Beam Height for Beam Bending Problem

M. Meštrović[1]
[1]Faculty of Civil Engineering, University of Zagreb, Zagreb, Croatia

Abstract— *The beam height is assumed to have spatial uncertainty. The formulation to determine the response variability in the beam structure due to randomness of the beam height is given. The concept of variability response function is extended to beam bending problem where the beam height is considered to be one-dimensional, homogenous stochastic field. The randomness of the beam height has than influence not only on the flexural rigidity of the beam, but also on the self-weight load of the beam. Through the proposed formulation, it becomes possible for the weighted integral stochastic finite element analysis to consider complete influence of uncertain geometrical property on response variability.*

Keywords: stochastic beam bending equation, uncertain geometrical property, weighted integral method, response variability

1. Introduction

The assumption that structures have deterministic material properties, is implicitly involved in the most calculation of standard finite element structural analysis of the structures. The material and geometrical properties of real structures have uncertainties, which have to be considered in structural analysis. The uncertainties of the structures are than considered through the increase of the safety factors using deterministic analysis.

The concept of variability response function was introduced in [1] and used in [2], [3], [4], [5]. The weighted integral method was introduced in [2] and generally applied in [3], [4]. The extension of the concept of variability function and weighted integral method on the plate bending problems was first presented in [7] and further developed with bicubic finite element in [8]. In those works the variability was involved through the variability of elastic modulus. The analysis of plate with uncertain plate height was introduced in [6].

This study is concentrated on the randomness in geometrical parameters (the beam height) and its influence on the both side of equation. The first and second moment are used to describe randomness of the input quantities. The new autocorrelation function is written for the flexural rigidity of the beam considering the randomness of the beam height. The variability response function is used to find spectral-distribution-free upper bounds of the response variability. The response variability is represented as second moment of the response deflection.

2. Variability of input quantities

We consider a beam, of the length L, elastic modulus E and with a spatially varying height of the beam $H(x)$. The height of the beam is assumed to constitute a homogenous one-dimensional random field in the following form:

$$H(x) = H_0 \left[1 + h(x)\right] \ , \tag{1}$$

where $H_0 = H_0(x) = const.$ is expectation of the height of the beam taken equal for any point on the beam and $h(x)$ is homogenous one-dimensional random field with expectation zero. This random field is represented with its variance σ_{hh}^2 and autocorrelation function

$$R_{hh}(\xi) = \mathrm{E}\left[h(x+\xi)h(x)\right] \ , \tag{2}$$

what leads to variance and coefficient of variation of beam height$H(x)$

$$\mathrm{Var}\left[H(x)\right] = H_0\sigma_{hh}^2, \quad \mathrm{COV}[H(x)] = \sigma_{hh} \ . \tag{3}$$

The flexural rigidity of the beam, $EI(x)$, is now also random field of the form

$$EI(x) = EI_0 \left[1 + d(x)\right] \ , \tag{4}$$

where $d(x)$ is homogenous one-dimensional random field with expectation zero defined as

$$d(x) = \left[1 + h(x)\right]^3 - 1 = 3h(x) + 3h^2(x) + h^3(x) \ . \tag{5}$$

Autocorrelation function of this random field is then according to [6]

$$\begin{aligned} R_{dd}(\xi) = 9\sigma_{hh}^4 + \left(9 + 18\sigma_{hh}^2 + 9\sigma_{hh}^4\right) R_{hh}(\xi) \\ + 18R_{hh}^2(\xi) + 6R_{hh}^3(\xi) \quad . \end{aligned} \tag{6}$$

The variance and the coefficient of variation are

$$\begin{aligned}\sigma_{dd}^2 &= 9\sigma_{hh}^2 + 45\sigma_{hh}^4 + 15\sigma_{hh}^6 \\ &= 9\sigma_{hh}^2\left(1 + 5\sigma_{hh}^2 + \frac{5}{3}\sigma_{hh}^4\right) , \quad (7)\end{aligned}$$

$$\mathrm{COV}\left[d((x)\right] = 3\sigma_{hh}\sqrt{1 + 5\sigma_{hh}^2 + \frac{5}{3}\sigma_{hh}^4} . \quad (8)$$

The variability of the height of the beam, obviously, leads to variability of loading. The self-weight of the beam directly depends on the height of the beam. We define the load as some prescribed load, $\overline{q} = \overline{q(x)} = const.$, added to self weight calculated with constant weight density of the beam structure, $\gamma = \gamma(x) = const.$, what leads on load expressed as linear combination of deterministic and stochastic part,

$$\begin{aligned}q(x) &= \gamma H(x) + \overline{q} \\ &= \gamma H_0\left[1 + h(x)\right] + \overline{q} \\ &= q_0\left[1 + \Delta q(x)\right] \quad , \quad (9)\end{aligned}$$

where deterministic part is expressed as

$$q_0 = \gamma H_0 + \overline{q} , \quad (10)$$

and stochastic part, with introduced supstitution

$$G = \gamma H_0/(\gamma H_0 + \overline{q}) , \quad (11)$$

follows as

$$\Delta q(x) = \frac{\gamma H_0}{\gamma H_0 + \overline{q}} h(x) = G h(x) . \quad (12)$$

Autocorrelation function of loading as random field is then

$$R_{qq}(\xi) = G^2 R_{hh}(\xi) , \quad (13)$$

and the variance is

$$\sigma_{qq}^2 = G^2\sigma_{hh}^2 . \quad (14)$$

3. Finite element formulation

Standard deterministic finite element formulation of the thin plate bending problem is

$$\mathbf{K}_0\mathbf{w} = \mathbf{q}_0 . \quad (15)$$

Involving the randomness, the formulation for the stochastic analysis is according to [2]

$$\left(\mathbf{K}_0 + \Delta\mathbf{K}\right)\mathbf{w} = \mathbf{q}_0 + \Delta\mathbf{q} , \quad (16)$$

where $\mathbf{K}_0$ and $\mathbf{q}_0$ are deterministic stiffness matrix and load vector respectively and $\Delta\mathbf{K}$ and $\Delta\mathbf{q}$ are stochastic parts of stiffness matrix and load vector respectively. The displacement vector, with assumption that variance is sufficiently footnotesize, is approximated with

$$\begin{aligned}\mathbf{w} = \mathbf{K}^{-1}\mathbf{q} &= \left[\mathbf{K}_0\left(\mathbf{I} + \mathbf{K}_0^{-1}\Delta\mathbf{K}\right)\right]\mathbf{q} \\ &= \left(\mathbf{I} + \mathbf{K}_0^{-1}\Delta\mathbf{K}\right)^{-1}\mathbf{K}_0^{-1}\mathbf{q} \\ &= \left[\sum_{n=0}^{\infty}(-1)^n\left(\mathbf{K}_0^{-1}\Delta\mathbf{K}\right)^n\right]\mathbf{K}_0^{-1}\mathbf{q} \\ &\approx \left(\mathbf{I} - \mathbf{K}_0^{-1}\Delta\mathbf{K}\right)\mathbf{K}_0^{-1}\left(\mathbf{q}_0 + \Delta\mathbf{q}\right) \\ &= \left(\mathbf{I} - \mathbf{K}_0^{-1}\Delta\mathbf{K}\right)\left(\mathbf{w}_0 + \mathbf{K}_0^{-1}\Delta\mathbf{q}\right) \\ &= \mathbf{w}_0 - \mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{w}_0 + \left(\mathbf{I} - \mathbf{K}_0^{-1}\Delta\mathbf{K}\right)\mathbf{K}_0^{-1}\Delta\mathbf{q} \\ &\approx \mathbf{w}_0 - \mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{w}_0 + \mathbf{K}_0^{-1}\Delta\mathbf{q} . \quad (17)\end{aligned}$$

The stochastic part of displacement vector is now

$$\Delta\mathbf{w} = -\mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{w}_0 + \mathbf{K}_0^{-1}\Delta\mathbf{q} . \quad (18)$$

The expression for expectation of displacement vector is

$$\mathrm{E}\left[\mathbf{w}\right] = \mathbf{w}_0 \quad (19)$$

and the covariance matrix of the response deflection $\mathbf{w}$ is given as

$$\begin{aligned}\mathrm{Cov}\left[\mathbf{w},\mathbf{w}\right] &= \mathrm{E}\left[\left(\mathbf{w}-\mathbf{w}_0\right)\left(\mathbf{w}-\mathbf{w}_0\right)^{\mathrm{T}}\right] \\ &= \mathrm{E}\left[\left(\mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{w}_0 - \mathbf{K}_0^{-1}\Delta\mathbf{q}\right)\left(\mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{w}_0 - \mathbf{K}_0^{-1}\Delta\mathbf{q}\right)^{\mathrm{T}}\right] \\ &= \mathrm{E}\left[\mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{W}_0\Delta\mathbf{K}\mathbf{K}_0^{-1}\right] + \mathrm{E}\left[\mathbf{K}_0^{-1}\Delta\mathbf{q}\Delta\mathbf{q}^{\mathrm{T}}\mathbf{K}_0^{-1}\right] \\ &\quad - \mathrm{E}\left[\mathbf{K}_0^{-1}\Delta\mathbf{K}\mathbf{w}_0\Delta\mathbf{q}^{\mathrm{T}}\mathbf{K}_0^{-1}\right] - \mathrm{E}\left[\mathbf{K}_0^{-1}\Delta\mathbf{q}\mathbf{w}_0^{\mathrm{T}}\Delta\mathbf{K}\mathbf{K}_0^{-1}\right] , \quad (20)\end{aligned}$$

where $\mathbf{W}_0 = \mathbf{w}_0\mathbf{w}_0^{\mathrm{T}}$. The first part of Eq.20 is same as in former analysis [2], [7]. The second part includes the randomness of the loading. Last two parts exist only when stochastic field of stiffness and stochastic field of load is correlated. If we represent self-weight load as function of the beam height, those fields are strictly correlated.

4. Weighted integral method

Weighted integral method was introduced in [3], [4]. Stochastic part of element stiffness matrix is represented as linear combination of NWI random variables $X_i^{(e)}$ called weighted integrals,

$$\Delta\mathbf{K}^{(e)} = \sum_{k=0}^{NWI} X_k^{(e)}\Delta\mathbf{K}_k^{(e)} . \quad (21)$$

The number of weighted integrals, NWI, depends on the choice of finite element. Using the standard beam (cubic) finite element follows ($NWI = 3$) weighted integrals. The weighted integrals $X_i^{(e)}, i = 0, 1, 2$ are defined as

$$X_i^{(e)} = \int_0^{L^{(e)}} \xi^i d(\xi) d\xi . \quad (22)$$

The element matrices, $\Delta\mathbf{K}_i^{(e)}$, are all deterministic and given in [3]. The stochastic part of element load vector is defined as

$$\Delta\mathbf{q}^{(e)} = \sum_{k=1}^{\mathrm{NWIQ}} Y_k^{(e)} \Delta\mathbf{q}_k^{(e)} \,, \tag{23}$$

where $NWIQ$ is the number of weighted integrals $Y_k^{(e)}$. Using the cubic finite element follows $NWIQ = 4$. The weighted integrals $Y_i^{(e)}, i = 0,1,2,3$ are defined as

$$Y_i^{(e)} = \int_0^{L^{(e)}} \xi^i h^{(e)}(\xi) d\xi \,, \tag{24}$$

where all vectors $\Delta\mathbf{q}_i^{(e)}$ are deterministic and

$$\Delta\mathbf{q}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} , \tag{25}$$

$$\Delta\mathbf{q}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}} , \tag{26}$$

$$\Delta\mathbf{q}_2 = \begin{bmatrix} \frac{-3}{L^{(e)2}} & \frac{-2}{L^{(e)}} & \frac{3}{L^{(e)2}} & \frac{-1}{L^{(e)}} \end{bmatrix}^{\mathrm{T}} , \tag{27}$$

$$\Delta\mathbf{q}_3 = \begin{bmatrix} \frac{2}{L^{(e)3}} & \frac{1}{L^{(e)2}} & \frac{-2}{L^{(e)3}} & \frac{1}{L^{(e)2}} \end{bmatrix}^{\mathrm{T}} . \tag{28}$$

5. Response variability

The response vector, the vector of unknown displacements $\mathbf{w}$, could be approximated with linear part of Taylor series around the expectation of weighted integrals $X_k^{(e)}$ and $Y_p^{(e)}$,

$$\begin{aligned}
\mathbf{w} &= \mathbf{w}_0 + \sum_{(e)=1}^{N^{(e)}} \sum_k X_k^{(e)} \left[\frac{\partial \mathbf{w}}{\partial X_k^{(e)}}\right]_{\mathrm{E}} \\
&+ \sum_{(e)=1}^{N^{(e)}} \sum_p Y_p^{(e)} \left[\frac{\partial \mathbf{w}}{\partial Y_p^{(e)}}\right]_{\mathrm{E}} \\
&= \mathbf{w}_0 - \sum_{(e)=1}^{N^{(e)}} \sum_k \mathbf{K}_0^{-1} \left[\frac{\partial \mathbf{K}}{\partial X_k^{(e)}}\right]_{\mathrm{E}} \mathbf{w}_0 X_k^{(e)} \\
&+ \sum_{(e)=1}^{N^{(e)}} \sum_p \mathbf{K}_0^{-1} \left[\frac{\partial \mathbf{q}}{\partial Y_p^{(e)}}\right]_{\mathrm{E}} Y_p^{(e)} \\
&= \mathbf{w}_0 - \sum_{(e)=1}^{N^{(e)}} \sum_k \mathbf{K}_0^{-1} \Delta\mathbf{K}_k^{(e)} \mathbf{w}_0 X_k^{(e)} \\
&+ \sum_{(e)=1}^{N^{(e)}} \sum_p \mathbf{K}_0^{-1} \Delta\mathbf{q}_p^{(e)} \quad ,
\end{aligned} \tag{29}$$

where the expressions $\left[\partial\mathbf{w}/\partial X_k^{(e)}\right]_{\mathrm{E}}$, $\left[\partial\mathbf{w}/\partial Y_p^{(e)}\right]_{\mathrm{E}}$, $\left[\partial\mathbf{K}/\partial X_k^{(e)}\right]_{\mathrm{E}}$ i $\left[\partial\mathbf{q}/\partial Y_p^{(e)}\right]_{\mathrm{E}}$ are the values of partial derivatives at the expectation of weighted integrals, $X_k^{(e)}$ and $Y_p^{(e)}$, $N^{(e)}$ is total number of finite elements, summation $k = 0,1,2$ summation $p = 0,1,2,3$ and $\mathbf{w}_0$ is the solution of deterministic problem.

Using the first-order approximation around the zero-mean value of weighted integrals, first-order approximation of covariance matrix of response vector follows as

$$\begin{aligned}
\mathrm{Cov}[\mathbf{w},\mathbf{w}] &= \mathrm{E}\left[(\mathbf{w}-\mathbf{w}_0)(\mathbf{w}-\mathbf{w}_0)^{\mathrm{T}}\right] \\
&= \sum_{(e,f)} \sum_{k,m} \mathbf{K}_0^{-1} \Delta\mathbf{K}_k^{(e)} \mathbf{W}_0 \left(\Delta\mathbf{K}_m^{(f)}\right)^{\mathrm{T}} \left(\mathbf{K}_0^{-1}\right)^{\mathrm{T}} \mathrm{E}\left[X_k^{(e)} X_m^{(f)}\right] \\
&- \sum_{(e,f)} \sum_{k,r} \mathbf{K}_0^{-1} \Delta\mathbf{K}_k^{(e)} \mathbf{w}_0 \left(\Delta\mathbf{q}_r^{(f)}\right)^{\mathrm{T}} \left(\mathbf{K}_0^{-1}\right)^{\mathrm{T}} \mathrm{E}\left[X_k^{(e)} Y_r^{(f)}\right] \\
&- \sum_{(e,f)} \sum_{p,m} \mathbf{K}_0^{-1} \Delta\mathbf{q}_p^{(e)} \mathbf{w}_0^{\mathrm{T}} \left(\Delta\mathbf{K}_m^{(f)}\right)^{\mathrm{T}} \left(\mathbf{K}_0^{-1}\right)^{\mathrm{T}} \mathrm{E}\left[Y_p^{(e)} X_m^{(f)}\right] \\
&+ \sum_{(e,f)} \sum_{p,r} \mathbf{K}_0^{-1} \Delta\mathbf{q}_p^{(e)} \left(\Delta\mathbf{q}_r^{(f)}\right)^{\mathrm{T}} \left(\mathbf{K}_0^{-1}\right)^{\mathrm{T}} \mathrm{E}\left[Y_p^{(e)} Y_r^{(f)}\right]
\end{aligned} \tag{30}$$

where the only unknowns are the values for expectations of weighted integrals products $\mathrm{E}\left[X_k^{(e)} X_m^{(f)}\right]$, $\mathrm{E}\left[X_k^{(e)} Y_r^{(f)}\right]$, $\mathrm{E}\left[Y_p^{(e)} X_m^{(f)}\right]$ and $\mathrm{E}\left[Y_p^{(e)} Y_r^{(f)}\right]$. The first expectation $\mathrm{E}\left[X_k^{(e)} X_m^{(f)}\right]$ is to find as in [7] what leads on

$$\begin{aligned}
\mathrm{E}\left[X_k^{(e)} X_m^{(f)}\right] &= \mathrm{E}\left[\left(\int_0^{L^{(e)}} \xi_e^k d^{(e)}(\xi_e)\, d\xi_e\right)\left(\int_0^{L^{(f)}} \xi_f^m d^{(f)}(\xi_f)\, d\xi_f\right)\right] \\
&= \int_0^{L^{(e)}} \int_0^{L^{(f)}} \xi_e^k \xi_f^m \mathrm{E}\left[d^{(e)}(\xi_e)\, d^{(f)}(\xi_f)\right] d\xi_e d\xi_f \,.
\end{aligned} \tag{31}$$

Considering that all finite elements are characterised by the same stochastic field $h(x)$, what leads than to the same stochastic field $d(x)$ for all elements, the expectation (31) can be expressed as in [7] with

$$\mathrm{E}\left[d(\xi_e)\, d(\xi_f)\right] = R_{dd}\left(\Delta_{fe} + L^{(f)}\xi_f - L^{(e)}\xi_e\right) , \tag{32}$$

where $R_{dd}(\xi)$ is autocorrelation function of stochastic field $d(\xi)$, and Δ_{fe} is given as the distance of the first knots of finite elements (e) and (f) expressed as

$$\Delta_{fe} = x_i^{(f)} - x_i^{(e)} \,. \tag{33}$$

After similar algebra and same simplification about field characterisation, the other expectations are

$$\begin{aligned}
\mathrm{E}\left[X_k^{(e)} Y_r^{(f)}\right] &= \mathrm{E}\left[\left(\int_0^{L^{(e)}} \xi_e^k d^{(e)}(\xi_e)\, d\xi_e\right)\left(\int_0^{L^{(f)}} \xi_f^r h^{(f)}(\xi_f)\, d\xi_f\right)\right] \\
&= \int_0^{L^{(e)}} \int_0^{L^{(f)}} \xi_e^k \xi_f^r \mathrm{E}\left[d^{(e)}(\xi_e)\, h^{(f)}(\xi_f)\right] d\xi_e d\xi_f \,,
\end{aligned} \tag{34}$$

$$\begin{aligned}
\mathrm{E}\left[Y_p^{(e)} X_m^{(f)}\right] &= \mathrm{E}\left[\left(\int_0^{L^{(e)}} \xi_e^p h^{(e)}(\xi_e)\, d\xi_e\right)\left(\int_0^{L^{(f)}} \xi_f^m d^{(f)}(\xi_f)\, d\xi_f\right)\right] \\
&= \int_0^{L^{(e)}} \int_0^{L^{(f)}} \xi_e^p \xi_f^r \mathrm{E}\left[h^{(e)}(\xi_e)\, d^{(f)}(\xi_f)\right] d\xi_e d\xi_f \,,
\end{aligned} \tag{35}$$

$$\begin{aligned}
\mathrm{E}\left[Y_p^{(e)} Y_r^{(f)}\right] &= \mathrm{E}\left[\left(\int_0^{L^{(e)}} \xi_e^p h^{(e)}(\xi_e)\, d\xi_e\right)\left(\int_0^{L^{(f)}} \xi_f^r h^{(f)}(\xi_f)\, d\xi_f\right)\right] \\
&= \int_0^{L^{(e)}} \int_0^{L^{(f)}} \xi_e^p \xi_f^r \mathrm{E}\left[h^{(e)}(\xi_e)\, h^{(f)}(\xi_f)\right] d\xi_e d\xi_f \,,
\end{aligned} \tag{36}$$

with

$$\mathrm{E}\left[d\left(\xi_e\right)h\left(\xi_f\right)\right] = R_{dh}\left(\Delta x_{fe}\right) \ , \tag{37}$$
$$\mathrm{E}\left[h\left(\xi_e\right)d\left(\xi_f\right)\right] = R_{hd}\left(\Delta x_{fe}\right) \ , \tag{38}$$
$$\mathrm{E}\left[h\left(\xi_e\right)h\left(\xi_f\right)\right] = R_{hh}\left(\Delta x_{fe}\right) \ , \tag{39}$$

and $\Delta x_{fe} = \Delta_{fe} + L^{(f)}\xi_f - L^{(e)}\xi_e$, autocorrelation function $R_{dh}(\xi) = R_{hd}(\xi) = R_{hh}(\xi)\left(1+\sigma_{hh}^2\right)$ and its variance $\sigma_{dh}^2 = \sigma_{hh}^2 + \sigma_{hh}^4$.

The variance vector of displacement vector $\mathbf{w}$ is than evaluated as

$$\begin{aligned}\mathrm{Var}[\mathbf{w}] = &\int_{-\infty}^{\infty} S_{dd}(\kappa_x)\mathrm{VRF}_1(\kappa_x)d\kappa_x \\ &- \int_{-\infty}^{\infty} S_{dh}(\kappa_x)\left[\mathrm{VRF}_2(\kappa_x) + \mathrm{VRF}_3(\kappa_x)\right]d\kappa_x \\ &\int_{-\infty}^{\infty} S_{hh}(\kappa_x)\mathrm{VRF}_4(\kappa_x)d\kappa_x \ ,\end{aligned} \tag{40}$$

where $S_{dd}(\kappa_x)$, $S_{dh}(\kappa_x)$ and $S_{hh}(\kappa_x)$ are spectral density function and the vectors $\mathrm{VRF}_i(\kappa_x)$ are the first-order approximations of the variability response function parts respectively

$$\begin{aligned}\mathrm{VRF}_1(\kappa_x) = &\sum_{(e),(f)}\sum_{k,m}\mathrm{diag}\left(\mathbf{K}_0^{-1}\Delta\mathbf{K}_k^{(e)}\mathbf{w}_0\right)\mathbf{K}_0^{-1}\Delta\mathbf{K}_m^{(f)}\mathbf{w}_0 \\ &\cdot\left[\left(CI_k^{(e)}CI_m^{(f)} + SI_k^{(e)}SI_m^{(f)}\right)\cos\left(\Delta_{fe}\kappa_x\right)\right. \\ &\left.-\left(SI_k^{(e)}CI_m^{(f)} - CI_k^{(e)}SI_m^{(f)}\right)\sin\left(\Delta_{fe}\kappa_x\right)\right] \ ,\end{aligned} \tag{41}$$

$$\begin{aligned}\mathrm{VRF}_2(\kappa_x) = &\sum_{(e),(f)}\sum_{k,r}\mathrm{diag}\left(\mathbf{K}_0^{-1}\Delta\mathbf{K}_k^{(e)}\mathbf{w}_0\right)\mathbf{K}_0^{-1}\Delta\mathbf{q}_r^{(f)} \\ &\cdot\left[\left(CI_k^{(e)}CI_r^{(f)} + SI_k^{(e)}SI_r^{(f)}\right)\cos\left(\Delta_{fe}\kappa_x\right)\right. \\ &\left.-\left(SI_k^{(e)}CI_r^{(f)} - CI_k^{(e)}SI_r^{(f)}\right)\sin\left(\Delta_{fe}\kappa_x\right)\right] \ ,\end{aligned} \tag{42}$$

$$\begin{aligned}\mathrm{VRF}_3(\kappa_x) = &\sum_{(e),(f)}\sum_{p,m}\mathrm{diag}\left(\mathbf{K}_0^{-1}\Delta\mathbf{q}_p^{(e)}\right)\mathbf{K}_0^{-1}\Delta\mathbf{K}_m^{(f)}\mathbf{w}_0 \\ &\cdot\left[\left(CI_p^{(e)}CI_m^{(f)} + SI_p^{(e)}SI_m^{(f)}\right)\cos\left(\Delta_{fe}\kappa_x\right)\right. \\ &\left.-\left(SI_p^{(e)}CI_m^{(f)} - CI_p^{(e)}SI_m^{(f)}\right)\sin\left(\Delta_{fe}\kappa_x\right)\right] \ ,\end{aligned} \tag{43}$$

$$\begin{aligned}\mathrm{VRF}_4(\kappa_x) = &\sum_{(e),(f)}\sum_{p,r}\mathrm{diag}\left(\mathbf{K}_0^{-1}\Delta\mathbf{q}_p^{(e)}\right)\mathbf{K}_0^{-1}\Delta\mathbf{q}_r^{(f)} \\ &\cdot\left[\left(CI_p^{(e)}CI_r^{(f)} + SI_p^{(e)}SI_r^{(f)}\right)\cos\left(\Delta_{fe}\kappa_x\right)\right. \\ &\left.-\left(SI_p^{(e)}CI_r^{(f)} - CI_p^{(e)}SI_r^{(f)}\right)\sin\left(\Delta_{fe}\kappa_x\right)\right] \ ,\end{aligned} \tag{44}$$

and for $(g) = (e), (f)$

$$CI_k^{(g)} = \int_0^{L^{(g)}} \xi_g^k \cos\left(\kappa_x\xi_g\right)d\xi_g \tag{45}$$

$$SI_k^{(g)} = \int_0^{L^{(g)}} \xi_g^k \sin\left(\kappa_x\xi_g\right)d\xi_g \ . \tag{46}$$

Consider a specific degree of freedom w_i and corresponding component of according response variabilities $\mathrm{VRF}_j^i(\kappa_x), j = 1,\ldots,4$, the coefficient of variation is bounded as

$$\mathrm{COV}\left[w_i\right] \le \sigma_{hh}\frac{\sqrt{V^{*,i}}}{\|\mathrm{E}[w_i]\|} \ , \tag{47}$$

where

$$\begin{aligned}V^{*,i} = &\left[9 + 45\sigma_{hh}^2 + 15\sigma_{hh}^4\right]\mathrm{VRF}_1^i(\kappa_x^*) \\ &-\left(1+\sigma_{hh}^2\right)\left[\mathrm{VRF}_2^i(\kappa_x^*) + \mathrm{VRF}_3^i(\kappa_x^*\right] + \mathrm{VRF}_4^i(\kappa_x^*) \ ,\end{aligned} \tag{48}$$

and (κ_x^*) is the point at which the function under square root takes its maximum value.

6. Numerical example

The unit length beam under uniformly distributed load, simply-supported on the both edges is considered in numerical example. The variation of the beam height is taken $\sigma_{hh} = 0.1$ what leads to $\sigma_{dd} \approx 0.3074$. The variability response function is calculated for the deflection in the middle of the beam, $w(0.5)$, by weighted integral method with 8 finite elements. The results of the proposed weighted integral method are compared arecompared with results of the classical Monte Carlo simulatioon (MCS).

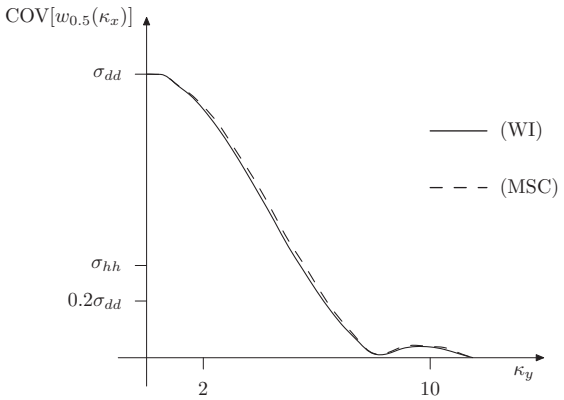


Fig. 1
COEFFICIENT OF VARIATION OF THE RESPONSE DEFLECTION AT THE MID-POINT, $w(0.5)$.

The results show that the randomness of the beam height has the influence on the structural response. It is also obvious that results obtained with weighted integral method are in good agreement with results obtained by MCS.

7. Conclusions

The concept of the variability response function based on weighted integral and local average method was extended to the beam bending problem with random beam height. It has been shown that randomness of the beam height has influence on the randomness of self-weight load. The influence on the variability of the response deflection was calculated according the concept of the variability response function. It has been shown very good agreement of the results calculated with weighted integral method with results obtained by MCS.

With proposed wighted integral formulation, that includes randomness of the beam height on the flexural rigidity and loading, it becomes possible for the weighted integral stochastic finite element analysis to consider complete influence of uncertain geometrical property on the structural response variability.

References

[1] G. Deodatis, M. Shinozuka "Bounds on response variability of stochastic systems", *ASCE J. Eng. Mech.*, vol. 115, pp. 2543–2563, Nov. 1989.

[2] G. Deodatis "Bounds on response variability of stochastic finite element systems", *ASCE J. Eng. Mech.*, vol. 116, pp. 565–585, Mar. 1990.

[3] G. Deodatis "Weighted integral method. I: stochastic stiffness matrix", *ASCE J. Eng. Mech.*, vol. 117, pp. 1851–1864, Aug. 1991.

[4] G. Deodatis, M. Shinozuka "Weighted integral method. II: response variability and reliability", *ASCE J. Eng. Mech.*, vol. 117, pp. 1865–1877, Aug. 1991.

[5] F.J. Wall, G. Deodatis "Variabilty response functions of stochastic plane stress/strain problems", *ASCE J. Eng. Mech.*, vol. 120, pp. 1963–1982, Sep. 1994.

[6] N.K. Choi, H.-C. Noh "Stochastic finite element analysis of plate structures by weighted integral method", *Struct. Eng. Mech.*, vol. 4, pp. 703–715, Jun. 1996.

[7] L. Graham, G. Deodatis "Variability response function for stochastic plate bending problems", *Struct. Saf.*, vol. 20, pp. 167–188, Jan. 1998.

[8] M. Meštrović, "Variability Response Function for Stochastic Thin Plate Bending Problem", in *Structural Safety and Reliability - Proceedings of* 8^{th} *ICOSSAR'01*, 2001.

Solving Kinematics Problems of a 6-DOF Robot Manipulator

Alireza Khatamian
Computer Science Department, The University of Georgia, Athens, GA, U.S.A

Abstract - *This paper represents an analytical approach for solving forward kinematics problem of a serial robot manipulator with six degrees of freedom and a specific combination of joints and links to formulate the position of the gripper by a given set of joint angles. In addition, a direct geometrical solution for the robot's inverse kinematics problem will be defined in order to calculate the robot's joint angles to pose the gripper in a given coordinate. Furthermore, the accuracy of the two solutions will be shown by comparing the results in a developed simulation program which uses the Unified System for Automation and Robot Simulation (USARSim).*

Keywords: Kinematics, Geometric, Robot Manipulators

1 Introduction

Automation and hazard reduction in the workplace are two important factors of driving the use of robotics, specially articulated arm robots, in order for decreasing human deaths and injuries, economically affordable production plus saving time. Therefore, high precision control is the basic and primary step of using these robots.

In brief, six degrees of freedom, three of which are used for posing the gripper at a specific position and the other three for orientation adjustment, is an essential dexterity characteristic of a space robot manipulator. Hence, the focus of this paper is on controlling a six degree-of-freedom arm robot with a particular series of links and joints.

[3] represents the kinematics of nine common industrial robots including forward kinematics, inverse kinematics plus Jacobian matrix which is used in higher level arm robots calculations. In [4], general equations for a human-arm-like robot manipulators have been presented and [5] introduces kinematics solutions for robot manipulators based on their structures. A novel recurrent neural network controller with learning ability to maintain multiple solutions of the inverse kinematics is introduced in [8]. [9] proposes a nonlinear programming technique for solving the inverse kinematics problem of highly articulated arm robots in which finding the joint angles is hard for a desired joint configuration (inverse kinematics). In [10] a new combined method is proposed to solve inverse kinematics problem of six-joint Stanford robotic manipulator by using genetic algorithms and neural networks in order to obtain more precise solutions and to minimize the error at the end effector final position. A novel heuristic method, called Forward And Backward Reaching Inverse Kinematics (FABRIK), is described and compared with some of the most popular existing methods regarding reliability, computational cost and conversion criteria in [11]. [12] represents a hybrid combination of neural networks and fuzzy logic intelligent technique to solve the inverse kinematics problem of a three degree of freedom robotic manipulator. In [13], an algorithm in which the independent components of link lengths are used as a medium to analyze the forward kinematics of a six-degree-of-freedom Stewart platform can be found and an extension to closed –loop inverse kinematics (CLIK) algorithm is introduced in [14] to meet some applications that require the joint acceleration.

A fundamental phase for controlling a robot manipulator is solving the robot's kinematics problems. Kinematics is the science of motion that treats the subject without regard to the forces that cause it [1]. Kinematics has two aspects: (1) Forward Kinematics and (2) Inverse Kinematics. The former is getting the position and orientation of the end effector of the robot by having the joint angles. The latter is setting joint angles to place the end effector of the robot in a given position and orientation.

Forward kinematics problem of a serial manipulator has a straight forward solution which formulates the position and orientation of the robot's end effector by a series of joint angles. On the other hand, having more degrees of freedom on a serial manipulator elaborates solving the inverse kinematics problem of the robot. There are two approaches to solve the inverse kinematics problem: (1) numerical method and (2) geometrical approach.

The purpose of this paper is to present an analytical method and a geometrical approach for solving forward and inverse kinematics problem of a particular six degree-of-freedom serial manipulator, accordingly. The target robot is a KUKA KR60 which is an industrial manipulator production of KUKA Corporation.

The rest of this paper is organized as follows: in section II, robot parameters and specifications are described, section III and IV represent forward and inverse kinematics solutions, accordingly. Experimental results and precision assessment are provided in section V, and section VI concludes the paper.

2 Robot Specification

To describe a robot manipulator specifically, joint types and the relation between links, which is the exposure mode of two operational axes of two consecutive joints (parallel or perpendicular), have to be explained.

KUKA KR60 is an industrial robot designed by seven links which are connected to each other by six revolute joints. All the joints of this robot are the same and there is no prismatic, cylindrical, planar or any other type of joint in the structure of the robot.

The functional state of each joint related to its successive joint in the design of this robot is as follows:

$$R_1 \perp R_2 \parallel R_3 \perp R_4 \perp R_5 \perp R_6 \tag{1}$$

in which R indicates a revolute joint and the indices describe the position of the joint relative to the base of the robot.

Figure 1 displays a symbolic structure of a KUKA KR60 and the attached frames to the joints in addition to the effective axis of the joints which is always along the Z axis and also length of the links.

A robot manipulator's crucial aspect is its Denavit-Hertenberg parameters or D-H parameter table. It specifies all the characteristics of a robot manipulator including link lengths and relative orientation of the joints. Table 1 shows the D-H parameters of a KUKA KR60.

3 Forward Kinematics

Forward kinematics problem is finding the position and orientation of the end effector of the robot by a given set of joint angles and also having D-H parameters of the robot. This section explains an analytical method for solving the forward kinematics problem of a KUKA KR60.

A robot manipulator's forward kinematics problem is solved by attaching a single frame to each joint along with the robot's base. Each frame describes the position and orientation of each joint of the robot relative to the base or any other global coordinate. Attaching these frames to the joints reduces the calculation of the robot's end effector's position and orientation to a coordinate translation problem which is solved by transformation matrices.

Therefore, every joint has a position and orientation relative to its previous joint. These relations are described by transformation matrices. A general formulation for calculation of these matrices is as follows:

TABLE 1
D-H parameters of a KUKA KR60

i	a	α	d	θ
0	350	0	–	–
1	0	$\frac{\pi}{2}$	−815	$0+\theta_1$
2	850	0	0	$-\frac{\pi}{2}+\theta_2$
3	145	$\frac{\pi}{2}$	0	$0+\theta_3$
4	0	$-\frac{\pi}{2}$	−820	$\pi+\theta_4$
5	0	$\frac{\pi}{2}$	0	$\pi+\theta_5$
6	–	–	170	$0+\theta_6$

$$^{i-1}_{i}T = R_X(\alpha_{i-1})D_X(a_{i-1})R_Z(\theta_i)D_Z(d_i) \tag{2}$$

in which $R_X(\alpha_{i-1})$ is a rotation matrix about the X axis by α_{i-1}, $D_X(a_{i-1})$ is translation matrix along the X axis by a_{i-1}, $R_Z(\theta_i)$ is a rotation matrix about the Z axis by θ_i, $D_Z(d_i)$ is translation matrix along the Z axis by d_i and α, a, θ and d are D-H parameters of the robot. So we have:

$$R_X(\alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_{i-1} & -\sin\alpha_{i-1} & 0 \\ 0 & \sin\alpha_{i-1} & \cos\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$D_X(a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$R_Z(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$D_Z(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

and $^{i-1}_{i}T$ is,

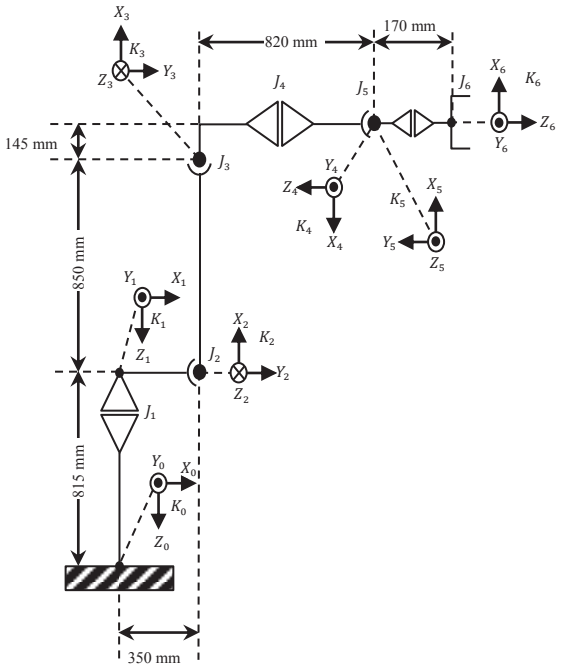


Figure 1. Symbolic structure of a KUKA KR60, link length, the attached frames to the joints and the operational axis which is along the Z axis

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \cos\alpha_{i-1}\sin\theta_i & \cos\alpha_{i-1}\cos\theta_i & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\alpha_{i-1}\sin\theta_i & \sin\alpha_{i-1}\cos\theta_i & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Hence, the following matrix multiplication computes the final transformation matrix that gives the position and orientation of the robot's end effector relative to its base.

$${}^0_6T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T \quad (8)$$

Let,

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

The position and the orientation of the end effector in roll-pitch-yaw representation are as follows:

$${}^0\mathrm{P}_6 = \begin{bmatrix} \mathrm{r}_{14} \\ \mathrm{r}_{24} \\ \mathrm{r}_{34} \end{bmatrix} \quad (10)$$

$$\mathrm{pitch} = \mathrm{Atan2}\left(\mathrm{r}_{13}, \sqrt{\mathrm{r}_{23}^2 + \mathrm{r}_{33}^2}\right) \quad (11)$$

$$\mathrm{roll} = \begin{cases} 0 & pitch = \frac{\pi}{2}, -\frac{\pi}{2} \\ \mathrm{Atan2}\left(-\frac{\mathrm{r}_{23}}{\cos(\mathrm{pitch})}, \frac{\mathrm{r}_{33}}{\cos(\mathrm{pitch})}\right) & o.w \end{cases} \quad (12)$$

$$\mathrm{yaw} = \begin{cases} \mathrm{Atan2}(\mathrm{r}_{32}, \mathrm{r}_{22}) & pitch = \frac{\pi}{2} \\ -\mathrm{Atan2}(\mathrm{r}_{32}, \mathrm{r}_{22}) & pitch = -\frac{\pi}{2} \\ \mathrm{Atan2}\left(-\frac{\mathrm{r}_{12}}{\cos(\mathrm{pitch})}, \frac{\mathrm{r}_{11}}{\cos(\mathrm{pitch})}\right) & o.w \end{cases} \quad (13)$$

4 Inverse Kinematics

Inverse kinematics problem of a robot manipulator is finding the joint angles of the robot by having the position and orientation of the end effector of the robot. Inverse kinematics problem of a serial manipulator is more important than the forward kinematics, as it is essential to move the gripper of the robot to a required position with a defined orientation in order to, for instance, grab an object in that position and orientation.

There are two approaches to solve the inverse kinematics problem of a robot manipulator; mathematical or algebraic and geometrical. The higher degrees of freedom requires the more complicated algebraic solution. Therefore, this section has been devoted to present a geometrical solution for the inverse kinematics problem of a KUKA KR60.

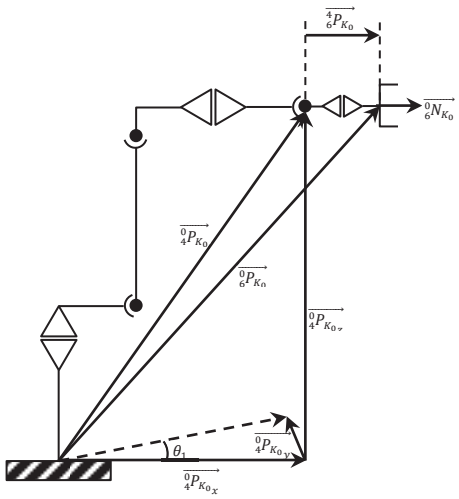


Figure 2. Geometrical representation of first joint angle calculation

In a geometrical method, vectors describe the robot's state to solve the problem which is the calculation of the joint angles of the robot. This section is divided into five subsections to illustrate the joint angles' computing method.

A. Joint 1

The first joint angle's calculation, as shown in Figure 2, is accomplished by the projection of a vector which originates from the origin of frame K_0 and ends to the origin of frame K_4 ($\overrightarrow{{}^0_4\mathrm{P}_{\mathrm{K}_0}}$) on the $\mathrm{X}-\mathrm{Y}$ plane of frame K_0.

Let ${}^0_{\mathrm{G}}\mathrm{T}$ be the target transformation matrix relative to the base which defines the target position and orientation.

$${}^0_GT = \begin{bmatrix} {}^0_GT_{11} & {}^0_GT_{12} & {}^0_GT_{13} & {}^0_GT_{14} \\ {}^0_GT_{21} & {}^0_GT_{22} & {}^0_GT_{23} & {}^0_GT_{24} \\ {}^0_GT_{31} & {}^0_GT_{32} & {}^0_GT_{33} & {}^0_GT_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \overrightarrow{{}^0_6N_{K_0}} = \begin{bmatrix} {}^0_GT_{13} \\ {}^0_GT_{23} \\ {}^0_GT_{33} \end{bmatrix} \quad (14)$$

then we have,

$$\begin{cases} \overrightarrow{{}^4_6P_{K_0}} = d_6 \times \overrightarrow{{}^0_6N_{K_0}} \\ \overrightarrow{{}^0_6P_{K_0}} = \begin{bmatrix} {}^0_GT_{14} \\ {}^0_GT_{24} \\ {}^0_GT_{34} \end{bmatrix} \end{cases} \quad (15)$$

$$\Rightarrow \overrightarrow{{}^0_4P_{K_0}} = \overrightarrow{{}^0_6P_{K_0}} - \overrightarrow{{}^4_6P_{K_0}} = \begin{bmatrix} {}^0_GT_{14} - d_6\,{}^0_GT_{13} \\ {}^0_GT_{24} - d_6\,{}^0_GT_{23} \\ {}^0_GT_{34} - d_6\,{}^0_GT_{33} \end{bmatrix}$$

so,

$$\theta_1 = \begin{cases} Atan2({}^0_GT_{24} - d_6\,{}^0_GT_{23}, {}^0_GT_{14} - d_6\,{}^0_GT_{13}) \\ Atan2({}^0_GT_{24} - d_6\,{}^0_GT_{23}, {}^0_GT_{14} - d_6\,{}^0_GT_{13}) + \pi \end{cases} \quad (16)$$

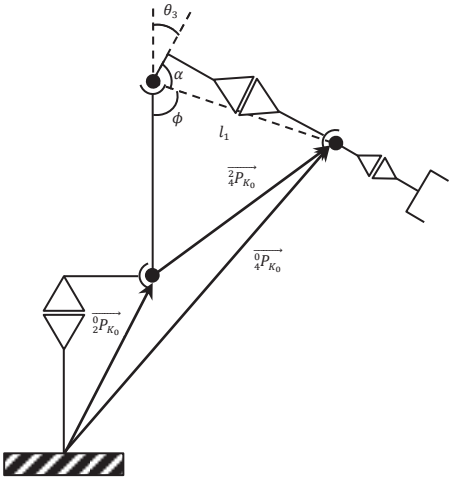


Figure 3. Visual representation of joint 3 angle calculation

B. Joint 3

Based on Figure 3's illustration, to calculate θ_3, first $\overrightarrow{{}^2_4P_{K_0}}$ needs to be calculated. In order to compute $\overrightarrow{{}^2_4P_{K_0}}$, $\overrightarrow{{}^0_2P_{K_0}}$ should be available, beforehand. By having $\overrightarrow{{}^2_4P_{K_0}}$ and l_1, ϕ can be calculated and then by using a simple geometric rule, which helps to compute the angle of between two edges of a triangle, α will be quantified.

Let $\theta_2 = 0$ and,

$$ {}^0_2T = \begin{bmatrix} {}^0_2T_{11} & {}^0_2T_{12} & {}^0_2T_{13} & {}^0_2T_{14} \\ {}^0_2T_{21} & {}^0_2T_{22} & {}^0_2T_{23} & {}^0_2T_{24} \\ {}^0_2T_{31} & {}^0_2T_{32} & {}^0_2T_{33} & {}^0_2T_{34} \end{bmatrix} \Rightarrow \overrightarrow{{}^0_2P_{K_0}} = \begin{bmatrix} {}^0_2T_{14} \\ {}^0_2T_{24} \\ {}^0_2T_{34} \end{bmatrix} \tag{17} $$

thus,

$$ \overrightarrow{{}^2_4P_{K_0}} = \overrightarrow{{}^0_4P_{K_0}} - \overrightarrow{{}^0_2P_{K_0}} = \begin{bmatrix} \overrightarrow{{}^2_4P_{K_0}}_x \\ \overrightarrow{{}^2_4P_{K_0}}_y \\ \overrightarrow{{}^2_4P_{K_0}}_z \end{bmatrix} \tag{18} $$

$$ \phi = Asin\left(\frac{\left(l_1^2 - a_2^2 + \left| \overrightarrow{{}^2_4P_{K_0}} \right|^2 \right)}{2 \left| \overrightarrow{{}^2_4P_{K_0}} \right| l_1} \right) + Asin\left(\frac{\left| \overrightarrow{{}^2_4P_{K_0}} \right| - \frac{l_1^2 - a_2^2 + \left| \overrightarrow{{}^2_4P_{K_0}} \right|^2}{2 \left| \overrightarrow{{}^2_4P_{K_0}} \right|}}{a_2} \right) \tag{19} $$

$$ \alpha = Atan2(-d_4, a_3) \tag{20} $$

So,

$$ \theta_3 = \begin{cases} \pi - \phi - \alpha \\ \pi + \phi - \alpha \end{cases} \tag{21} $$

C. Joint 2

θ_2 is computed by $\overrightarrow{{}^2_4P_{K_2}}$, β_1 and β_2, as Figure 4 displays.

$$ \overrightarrow{{}^2_4P_{K_2}} = {}^2_0R \overrightarrow{{}^2_4P_{K_0}} = {}^0_2R^{-1} \overrightarrow{{}^2_4P_{K_0}} \tag{22} $$

$$ {}^0_2T = \begin{bmatrix} {}^0_2R & {}^0_2P_{ORG} \\ 0 & 1 \end{bmatrix} \Rightarrow {}^0_2R = \begin{bmatrix} {}^0_2T_{11} & {}^0_2T_{12} & {}^0_2T_{13} \\ {}^0_2T_{21} & {}^0_2T_{22} & {}^0_2T_{23} \\ {}^0_2T_{31} & {}^0_2T_{32} & {}^0_2T_{33} \end{bmatrix} = {}^2_0R^{-1} \tag{23} $$

$$ \overrightarrow{{}^2_4P_{K_2}} = \begin{bmatrix} {}^0_2T_{11} & {}^0_2T_{12} & {}^0_2T_{13} \\ {}^0_2T_{21} & {}^0_2T_{22} & {}^0_2T_{23} \\ {}^0_2T_{31} & {}^0_2T_{32} & {}^0_2T_{33} \end{bmatrix} \begin{bmatrix} \overrightarrow{{}^2_4P_{K_0}}_x \\ \overrightarrow{{}^2_4P_{K_0}}_y \\ \overrightarrow{{}^2_4P_{K_0}}_z \end{bmatrix} \tag{24} $$

Thus,

$$ \beta_1 = Atan2\left(\overrightarrow{{}^2_4P_{K_2}}_x, \overrightarrow{{}^2_4P_{K_2}}_y \right) \tag{25} $$

$$ \beta_2 = Asin\left(\frac{a_2^2 - \left| \overrightarrow{{}^2_4P_{K_0}} \right|^2 + l_1^2}{2 l_1 a_2} \right) + Asin\left(\frac{l_1 - \frac{a_2^2 - \left| \overrightarrow{{}^2_4P_{K_0}} \right|^2 + l_1^2}{2 l_1}}{\left| \overrightarrow{{}^2_4P_{K_0}} \right|} \right) \tag{26} $$

and then,

$$ \theta_2 = \begin{cases} \frac{\pi}{2} - (|\beta_1| + \beta_2) \\ \frac{\pi}{2} + (|\beta_1| - \beta_2) \end{cases} \tag{27} $$

D. Joint 5

In order to calculate θ_5, ${}^0_4T$ is computed by assuming θ_4 is equal to 0. Then by using the definition of dot product of two normal vectors which are shown in Figure 5, θ_5 is obtained.

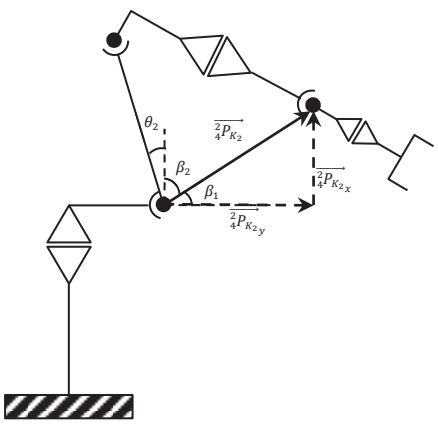


Figure 4. Joint 2 angle calculation vector representation

$$ {}^0_4T = \begin{bmatrix} {}^0_4T_{11} & {}^0_4T_{12} & {}^0_4T_{13} & {}^0_4T_{14} \\ {}^0_4T_{21} & {}^0_4T_{22} & {}^0_4T_{23} & {}^0_4T_{24} \\ {}^0_4T_{31} & {}^0_4T_{32} & {}^0_4T_{33} & {}^0_4T_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \overrightarrow{{}^0_4N_{K_0}} = \begin{bmatrix} {}^0_4T_{13} \\ {}^0_4T_{23} \\ {}^0_4T_{33} \end{bmatrix} \quad (28) $$

So we have,

$$ \theta_5 = \pi - Acos(\overrightarrow{{}^0_4N_{K_0}}. \overrightarrow{{}^0_6N_{K_0}}) \quad (29) $$

E. Joint 4 and 6

To obtain θ_4 and θ_6, rotation matrix ${}^4_6R$ is used. On the one hand, ${}^4_6R$ is:

$$ {}^4_6R = {}^0_4R^{-1}{}^0_6R = {}^4_0R{}^0_6R \quad (30) $$

and on the other hand,

$$ {}^4_6R = Rot_z(\theta_4)Rot_y(\theta_5 + \pi)Rot_z(\theta_6) \quad (31) $$

in which,

$$ Rot_y\,(\theta_5 + \pi) = \begin{bmatrix} \cos(\pi + \theta_5) & 0 & \sin(\pi + \theta_5) \\ 0 & 1 & 0 \\ -\sin(\pi + \theta_5) & 0 & \cos(\pi + \theta_5) \end{bmatrix} \quad (32) $$

$$ Rot_z(\theta_6) = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (33) $$

thus,

$$ {}^4_6R = \begin{bmatrix} -c_4c_5c_6 - s_4s_6 & c_4c_5s_6 - s_4c_6 & -c_4s_5 \\ -s_4c_5c_6 + c_4s_6 & s_4c_5s_6 + c_4c_6 & -s_4s_5 \\ s_5c_6 & -s_5s_6 & -c_5 \end{bmatrix} \quad (34) $$

in which c_4 is corresponding to $\cos(\theta_4)$, s_4 is $\sin(\theta_4)$ and so forth.

For the sake of simplicity, let:

$$ {}^4_6R = \begin{bmatrix} {}^4_6R_{11} & {}^4_6R_{12} & {}^4_6R_{13} \\ {}^4_6R_{21} & {}^4_6R_{22} & {}^4_6R_{23} \\ {}^4_6R_{31} & {}^4_6R_{32} & {}^4_6R_{33} \end{bmatrix} \quad (35) $$

So, we have,

$$ \theta_4 = Atan2(-{}^4_6R_{23}, -{}^4_6R_{13}) \quad (36) $$

$$ \theta_6 = Atan2(-{}^4_6R_{32}, {}^4_6R_{31} \quad (37) $$

5 Experimental Results

In order for testing the forward and inverse kinematics solutions, a simulation program has been developed under the Unified System for Automation and Robot Simulation (USARSim).

A feedback testing approach by the following steps has been selected to estimate the accuracy of the proposed forward and inverse kinematics solutions:

(1) Moving the end effector of the robot to a specific location and orientation.
(2) Calculating the joint angles by the inverse kinematics solution.
(3) Changing the joint angles to the calculated values.
(4) Getting the gripper position and orientation by using the forward kinematics solution.
(5) Finally, computing the Euclidean distance between the initial position/orientation and the final position/orientation to get the error of the two solutions.
(6) Running the above steps 50 times with random generated initial position and orientation.

Figure 6 shows the calculated errors during 50 runs of the program. As it can be seen, the accuracy of these solutions is reasonably fair. The 50 runs only manifest three drastic errors in the position and orientation calculation method.

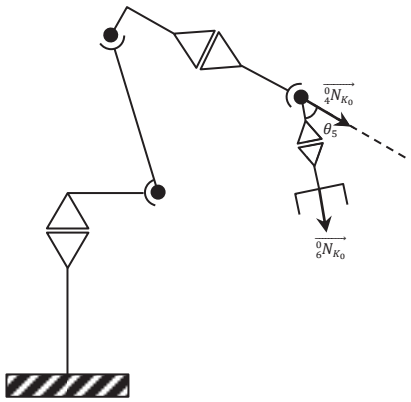


Figure 5. Joint 5 angle calculation geometrical visualization

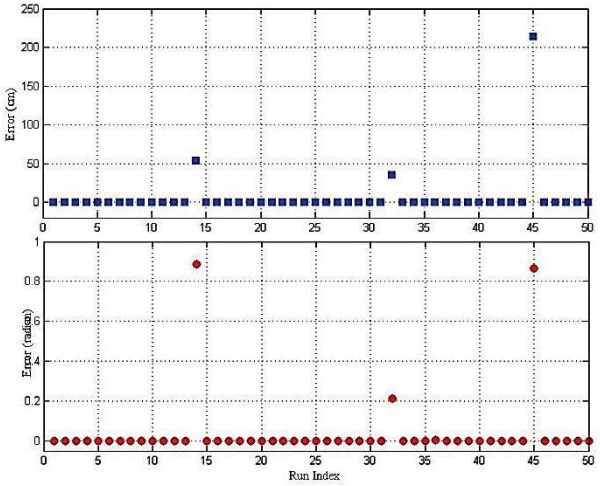


Figure 6. Position and orientation calculation error in 50 random runs in a simulated area using feedback loop testing

6 Conclusion

To conclude, this paper proposed a mathematical method and a geometrical approach for solving the forward and the inverse kinematics problems of an industrial arm robot, KUKA KR60. The experimental result obtained by feedback testing showed these solutions are less erroneous and more accurate. In the simulated programming application which with this method has been tested, all the steps have been implemented and therefore the result is based on the accuracy of the models in the simulation environment. Even the results are based upon simulation; one can conclude that the measurement has enough accuracy for practical usage.

7 Acknowledgment

I would like to thank Dr. Behrouz Minaei who was my advisor in my bachelor's degree at the Iran University of Science and Technology and Dr. Nasser Mozayani whose unforgettable effort was an absolute motivation and support during my bachelor's study. I also want to thank Dr. Hamid R. Arabnia and Dr. Walter D. Potter, faculty members of The University of Georgia, and Mr. Jeffrey Thompson who helped me to write this paper.

8 References

[1] J. J. Craig, Intrduction to Robotics: Mechanics and Manipulations, 3rd ed., 2004.

[2] A. Khatamian, "Virtual Manufacturing Automation by 6-DOF Manipulator in USARSim Simulation Area under Unreal Engine 3 (BSc. Thesis in Persian)," Iran University of Science and Technology, Tehran, 2011.

[3] V. H. John Lloyd, "Kinematics of common industrial robots," vol. 4, no. 2, pp. 169-191, June 1988.

[4] R. L. Ahmad Hemami, "Kinematic equations and solutions of a human-arm-like robot manipulator," vol. 4, no. 1, pp. 65-72, March 1988.

[5] T. K. L. Kesheng Wang, "Structure design and kinematics of a robot manipulator," vol. 6, no. 4, pp. 299-309, October 1988.

[6] A. P. M. L.-W. Tsai, "Solving the Kinematics of the Most General Six- and Five-Degree-of-Freedom Manipulators by Continuation Methods," pp. 189-200, 01 Jun 1985.

[7] C. R. T. A. E. S. Stuart R. Lucas, "On the Numerical Solution of the Inverse Kinematic Problem," 01 Dec 200.

[8] R. F. a. J. J. S. Reinhart, "Neural Learning and Dynamical Selection of Redundant Solutions for Inverse Kinematic Control," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference*, 2011.

[9] J. a. N. I. B. Zhao, "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures," *ACM Transaction on Graphics (TOG),* vol. 13, no. 4, pp. 313-336, 1994.

[10] R. Köker, "A Genetic Algorithm Approach to a Neural-Network-Based Inverse Kinematics Solution of Robotic Manipulators Based on Error Minimization," *Information Science,* vol. 222, pp. 528-543, 2013.

[11] J. L. Andreas Aristidou, "FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem," *Graphical Models,* vol. 73, no. 5, pp. 243-260, 2011.

[12] S. V. A. a. B. C. N. Manjaree, "Inverse Kinematics Using Neuro-Fuzzy Intelligent Technique for Robotic Manipulator," *International Journal of Computers, Communications and Control–Communicated,* 2013.

[13] J. H. H. G. Zhelong Wang, "Forward Kinematics Analysis of a Six-Degree-of-Freedom Stewart Platform Based on Independent Component Analysis and Nelder–Mead Algorithm," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions,* vol. 41, no. 3, pp. 589-597, 2011.

[14] Y. L. a. X. Z. Jingguo Wang, "Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm," *International Journal of Advanced Robotics Systems,* 2010.

Programmable reference generator to control a DC/AC converter for solar applications

David Navarro[1], **Nimrod Vazquez**[2], **and Domingo Cortes**[2]
[1]Instituto Politecnico Nacional
Superior School of Electrical and Mechanical Engineering
Av. Santa Ana No. 1000 Mexico City, Mexico.
[2]Instituto Tecnologico de Celaya
Department of Electronics
38010 Celaya, GTO, MÃl'xico
[1]david.navarro.d, [3]domingo.cortes@gmail.com, [2]n.vazquez@ieee.org

Abstract— *Renewable energies are the energy sources of the future. In recent decades it has invested in the development of solar energy systems, but for a better performance this implementations require a single phase inverter to regulate the output voltage. There have been developed a lot of inverter topologies, but most of them supply a differential output voltage. The differential voltage output is a disadvantage for photovoltaic panels applications, because the panel constitution has parasitic capacitances that cause leakage currents that reduce system performance and and reduce the panel lifetime. To overcome this issue many topologies have been developed to reduce the leakage currents and improve the solar energy efficiency, particularly the inverter topology developed in [12] accomplish this goal by eliminating the differential output voltage. Although this topology reduces considerably the leakage currents, the number of inductors an capacitors is increased as a consequence complicates the design of a controller.*

In this paper the single phase inverter is modeled by discontinuous state space equations, from which can be appreciated the difficulty to find the state references. An algorithm to find the references is developed from the state space equations, just considering the desired sinusoidal output voltage for the inverter. The algorithm can be programmed on a digital device using numerical approximations and so create a reference generator. The implementations of a control that regulates the output voltage of the inverter is easy if the state references are known parameters. So the reference generator is simulated and implemented on simulation to the single phase inverter, a sliding mode control is used to track the reference and try to regulate the output voltage.

Keywords: Programmable algorithm, control, calculations, power electronic control

1. Introduction

It is expected that in the year 2050 the population will rise to 9 million habitants worldwide. At the same time the economy will grow almost four times, as a consequence the energy and natural resources demand will increase [11]. Now a days the energy consumption is asociated to the population growth and the progress, that is countries with poverty have low energy consumption and richer countries haver higher energy consumptions.

According to an study realized by BP oil company on June 2012 [1], the world energy consumption mostly comes from fossil fuels and is about 87%, the nuclear energy is the 5%, hydroelectric is 6% and the 2% comes from other renewable sources (incluidng eolic, solar, biodiesel among others). Because most of the energy comes from non renewable resources that have begun to run out, the exploit of renewable sources is imminent. For such reason there are some measures to reduce the fossil fuel usage and at the same time ensure continuous and sustainable supply of energy, so the CO_2 emissions reduction goals can be achieve in 2050 and avoid climate change.

Renewable energies are expected to be the energies of the future because present a continuous regeneration, do not produce a significant environmental impact an are clean energies. The renewable energies are obtained from wind turbines, micro hydro turbines, hybrid systems, photovoltaic panels among many others. Actually the photovoltaic systems have great impact, and so many countries and companies are investing on the development of this technology.

The solar energy systems can not operate properly without single phase inverters, the importance of this kind of converters comes from the need to transform the direct current (DC) provided by the photovoltaic panel to alternate current (AC) used on many applications. In photovoltaic applications two kinds of inverters are distinguished, the isolated and the electric grid connected. Isolated converters don not have connection to the electric grid, those are commonly used on

distant locations in which is difficult to be reached by the electric grid, some examples could be pumping systems for water wells, lighting, among others. In some circumstances isolated photovoltaic inverter could be used to supply energy to buildings. However, the most common use of photovoltaic inverters is connected with the electric grid. The main goal of the grid connection is to reduce the energy consumption from the company that supplies the service.

The single phase inverters connected to the gird transform the DC supplied by the photovoltaic pane (PV) and inject the energy on the electric grid directly. According to the European Photovoltaic Industry Association by the end of year 2007 the energy supplied by photovoltaic systems was more than 7 GW, by the end of 2010 such energy overcame 100 GW, making a clear statement that there is a strong market for the development and use of this technology [4].

A very common disadvantage discussed in several papers is the fact that because of its connection most of the inverters have a leakage current to ground through parasitic capacitances (C_p) generated by the photovoltaic pane nature [13]. The leakage current can be reflected in low efficiency of the whole system, output current distortion, electromagnetic interference, security issues and eventually reduce the solar panel life time.

Some measures that have been presented to reduce the leakage currents are: the common mode voltage to be constant (without tension variations), apply a low frequency voltage to the parasitic capacitances and unplug the solar panel from the grid at the times where the leakage current could be higher, in other words when the load voltage is zero. Al those proposals have a slightly improvement to the inverter performance, but the main issue has not been properly solve. Another choice was to developed new inverter topologies to reduce the leakage currents and increase the inverters efficiency and provide a longer life time to the solar panel, in all the cases the traditional inverter has been considered as the basis [6], [5], [13].

In [12] a topology that eliminates the differential voltage output has been developed, and so the leakage currents are reduced increasing the panel life time. Also the new topology reduces the number of switches required, but the inductors and capacitors are doubled. To prove the functionality on [12] a sliding mode control was developed to regulate the output current, but in most of the applications the output voltage is the desired parameter to control. From some decades ago it is well known that current mode control is faster than voltage mode control [7], [8], on power electronic converters if the output voltage is considered as the system output the analysis and control is complicated because the converters are non minimum phase systems. Another trouble with power electronics is their discontinuous nature caused by the switches, which in many cases is associated to non linear issues and complicates the controller design.

The topology proposed in [12] is an interesting control problem, because the discontinuities, the fourth order system topology, many references are unknown, the non minimum phase behavior of many power converters make an interesting challenge. In this paper a programmable algorithm is developed to obtain the inverter references just with the desired output value. To prove the functionality of the reference generator a simple sliding mode control is developed to prove the reference generator could be used on practical applications. Commonly the power electronics required a fixed frequency, but on sliding mode control the frequency could not be fixed, so a result to adjust sliding mode control to a fixed frequency has been developed on [2]. In conjunction all this elements are put together and the control of the inverter is performed with a programmable algorithm that is used a reference generator for the sliding mode control developed to regulate the converter output voltage.

The paper is organized as follows, on section 2 a brief review of single phase photovoltaic inverters is presented, considering the development of some topologies to overcome the leakage current problem. On section 3 a single phase inverter topology is modeled and studied, an equilibrium points analysis is presented to know the converter capabilities on the output voltage. On section 4 the control problematic is presented, in which is shown that is difficult to obtain the references for the system and a programmable algorithm is developed to overcome this issue and control the inverter. On 4.1 the programmable reference generator is simulated and the simulated references to provide a desired output voltage are presented. On section 5 the programmable reference generator is implemented among a sliding mode control, a sliding surface is proposed to be implemented on the inverter to reach the desired output voltage. On section 5.1 the sliding mode control and reference generator are presented, showing that by this procedure is possible to find the system references. At last on section 6 the conclusions obtained from this research are presented.

2. Single phase inverter

A single phase inverter is a power electronic converter which converts DC current to AC current. There has been many topologies in which the performance is determined by the switches (BJTs, MOSFETs, IGBTs, MCTs, SITs y GTOs), so is important to choose the different transistors for each applications and the proper modulation technique to reduce the harmonic content [9]. To improve the performance on the inverter many topologies have been developed, but most of this topologies are variations from the traditional inverter presented on Figure 1. Almost all the inverter topologies supply the output by a differential voltage and for photovoltaic applications this is a disadvantage, because the panel constitution has parasite capacitances which create leakage currents reducing the application performance. To overcome the leakage currents some modifications have been

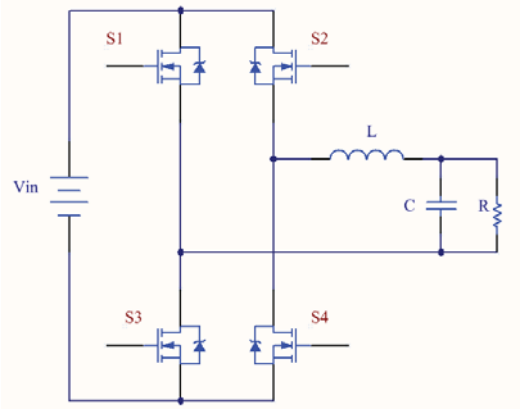


Fig. 1: Traditional single phase inverter

done to the inverters, there have been improvements but the problem has not been solved yet.

The firs attempts to improve the traditional power inverters was the H bridge inverter with MOSFET, and another step was to use MOSFET on the lower part of the H bridge and IGBT on the upper part of the bridge. Both topologies presented an improvement to the power electronic inverters, but only when galvanic isolation, without it the leakage currents were increased. The H5 topology is based on an H bridge, the main difference is a switch on the DC side which operates at high frequency, again the leakage currents are increased without galvanic isolation. The HERIC topology is designed from a common H bridge with a pair of switches in opposite directions, this switches are parallel to the output filter and the load. This topology reduces the leakage currents compared to the H5 or Hybrid inverter, but the leakage currents are still considerable on photovoltaic applications. An H6 inverter is presented on [13], [14] which is developed using 6 transistors, 3 on each inverter totem, this presents an advantage because dead time is not required because the three transistor will never be on at the same time. The parasitic currents are less than the traditional H bridge, but in practical applications without isolation the leakage currents are high.

The topology proposed in [12] minimizes the leakage currents, because the differential output voltage is eliminated by the connection between the output and the source ground. Also on this topology the commutation losses are decrease by the reduced number of switches, but even the topology has some advantages the main problem is that this new topology has a complicated operation that makes a difficult task to design a controller. In the next section the inverter is modeled and analyzed to determine their properties and capacities.

3. Single phase inverter model

In pohtovoltaic applications a common ground between the source and the output has important advantages to

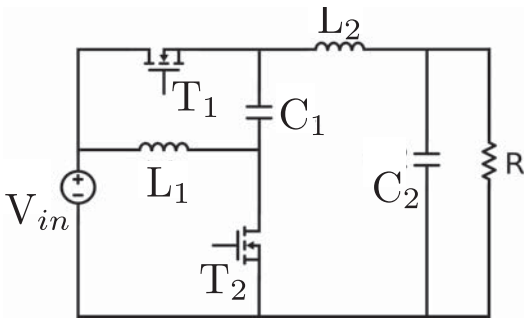


Fig. 2: Single phase inverter proposed topology

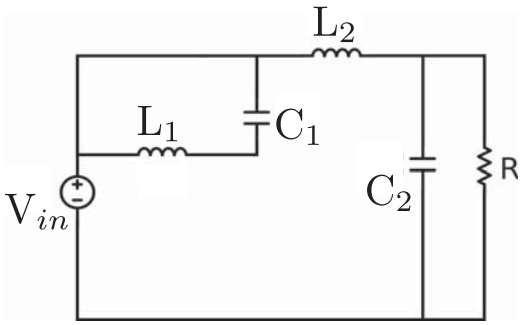


Fig. 3: Single phase inverter topology considering $u = 1$

harness the generated energy, a topology with such characteristics was developed in [12]. Another advantage that this topology presents is that only requires two switches, unlike other inverters that require at least four transistors. But in turn it requires two capacitors and two inductors, causing the increase of the system order, which could complicate the design of a controller.

The new single phase inverter topology is presented on Figure 2, from the schematic diagram it is possible to appreciate that the inverter output and the source have a common ground.

Consider from Figure 2 transistor T_1 is on and transistor T_2 is off, the equivalent circuit is presented on Figure 3,

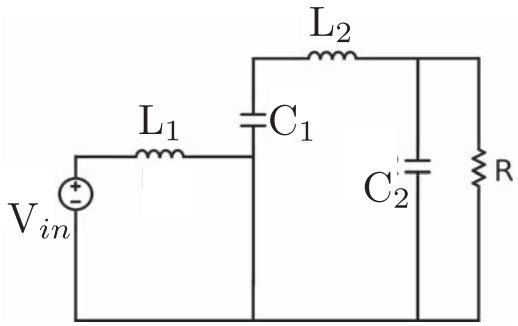


Fig. 4: Single phase inverter topology considering$u = 0$

from which are obtained equations

$$\dot{x}_1 = \frac{x_2}{L_1} \tag{1a}$$
$$\dot{x}_2 = \frac{x_1}{C_1} \tag{1b}$$
$$\dot{x}_3 = \frac{V_{in} - x_4}{L_2} \tag{1c}$$
$$\dot{x}_4 = \frac{x_3}{C_2} - \frac{x_4}{RC_2} \tag{1d}$$

where x_1, x_2, x_3 y x_4 are the current on inductor L_1, the voltage on capacitor C_1, the current on inductor L_2 and the voltage on capacitor C_2 respectively. The inductances L_1 y L_2, capacitances C_1 y C_2, and the input voltage V_{in}, are known constants. While resistance R remains as an unknown constant for analysis purposes.

Now consider from Figure 2 the transistor T_1 is off and transistor T_2 is on, the equivalent circuit is presented on Figure 4, from which are obtained equations

$$\dot{x}_1 = \frac{V_{in}}{L_1} \tag{2a}$$
$$\dot{x}_2 = \frac{x_3}{C_1} \tag{2b}$$
$$\dot{x}_3 = \frac{x_2 - x_4}{L_2} \tag{2c}$$
$$\dot{x}_4 = \frac{x_3}{C_2} - \frac{x_4}{RC_2} \tag{2d}$$

Combining models (1) and (2) is obtained

$$\dot{x}_1 = \frac{V_{in}\,(1-u) - ux_2}{L_1} \tag{3a}$$
$$\dot{x}_2 = \frac{ux_1 + (1-u)\,x_3}{C_1} \tag{3b}$$
$$\dot{x}_3 = \frac{V_{in}u - (1-u)x_2 - x_4}{L_2} \tag{3c}$$
$$\dot{x}_4 = \frac{x_3}{C_2} - \frac{x_4}{RC_2} \tag{3d}$$

considering that $u \in \{0, 1\}$.

From model (3) is possible to know the inverter capabilities, and most important stablish the inverter output limits. An equilibrium points analysis is developed to determine the inverter capabilities, the equilibrium points could be find considering the derivatives on (3) equal to zero obtaining

$$0 = V_{in}\,(1-u) - ux_2 \tag{4a}$$
$$0 = ux_1 + (1-u)\,x_3 \tag{4b}$$
$$0 = V_{in}u - (1-u)x_2 - x_4 \tag{4c}$$
$$0 = x_3 - \frac{x_4}{R} \tag{4d}$$

rearranging and solving for each state variable as a function of u is obtained

$$x_1 = \frac{V_{in}}{R}\left(\frac{-1 + 3u - 2u^2}{u^2}\right) \tag{5a}$$
$$x_2 = \frac{(1-u)\,V_{in}}{u} \tag{5b}$$
$$x_3 = \frac{V_{in}}{R}\left(\frac{2u-1}{u^2}\right) \tag{5c}$$
$$x_4 = V_{in}\left(\frac{2u-1}{u}\right) \tag{5d}$$

where x_4 represents the inverter output voltage, so from (5d) is possible to obtain the output voltage range that the inverter is capable to supply. If $u = 1$ in (5d) the output voltage is V_{in}, meanwhile if $u = 0$ the output voltage is $-\infty V_{in}$. From output voltage limits provided by the inverter is possible to conclude that the converter can generate a sinusoidal wave of $V_{in}\sin(\omega t)$. However, even when it has been shown that the inverter can supply the desired output voltage is not enough, a control law should be provided.

Knowing the inverter can supply a desired output voltage the next step is the design of a control law that achieves such goal. Most of the inverter topologies are second order systems and so a control law could be implemented without complications. In the proposed topology the case is different because in this is a fourth order systems, so the design of the control law is not so easy and considering the non linear and non minimum phase characteristics the power electronic converters commonly have the controller design is a challenging duty. In the next section the controller is designed analyzing the troubles generated by the difficulty to find the references for each state, so a programmable reference generator is developed.

4. Design of a reference generator for an inverter

The desired performance of a single phase inverter is to commutate the transistors such that the output voltage is a sinusoidal wave with desired amplitude and frequency. To control the inverter in [12] a sliding mode control was developed to test experimentally the inverter capabilities, but some states were not considered in the controller design. Even when the experiment show that the inverter could generate a sinusoidal wave at the output, is important to consider and verify that all the states on the converter are under bounded limits because of the non minimum phase characteristics of power electronic converters . For the single phase inverter is simple to obtain the references for states (x_3 y x_4), but to find the references for the states (x_1 y x_2) is a complicated labor, that is why the controller designed in [12] is based on x_3.

From model (3) is appreciated that the references for states x_3 y x_4 are easy to find, but the references for the states x_1 y x_2 are not trivial. The reference for the state x_4 is a simple

one because is the desired output voltage, which is expected to be $k \sin \omega t$, where k is the desired amplitude and ω is the desired frequency. Knowing this now the main issue is how the references x_1, x_2, y x_3 should be when $x_4 = x_{4d}$.

To know the value that x_3 should have when $x_4 = x_{4d}$ could be obtained from (3d), where

$$x_4 = x_{4d} = k \sin(\omega t) \tag{6}$$

and the value for $\dot{x}_4$ could be found derivating (6) and so obtaining

$$\dot{x}_4 = \dot{x}_{4d} = k\omega \cos(\omega t) \tag{7}$$

substituting (6) and (7) in (3d) and solving for x_3 is obtained

$$x_3 = \dot{x}_{4d} C_2 + \frac{x_{4d}}{R} \tag{8}$$

that is the current that should flow on inductor L_2 when $x_4 \rightarrow x_{4d}$. The references for the states x_1 and x_2 are more complicated to be found, because both are dependent from the control variable u and at this point is an unknown parameter. To find the expression for u when (6) could be obtained from (3c), see that $\dot{x}_3$ is an unknown parameter, but it could be find by derivating (8) and is obtained

$$\dot{x}_3 = \ddot{x}_{4d} C_2 + \frac{\dot{x}_4}{R} \tag{9}$$

from (9) the term $\ddot{x}_{4d}$ is unknown, but could be found derivating (7) and so is obtained

$$\ddot{x}_4 = \ddot{x}_{4d} = -k\omega^2 \sin(\omega t) \tag{10}$$

Substituting (9) and (6) in (3c) and solving for u is obtained

$$u = \frac{\dot{x}_3 L_2 + x_2 + x_4}{V_{in} + x_2} \tag{11}$$

in which all the terms except for x_2 are known parameters.

With the expression for u to generate the output (6) the current issue is to find a expression for x_1 y x_2. From system (3) equations (3c) and (3d) have been occupied to obtain (8) and (11), leaving available

$$\dot{x}_1 = \frac{V_{in}(1-u) - u x_2}{L_1} \tag{12a}$$

$$\dot{x}_2 = \frac{u x_1 + (1-u)}{C_2} \tag{12b}$$

the procedure to obtain the values for x_3 and x_4 can not be used on model (12), because mathematically is a problem to solve two equations with 4 variables (x_1, x_2, $\dot{x}_1$ y $\dot{x}_2$) and the problem can not be solved by that mean.

A simple way to find the references x_1 and x_2 is by considering (12) as a system of first order differential equations, so a solution could be found by numerical approximations considering that u is obtained as (11). Solving the system (12) by numerical approximations is a good solution that can be used on programmable devices to implement real applications. This methodology has presented a proper manner to find the references for some DC-DC power electronic converters with good results [2].

V_{in}	$350V$
L_1	$2mH$
C_1	$110\mu F$
L_2	$1mH$
C_2	$2.2\mu F$
R	7Ω
x_{4d}	$127\sin(2\pi 60t)$
$\dot{x}_{4d}$	$47877.87\cos(2\pi 60t)$
$\ddot{x}_{4d}$	$18049532.52\sin(2\pi 60t)$

Table 1: Design parameters

Solving the system by numerical approximations is not enough, because the initial conditions allow the solution to be achieved with less iterations and so the converter to achieve the desired output voltage faster. The proper initial conditions of system could be found from the equilibrium points (5). So to find the appropriate initial conditions the initial output voltage (x_4) should considered, and so solving (5d) for u and substituting in (5a) and (5b) the initial conditions for x_1 y x_2 could be found.

Since most of power electronic converter have non minimum phase characteristics solving numerically the system (12) could occur that the solution is unstable. See that the system (12) is not an exception and simulated considering t will give an unstable response, but if the system is simulating considering $\tau = -t$ the solution is stable and so the references for x_1 y x_2 are found.

4.1 Simulation

To validate the results presented a simulation presented in this section, the references for the states x_1 and x_2 are generated from model (12) and the system is simulated considering the parameters of Table 1. The initial conditions are important parameters that should be taken with care, so in the simulation the initial conditions are selected in such a way that the system reaches faster to the solution expected. If the parameters from Table 1 are substituted in (5) is obtained

$$x_1(0) = 0 \tag{13a}$$

$$x_2(0) = 350 \tag{13b}$$

as the system initial conditions. The differential equations are solved with the runge-kutta method using a $1\mu s$ step. The simulation is developed considering the time τ so the system converges to a bounded solution. On Figure 5 the references for states x_1 and x_2 when the output voltage is $x_4 = 127\sin(2\pi 60t)$ are presented. It is possible to appreciate that this references do not have sinusoidal form, but those signals maintain a constant frequency equal to the reference frequency.

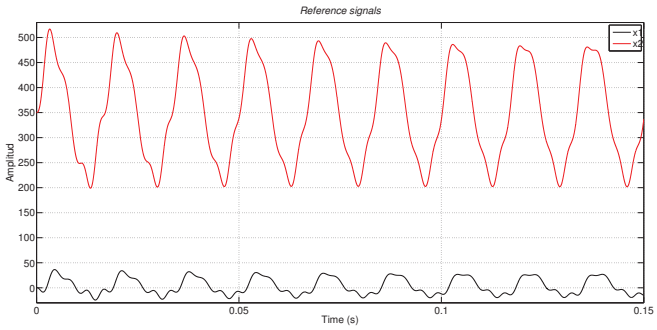


Fig. 5: Reference signals

The simulation of system (12) with time τ allows to numerically obtain the references for the states x_1 and x_2 considering the system is forced to $x_4 = k \sin(2\pi f t)$, so this development could be considered as a reference generator for the inverter on Figure 2. Commonly if the reference for all the states are known the design of a controller is a simpler task and so many control techniques could be applied. On the next section the reference generator is tested under a sliding mode control for the single phase inverter.

5. Sliding mode control

The converter natural discontinuities match with the discontinuous control technique sliding mode control. In this kind of control methodology the most difficult part is to find a proper sliding surface, but for the single phase inverter is not a very complex process because all references are well known.

Since many decades ago many results have proved that current mode control provides faster responses [7], [8] for such reason a control law that would enhance the performance of the single phase inverter is

$$u = \begin{cases} 1 & if\ \sigma > 0 \\ 0 & if\ \sigma < 0 \end{cases} \tag{14}$$

considering the sliding surface

$$\sigma(x) = x_1 - x_{1ref} \tag{15}$$

An issue applying sliding mode control to power electronic converters is the commutation frequency, because in the sliding mode control the frequency can not be regulated, and the power electronic converters are designed to operate at a specific frequency. To adjust sliding mode control to a fixed frequency has been implemented on DC-DC converters just by introducing the sliding surface to a PWM. As a result in [3], [10] has been demonstrated that the equivalent control of sliding mode control is in fact the duty cycle of linear models. This result could be applied to the single phase inverter presented on this paper, because the inverter has two transistors and can be operated by a simple PWM.

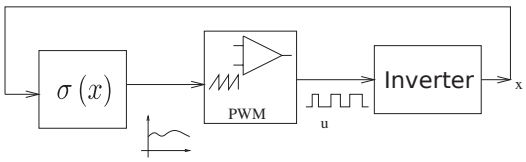


Fig. 6: Implementation of a sliding mode controller with fixed frequency

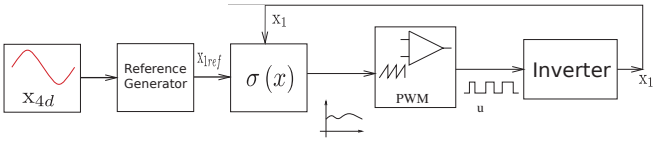


Fig. 7: Controller implementation using the reference generator

On Figure 6 is represented the implementation of a sliding mode controller with fixed frequency, but to implement the controller with the reference generator developed on this paper some modifications are required. On Figure 7 is presented the controller implementation with the reference generator, this scheme will be simulated and the results are presented on the next section.

5.1 Simulation

In this section the sliding mode control for the single phase inverter is simulated considering the reference generator as illustrated on Figure 7, the simulation parameters are presented on Table 2.

The reference generator (12) and (11) with controller (14) and (15) are simulated, on Figure 8 is presented the simulation results presenting the output current, output voltage and reference voltage. See that on the voltage a de-phase with respect to to the reference, on the amplitude there is a very insignificant difference in which the reference between the output that not exceeds the $5V$. As a conclusion the results presented on Figure 8 show that the implementation of the reference generator and the sliding mode control allow the inverter to supply a derided voltage.

As a comparison, the current measured on the single phase inverter is presented and compared with the reference provided by the reference generator. On Figure 9 is possible to appreciate that the current on inductor L_1 and the reference match on every point. So the output voltage difference is associated to the components dynamics and their values, or inclusively a better sliding surface should eliminate the steady state error.

To show how components parameters affect the system response, a new simulation is presented but in this case the inductor $L_1 = 2mH$, on Figure 10 is presented the output voltage compared with the reference voltage. See that the de-phase and the amplitude are reduced considerably, so the parameters ares a important to the system. So the

V_{in}	$350V$
L_1	$1mH$
C_1	$110\mu F$
L_2	$1mH$
C_2	$2.2\mu F$
R	7Ω
V_{ref}	$127sin(2\pi 60t)$
f_{sw}	$40kHz$

Table 2: Design parameters

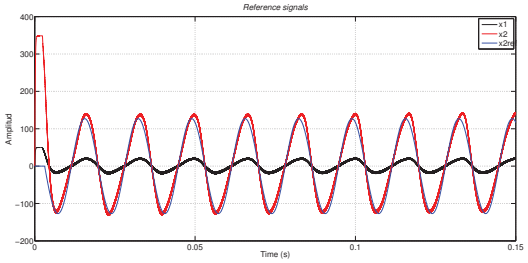


Fig. 8: Single phase inverter output

programmable reference generator is a good starting point to control systems with unknown state references, but of course the control technique could be improved.

6. Conclusions

On this paper has been presented a methodology to developed a programmable reference generator for control applications. This solution is very useful when the system references are unknown or hard to be found. In order to proved present a practical example the single phase inverter is controlled by a sliding mode control scheme, showing a good result. The simplicity of this methodology makes viable

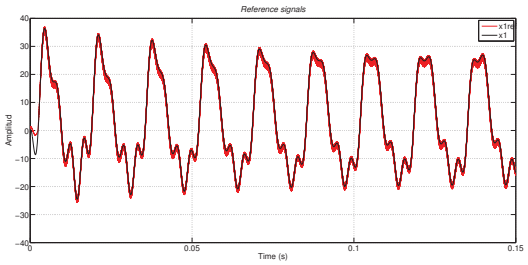


Fig. 9: Reference current vs real current

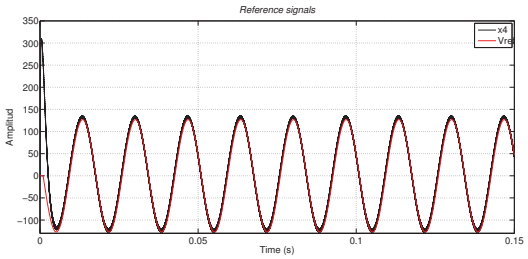


Fig. 10: Output voltage vs reference voltage

to implement this reference generator on programmable devices for physical implementations.

This result should give important benefits on the control area, where in many cases the principal issue is how to find the references for the system. So the reference generator could be implemented with so many control techniques to evaluate the better performance. As an example on this paper is shown how a difference on the parameters could be reflected on the system performance.

Acknowledgment

This research has been partially supported by Mexico City Secretariat of Science, Technology and Innovation under grant ICYTDF-198-2012.

References

[1] BP Statistical Review of World Energy. June 2010.

[2] J. Alvarez, D. Cortes, and J. Alvarez. Indirect control of high frequency power converters for ac generation. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 4, pages 4072–4077 vol.4, 2000.

[3] Domingo Cortes, Jaime Alvarez, and Joaquin Alvarez. Output feedback and dynamical sliding-mode control of power converters. *International Journal of Electronics*, 98(4):505–519, 2011.

[4] European Photovoltaic Industry Association EPIA. Global Market Outlook: For Photovoltaics 2013-2017. *http://www.epia.org/home/*.

[5] Bin Gu, J. Dominic, Jih-Sheng Lai, Chien-Liang Chen, T. LaBella, and Baifeng Chen. High reliability and efficiency single-phase transformerless inverter for grid-connected photovoltaic systems. *Power Electronics, IEEE Transactions on*, 28(5):2235–2245, May 2013.

[6] T. Kerekes, R. Teodorescu, P. Rodriguez, G. Vazquez, and E. Aldabas. A new high-efficiency single-phase transformerless pv inverter topology. *Industrial Electronics, IEEE Transactions on*, 58(1):184–191, Jan 2011.

[7] Yan-Fei Liu and P.C. Sen. A general unified large signal model for current programmed dc-to-dc converters. *Power Electronics, IEEE Transactions on*, 9(4):414–424, Jul 1994.

[8] R.D. Middlebrook. Modeling current-programmed buck and boost regulators. *Power Electronics, IEEE Transactions on*, 4(1):36–52, Jan 1989.

[9] Muhammad H. Rashid. *Power electronics: circuits, devices, and aplications*. Ed. Pearson Prentice Hall, Tercera edición.

[10] Eva M. Navarro-López, Domingo Cortés, and Christian Castro. Design of practical sliding-mode controllers with constant switching frequency for power converters. *Electric Power Systems Research*, 79(5):796 – 802, 2009.

[11] OECD. OECD Environmental Outlook to 2050: The Consequences of Inaction. *OECD Publishing, doi:10.1787/9789264122246-en., 2012.*

[12] Marco Luis Rosas Compeán. *Convertidor CD/CA sin aislamiento con conexión a la red eléctrica para aplicaciones en paneles solares*. PhD thesis, IPICYT, 29 September 2014.

[13] Wensong Yu, Jih-Sheng Lai, Hao Qian, C. Hutchens, Jianhui Zhang, G. Lisi, A. Djabbari, G. Smith, and T. Hegarty. High-efficiency inverter with h6-type configuration for photovoltaic non-isolated ac module applications. In *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE*, pages 1056–1061, Feb 2010.

[14] Li Zhang, Kai Sun, Yan Xing, and Mu Xing. H6 transformerless full-bridge pv grid-tied inverters. *Power Electronics, IEEE Transactions on*, 29(3):1229–1238, March 2014.

Online Auction and Secretary Problem

Greg Harrell, Josh Harrison, Guifen Mao, and Jin Wang
Department of Mathematics and Computer Science
Valdosta State University, Valdosta, GA 31698, USA

Abstract - *In this study, we are focused on a set of problems of a very specific and popular topic; The Online Auction, Secretary Problem, and K-Secretary Problem. Detailed discussion of methods is used to obtain each problems optimal solution. Also, models to help guide the explanation. We describe the relevance of each and how they relate to one another in finding probability as well as the optimal solution. Last but not least, research in the algorithms associated in the method was discussed for finding the optimal solution.*

Keywords: Secretary Problem; Online Auction; K-Secretary Problem

1. Introduction

You are in need of a home and you are trying to find the perfect environment to live in. You are given a set of locations to choose from. They are revealed to you one by one and you must make a decision to take that current location or pass on it for the remaining choices, without going back. Think about it. How would you handle this situation to have the highest probability in acquiring the best place to live in and not end up with a location that is not ideal? Choices like this and the solution, is what will be discussed in detail.

2. Secretary Problem

The Secretary Problem is a very popular problem that has great use. It aroused from trying to decide the best secretary out of a group of individuals. However, it is not just as simple as a quick selection. The challenge arrives when you have no knowledge of them, until you interview each individual one by one, each having the same probably of being selected, and you must decide right then rather to accept them or to deny with no going back.

2.1 History

This problem first appeared in the late 1950's and early 1960's. It is a problem that spread throughout. It was similar to the current problems at the time, for example the marriage problem. However, it had a shocking solution. With the amount of attention and growth this problem was getting, it became a field of study. In research papers, Statisticians Lindley (1961), Dynkin (1963), Moritguti, Chow, Samuels and Robbins (1964), and Gilbert and Mosteller (1966), were trying to solve this problem. They were at a race to see who would be the first to solve the problem (Thomas page 282).

2.2 Optimal Solution

The primary goal is to find the best contestants for the secretary job position. This is what you call, the optimal solution. In other terms, the optimal stopping is *r*. To acquire such solution, a method or algorithm is necessary. For this particular problem, the strategy in finding the optimal *r* is to use the stop method, or the stopping rule. Selecting a set of $r - 1$ contestants, set S, of *n* individuals and deny them the position automatically. Then you compare *r* contestants to the set S. If the current contestant is superior then set S, you select that contestant as the solution. If it was not superior, then you move on to the next one in line, selecting the *n*th contestant if no other options were superior. Varying the size of set S will give you multiple results, vary the probability of succeeding in finding the best contestant. Article [5] provides the following algorithm that will provide the probability of success, given *r*:

$$P(r) = \sum_{i=1}^{n} P(\text{applicant } i \text{ is selected} | \text{applicant } i \text{ is the best}) * P(\text{applicant } i \text{ is the best}) \quad (1)$$

$$= \sum_{i=r}^{n} (r-1)/(i-1) * \frac{1}{n} = \frac{r-1}{n} \sum_{i=r}^{n} \frac{1}{i-1}.$$

Article [1] states, "Lindley [1961] and Dynkin [1963] proved that a generalization of this strategy to a setting with *n* applicants yields a probability approaching 1/e ≈ .37 of hiring the best secretary, and that this is the best possible guarantee." There has been results that help prove this statement. In article [5], it also states, "Letting *n* tend to infinity, writing *x* as the limit of *r/n*, using *t* for *i/n* and *dt* for 1/*n*", which can be represented by the following integral:

$$P(x) = x \int_{x}^{1} 1/t \, dt = -x log(x). \quad (2)$$

The derivative of this integral in respect to *X*, will prove the optimal x is equal to 1/e, when solving for *x* and setting it equal to 0.

2.3 Secretary Model

A model was constructed to simulate the Secretary Problem using Java, a programming language. In the model, algorithm (1) was used on a set of 10 individuals, ranking them from 0-9 and rank 9 being the optimal solution. The model tested each value of *r* to use for the stopping rule, 2-7, 1 million times and the Figure 1 resembles the average results:

r	2	3	4	5	6	7
Model: P(*r*)	0.2827	0.36639	0.39848	0.39843	0.37249	0.32633
Algorithm: P(r)	0.2828	0.36579	0.39869	0.39823	0.37281	0.32738

Figure 1

The results from the model, compared with the formula, are almost identical. As the trials increase, the Model result will tend to the result from the formula, algorithm (1).

3. K-Choice Secretary Problem

The K-Choice Secretary Problem is similar to the Secretary Problem, note the name. However, there is a big difference between the two. Previously, you experienced the Secretary Problem and also seen the results of the simulation. You were introduced to a method that could be used to select the best choice of a set of n items, algorithm (1). Now, what if you want a set of k elements that are the best? This is where the K-Choice Secretary Problem was originated.

3.1 Optimal Solution

There is a process in finding the optimal solution for this problem. It is similar to the Secretary Problem, however, there is a few more steps you have to do in this situation. Article [4] states the following algorithm:

(a) Observe the first [*n/e*] elements. (3)

(b) Remember the best k elements among these first [*n/e*], and call this set T. If $k >$ [*n/e*], then let T consist of the first [*n/e*] elements observed, together with k - [*n/e*] "dummy elements" of zero value.

(c) Whenever an element arrives whose value is greater than the minimum-value element in T, select this element and delete the minimum-value element from T.

3.2 Theory

The theory behind the previous solution, algorithm (3), is a very ideal and optimal approach to the problem. Taking the best k elements of the set of elements that were observed based on the stopping rule, algorithm (1), and comparing them with the following elements, will result in the highest probability of obtaining the optimal solution. Theoretically this seems correct, analyzing the first set of elements to get an idea of value of possible options and comparing the rest of the elements with the top k analyzed elements. As long as you follow the stopping rule, it will appear optimal. With further study, discussed in article [4], it was discovered that the constant *e* becomes less optimal as *k* approaches infinity.

3.3 Example

Branching off from the previous example discussed, you want to hire a team of the top two candidates. Ten people applied for the team. Just like previously, you are unaware of the value of each candidate, until you interview them one by one. During the interview, you have to make a decision to hire them or to let them go for good. The optimal way of solving this problem is to first use the stopping rule, algorithm (1), to determine the cut off for the set that will be analyzed. Then, you form a subset, team, of *k* candidates with the best valued options. In this example, $k = 2$ and the cut off will be 3. The team will consist of the top 2 of the set of 3 that were analyzed, set T. Once you have set T, compare each of the remaining candidates to the team. If the candidate has a greater value than the minimum member of the team, you select that candidate and remove the minimum member from the team. You continue this method until that you have selected 2 candidates or until *k* – selected element(s) are left.

The following figure is a visual of the example:

Ten candidates in random order, ranked from 1-10 in value	4, 10, 5, 3, 9, 7, 6, 2, 8, 1
Cut off set based on stopping rule, r = 4.	4, 10, 5
Subset, set T	5, 10
End result after comparing the remaining candidates	9, 1

Figure 2

As you can see, in this example, the optimal team was not selected due to probability, the 10 was in the cut off set. After selecting 9 because the value was greater than 5, no remaining candidate had greater value then the remaining candidate with the rank of 10 in set T. So, the last k-selected remaining element(s) was selected along with the 9. This is the result in 9, 1.

4. Online Auction

4.1 History

Online Auction is the best source for an example of modern markets. It ranges from all different categories, more specifically, networked markets. It is a method used to sale or purchase goods on a joined network. This gives business an option in being in a more preferred environment for the buyers or even the sellers. Also, being on the web can give you more access as far as buyers, or even goods to browse through. As a seller, using the optimal stopping rule, theory, can be a great power in the online auction environment as well. Article [2] states, "This first has been done by Hajiaghayi et al. [2004] who considered the well-known *secretary problem* in online settings."

4.2 Relations with K-Choice Secretary

Online Auction and the K-Choice Secretary Problem have many similarities. This is why the K-Choice Secretary Problem is a very powerful tool in the online setting. The same optimal logic that is involved with finding the best way to handle the K-Choice Secretary Problem can be used for online auction. In setting of an online auction, as a seller, you are trying to bid off product that you own at the greatest sale price possible. However, you do not want to turn down a bid that may be the highest you will ever get with the current auction. As you can see, this is a similar scenario as the K-Choice Secretary Problem. When you are having an auction online, buyers will randomly approach and bid on the current item. It is unknown how much the next person will bid, or even if there will be anyone else. So, a decision has to be made on the spot to take the current bid or not, just like the Secretary Problem or the K-Choice Secretary Problem, if it is an auction of multiple items.

5. Apply K-Choice With Online Auction

This section a Model will be built to apply the K-Choice Secretary Problem with the online setting of an auction to show that the secretary type problems can be used for an optimal strategy for Online Auction environment.

5.1 Application Model

The model simulation is based on an auction of cars. There will be a total of 100 cars that are up for auction that individuals can bid on. For testing purposes, there will be a set of 300 buyers that will be randomly selected to bid one by one. This process will continue until there are no more cars.

5.2 Optimal Strategy

To find the optimal solution to this application model, the same strategy will be used as the one used for the K-Choice Secretary Problem. Review algorithm (3) for details. Once a buyer is found that has the bid amount needed based on the analysis, we will accept that buyer and sell one car from auction.

5.3 Algorithm: Pseudo Computer Code

Int k = 100; Int n = 300; Int r = 111, based on cut off rule $1/e$;

Array[] buyers = Group of random ordered buyers with one bid value;

Array[] setT = Top k buyers from the 110 (r – 1) analyzed set based on cut off rule, r.

Array[] acceptedBuyers = The buyers that are accepted based on comparison of first analyzed set of buyers.

```
While( k != 0){
Current = the next available buyer that appears with their bid.
        If( number of left over buyers is equal to number of left over cars left for auction)
                Break, and default accept all the left over buyers.
        If( Current's bid > then minimum buyer's bid in set T)
                Remove minimum buyer from set T;
                Add Current to acceptedBuyers set.
                Sell the car the current buyer, k – 1;
}
Return acceptedBuyers;
//No more cars to auction
```

6. Extended Research

Recently there was discussion about the optimal solution for the secretary problem and the corresponding optimal solution. There were two algorithms, one for small amount of n elements, algorithm (1), and another that shows as n attends infinity, the cut off rule is $1/e$, algorithm (2). I decided to research and use a model to see what happens to the optimal r from algorithm (2) as n tends to infinity.

6.1 Model

The model to obtain information on what happens to the accuracy of the optimal r produced by algorithm (2) was designed to use both algorithms and analyze both

results. There will be multiple tests on multiple n sizes. The model will produce the portion of P(r) results from algorithm (1) around the optimal r produced by the algorithm (2), limiting excess data.

6.2 Results Discussion

The results of the previous model can be found on next page. There were four tests produced using this model. Examining each one, you can see the lower the n value, the smaller the variance. Using 1/*e* to determine the cut off rule seemed to be accurate, when rounded, roughly below 100. With further testing, you can see the variance increase from the optimal r produced by both algorithms. Testing the value n = 24,333, algorithm (2) produced r = 9003.210. However, based on the results from algorithm (1), the optimal r was found $\approx$ 8934 – 8970. Now, the variance may become minuscule compared to the size on n, so using algorithm (2) is very useful in terms of cost.

7. Summary

A quick overview to sum up the overall information discussed in this paper. The Secretary Problem is a very power tool that can be used in multiple different ways and be the fundamentals of a solution to other problems. The background and history of where it first appeared were discussed. It was shown that it can be solved in multiple ways as far as the algorithm used. K-Choice Secretary problem is another form of the Secretary Problem and how it was used to find an optimal r for multiple selections. Online Auction and the explanation of its type of environment and how the Secretary / K-Choice Secretary problem was a major tool in the online environment, giving an optimal solution. Models of each mentioned to show how they relate to one another and explanation of finding the probability and optimal r. Also, research in optimal comparison was studied between algorithms (1) and (2) by examining their results in multiple *n* tests.

8. References

[1] Baibaioff, M, Immorlica, N., Kempe, D., and Kleinberg, R. *Online Auctions and Generalized Secretary Problems.* ACM, Inc Copyright, June 2008. url: http://www.sigecom.org/exchanges/volume_7/2/babaioff.pdf

[2] Bateni, M, Hajiaghayi, M., and Zadimoghaddam, M. 2013. Submodular Secretary Problem and Extensions. ACM Trans. Algor. 9, 4, Article 32 (September 2013), 23 pages.

[3] Ferguson, Thomas S. *Who Solved the Secretary Problem?* Institute of Mathematical Statics, 1989, 282-289.

[4] *Secretary Problem, Wikipedia:* http://en.wikipedia.org/wiki/Secretary_problem

Low-Noise Fast Digital Differentiation Filters

Dr. Abdulwahab A. Abokhodair
King Fahd University of Petroleum and Minerals
Department of Earth Sciences
Dhahran 31261, Saudi Arabia
akwahab@kfupm.edu.sa

Abstract *- Differentiation of discrete data is a classical problem of data analysis which arises in many scientific fields ranging from biology to chemistry and the geosciences. Because of its step-size sensitivity, conventional FD method is not suitable for discrete data collected at a preset sampling frequency. This paper introduces a class of differentiation filters known as least-squares digital differentiators (LSDD). I discuss methods of their fast generation for 1D and 2D data and examine their properties in the space and frequency domains. The filters have a range of desirable properties which include ease of generation with simple integer coefficients thus reducing the risk of cumulative round-off errors. They are low pass linear phase, maximally flat and moment preserving. The low-pass nature of the filters renders them noise suppressant with very low noise amplification factor, hence the filters are actually multitasking, performing data smoothing and differentiation simultaneously. They are easily generated for any order derivative with arbitrary length to suite any desired sampling frequency.*

Keywords: Digital Filters, Derivatives, Gradients, Separable filters.

1. Introduction

Digital differentiation is widely used in many scientific fields for signal processing, imaging and data analysis. In potential field geophysics, for example, the usefulness of the spatial gradients as effective interpretation tools has long been recognized. Compared to the measured fields, gradients of the fields have greater spatial resolution, better definition of lateral boundaries, added depth discrimination and filtering properties, and better structural indicators.

Examples of the use of gradients for detailed interpretations of specific geologic structures may be found in [1-5]. The high detectibility and resolution power of gradients are illustrated in [6-9], where gradients are utilized for locating and mapping near-surface cultural and archaeological artifacts.

With recent advances in the power and graphics capabilities of modern computers, new gradient-based technologies have emerged. These include high resolution detection of geologic boundaries, Werner deconvolution for source depthing, Euler deconvolution and its extended form for the calculations of physical properties contrasts, dip information, location and depth of source, analytic, enhanced analytic signal and local wave numbers for source characterization and imaging [10-15]. The success of these new technologies made numerical computation of the spatial gradients and higher derivatives a basic geophysical data processing step.

This paper introduces a class of differentiation filters known as least-squares digital differentiators (LSDD). These are very popular in absorption spectroscopy, chromatography and medical technologies, but are virtually unknown in the geosciences literature. Least-squares filters may be constructed and applied in a computationally efficient way. They are in effect multi-tasking, performing both smoothing and differentiation simultaneously.

2. Filter Generation

LSDD filters are based on the principle of least squares data fitting. The underlying idea is to fit a vector **x** of equally spaced data of length 2n+1 to a polynomial $\mathrm{p_d}(k)$ of degree *d* in the integer index *k*, such that:

$$\mathbf{x} \simeq \mathbf{V}\mathbf{c}\,, \qquad (1)$$

where **V** is Vandermonde matrix with elements $v_{kj} = k^{\,j}$, $(j = 0,1,...,d)$ and **c** is the (*d*+1) vector of polynomial coefficients. The least-squares solution is the familiar normal equations:

$$\hat{\mathbf{c}} = \left[\left(\mathbf{V}^T\,\mathbf{V}\right)^{-1}\mathbf{V}^T\right]\mathbf{x} = \mathbf{H}\,\mathbf{x}\,, \qquad (2)$$
$$\mathbf{H} = \left(\mathbf{V}^T\,\mathbf{V}\right)^{-1}\mathbf{V}^T$$

The individual filter operators are the elements of $\hat{\mathbf{c}}$ given by equation (2). Each filter may be extracted explicitly by impulsing the matrix **H** with a unit impulse of appropriate delay, or the entire set of filters may be extracted at once by post multiplying **H** with an identity matrix of appropriate size. The Matlab script below generates the 1D filters for a given data window half-width (*hw)* and any polynomial degree (*d)*.

```
function H = ls1D(hw,d)
% Usage: H=sg1D(hw,d)
% Output
% H = Filter coeffecients
 W = 2*hw+1 ;
Nc = d+1;

if(Nc > W)
    error('Data window too small')
end
[i,j]=meshgrid(-hw:hw,0:d)
 G = (i.^j)';
id = eye(W);
 D = G\id;
 f = repmat(factorial((0:d)'),1,W);
H = (f.*D)';
```

3. Properties of LSDD

LSDD filter kernels for 1D and 2D data are compared in Figure 1. It is clear from the figure that the 1D filters are principle sections of their corresponding 2D kernels along the differentiation axis. This suggests that the properties of the 2D filters may be completely investigated from their 1D counterpart.

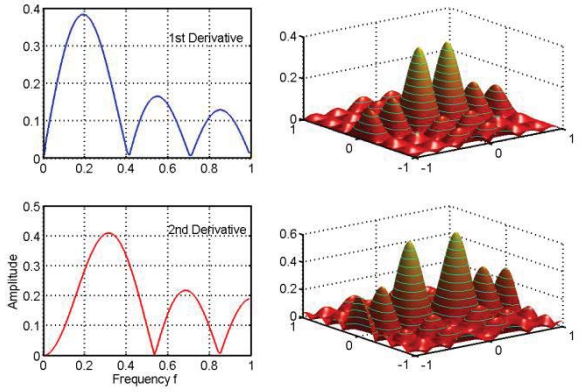


Figure 1: *Amplitude spectra of first (top) and second derivative (bottom) filters for 1D (Left) and 2D (right) data.*

3.1. General properties

The impulse responses of first- and second-order derivative filters are the same for any two consecutive degrees of the underlying polynomials. Thus the polynomial pairs of degrees (1,2), (3,4), (5,6), …, etc. produce the same first order differentiators, while polynomials of degree (2,3), (4/5), (6/7), …, etc. produce the same second order differentiators. Thus only even-degree polynomials produce unique first and second order derivative operators. Filters of both derivative orders are linear phase, non-recursive, FIR, highly stable and self-damping. The amplitude spectra of the filters are maximally flat closely approximating the ideal low-pass digital differentiation filters (IDD) spectra at low frequency and attenuate rapidly at high frequencies (Figures 2). Thus, unlike IDD which strongly amplify noise especially at high frequencies, LSDD filters are low-pass filters, which is a significant advantage in practical applications.

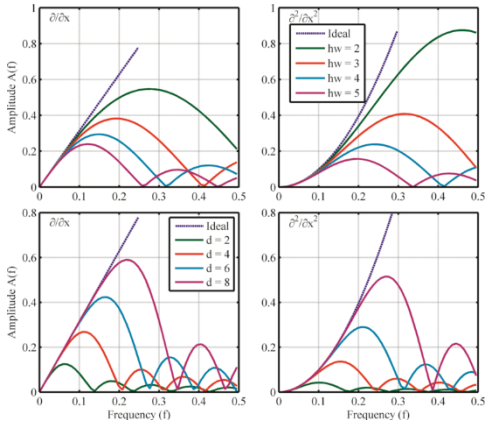


Figure 2: *Amplitude responses of first derivative (left) and second derivative (right) filters of different polynomial degrees (d) (left panel) and filter half-width (hw) (right panel compared to the responses of IDDs.*

3.2. Spectral properties

The major spectral properties of LSDD filters are primarily determined by two parameters - the degree of the generating polynomial (*d*) and the filter half-width (*hw*). The initial slope of the main lobe, the width of the pass-band, the roll-off rate and the cut-off frequency, all vary with variations in the parameters of the underlying polynomial.

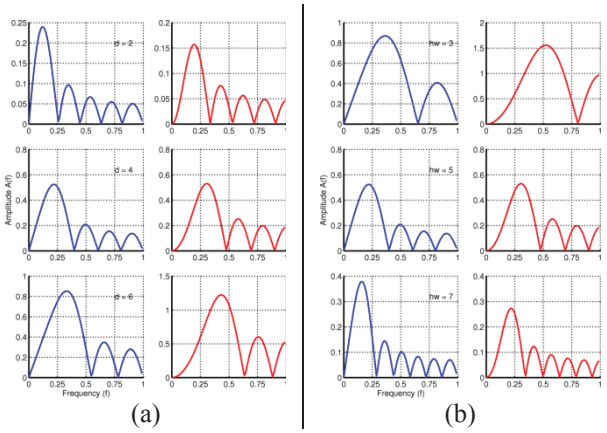


Figure 3: *Variations of spectral characteristics of first (Blue) and second (Red) derivative filters with: (a) degree (d) of generating polynomial and (b) filter size (hw).*

Figure 3a depicts the amplitude spectra of first- (blue) and second (red) order derivative operators of the same length (*hw* = 5) but of different polynomial degree (*d*) (left panel). Note that with increasing degree, the initial slopes of the spectra decrease while the pass-band, the roll-off rate and the cut-off frequencies

increase. The opposite trend is seen in Figure 3b, which displays the spectra for a fixed degree (d = 4) and varying filter length (hw).

3.3. Noise amplification

The performance of filters in the presence of random noise (errors) is an important consideration in geophysical applications. One of the aims of the present study is to improve our understanding of LSDD with regards to their noise propagation and amplification characteristics. It is well known that FIR filters are generally less susceptible to round-off noise than IIR. Nonetheless, it is essential to understand the response of LSDD filters to noise of different structures. Much of the earlier work on noise transmission in FIR filters relates to numerical noise or round-off errors. These errors, however, are very small compared to measurement and other experimental errors typical of observational data. The relation between the noise variance σ_ε^2 and the variance of the output σ_y^2 of a filter is given by (e.g. Rabiner and Gold, 1975; Hamming, 1989):

$$\sigma_y^2 = \sigma_\varepsilon^2 \sum_{i=-hw}^{hw} |h_i|^2 \qquad (4)$$

Thus, the noise amplification factor (NAF) is proportional to the inner product of the filter vector. Figure 5 shows the dependence of NAF on the parameters (hw, d) of the generating polynomial for first and second order derivative operators. It is clear from the figure that these parameters have opposing influence on error amplification. For a fixed filter length (hw), noise is amplified rapidly with increasing degree of the generating polynomial. Inversely, for a fixed degree polynomial, noise is attenuated rapidly with increasing filter half-width (hw) for both order derivatives. This last behavior contrast sharply with the behavior of central finite difference (CFD) filters whose NAF is considerably larger and increase with increasing filter size as indicated in figure 6. It should be pointed out here that LSDD are moment preserving, implying that they are guaranteed to have optimum noise removal while preserving spectral details of the input signal.

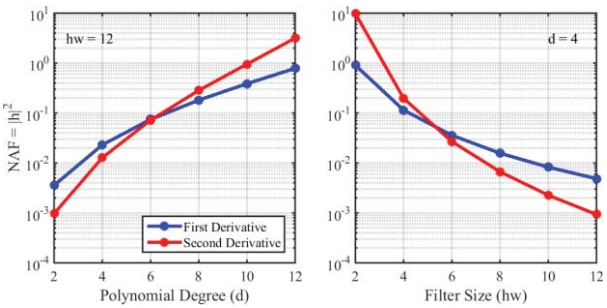


Figure 5: *Dependence of NAF of LSDD filters on polynomial degree (d) and filter size (hw).*

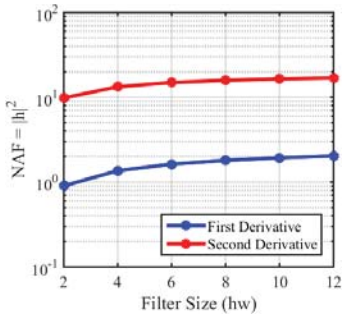


Figure 6: Variation of NAF of CFD filters with size (hw).

3.4. Comparison with IDD

In designing optimal filters, a standard criterion often used is the requirement that the frequency response of the desired filter closely approximates that of the ideal filter. It is, therefore, informative to investigate how closely the LSDD approximate their corresponding IDDs. The frequency response of the ideal low-pass DD filters for first and second order derivatives are respectively given by:

$$\begin{aligned} H_{id}(\omega) &= i\omega \quad |\omega| \le \alpha\pi \\ H_{id}(\omega) &= -\omega^2 \quad |\omega| \le \alpha\pi \end{aligned}, \qquad (5)$$

where the cut-off frequency parameter $0 \le \alpha \le 1$. And the frequency responses of the corresponding LSDD for first and second order derivatives respectively are:

$$\begin{aligned} H_{ls}(\omega) &= -2i \sum_{k=1}^{hw} h(k) \sin(k\omega) \\ H_{ls}(\omega) &= h(0) + 2\sum_{k=1}^{hw} h(k) \cos(k\omega) \end{aligned}. \qquad (6)$$

As a measure of "closeness", I use the mean square error (MSE) defined by:

$$MSE(\alpha) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[H_{id}(\omega) - H_{ls}(\omega) \right] d\omega, \qquad (7)$$

where $H_{id}(\omega)$ and $H_{ls}(\omega)$ are the frequency responses of the ideal and LSDD filters. Substituting from Equations (5) and (6) into (7) and carrying out the integration yields for first and second derivatives respectively:

$$\begin{aligned} MSE_1(\alpha, hw, d) &= q_1 + \frac{4}{\pi} \sum_{k=1}^{hw} \frac{h(k)}{k^2} \left(\sin\phi_k - \phi_k \cos\phi_k \right) + 2\sum_{k=1}^{hw} h^2(k) \\ MSE_2(\alpha, hw, d) &= q_2 + \frac{4}{\pi} \sum_{k=1}^{hw} \frac{h(k)}{k^3} \left[\left(\phi_k^2 - 2 \right) \sin\phi_k + 2\phi_k \cos\phi_k \right] + \sum_{k=1}^{hw} h^2(k) \end{aligned} \qquad \ldots\ldots (8)$$

where,

$$\phi_k = \alpha\pi k \quad q_1 = \frac{1}{3\pi k^3} \phi_k^3; \quad q_2 = \frac{2\phi_k^3}{k^3} \left(\frac{1}{5k^2} \phi_k^2 + \frac{2}{3} h_o \right) + h_o^2 \cdot$$

The behavior of MSE(α) is depicted in figure 7 for first and second derivative filters of different lengths (*hw*) and a fixed polynomial degrees (*d* = *4*). The general trend shown by the figure is of increasing deviation of the LSDD filters of both orders from their corresponding IDDs at all filter lengths. Moreover, the increase in MSE of both derivative orders is smaller the larger the filter size. However, as indicated by the figure, the deviations of LSDD filters from ideal behavior are small ranging between 2 to 15% over the entire range of α.

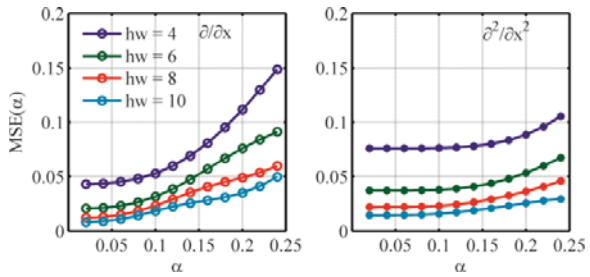


***Figure 7**: Variation of MSE(α) of first and second derivative filters with filter size (hw) for a polynomial degree d=4.*

An important outcome of this analysis is the optimum cut-off frequency parameter (α_o) for a given filter size (*hw*) and polynomial degree (*d*). This is obtained by optimizing the expressions in equations (8) with respect to α. The results are shown in figure 8 for first and second derivative filters. Operators of both derivative order show similar trends of decreasing optimum parameter with increasing filter size; and for any filter size *hw*, optimum α_o is higher the higher the degree (*d*) of the parent polynomial.

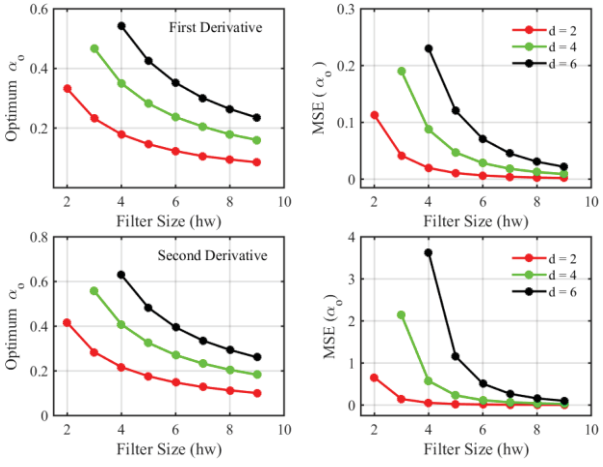


***Figure 8**: Variation of α_o (left panel) and MSE(α_o) (right panel) with filter size (hw) and degree (d) of generating polynomial of first (top row) and second (bottom row) derivative filter.*

Similarly, MSE(α_o) decreases with increasing filter size and decreasing polynomial degree. The rate of decrease of MSE is steeper for second order derivative operators than for first derivative operators.

3.5. Comparison with CFD

Since finite difference is the differentiation method of choice of most researchers, it is instructive, therefore, to examine their spectral characteristics and compare them with LSDD. The amplitude spectra of central finite difference (CFD) filters for first and second order derivatives are compared in figure 9 with their LSDD equivalents. The differences in the passband and attenuation characteristic of the two types of filters are immediately apparent. Whereas LSDD operators are low-pass with filter-length-dependent cutoff frequencies, the CFD filter are allpass attaining their zero-value at the end of the Nyquist interval. Moreover, CFD are amplifying filters with spectral maxima greater than unity for any filter size (*hw*), and increase with increasing filter length. LSDD filters, on the other hand, are attenuating filters as evident from their spectral magnitude maxima of less than unity and which decrease with increasing filter size.

These contrasting spectral characteristics of the two filter types determine their filtering performances. Because of their lowpass and attenuating characteristics, the LSDD filters are noise suppressant performing both smoothing and differentiation simultaneously. CFD filters, in contrast, are high-noise allowing much of the high frequency noise to pass amplified. These same contrasting characteristics explain the contrast in the noise amplification factors of the two types of filters (see figures 5 and 7). Finite difference schemes, therefore, perform best on exact functions and noise-free digital data.

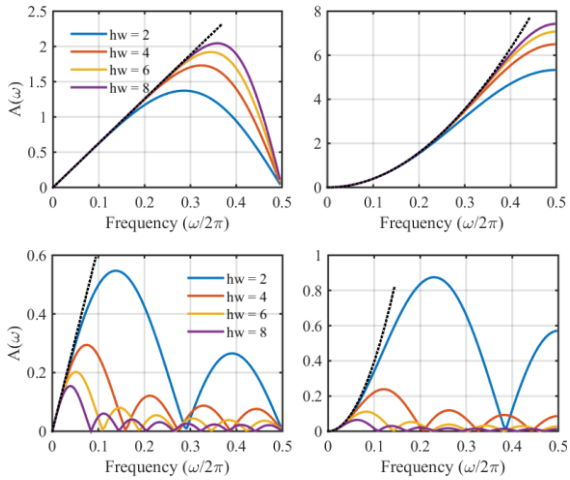


***Figure 9**: Amplitude spectra of CFD (top) and LSSD (bottom) filters of different lengths for first (right) and second (right) order derivatives. The dotted curve is the IDD.*

4. Performance Test

I tested the performance of the LSDD filters using a sine wave contaminated with a Gaussian white noise of zeros mean and 0.1 standard deviation. The noisy data were then differentiated using first derivative LSDD filter of length 21 and degree 2. The same data set was also differentiated using an equivalent CFD filter. As a measure of performance quality, I use the variance of the filter output. The results of these numerical experiments are shown in figure 10.

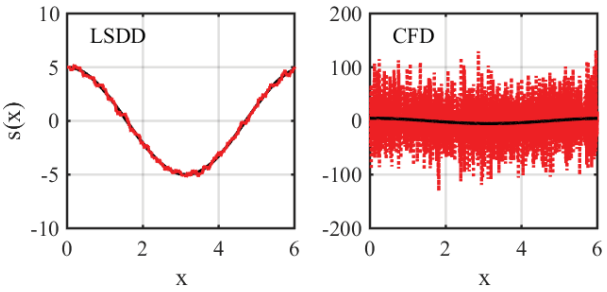


***Figure 10**: Output of first derivative LSDD and CFD filters.*

As is clear from the figure, the LSDD results is of acceptable quality with output variance of 0.02, i.e. equivalent to a noise amplification factor of about 0.03. In contrast, the CFD filter has amplified the high frequency noise several orders of magnitude obscuring the output and rendering it useless.

5. Filter Selection Criterion

From the previous discussion, it is apparent that LSDD differentiation filters can yield excellent results provided the length (*hw*) and degree of the parent polynomial (*d*) are correctly chosen. To facilitate this task for interested user I have constructed contour plots (Figure 11) of the variation of the cut-off frequency versus (*hw*, *d*). In these maps, 'cut-off' frequency is defined as the frequency at half maximum on the amplitude spectra of the filter.

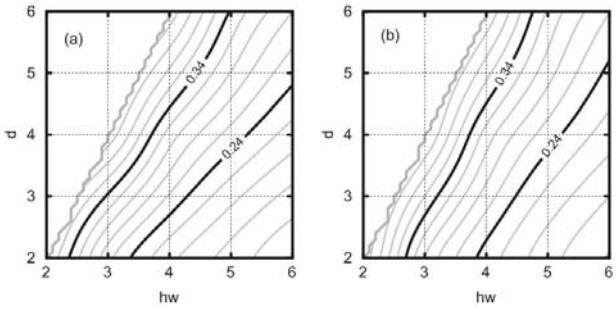


Figure 10: Contour plots of cut-off frequency versus filter half-length (*hw*) and degree (*d*) of generating polynomial.

6. Conclusion

Detailed examination of the spectral properties of LSDD filters has shown that this class of differentiators is low-pass, linear phase, highly stable and self-damping. The amplitude spectra of the filters are maximally flat closely approximating the ideal low-pass digital differentiators (IDD). Comparison with ordinary finite difference filters shows that LSDD filters perform considerably better on noisy data with high degree of noise reduction and smoothing of the output. Therefore, LSDD are more suitable for use with noisy data than FD schemes.

7. Refernces

[1] H. M. Evjen, "The place of the vertical gradient in gravitational interpretations", Geophysics, 1 (1936), p.127.

[2] S. Hammer and R. Anzoleaga, " Exploring for stratigraphic traps with gravity gradients", Geophysics, 40 (1975), p.256.

[3] E.E. Klingele, I. Marson, and H. G. Kahle, "Automatic interpretation of gravity gradiometric data in two dimensions -Vertical gradient", Geophys. Prosp., 39 (1991), p. 407.

[4] W.M. Moon, A. Ushah, V. Singh, and B. Bruce, "Application of 2-D Hilbert transform in geophysical imaging with potential field data", IEEE Trans. on Geoscience and Remote Sensing, 26 (1988), p. 502.

[5] R. T.Shuey, R. T., "Applications of Hilbert transforms to magnetic profiles", Geophysics, 37(1972), p. 1043.

[6] I. I. Mueller, "The gradients of gravity and their applications in geology", Ph.D. dissertation (1960), Ohio

[7] Z. J. Fajklewicz, "Gravity vertical gradient measurements for the detection of small geologic and anthropomorphic forms", Geophysics, 41 (1976), p.1016.

[8] C. A. Heiland, "A rapid method for measuring the profile components of horizontal and vertical gravity gradients" , Geophysics, 8 (1943), p. 119.

[9] D. K. Butler, "Microgravimetry and the theory, measurement, and application of gravity gradients", Ph.D. dissertation (1983), Texas A&M University.

[10] Q. Shuang, "An analytic signal approach to the interpretation of total field magnetic anomalies", Geophysical Prospecting, 44 (1996), p. 911.

[11] R. J. Blakely, " Potential theory in gravity and magnetic applications", Cambridge University Press (1995.(

[12] R. J. Blakely and R. W. Simpson, "Approximating edges of source bodies from magnetic or gravity anomalies", Geophysics, 51 (1986), p. 1494.

[13] S. K. Hsu, J. C. Sibuet and C. T. Shyu, "High-resolution detection of geological boundaries from potential-field anomalies: An enhanced analytic signal technique", Geophysics, 61 (1996), 373–386.

[14] I. N. MacLeod, K. Jones, and T. F.Dai, "3-D analytic signal in the interpretation of total magnetic field data at low magnetic latitudes", Exploration Geophysics, 24 (1993), p. 679.

[15] A. Salem, D. Ravat, T. J. Gamey, and K. Ushijima, "Analytic signal approach and its applicability in environmental magnetic applications", Journal of Applied Geophysics, 49 (2002), 231–244.

Using π digits to Generate Random Numbers: A Visual and Statistical Analysis

Ilya Rogers, Greg Harrell, and Jin Wang
Department of Mathematics and Computer Science
Valdosta State University, Valdosta GA 3198, USA

Abstract - *Monte Carlo simulation is an important method with widely applications in real-world problem modeling, solving, and analysis. Random numbers are key part of this method. A good random number generator should have the following qualities: randomness, speed, simplicity, and large period. In this research, we study using pi database to generate random numbers. Our study shows that this method is efficient and simple with large period. The pi database is a free resource on the internet with 12.1 trillion digits. The special structure of the pi random number generator made it simple and fast with almost no cost. Is pi a good random number generator? The most important thing is the randomness. Based on our experiment outputs, the 2D and 3D plots indicate that the randomness of pi is pretty good comparing with the existing popular LCG pseudo random number generates in computer simulation community. Finally we use this pi random number generator to simulate the true pi value. Our result shows that the pi approximation is very accurate.*

Keywords: Monte Carlo Simulation; Random Number Generator.

1 Introduction

1.1 History of π:

π is an irrational number that is extremely helpful in calculating area of a circle. With the ability of calculating area of a circle gives us unparalleled ability to apply the idea in numerous applications. Such applications include: engineering, measuring sound waves, simulation, GPS, and pretty much anything that has a "curved" surface. π, more specifically, is a ratio of circles' circumference and diameter

$$\pi = C/d$$

which allows us to closely estimate circumference and area of a given circle with radius r:

$$C = 2 * \pi * r$$

$$A = \pi r^2$$

History of π is extremely rich and diverse and yet still π hold many mysteries that have not yet have been discovered. It is not known who has originally come up with concept of π but the earliest record of a civilization trying to find the ratio is about 4000 years old belonging to Babylonians and Egyptians. It is speculated that a rope was being used to measure the circumference and the diameter after which they have estimated that π is slightly larger than 3, more specifically approximately 3.125. [1]. Next appearance of π is in a Egyptian Papyrus dated back 1650BCE. The papyrus outlines a list of problems for students to solve one of which required a student to figure out an area of a circle inside of a square [2]. This problem calculated π to be about 3.1605 or 3 and 13/81. The approximate value of π as we know it today was calculated by Archimedes by taking 2 hexagons and doubling the sides 16 times. The final result came to about π = 3.1415926535[1]. Fast forwarding to more recent events, the creation of computers and the ability to calculate more decimal digits of π the current record holder as of December 2013 is 12 trillion digits held by Alexander J. Yee & Shigeru Kondo[3].

1.2 How to calculate π?

π, being complex number it is, can be fairly easy but costly to calculate. The problem lies in how precise of decimal places you want it to be. There are multiple ways of calculating π. It is possible to compute π using Numerical methods such as 22/7 or drawing hexagons and multiplying their sides; more sides equal more precise value of π. Another way to compute π is to use computers and algorithms to automate the process. Last but not least option is by using a random number generator to simulate π. Geometrical way of calculating π is by inscribing and circumscribing n number of polygons and the calculating their perimeter and areas. Archimedes used this technique to estimate π being roughly 3.1416. More modern way of calculating π is using Gregory's formula:

$$\int_0^x \frac{dt}{1+t^2} = \arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots$$

Evaluating for x = 1 we get:

$$\frac{\pi}{4} = 1 + \frac{1}{3} - \frac{1}{5} + \frac{1}{7} - \cdots$$

This method was used by Abraham Sharp to calculate π to 72nd decimal place [2]. With computer age the possibilities of computing decimal places for π has significantly increased. Instead of spending years of computing π to several thousandth place early computer could do it in matter of hours. One of the computer algorithms using ENIAC in 1950 to calculate π to the 2037 digits used the following algorithm [5]:

$$\frac{\pi}{4} = \sum_{n=0}^{\infty} (-1)^n * \left[\frac{100(0.2)^{2n+3} - \left(\frac{1}{239}\right)^{2n+1}}{2n+1} \right]$$

It has taken the machine 70 hours to finish the computation. That record was beaten in 1955 by using the same formula but with a better machine. As the computers evolved at exponential rate (Moore's Law) the possibility of calculating π to higher number of decimal places has grown along with it. Lastly, it is possible to calculate π using simulation; the method is called "Monte Carlo π". The Monte Carlo method calculates π/4. We begin by drawing a 1 by 1 square on a coordinate plane, and then we inscribe a circle inside of the square. Next, use LCG (Linear Congruential Generator) RNG to randomly generate X and Y value plotting them in the first quadrant. Using a computer algorithm we check if $X^2 + Y^2 \leq 1$ meaning that the point is inside/on the line the quarter-circle and we increment our "success" or k counter which is represented by red points in picture above. After the simulation, depending on number of samples we calculate our p̂ where n is number of trials and k is number of successes:

$$\hat{p} = k/n$$

With enough rounds we will start to see that p̂ will begin to look like true π value decimal place by decimal place. Due to the nature of simulation and equation of standard error:

$$ste = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

We would have to run the simulation 100 times more in order to gain one decimal place accuracy every time. After a while it is obvious that calculating decimal places of π using simulation can get extremely costly in terms of time and resources.

1.3 Trillion digits π value

The record as of December 2013 in calculating decimal points of π is 12.1 trillion digits achieved by Alexander J. Yee & Shigeru Kondo. Yee and Kondo have built a computer [4]. Note the amount of RAM memory and HDD space. Calculating π to approx. 12 trillionth digit is no easy task and requires tremendous resources. The resources required go up as the number of digits increases, especially HDD space requirements. Yee and Kondo used Chudnovsky algorithm displayed below:

$$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{(-1)^k (6k)! \, (54514013k + 13591409)}{(3k)! \, (k!)^3 * 640320^{\frac{3k+3}{2}}}$$

After the algorithm completes one simply takes the inverse of the result giving them the value of π to the n number of decimal places. Implementing this algorithm in a computer along with some I/O operations to write data it has taken Yee and Kondo 94 days to calculate 12.1 trillion digits of π… then they ran out of HDD space. It is clear that any computer can compute π to extremely high number of decimal places, however, hardware plays major role in terms of time and storage.

2 π RNG Testing and Analysis

2.1 Generating Random number using π values

Talking about generating π to an astounding number of decimal places is great, however, to keep on the to πc we must shift our attention to actual random number generators (RNG). There are numerous random number generators on the market today. Some are quite good (LCG) and some are notoriously bad UNIVAC which as only 5 numbers in the cycle. The optimal RNG produces truly random number and does not have a cycle. Due to the realities of the real world and limitations of computer hardware producing truly random numbers is extremely difficult. Instead algorithm based RNGs were developed. The problem with algorithm RNG is that we can predict next random number if we have the seed and the iteration number, and that those usually have a cycle. The larger the cycle the better RNG is considered due to the fact that there are more numbers to pick from. Speaking in terms of π, there is no said cycle proven to date in π. Theoretically we can calculate π infinitely but due to hardware limitations we only have 12.1 trillion digits. Even still no strong patterns were found in that impressive number. If we continue calculating π past the 12 trillion it is going to be nearly impossible to predict which values will come next. A good RNG is measured on following criteria:

- Uniform distribution
- Memory requirement
- Speed
- Reconfigurable
- Portable
- And implementation easiness

I am going to run some tests and grade the π generator on above mentioned criteria along with some other things. The objective of this paper is to determine whether π decimal digits can be used as random numbers. To achieve my objective I am going to compare my π RNG against a Linear Congruential Generator (LCG) which uses a seed and an algorithm to generate a random number. My hypothesis is that π can be used as a cycle free RNG with similar success as the LCG.

2.1.1 Test 1: 3D uniform distribution of π vs. LCG RNG visual comparison

All of the Java code to create graph plots can be found on my GitHub repository. [8]. The Test method for π RNG is as follows:

- Using y-cruncher ver. 0.5.5. [3] I am going to generate 1 billion decimal places of π and save them into a text document. Y-cruncher saves database file as text file by default.

- Use code to read in the π database file
- Initialize beginning pointer at the beginning of the π value (Number 3)
- Use slice size of 5 digits and n size of 5000, 10^4, and 10^5 resetting RNG back to the beginning of π (Init. pointer = 0) beginning every new number of n.
- Output generated RNs into another output file.
- Using data in the output file I am going to plot the resulting number of samples in 3D scatterplot to try to identify patterns
- As I am generating the random number data for a specific slice I am keeping track of the pointer resetting it only when I am generating numbers for a different slice size.
- Plot the x, y, and z values using Java code and displaying them in a 3D graphs.

Test method for LCG RNG

- Perform same exact procedures as for π RNG.

The algorithm for π is outlined below:

- Initialize all input and output streams and any relevant variables.
- Initialize number array to set number and string array to slice number
- Skip to the initialization decimal place of π database file.
- Loop for number of sets generating first row of random numbers and storing them in a number array of size set
    - Loop for slice number
        - Read 1 Byte of the file converting it from ASCII to a readable character
        - If character = '.'
            - Throw away the character and read next Byte
        - Add the character to string array building a number/character string
    - Convert each containing string to a number then divide by 10^{slice}
    - Place resulting number in number array[i]
    - Write the number into the file on the same row
- Skip down to next row of the output file
- Begin main while loop running until number of trials-1 or end of input file
    - Loop for number of sets-1 times
        - Copy contents of number array[i+1] to number array[i]. That leaves last spot open for newly generated RN
    - Loop for slice number
        - Read 1 Byte of the file converting it from ASCII to a readable character
        - If character = '.'
            - Throw away the character and read next Byte
        - Add the character to string array building a number/character string
    - Add the resulted string to the last spot of the number array
    - Loop for the number array length
        - Convert each containing string to a number then divide by 10^{slice}
        - Write resulting number to output file
    - Skip down to next row of the output file
    - Decrement/increment while control variable
- Flush and close all output/input streams

The algorithm for LCG U16807 RNG:

- Initialize all input and output streams and all relevant variables
- Initialize m, c, and a to $2^{31}-1$, 0, and 16807 respectively; w0 is seed variable for the RNG.
- Initialize number array to set size
- Loop for number of sets generating first row of random numbers (Explained in while loop) and storing them in a number array of size set
    - Calculate a temp variable using equation: $\text{temp} = (\text{a} * \text{w0} + \text{c}) \bmod \text{m}$
    - Copy temp variable to w0 variable
    - Write the number into the file on the same row
- Skip down to next row of the output file
- Begin main while loop running until number of trials-1
    - Loop for number of sets-1 times
        - Copy contents of number array[i+1] to number array[i]. That leaves last spot open for newly generated RN
    - Calculate a temp variable using equation: $\text{temp} = (\text{a} * \text{w0} + \text{c}) \bmod \text{m}$
    - Copy temp variable to w0 variable
    - Fill in last spot of the number array with $\text{w0}/\text{m}$ value
    - Loop for the number array length
        - Write contents of the number array to file as 1 row
    - Skip down to next row of the output file
    - decrement/increment while control variable
- Flush and close all output/input streams

We generate 3D graphs where $U_i = x$, $U_{i+1} = y$, and $U_{i+2} = z$. First I am going to display the distribution of slice =

5 and sample size = 5000, 10^4, and 10^5 for π RNG in figures 3-5 respectively. Graphs generated by the U16870 Generator are displayed in figures 6-8. Do note each set of graphs are displayed from different angles of view.

Figure 1. U(0,1) Slice = 5 and Sample size = 5000

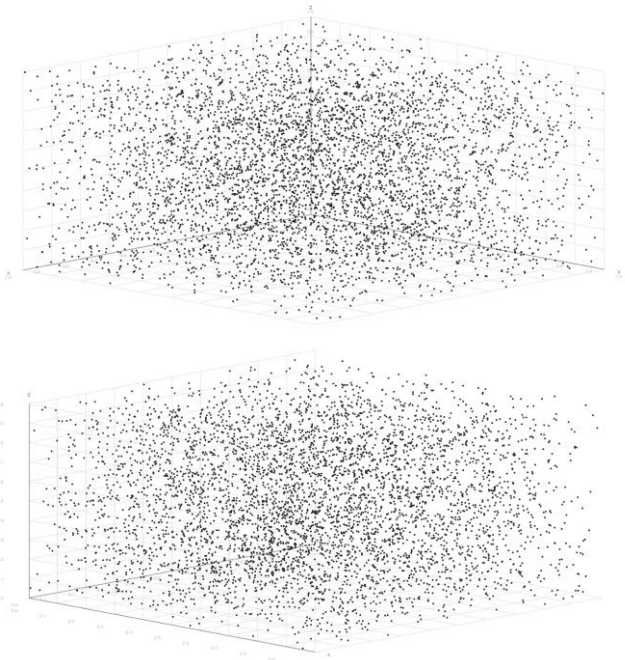


Figure 2. U(0,1) Slice = 5, Sample size = 10,000

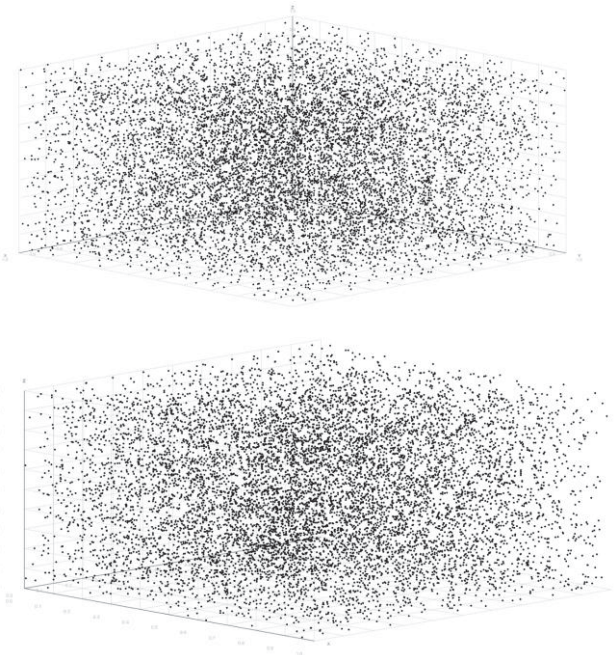


Figure 3. U(0,1) Slice = 5, Sample Size = 100,000

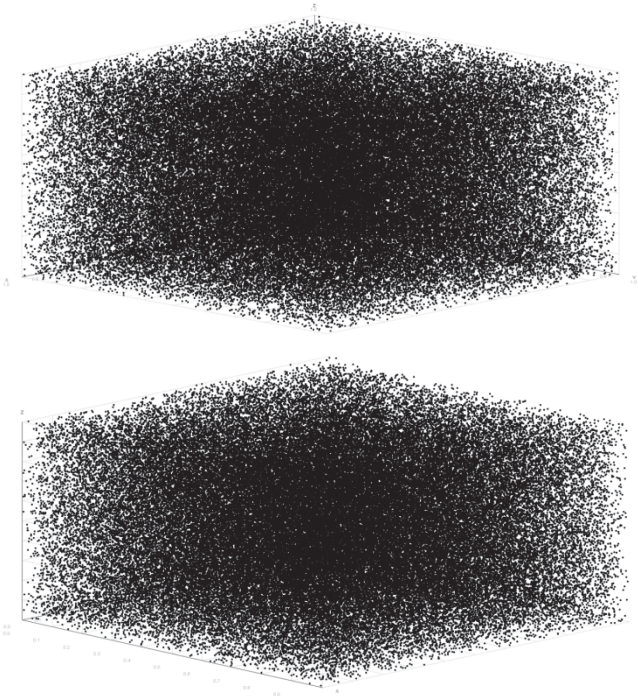


Figure 4. U(0,1) w0 = 1, Sample size = 5000

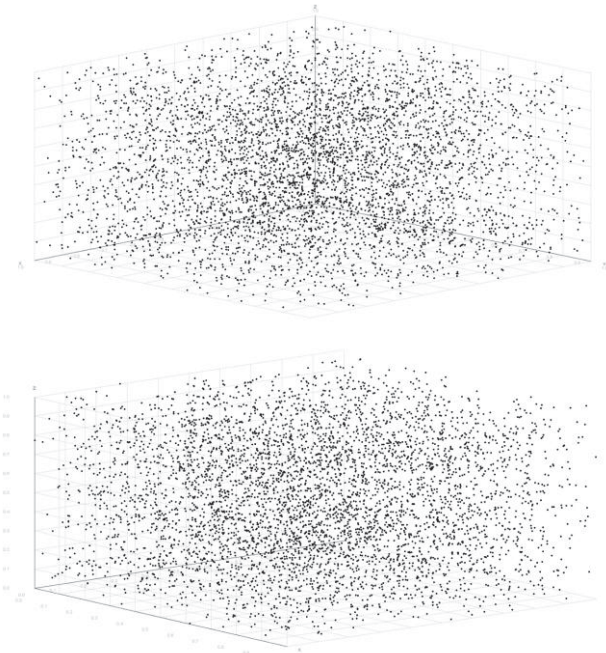


Figure 5. U(0,1) w0 = 1, size = 10,000

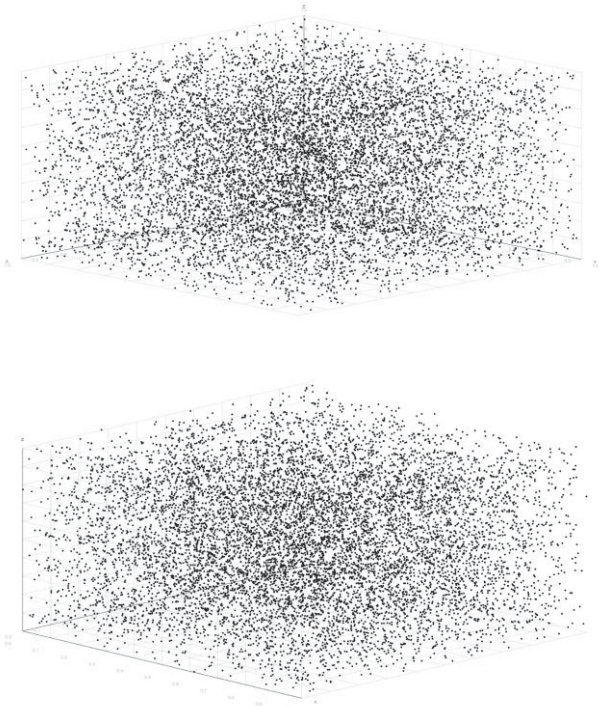
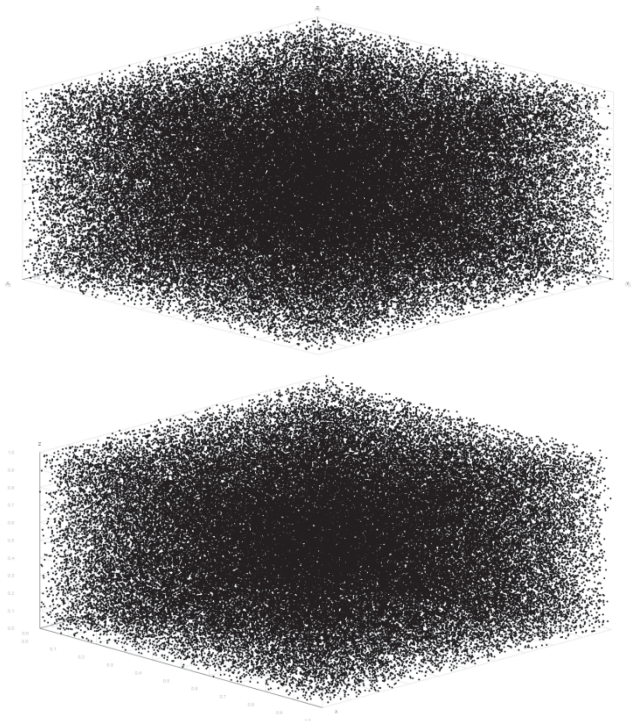


Figure 6. U(0,1) w0 = 1, Sample size = 100,000



Test 1 Summary:

Comparing the uniform distribution graphs of the π and U16807 generators there is a minimal difference. The distribution is uniform across all of the tests. If done for higher number of iterations a solid rectangle would appear indicating that the distribution is all the way across U(0,1). There are no observable patterns indicating any cycles or "preferred" numbers. Final assessment is that π RNG is identical to U16807 in terms of uniform distribution and does generate good random numbers.

2.1.2 Generating π Monte Carlo method using U16807 and π RNG and statistical analysis

Second test I am going to perform is Monte Carlo π. I am going to use both RNGs to emulate a real world problem. I am going to use π and LCG RNGs to generate π. I am going to use a circle inscribed inside 1 by 1 square method to approximate π. Each RNG will run for $10^4, 10^6$, and 10^8 iterations then will be compared in terms of p̂ vs. true π value using swing digit method to approximate cut-off decimal place. Swing digit method works by removing unnecessary decimal places for generated p̂ value. Suppose:

$$\text{Generated } \hat{p} = 3.112343452$$
$$Ste = 0.02346434$$

Look at ste value from left to right and count all the zeros without break. If a non-zero digit is encountered stop and that is how many fist digits of p̂ we will keep. Idea is that the first non-zero digit is where the actual uncertainty error is so the p̂ will fluctuate that that decimal place which is not what we want.

$$\text{After swing digit: } \hat{p} = 3.11$$

Method (Same for both generators):

- Each generator will be run for 10^4, 10^6, and 10^8 iterations.
    - Due to the fact that I have to run 100 times more got get extra decimal point
- LCG will start at $w0 = 1$ and π RNG will start at pointer location 0 for each iteration test.
- π RNG will have a slice of 5 for each RN
- All data will be imported from corresponding RNG output files
    - Must generate 2 times number of samples due to reading *X* and *Y* value per iteration.
        - 1bn π number is just enough to do large test
    - Single set
- Each RN sample set will be run through π simulator
- Success counter will be incremented only if condition is $x^2 + y^2 \leq 1$
    - $A = \pi * r^2$
- $\hat{p}$ standard error will be calculated at the end of the run
    - $\hat{p} = k/n$
    - $ste = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

- Results will be recorded for comparison where $\hat{p}$ will be cut off using swing digit method.

Algorithm:

The algorithm is largely the same for both π and U16807. Due to both programs grabbing values from pre-generated list of random numbers

- Pass in the file containing a list of pre-generated random numbers
  - For π it's the list generated with π RNG and with U16807 is the list generated with U16807 RNG
- Initialize file reader
- Take input on how many iterations to perform
- Initialize success variables
- Start while loop running for entered number of iterations
  - X = first random number read from the list
  - Y = second random number read from the list
  - If $X^2 + Y^2 \leq 1$
    - Increment success variable
  - Increment/decrement loop control variable
- Calculate and store p̂
  - Success/iterations
- Calculate and store true p
  - Use system provided variable for π. Java = Math.PI Due to the fact that we are only generating to 2-4 decimal places there is no need to have a large decimal point of π.
- Calculate and store standard $\hat{p}$
  - $\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$
- Return a string/print all above mentioned variables.

Results:

Table 1. Results for sample size 10^8, π slice = 5, and $w0$ = 1.

RNG	Raw $\hat{p}$	True p	ste	Swing digit $\hat{p}$	Time (ms)
π	3.14151212	3.141592653589793	1.6422393491489316E-4	3.1415	117152.0
U16807	3.14143824	3.141592653589793	1.642290700292035E-4	3.1414	205539.0

Table 2. Results for sample size 10^6, π slice = 5, and $w0$ = 1.

RNG	Raw $\hat{p}$	True p	ste	Swing digit $\hat{p}$	Time (ms)
π	3.142004	3.141592653589793	0.0016418973366151735	3.142	3846.0
U16807	3.142096	3.141592653589793	0.0016418333431819441	3.142	11836.0

Table 3. Results for sample size 10^4, π slice = 5, and $w0$ = 1.

RNG	Raw $\hat{p}$	True p	ste	Swing digit $\hat{p}$	Time (ms)
π	3.1528	3.141592653589793	0.01634335387856483	3.15	1787.0
U16807	3.1496	3.141592653589793	0.016365878650411655	3.14	1622.0

Test 2 Summary:

For all tests U16807 and π RNG performed calculations in the similar manner have produced rather close results. Overall, both RNGs calculated π to the same STE thus calculating same "accurate". Do take note that the smaller the sample size that I used the less accurate swing digit p̂ became indicating that in order for us to get 1 extra decimal place accuracy we have to run the simulation 100 times more than previous trial. Both generators have performed at the same success rate and efficiency. One thing to mention is time. Due to U16807 producing larger decimal place numbers it has taken slightly longer to process opposed to π RNG where it was calculated to 5 decimal places. Overall result is that π and U16807 RNG performed the same.

2.2 Technical Issues

π generator was a unique generator to implement. It has required me to extend my Java knowledge to new levels. First major issue was generating the actual π number. Due to calculation intensity it has taken me substantial amount of resources to calculate π to 1 billion decimal places. Not to say I have a bad computer but it was extremely surprising to see that that calculation has taken up almost all of my RAM memory 6/8GB which resulted in my computer nearly halting for the duration of the calculation. After the calculation finished in roughly 5 min I was surprised to find a file size of 1GB+ in my directory. For commercial implementation π will have to be calculated to much greater decimal places in comparison resulting in numerous terabytes or even petabytes of space being taken. Transferring 1 GB file between

directories on 7200RPM HDD was tedious in terms that it would take the computer some minutes to copy the file somewhere else. Next issue I have encountered was actually reading such big file. Instead of Java Scanner class I was forced to use file input streams due to Java running out of heap memory. Using input streams has its advantages however, now my code can read files of theoretically unlimited size. Last major issue was again, the file size except in this scenario it was my output files. I, again, had to use buffered output streams to properly write output files. In some instances generating my RN sets from 1GB file yielded 2-4GB files which could pose much greater issue in commercial sense. Writing to those file have also given me some issues specifically by buffered output streams. For buffered output or input one must flush the stream before exiting the stream, otherwise, you will end up with incomplete set of RNs in your output file. Overall computer hardware plays an immense role in success of π RNG. The relation of computer hardware, specifically RAM, CPU, and HDD to π RNG is that the better the hardware the better π RNG you will have.

2.3 Is π a good random number generator?

In summary, I have performed three tests, each has put π RNG against one of the more popular U16807 RNG. π RNG has proven to be competitive in visual test, iteration (π value calculation) test, and probability calculation test. In terms of uniform distribution both generators perform the same. Memory requirement U16807 has the advantage due to when we generate RNs using Chudnovsky formula we use quite a bit of memory and other resources. To generate 1 billion digits of π it has taken my computer over 8GB of memory; the higher value of decimal numbers I wanted that memory requirement gone up. In terms of speed π generator loses to U16807 in the same manner as mentioned in memory requirement. The more RNs I want the heavier calculations have become. π wins the reconfiguration criteria over U16807. π I can specify start point and slice size giving me different random numbers each time where U 16807 I can only specify w0 which will only place me as some point of the cycle giving me the same RNs if I run it long enough. U16807 is more portable than π generator. π generator requires huge database size, 12.1 trillion ≥ 20TB, in order to have a decent pick of random number whereas U16807 is limited by computers word size and is not backed by database. π generator wins in ease of implementation. If I have a large database all I need to do is read it where in U16807 I have to implement a function for it run properly.

3 Summary

π RNG is an unconventional random number generator however it offers unprecedented speed and accuracy of commercially created RNGs; assuming database is not an issue. π generator certainly has great potential however, there are few issues that can keep it from being as "convenient" as U16807. As already mentioned, π RNG requires a very large database to read the random numbers from in order for it to work well and indefinitely. Due to π having no proven strong patterns in its number sequence to date π does give us luxury of having a good cycle free generator. Overall, provided that the π digit database is large enough or resources for calculating π as you generate RNs is not a factor π can be considered an excellent random number generator.

4 References

[1] http://www.math.rutgers.edu/~cherlin/History/Papers2000/wilson.html

[2] http://www.math.tamu.edu/~dallen/masters/alg_numtheory/ π.pdf

[3] http://www.numberworld.org/misc_runs/ π-10t/details.html

[4] http://mathforum.org/library/drmath/view/57045.html

[5] http://www.jstor.org/stable/pdfplus/1403789.pdf?&acceptTC=true&jpdConfirm=true

[6] http://mathfaculty.fullerton.edu/mathews/n2003/montecarlo πmod.html

[7] https://code.google.com/p/jmathplot/ -- JMathPlot A Π For 3D Java plotting

[8] https://github.com/ikrogers/Operations-Research- π-RNG-Java-Source

SESSION

POSTER PAPERS

Chair(s)

TBA

Numerical Simulation of the Internal Flow of a Three-dimensional Thrust-Vectoring Nozzle

Tsung Leo Jiang, En-Yu Yeh, and Hsiang-Yu Huang
Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan, Taiwan, ROC

Abstract-*In the present study, a simulation model of a three-dimensional thrust-vectoring nozzle has been developed, and the simulation analyses for the flow of a three-dimensional thrust-vectoring nozzle have been performed under various deflecting angles. The results obtained from the present numerical simulation show that for a small radius of curvature of the deflector section, the predicted thrust decreases significantly with increasing deflecting angles, since a recirculating flow exhibits locally when the deflecting angle is large, reducing the average exhaust velocity. By enlarging the radius of curvature of the deflector section, the recirculating flow occurring at large deflecting angles vanishes gradually, and the average exhaust velocity would not drop significantly. The thrust at a large deflecting angle is thus maintained at the same level as that without deflection. It is also noted that the thrust is lower with a larger angular speed of the deflecting nozzle at the same deflection angle.*

Keywords: Thrust-vectoring nozzle, Three-dimensional Flow, Numerical Simulation

1 Introduction

In the modern development of propulsion technology, the expansion nozzle of the aircraft engine as been designed to have the capability of changing the direction of the exhaust flow, making the change of thrust direction possible [1]. The expansion nozzles with the capability of changing the thrust direction are known as thrust-vectoring nozzles. With a thrust-vectoring nozzle installed, the aircraft would have better manipulations under low-speed flight [2]. The reversing thrust of a thrust-vectoring nozzle can even be used as a brake. As a result, the new-generation fighters, such asF-22 and F-35 of USA as well as Su-30 and Su-37 of Russia, all employ the technology of the thrust-vectoring nozzle to improve their combat advantage.

In the present study, the software FLUENT adopting the SST-k-ω turbulence model and the dynamic moving-grid system [3] is employed for the simulation of a three-dimensional thrust-vectoring nozzle, as shown in Fig. 1. The boundary conditions are shown in Table 1.

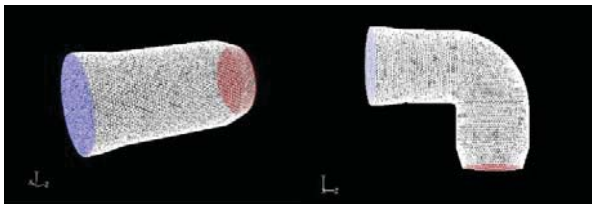


Fig.1 The grid system of the three-dimensional thrust-vectoring nozzle

Table 1 Boundary conditions

	Temperature(K)	Pressure(psi)
Inlet conditions	1070.6	23.74
Wall condition	adiabatic	

2 Results and Discussion

The thrust versus the deflection angle curve at various radii of curvature is shown in Fig. 2. With a smaller radius of curvature of the deflector section (R=6.92 mm), the predicted thrust decreases significantly with the increasing deflecting angle, since a recirculating flow exhibits locally (Fig. 3a) when the deflecting angle is large, reducing the average exhaust velocity. By enlarging the radius of curvature of the deflector section, the recirculating flow vanishes (Fig. 3b), and the average exhaust velocity would not drop significantly. The thrust at a large deflecting angle is thus maintained at the same level as that without deflection.

The thrust versus the deflection angle curve at various angular speeds of the deflecting nozzle is shown in Fig. 4. The thrust is lower with a larger angular speed at the same deflection angle. This is due to the fact that the flow is unable to accommodate the fast deflection of the nozzle, tending to separate from the wall. As evidenced by the pressure contours and velocity vectors shown in Figs. 5-7, flow separation exhibits when the angular speed ω is 1.5708 rad/sec (Fig. 7), in comparison to the smoother flow at lower angular speeds of 0.7854 rad/sec (Fig. 6) and 0.3927rad/sec (Fig. 5). It is also noted in Fig. 4 that the thrust produced by a turning thrust-vectoring nozzle is generally lower than that at steady state.

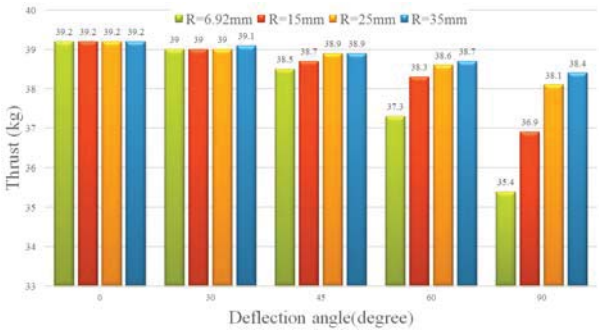
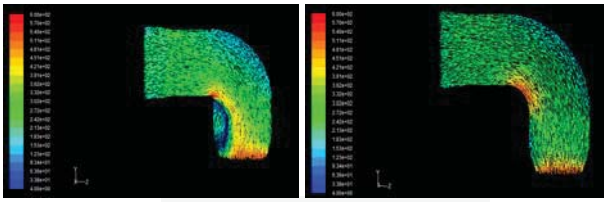


Fig.2 The thrust versus the deflection angle at various radii of curvature



(a) (b)

Fig.3 The velocity vectors for a radius of curvature of (a) R=6.92 mm and (b) R=35 mm

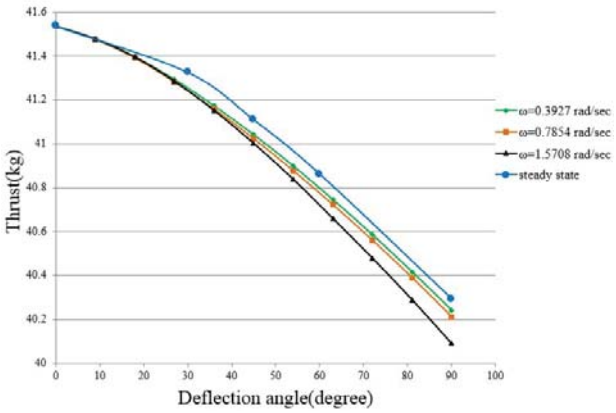


Fig.4 The thrust versus the deflection angle at various angular speeds

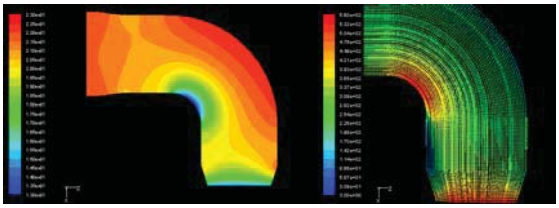


Fig.5 The pressure contour and velocity vectors with ω = 0.3927rad/sec.

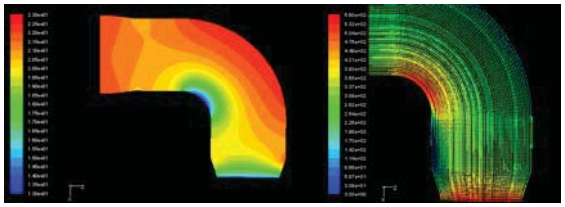


Fig.6 The pressure contour and velocity vectors with ω = 0.7854rad/sec.

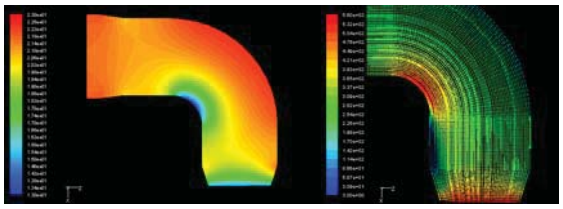


Fig.7 The pressure contour and velocity vectors with ω = 1.5708 rad/sec.

3 Conclusions

The dynamic moving-grid simulation mechanism has been developed successfully, and has been employed to carry out the analyses of the internal flow of a three-dimensional thrust-vectoring nozzle. The numerical results show that for a small radius of curvature of the deflector section, the predicted thrust decreases significantly with increasing deflecting angles, since a recirculating flow exhibits locally when the deflecting angle is large, reducing the average exhaust velocity. By enlarging the radius of curvature of the deflector section, the recirculating flow occurring at large deflecting angles vanishes gradually, and the average exhaust velocity would not drop significantly. The thrust is lower with a larger angular speed of the deflecting nozzle at the same deflection angle.

4 References

[1] C. W. Alcorn, M. A. Croom, M. S. Francis, and H. Ross, "The X-31 Aircraft: Advances in Aircraft Agility and Performance," Prog. Aerosp. Sci., Vol. 32, No. 4, pp. 377-413, 1996.

[2] A. J. Steer, "Low Speed Control of a Second Generation Supersonic Transport Aircraft Using Integrated Thrust Vectoring," Aeronaut. J., Vol. 104, No. 1035, pp. 237-245, May 2000.

[3] FLUENT 6.3, User Guide, FLUENT Incorporated, 2006.

Effect of reduced-grid on the global-domain Fourier Finite-Element model

Hyeong-Bin Cheong* **and Han-Byeol Jeong**
Department of Environmental Atmospheric Sciences, Pukyong National University
Yongso-ro 45, Namgu, Busan, Korea, 608-737
* Corresponding author (hbcheong@pknu.ac.kr)

Abstract *In this paper the reduced grid for the Fourier Finite Element model on the spherical surface is investigated, where the Fourier method and the Finite Element method (FFEM) are adopted in the zonal and meridional direction, respectively. The reduced grid is produced in such a way that the number of zonal grid for an equiangular lat-lon grid is reduced by discarding insignificant values of zonal-Fourier transformed variables, taking the attenuating behavior towards poles of P(M,m), the Legendre function of degree M and order m into consideration. Reducing the number of gridpoints in the FFEM is equivalent to placing artificial boundaries to the zonal-Fourier coefficients of field variables. For the reduced grid, the total number of grids on the globe is reduced by about 30%. The reduced grid is tested in terms of the differentiation and advection equations. Test results demonstrated that the reduced grid model performs with almost the same accuracy as the full lat-lon grid model without inviting discontinuity at the meridional boundaries.*

Keywords: Reduced grid, spherical surface, Fourier Finite Element method, advection equation, high-order elliptic equation, quasi-uniform grid.

1 Introduction

The lat-lon grid, which adopts the nodes aligned along the spherical coordinate axes, has many advantages over non lat-lon grid in spite of clustering of nodes near the poles. For instance, it is rather easy to apply the spectral Galerkin method, which is robust and provides high accuracy, to the lat-lon grid than any other grid system. The disadvantage of node clustering near poles for the lat-lon grid, can be overcome by reducing the grid-points near poles, i.e., adopting reduced grid. For spectral methods, this is equivalent to reducing the maximum zonal wavenumber resolved in the model. This strategy is practiced in the global weather prediction models, such as IFS of the European Center for Medium Range Weather Forecast (ECMWF). The reduced grid system can also be applied to the Fourier finite-element method with ease because it adopts localized basis functions in the meridional direction to expand the Fourier coefficient. In this study, the reduced grid is incorporated into the Fourier finite-element model, and the accuracy is tested in terms of rather simple differential equations such as advection equation and elliptic equation.

2 Fourier finite-element method and the reduced grid

The Fourier finite-element method on the spherical surface is well described in Cheong et al. (2015). The first step for discretization of differential equation is to get the Fourier transform of given function. And, then the finite element method is applied to the Fourier coefficients. The reduced grid is produced based on the associated Legendre function, which attenuates towards the poles, more severely for higher zonal wave-number. Detailed procedure to set the reduced grid is as follows.

(i) Set the equiangular lat-lon grid with the number of zonal grid being N.

(ii) Set the maximum zonal wavenumber M for a prescribed spectral truncation.

(iii) For each zonal wavenumber (m) and meridional nodes (j), generate the discrete Legendre functions, $P_n^m(M,m,j)$.

(iv) Let $g(m,j)$ be the zonal Fourier coefficients at the gridpoint j of $g(\lambda,j)$, then set g(m,j)=0 if $P_n^m(M,m,j) \le E$. E is the threshold value for the reduced grid, which is prescribed as 10^{-15}, the round-off for the double precision computation.

The spectral truncation is dependent on the largest order of differential equations included in the model. Quadratic and cubic truncation, which require M=N/3-1 and M=N/4-1, respectively, are considered. In Fig. 1, the reduced grid is illustrated for N=1024 for quadratic spectral truncation.

Discrete Legendre functions are calculated by recursion equation:

$$P_n^m(x) = \varepsilon_{n,m}^{-1}\left[xP_{n-1}^m - \varepsilon_{n-1,m}P_{n-2}^m\right], \quad (1)$$

where $P_n^m(x)$ is the Legendre function, $\varepsilon_{n,m} = \sqrt{(n^2-m^2)/(4n^2-1)}$, and x is sine of latitude. For high-resolution model, the calculation of the all Legendre functions needs considerable computation time and the accuracy of them is deteriorated significantly. Taking these into consideration, an approximate formula of the boundary curve for the retained- and discarded- wave components (reduced grid; upper part in Fig. 1) is introduced. The approximate formula, which was deduced empirically from calculated $g(m,j)$, is written as follows:

$$y = (0.75\sqrt{r} + 0.25\sin\pi r)$$
$$r = (M-m)/(M-m_c),\ m_c \le m \le M \quad . \qquad (2)$$
$$y = (j-J_c)/(N/4-J_c),\ J_c \le j \le N/4$$

The symbol m_c represents the zonal wavenumber at which the Legendre function $P_M^m(x_M)$ first becomes less than the double precision machine roundoff as m is increased from zero. J_c implies the meridional grid point at which the condition $P_M^M(\cos\theta_j) < \sqrt{M}\times 10^{-15}$ is first met as j is increased from zero towards pole. In this step, $P_M^M(x) \approx \sqrt{M}\,(1-x^2)^{M/2}$ was used.

3 Test results

The effect of reduced grid is tested in terms of meridional differentiation, elliptic equations, and the cosine-bell advection equation (Williamson et al. 1992). Three model resolutions are used for the tests: N=180, 360, and 720. Wave truncation is determined based on the two third rule, which is used commonly for the quadratic order of differential equations. Accuracy of the differentiation for the reduced grid was found to be the same as the full lat-lon grid model (Cheong et al. 2015) in terms of the absolute error and its convergence rate; As theoretically expected, the convergence rate turned out to be the fourth order. The error and its convergence rate for the elliptic equation are also found to be the same as the full grid model. In Fig. 2, the cosine bell at initial and the grid-point error by day 12 (after one rotation along great circle tilted towards North pole by $\pi - 0.05$) is shown for the resolution of N=360. The maximum error and the spatial patterns by the day is indistinguishable from those of Cheong et a. (2015), indicating that the reduced grid was set appropriately. Also is confirmed is the error convergence rate, being indicative of slightly larger than the second-order.

4 Conclusions

The effect of reduced grid on the Fourier Finite Element model on the spherical surface was investigated. The error was evaluated for the differentiation, elliptic equation, and the cosine bell advection. The reduced grid was demonstrated to produce results as accurate as the full lat-lon grid without giving any discontinuity at the meridional (artificial) boundaries which were set purely based on numerical reasons. The reduced grid will be tested using global hydrostatic Fourier finite element model in the near future.

5 References

[1] Cheong, H.-B., H.-J. Kong, H.-G. Kang, and J.-D. Lee. Fourier Finite-Element Method with Linear Basis Functions on a Sphere: Application to Elliptic and Transport Equations. Mon. Wea. Rev., 143, 1275-1294, 2015.

[2] Williamson D. L. and coauthors. Standard test suite for shallow water equations on the sphere. J. Comput. Phys., 102, 211-224, 1992.

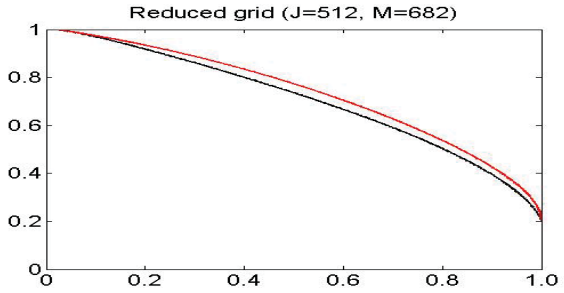


Fig. 1 Plot of boundary of retained- (below curve) and discarded (above curve) zonal Fourier transform $g(m, j)$ for the reduced grid. The horizontal and vertical axes represent the zonal wavenumber (m) and the meridional grid index (j), normalized by M and N/4, respectively. Black (red) line indicates the boundary curve from calculation (empirical formula).

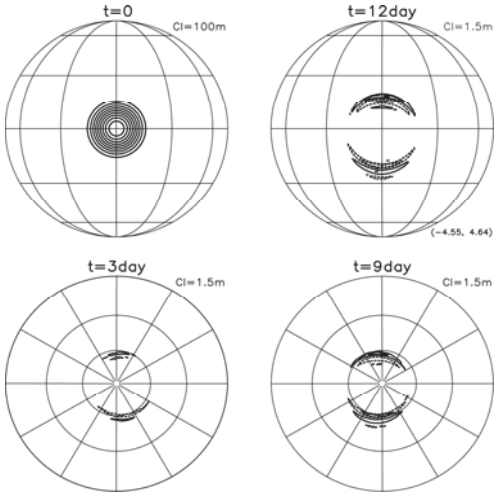


Fig. 2 The cosine bell at initial and the grid-point error by day 3, 9, and 12 for N=360. Positive (negative) values are in solid (dashed) lines. Contour interval by day 0 (12) is 100m and 1.5m, respectively. Outermost parallel of day 3 and 9 is 30 degrees of northern and southern hemisphere, respectively.

6 Acknowledgement

This research was carried out under the financial support from KMIPA (2015-5130).

SESSION

LATE BREAKING PAPERS: SCIENTIFIC COMPUTING AND APPLICATIONS

Chair(s)

TBA

Orthogonality and Computation

D. M. Topa[1,2,3] **and P. F. Embid**[1,2]
[1]Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM, USA
[2]Los Alamos National Laboratory, Los Alamos, NM, USA
[3]Engility Corporation, U.S.A.C.E. Engineer Research and Development Center,
Information Technology Laboratory, Vicksburg MS, USA

Abstract— *The property of orthogonality is predicated upon the specifications of a domain and a topology. The orthogonality of the continuum is violated in the computational domain as evidenced by poor convergence and numerical oscillations. Penalties are significant numerical errors and a substantial increase in computation time. By using linear independence, exact solutions are found in specific instances.*

Keywords: orthogonality, linear independence, Hilbert space, Lebesgue integration, domain topology

1. Introduction

Orthogonality and projection are two facets of the same gem. They are foundation concepts in many areas of science and engineering. For example, the mathematics of quantum mechanics and quantum field theory are the embodiment of the power of orthogonal projection. The least squares method delivers an orthogonal projection. A great deal of mathematics is based upon orthogonality. Computations are too often corrupted by the inappropriate presumption of orthogonality. This paper explores how this malady distorts numerical computations and presents a more suitable schema.

Common knowledge is that the trigonometric functions $\sin nx$ and $\cos nx$ (and their linear combination e^{inx}) are orthogonal over both the continuous domain $\Omega = \{x\colon -\pi \le x \le \pi\}$ and the uniformly sampled discrete domain $\sigma = \left\{x\colon x_\nu = -\pi + \frac{\nu}{n}\pi\right\}_{\nu=0}^{2n}$. A branch of mathematics built on these principles is Fourier analysis, which approximates arbitrary functions as

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx}.$$

Uncommon knowledge is that the sine and the cosine are the *only* functions orthogonal in *both* the continuum and in discrete space. This explains why there is no such thing as a discrete Legendre transform or a fast discrete Zernike transform.

Part of the confusion may stem from the famous case, Parseval's identity, where the two spaces are connected. The relation between smoothness in physical space and decay of the Fourier amplitudes is so fundamental in mathematics that it gives rise to the so-called Sobolev spaces, where the smoothness of the function $f(x)$ is understood in terms of the L^2-norm, and the corresponding decay of the coefficients is given in the related l^2-norm via

$$\int_{-\pi}^{\pi} \left|\frac{d^k f}{dx^k}\right|^2 dx = \sum_{n=-\infty}^{n=\infty} n^{2k} \left|c_n\right|^2 \ . \tag{1.1}$$

The spaces L^2 and l^2 are distinct and, excluding sine and cosine, there are no functions which are orthogonal in both.

2. Spaces and Topologies

Troubles arise when moving from the continuous space L^2 to the discrete space of l^2. This corresponds to moving from the continuum, the theoretical realm of the chalkboard, to discrete space, the realm of computer calculation. Either measurement or computation imply a discrete topology which sacrifices orthogonality.

2.1 Continuous vs. discrete

The stage is set with definitions of domains, norms, and membership for both topologies. Start with Ω, a continuous interval on the real number line. Given the boundary points, $[a, b]$ where $a < b$ this domain is formally defined as

$$\Omega = \{x \in \mathbb{R}\colon a \le x \le b\}\,. \tag{2.1}$$

For the discrete case there is but a finite sequence of μ points

$$\sigma = \{x_1, x_2, \ldots, x_\mu\} = \{x_\nu\}_{\nu=1}^{\mu} \tag{2.2}$$

ordered such that

$$a \le x_1 < x_2 < \cdots < x_\mu \le b. \tag{2.3}$$

Such a set represents a partition of the continuous interval Ω. The natural norms are introduced to measure distance. For continuous topologies

$$\|F(x)\|_{L^2}^2 = \int_\Omega F(x)\overline{F(x)}dx. \tag{2.4}$$

Integration is in the Lebesgue sense. For discrete topologies

$$\|f(x)\|_{l^2}^2 = \sum_{x\in\sigma} f(x)\overline{f(x)}. \tag{2.5}$$

The set σ may form a uniform mesh where $x_{k+1} - x_k = \Delta$, $k = 1, 2, \ldots, \mu - 1$. Reserve capital letters for the continuum, e.g., L^2, and lower case letters for discrete spaces, e.g., l^2.

The collection of all functions which are square integrable in the continuum is $L^2(\Omega)$. A function $F(x)\colon \mathbb{R} \mapsto \mathbb{C}$ is an element in this space iff the norm is finite,

$$F(x) \in L^2(\Omega) \quad \Longleftrightarrow \quad \int_\Omega F(x)\overline{F(x)}dx < \infty. \tag{2.6}$$

The collection of all functions which are square summable is $l^2(\sigma)$. A function $f(x)\colon \mathbb{R} \mapsto \mathbb{C}$ is an element in this space iff the norm is finite,

$$f(x) \in l^2(\sigma) \quad \Longleftrightarrow \quad \sum_{x\in\sigma} f(x)\overline{f(x)} < \infty. \tag{2.7}$$

With the stage set, the task at hand is resolving a target function in the bases of a complete and linearly independent set of functions over the domain. Let the highest order in the expansion be d for degree of fit, and for the continuum case where $x \in \Omega$ the target function is F, approximated as

$$F(x) \approx a_0 G_0(x) + a_1 G_1(x) + \cdots + a_d G_d(x). \tag{2.8}$$

In discrete space $x \in \sigma$ the target function f is approximated as

$$f(x) \approx b_0 g_0(x) + b_1 g_1(x) + \cdots + b_d g_d(x). \tag{2.9}$$

2.2 Riesz–Fischer theorem

The Riesz–Fischer theorem [2, p. 330] is a powerful result about the convergence of Cauchy sequences in L^2 spaces:[1]

Theorem 1 (Riesz–Fischer): Let $\{\phi_n\}$ be an orthonormal sequence of functions on Ω and suppose $\sum |a_n|^2$ converges. Denote the partial sum as

$$s_d = a_0\phi_0 + a_1\phi_1 + \cdots + a_d\phi_d.$$

There exists a function $F \in L^2(\Omega)$ such that $\{s_d\}$ converges to F in $L^2(\Omega)$, and such that

$$F = \sum_{k=0}^{\infty} a_k \phi_k, \tag{2.10}$$

almost everywhere.

Proof: The proof is both a staple in books on functional analysis and outside the scope of this article. ■

This existence theorem motivates a search for the amplitudes a_k. Section §4.3.2 exhibits the problems that arise when the basis set of functions is not orthogonal.

3. Background

This section provides the rudiments for finding a least squares solution for the amplitudes and discusses the benefits of orthogonality and linear independence. (Insightful background can be found in [3, §4.6, §5.13], [4, ch 8].)

[1] The theorem generalizes to $1 \le p < \infty$.

3.1 Least squares in L^2

Given a domain such as Ω in (2.1), a function $F(x) \in L^2(\Omega)$, and a sequence of basis functions $\{G_k\}_{k=0}^d \in L^2(\Omega)$, the least squares solution a_{LS} in the continuum is cast as

$$a_{LS} = \Big\{ a \in \mathbb{C}^{d+1} : \int_\Omega \left| F(x) - \sum_{k=0}^{d} a_k G_k(x) \right|^2 dx \text{ is minimized} \Big\}. \tag{3.1}$$

3.1.1 Linear independence

If the sequence of functions $\{G_k\}$ is *linearly independent* the resulting linear system,

$$\mathbf{A}^*\mathbf{A}a = \mathbf{A}^*F, \tag{3.2}$$

has full rank and is not singular. A simpler tool such as the normal equations can be used to find the solution vector a. The matrix form of these normal equations is

$$\underbrace{\begin{bmatrix} \langle G_0|G_0\rangle & \langle G_0|G_1\rangle & \cdots & \langle G_0|G_d\rangle \\ \langle G_1|G_0\rangle & \langle G_1|G_1\rangle & \cdots & \langle G_1|G_d\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle G_d|G_0\rangle & \langle G_d|G_1\rangle & \cdots & \langle G_d|G_d\rangle \end{bmatrix}}_{\mathbf{A}^*\mathbf{A}} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix} = \underbrace{\begin{bmatrix} \langle F|G_0\rangle \\ \langle F|G_1\rangle \\ \vdots \\ \langle F|G_d\rangle \end{bmatrix}}_{\mathbf{A}^*F}. \tag{3.3}$$

The inner product defines the matrix elements as

$$\langle G_j|G_k\rangle_{L^2} = \int_\Omega G_j(x)\overline{G_k(x)}dx. \tag{3.4}$$

The product matrix $\mathbf{A}^*\mathbf{A}$ is symmetric positive definite, and the eigenvalues are non-negative. The least squares solution is

$$a_{LS} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*F \tag{3.5}$$

3.1.2 Orthogonality

The Reisz-Fischer theorem motivates the use of an *orthogonal set* of basis functions. In this case, the inner products are simplified to

$$\langle G_j|G_k\rangle = \xi_j \delta_{jk}, \tag{3.6}$$

where δ_{jk} is the Kronecker delta function. Now the linear system decouples and the amplitudes can be solved mode-by-mode.

$$\begin{bmatrix} \xi_0 & 0 & \cdots & 0 \\ 0 & \xi_1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & \xi_d \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} \langle F|G_0\rangle \\ \langle F|G_1\rangle \\ \vdots \\ \langle F|G_d\rangle \end{bmatrix}. \tag{3.7}$$

The amplitudes are computed independently of one another

$$a_k = \frac{\langle F|G_k\rangle_{L^2}}{\langle G_k|G_k\rangle_{L^2}} = \frac{\langle F|G_k\rangle_{L^2}}{\xi_k}, \quad k = 0, 1, 2, \ldots, d. \tag{3.8}$$

For clarity, the inner products are marked to remind us of the host space topology.

A great benefit of decoupled systems is the amplitudes do not change as the order of fit is increased; given $d_1 \leq d_2$, $\{a_k\}_{k=0}^{d_1} \subseteq \{a_k\}_{k=0}^{d_2}$.

3.2 Least squares in l^2

Given a domain such as σ in (2.2), a function $f(x) \in l^2(\sigma)$, and a sequence of basis functions $\{g_k\}_{k=0}^{d} \in l^2(\sigma)$, the least squares solution b_{LS} in the discrete space is written as

$$b_{LS} = \Big\{ b \in \mathbb{C}^{d+1} : \sum_{\nu=1}^{\mu} \left(f(x_\nu) - \sum_{k=0}^{d} b_k g_k(x_\nu) \right)^2 \text{ is minimized} \Big\}. \tag{3.9}$$

as seen in (3.1). Different capitalization distinguishes the basis functions in $G_k(x) \in L^2\Omega$ from the basis functions $g_k(x) \in l^2(\sigma)$. For orthogonal functions, the *only* time $G_k(x) = g_k(x)$ is when they are sines and cosines.

3.2.1 Linearity

When the function sequence $\{g_k\}_{k=0}^{d}$ is linearly independent, the normal equations create a full rank linear system as seen in 3.3

$$\begin{bmatrix} \langle g_0|g_0\rangle & \langle g_0|g_1\rangle & \cdots & \langle g_0|g_d\rangle \\ \langle g_1|g_0\rangle & \langle g_1|g_1\rangle & \cdots & \langle g_1|g_d\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle g_d|g_0\rangle & \langle g_d|g_1\rangle & \cdots & \langle g_d|g_d\rangle \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_d \end{bmatrix} = \begin{bmatrix} \langle f|g_0\rangle \\ \langle f|g_1\rangle \\ \vdots \\ \langle f|g_d\rangle \end{bmatrix}, \tag{3.10}$$

with the matrix elements as

$$\langle g_j|g_k\rangle_{l^2} = \sum_{x\in\sigma} g_j(x) g_k(x). \tag{3.11}$$

3.2.2 Orthogonality

In cases[2] where the basis functions are orthogonal, the linear system decouples, and amplitudes can be computed mode-by-mode

$$\begin{bmatrix} \eta_0 & 0 & \cdots & 0 \\ 0 & \eta_1 & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & & \eta_d \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_d \end{bmatrix} = \begin{bmatrix} \langle f|g_0\rangle \\ \langle f|g_1\rangle \\ \vdots \\ \langle f|g_d\rangle \end{bmatrix}. \tag{3.12}$$

The solution set is

$$b_k = \frac{\langle f|g_k\rangle_{l^2}}{\langle g_k|g_k\rangle_{l^2}}, \quad k = 0, 1, 2, \ldots, d. \tag{3.13}$$

In contrast to (3.4) the inner product here is

$$\langle f|g_k\rangle_{l^2} = \sum_{j=0}^{d} f(x_j)\,\overline{g_k(x_j)}. \tag{3.14}$$

The linear system of normal equations in (3.10) is effectively diagonalized.

3.3 Summary

An orthogonal set of basis functions presents a significant advantage. The linear system decouples and the amplitudes can be computed directly using either (3.7) or (3.12) depending upon the topology. An orthogonal basis is also a linearly independent basis. Without orthogonality, and with only linear independence, a large linear system such as (3.3) or (3.10) must be solved. Without orthogonality, and without linear independence, the linear systems lack full rank and the solution demands tools like **QR** decomposition or the singular value decomposition.

4. Demonstration

The choice of a polynomial basis is driven by the domain, the domain of choice is the continuous interval $\Omega = \{x \in \mathbb{R} : -1 \leq x \leq 1\}$. The following demonstration uses the Legendre polynomials to illustrate points of emphasis.

4.1 Definitions

These functions are monic polynomials, orthogonal over the continuous set $\Omega = \{x : -1 \leq x \leq 1\}$. As expected, they are not orthogonal over the discrete and well-ordered set $\sigma = \{x_k : -1 \leq x_k \leq 1\}$, $k = 1, 2, \ldots, \mu$.

The polynomials can be defined as the set of functions which solve Legendre's differential equation,

$$\frac{d}{dx}\left((1-x^2)\frac{d}{dx} P_n(x) \right) + n(n+1) P_n(x) = 0. \tag{4.1}$$

A recipe for constructing the set starts with $P_0(x) = 1$ and $P_1(x) = x$ and employs the recursion relationship

$$P_{n+1}(x) = x P_n(x) - \frac{n^2}{4n^2-1} P_{n-1}(x). \tag{4.2}$$

This format demonstrates the definite parity of the polynomials, which toggles as the order increases in unit steps. The Legendre polynomials through order d are generated by

[2] Of course sine and cosine are orthogonal; other functions can be constructed to be orthogonal over a discrete domain [1, §8.2].

applying a Gram-Schmidt orthogonalization [3, p. 309] to the monomial functions $\left\{x^k\right\}_{k=0}^{d}$ through order d,

$$\mathcal{P}_d(x) = x^d - \sum_{k=0}^{d-1} \langle \mathcal{P}_k(x) | x^d \rangle. \tag{4.3}$$

$\{\mathcal{P}_n(x)\}_{n=0}^{\infty}$ is a sequence of orthogonal functions while $\{P_n(x)\}_{n=0}^{\infty}$ is a sequence of orthogonal and monic functions; the difference is a set of scaling factors.

As evidenced in (4.2), the parity of the Legendre polynomials is the same as the parity of the index. For example, when n is an odd number $P_n(x)$ is an odd function,

$$P_n(-x) = \begin{cases} P_n(x) & n \text{ even} \\ -P_n(x) & n \text{ odd} \end{cases}. \tag{4.4}$$

The norm is given as

$$\langle P_m | P_n \rangle = \frac{2}{2n+1} \delta_{mn}. \tag{4.5}$$

4.2 Computational efficiency

The Legendre polynomials are orthogonal in the continuum, which leads to a trivial form for the amplitudes,

$$\begin{aligned} a_k &= \frac{\langle F | P_k \rangle_{L^2}}{\langle P_k | P_k \rangle_{L^2}} \\ &= \frac{2k+1}{2} \int_{-1}^{1} F(x) P_k(x) dx, \quad k = 0, 1, 2, \ldots, d, \end{aligned} \tag{4.6}$$

as seen in (3.8). But in the discrete domain of computation, the off-diagonal matrix elements in (3.10) such as

$$\langle P_2(x) | P_4(x) \rangle = \sum_{\nu=1}^{\mu} P_2(x_\nu) P_4(x_\nu) dx \tag{4.7}$$

are nonzero, demanding solution of the complete linear system. Denuded of orthogonality, the Legendre polynomial set is a cumbersome combination of monomials. Table 1 compares functional forms for monomials and Legendre polynomials.

Table 1: Functional forms for the monomials and the Legendre polynomials.

order	monomial	Legendre
0	1	1
1	x	x
2	x^2	$\frac{1}{2}\left(3x^2 - 1\right)$
3	x^3	$\frac{1}{2}\left(5x^3 - 3x\right)$
4	x^4	$\frac{1}{8}\left(35x^4 - 30x^2 + 3\right)$
5	x^5	$\frac{1}{8}\left(63x^5 - 70x^3 + 15x\right)$

Comparing the monomials to the Legendre polynomials order by order, the succinctness of the monomials is apparent.

4.2.1 Monomials

Showing that the monomials are a minimal spanning set requires establishing that they are linearly independent.

Theorem 2 (Linear independence of the monomials): The monomial set $\left\{1, x, x^2, \ldots, x^d\right\}$ with $d \geq 1$ is linearly independent over any continuous interval including the origin.

Proof: The crux of the proof uses induction to show that the Wronskian

$$W(x) = \begin{vmatrix} 1 & x & x^2 & \cdots & x^d \\ 0 & 1 & 2x & & dx^{d-1} \\ 0 & 0 & 2 & & d(d-1)x^{d-2} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & & & d! \end{vmatrix} \tag{4.8}$$

evaluates to $W(0) = \prod_{k=0}^{d} k! \neq 0$. ■

4.2.2 Spans

Consider two different spans for the Legendre polynomials S_P, and for the monomials S_m,

$$\begin{aligned} S_P &= \text{sp}\left\{ \begin{bmatrix} P_0(x) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ P_1(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ P_d(x) \end{bmatrix} \right\}, \\ S_m &= \text{sp}\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ x \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ x^d \end{bmatrix} \right\}. \end{aligned} \tag{4.9}$$

Both spans are linearly independent. In the limit $d \to \infty$, both spans are complete. The span S_P is orthogonal over Ω, neither span is orthogonal over σ.

Theorem 3 (Equivalence of spans): Let $\{S_m\}^d$ be the span of a Hilbert space described by the monomial set $\left\{1, x, x^2, \ldots, x^d\right\}$, and let $\{S_P\}^d$ be the span of a Hilbert space described by the Legendre polynomial set $\{P_0(x), P_1(x), P_2(x), \ldots, P_d(x)\}$. The spans $\{S_m\}^d$ and $\{S_P\}^d$ are equivalent.

Proof: Proof is based upon the construction process in (4.3). ■

4.2.3 Transformations

Given the equivalence of the Hilbert spaces spanned by S_m and S_P, there exists an affine transformation to move between the Legendre basis and the monomial basis. One

such transformation matrix is shown in (4.10) for $d = 5$,

$$T_5 = \frac{1}{8}\begin{array}{c} \begin{matrix} P_0 & P_1 & P_2 & P_3 & P_4 & P_5 \end{matrix} \\ \begin{bmatrix} 8 & 0 & -4 & 0 & 3 & 0 \\ 0 & 8 & 0 & -12 & 0 & 15 \\ 0 & 0 & 12 & 0 & -30 & 0 \\ 0 & 0 & 0 & 20 & 0 & -70 \\ 0 & 0 & 0 & 0 & 35 & 0 \\ 0 & 0 & 0 & 0 & 0 & 63 \end{bmatrix} \end{array}. \quad (4.10)$$

Two important points are apparent: the transformation is exact in this integer form and the inverse transformation is inexpensive because T_d is an upper triangular matrix.

4.3 Probe functions

The first example extols the main point; the presumption of orthogonality destroys performance and provides an inferior result. Next, a discontinuous function reveals the computational troubles encountered outside of Riesz–Fischer.

4.3.1 Example I: smooth function

Start with the amplitude vector $\alpha^T = (0, 1, 2, 3, 0, 0, 0)$ to define a C^1 function,[3]

$$\begin{aligned} f(x) &= \alpha \cdot \{P_n(x)\}_{n=0}^{6} \\ &= P_1(x) + 2P_2(x) + 3P_3(x) \\ &= \frac{1}{2}\left(-1 - x + 3x^2 + 5x^3\right). \end{aligned} \quad (4.11)$$

Problem– What problems arise from the assumption that the Legendre polynomials are orthogonal over $l^2[-1, 1]$? Equation (4.6) is used in a corrupted form,

$$a_k = \frac{2k+1}{2}\Delta \sum_{x\in\sigma} f(x) P_k(x), \quad (4.12)$$

where Δ represents a uniform mesh spacing. The results are shown in Figure 1. The solid symbols are for amplitudes of input terms which are nonzero, open symbols indicate spurious modes created by using an inappropriate numerical method. The method converges linearly, which implies that reasonable accuracy is very expensive. Spurious modes are created, modes with amplitude 0 have nonzero values.

Solution– Switch to the monomial basis $\left(g_k(x) = x^k\right)$, and compute the amplitudes using (3.10). Use an affine transformation to recover the Legendre amplitudes, for example $b = T_6 a$. The algorithm is exact. The test case used $\Delta = 0.25$ for a double precision computation in *Mathematica* and recovered exact answers[4] for $\alpha_0 - \alpha_6$.

Source of error– The root of the error is analytic integrals do not equal their numeric counterparts; the off-diagonal elements in (3.10) are not zero and the solution methods

[3]Functions with a continuous first derivative.

[4]The exact answer is attributed to the twin blessings of a high quality linear solver and the x values having exact binary representation.

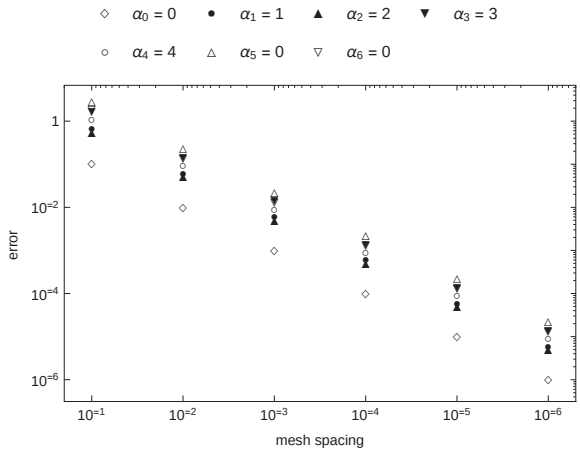


Fig. 1: Error in computing the Legendre amplitudes for the function in (4.11).

for (3.12) don't work. For example, the product function $P_0(x) P_2(x)$, which corresponds to $\mathbf{A}^*\mathbf{A}$ matrix element $(r, c) = (1, 3)$ in (3.3),

$$\int_{-1}^{1} P_0(x) P_2(x)\, dx \neq \sum_{\nu=0}^{2n} P_0(x_\nu) P_2(x_\nu)\, \Delta. \quad (4.13)$$

The left-hand side is an analytic integral exactly equal to 0 and the right-hand side is a numerical summation. For $\Delta = \frac{1}{4}$ the integral is approximated as

$$\sum_{\nu=1}^{8} \frac{1}{2}\left(3x_\nu^2 - 1\right)\Delta = \frac{1}{64} \neq 0,$$

a statement of orthogonality lost. Figure 2 contrasts the two integration methods.

4.3.2 Example II: discontinuous function

Stepping outside of the strictures of the Riesz–Fischer theorem, consider a discontinuous function,

$$\operatorname{sgn} x = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}. \quad (4.14)$$

The sign function has odd parity so the approximation contains only odd terms.

Figure 3 compares fits with maximum order $d = 50$, $d = 100$, and $d = 200$ against the sign function. The amplitudes for the approximations are shown in Figures 4 and 5.

Amplitudes for the monomial approximations are shown in Figure 4. To use a logarithmic scale, the absolute value of the amplitude $|a|$, is shown. Filled rectangles indicate the sign of the amplitude is positive, open rectangles indicate negative values. Amplitudes for even functions have value 0 and are not plotted. The amplitudes grow without bound.

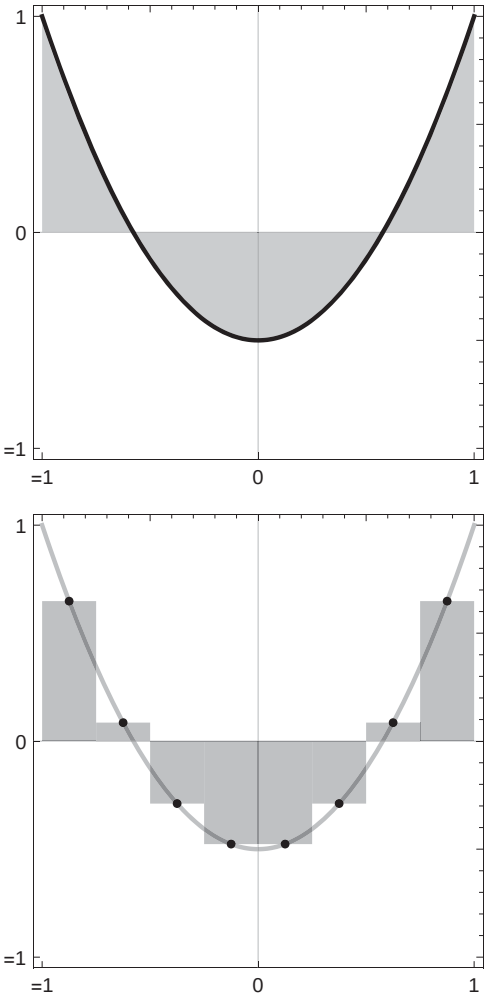


Fig. 2: Error in computing matrix cross terms. The function is $P_0(x)P_2(x)$.

Notice the invariance in form of the amplitudes; the scale is changing, but not the shape.

Amplitudes for the Legendre approximations are shown in Figure 5. Positive values are shown with filled rectangles, 0 values with open diamonds, and negative values with open rectangles. These amplitudes are bounded. The plots demonstrate that the values of the lowest order terms do not change as the order of fit is increased. Notice the invariance of amplitude values; the lower order amplitudes do not change as the fit order increases.

Problem– Monomial amplitudes grow without bound quickly exhausting the range of double precision computation. In such cases, orthogonal polynomials provide infinity insurance; Legendre amplitudes are bound. The monomial linear system is ill-conditioned. After all, these functions are merely linearly independent. But the Legendre polynomials present a much better conditioned linear system as they are closer to being orthogonal in l^2.

Solution– Use the Legendre basis $(g_k(x) = P_k(x))$ and

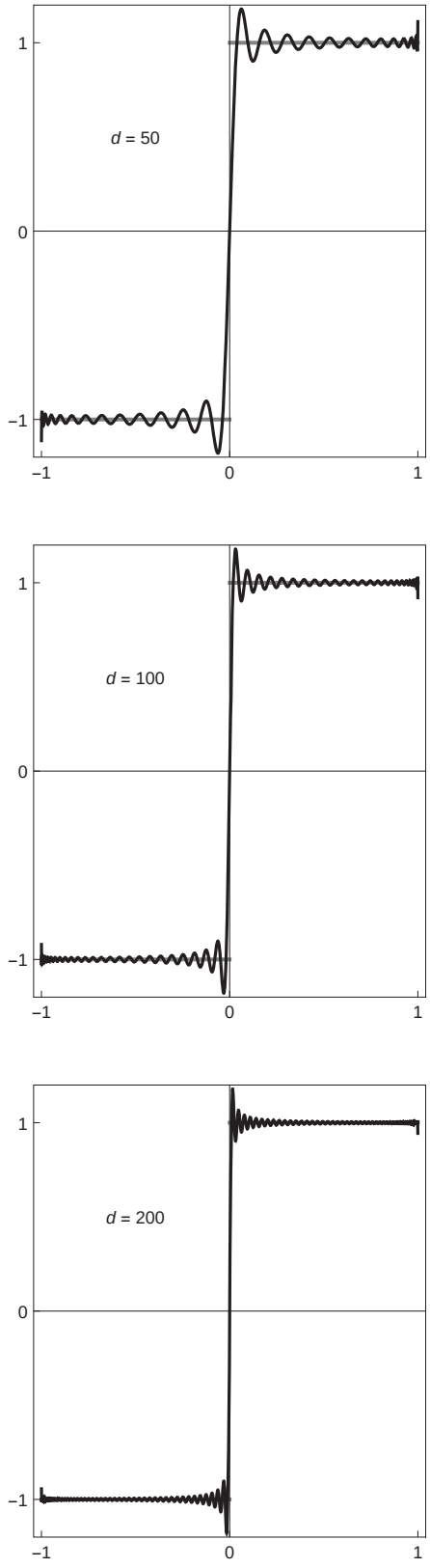


Fig. 3: Approximation of the sign function showing the Gibbs phenomena at the midpoint and boundaries.

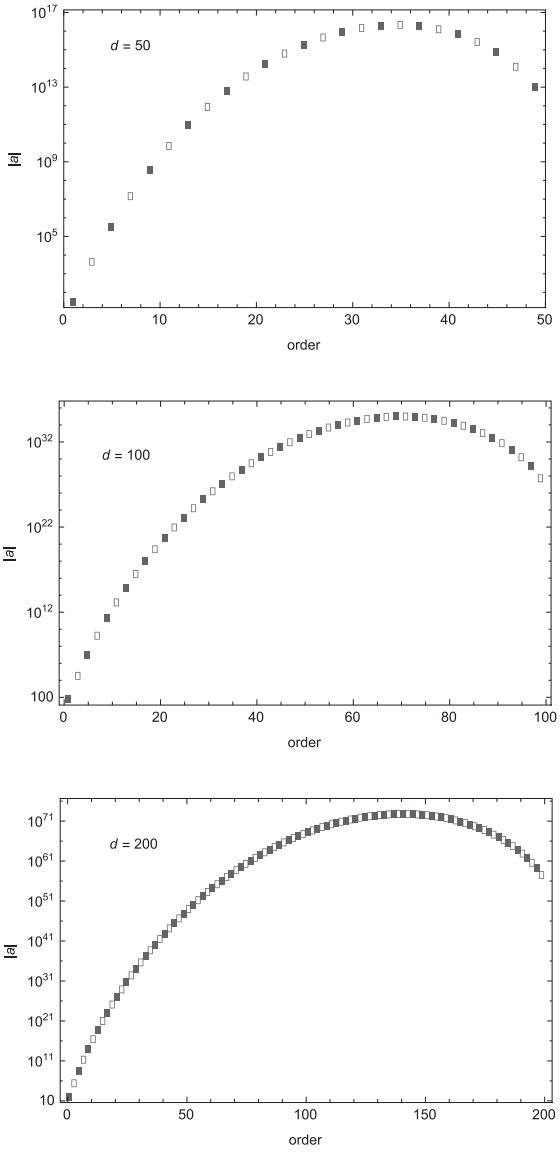


Fig. 4: Amplitudes for the monomial approximation of the sign function in (4.14) found by solving (3.10).

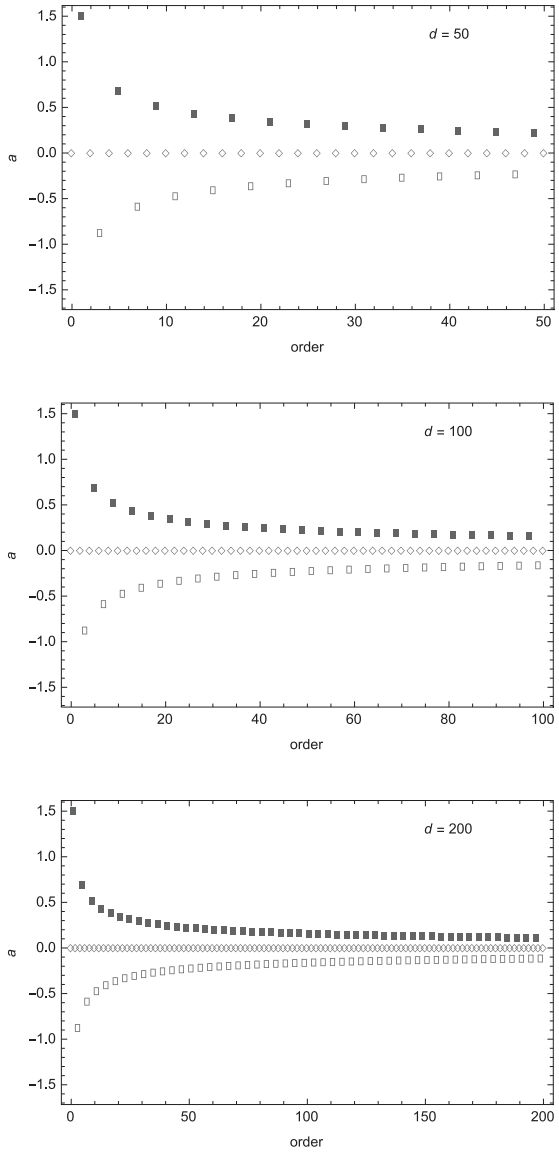


Fig. 5: Amplitudes for the Legendre approximation of the sign function in (4.14) found by solving (3.10).

compute the amplitudes using (3.10). Affine transformations recover the monomial amplitudes, for example $a = T_{200}^{-1} b$.

5. Conclusion

Computation demands discrete evaluation and finite meshes. Outside of sine and cosine, there are no functions continuous in both L^2 and l^2. The presumption of orthogonality in discrete space introduces extreme numerical errors. Instead, rely upon linear independence to create and solve the linear systems. Affine transformations connect the function spaces of choice. When resolving discontinuous functions, L^2-orthogonal functions can create a dense linear system with improved conditioning.

One of the authors (PFE) thanks Los Alamos for support during the summer of 2012.

References

[1] Bevington, P. R., *Data Reduction and Error Analysis for the Physical Sciences* 1e, McGraw Hill, 1969.
[2] Rudin, W., *Principles of Mathematical Analysis* 3e, McGraw-Hill, 1976.
[3] Meyer, C. D., *Matrix Analysis and Applied Linear Algebra*, SIAM, 2004.
[4] Laub, A. J., *Matrix Analysis for Scientists and Engineers*, SIAM, 2005.
[5] Weisstein, E.W., "Legendre Polynomial." MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/LegendrePolynomial.html

Analysis of Algorithms for Rapid Curve Characterization

G. G. Wood[1], **M. Hipskind**[1], **and T. Hampson**[1]
[1]Department of Physics, California State University Channel Islands, Camarillo, CA, USA

Abstract— *We have developed rapid, non-iterative, algorithms to extract characteristics from noisy signals. The center, width, height and noise are estimated from a sigmoid function with Gaussian distributed noise. Several algorithms are studied to find behavior as the signal to noise ratio decreases. The goal is to develop a heuristic to choose which algorithm best characterizes the signal in each signal-to-noise regime.*

Keywords: Fitting, Parameter Optimization, Nonlinear Model, Data Analysis

1. Introduction

For hundreds of years, people have tried to explain noisy experimental data with idealized mathematical functions with adjustable parameters, found by best fit to data. The idea of minimizing the squares of the differences between experiment and theory was formalized by Legendre over 200 years ago[1]. With the widespread use of computers, a wide variety of fitting procedures were developed in the 1960's and 1970's such as fitting to sub-regions within the data[2], elimination of troublesome data points[3], combining two different methods of fitting over subsets of the data by determining on a case by case basis which is best to use[4], and employing multi-step algorithms to optimize solutions in the absence of complete data[5]. Development and use of fitting is ongoing[6]These algorithms give good results in many cases, but have challenges which inspire many new algorithms and modifications to existing ones. Among these challenges are: pre-knowledge of the functional form, long time to convergence, and and finding local, instead of global, optimal solutions. In this paper, we propose a rapid, alternative algorithm which requires no iteration, no pre-knowledge of the exact functional form of the data.

In many experiments, data is gathered with an unknown functional form. In this work, we consider one type of signal: the sigmoid (or S-shaped) curve and consider how information can be extracted from a limited, or noisy, signal without knowledge of the exact functional form. This kind of signal is intimately related to another common shape, the peaked function, like a gaussian. The two are related by elementary calculus: the peaked function is the derivative of the sigmoid function and these algorithms for the sigmoid can be used on other data.

Often in experiments, a very limited amount of data is available. As examples, certain magnetic systems (called spin glasses) have long time decays, predicted to be on the order of the age of the universe[7], and so the experimental data is limited - but nonetheless the location of the the centroid of the sigmoid shaped transition is of key importance. When plotted as magnetization versus the logarithm of time, the data has a sigmoid shape. Another example is the helix to coil phase transition in peptides[8], as a function of temperature. The peptide gives a rotation of the angle of polarized light passed through it which is proportional to helix content of the peptide. The peptides must be in solution to undergo this change, and so the temperature range is quite narrow over which experiments can be meaningfully conducted. Generally, in this kind of experiment, a series of mutants of the peptide are studied and each mutant has a slight difference in centroid location (which in this case would be transition temperature) which effect the stability of the peptide.

The algorithms proposed are designed to work under such cases of very limited, highly incomplete data series, and give reliable estimates for centroid position and width of transition without knowledge of functional form.

With massively parallel on-chip chemical reactions now possible[9], fast and reliable algorithms to detect a change in signal which corresponds to a chemical reaction can be useful. Rapid algorithms, such as those developed in this work, can be used to screen experimental results and flag a small number of promising candidates for further, more careful, analysis.

1.1 Sigmoid Functions

Many different functions are sigmoidal. Examples are the logistic function, $\frac{1}{1+e^x}$, the arctangent, the hyperbolic tangent, the error function (which is the integral of the gaussian, and in fact the integral of any single peaked, gaussian-like function will be sigmoidal), and algebraic functions such as $\frac{x}{1+|x|}$. Each differs slightly, with some more rapidly saturating toward the baselines then others.

2. Algorithms to Extract Features

The first step in analysis of the signal is to find the total range of y-values. An arbitrary midpoint of y is found by averaging the highest and lowest y-values from the signal, which we will call y_m. Now two clouds of data are formed: all data with y-value above y_m, and below.

A first, but rejected, algorithm was to fit a straight line between the average x and y values of each cloud, and find the centroid of the signal (x-location) where this line passes

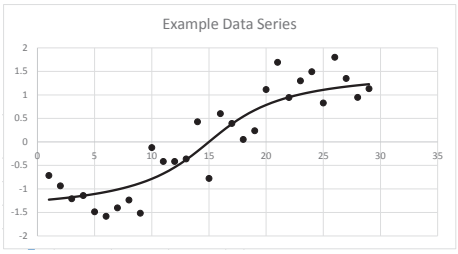


Fig. 1: An arctangent signal with gaussian noise. Solid line shows the arctangent function modeled with this data, which is the arctangent((x-15)/5), where the x-axis values range from one to 29. Note that the y-axis values of the arctangent should run from about +1.57 to -1.57, but only achieves +/- 1.23 over this range. Most of the transition is done, but a significant amount is still to be done. The standard deviation of the noise is about 0.35, giving a signal to noise ratio of about 9.

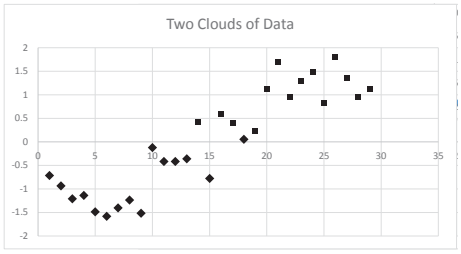


Fig. 2: Same data from Fig. 1 above, but showing the grouping of data points from the two clouds algorithm. Diamonds indicate the data with lower y-values, and squares that with larger y-values. In this case, there are 15 diamonds, 14 squares, but in general there can be a larger difference in the number of data points in each region.

through y-value equal to y_m. The reason for rejection is that this algorithm is biased to move the centroid to the side with more data points. To resolve this problem, the same number of data points are used from each cloud. But how to choose which data? The average x and y position, and number of data points in each cloud are computed. Let us call set 1, with N_1 points, average (x,y) values of (x_1, y_1), the set with fewer data points, so $N_1 < N_2$. If $x_1 < x_2$, then the data set begins (from low x value) with fewer data points, so the N_1 data points from set 2 are selected which have the lowest x-values. On the other hand, if $x_1 > x_2$, the data set begins with a larger group (group 2) of data points then transitions to the smaller. So the N_1 data points from set 2 are chosen which have the highest x-values. In the happy case that $N_1 = N_2$, no added step need be done. In any case, the *subset of cloud 2 are chosen that are closest to the average values of group 1*. These points are be closer to (and within, and across) the transition region from group one to two. The new average values of the small group 2 are computed which we denote as x_2' and y_2'. To estimate the x-coordinate of the midpoint, which we will cal x_m of the transition, we average x_1 with x_2'. Fig. 3 shows the selections of the two clouds of data from the original example data of Fig. 1.

Next, estimates of the width (in x) and height (in y) of the transition are required. First, two estimates of the height can be attained via: (1) the difference between the maximum and minimum values within the data set, and (2) the difference between y_1 and y_2'. The first is almost surely an overestimate, the second an underestimate.

To estimate the width of the transition, we divide the y-axis into four equal ranges, from the lowest to highest y-values in the set, and count the number of data points with y values within the center two ranges, or the number in the transition region, which we will call N_t. In a high signal to noise region, these would be the data closest to the centroid, but in general, these data points could be anywhere. Thus we choose the N_t data points closest to the centroid position, above, and fit them to a straight line, as seen in Fig. 3. The width, w, is estimated to be: the x-distance from where this line hits the lowest to highest y-values within the data set.

2.1 From Two to Three Clouds of Data

Now that crude estimates of all parameters have been established, we divide the data into three groups, or clouds of data and refine the estimates. All data with x-values within the range $x_m - w/2$ to $x_m + w/2$, or, in words, within one width of the center position, are grouped into the center cloud, II, with N_{II} data points, and average x and y values x_{II} and y_{II}. All data with x-values lower then this region are in one group, with N_I data points, average x and y values x_I and y_I, and above in a third group denoted with III. Fig. 4 shows the three groups of data for the example data from Fig. 1 above.

Improved estimates of the height and width of transition are derived as follows: first the height is estimated as: $y_{III} - y_I$. To improve the width estimate, beginning with the centroid location, pairs of data points equidistant from the centroid are considered. The width expands to encompass these data points if the difference in their y values is less then the height, $y_{III} - y_I$. The problem with this is that very noisy signals will have artificially low width. To help this, two pairs of points, at the new width, w, to be considered,

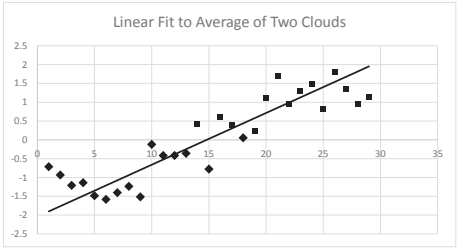


Fig. 3: Same data from Fig. 1 above, showing the linear fit between the two clouds of data. Ideally, this passes through the centroid of the transition, which in this example is the point (15,0), and it does come quite close. Note that before preforming this fit, the two clouds of data are trimmed to the same size, in this case the lower y-value data points contain one additional data point, so the point at the far left is excluded (in general, the points farthest from the transition point are trimmed).

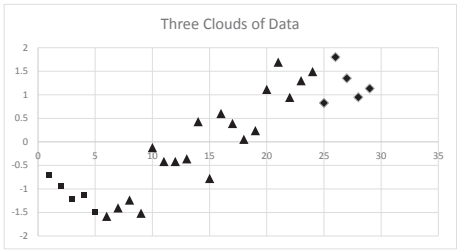


Fig. 4: Same data from Fig. 1 above, showing the selection of three regions of data points via the three clouds algorithm: squares for the low y-value baseline, triangles for the transition region, and diamonds for the high y-value data. The central transition region (triangular data points) is much broader then the full width at half maximum.

and just inside this width, are used together, and the width is expanded until the pairs of points violate the rule twice.

3. Results

Two sets of data are explored in detail, below. The first is a series of data sets with increasing signal-to-noise ratio (SNR). As expected, as the SNR increases, the results more closely match the expected values. This first data set employs a very limited window of the data: the data is nowhere near the baselines in the range of data used. This reflects the typical case in experiments where the x-axis is limited by experiment. As examples, in the helix-coil transition experiments[8], the x-axis is temperature and is limited to the range of values where the solution neither freezes nor boils, so for pure water from 273 K to 373 K. In magnetism experiments[7], the x-axis is time on a logarithmic scale, so the lowest possible time is limited by the experiment coming to thermal equilibrium and usually requires 3×10^2 seconds at the very least, and the longest time the equipment can run is only about four days, about 3×10^5 seconds, due to the limited capacity of the liquid refrigerants. Refilling vibrates the system so badly that measurement cannot continue. In this case, the x-axis can only run from two to five (on a log base ten scale). In both cases, it is highly impractable to conduct an experiment where both baselines are accessible.

The second data series holds the SNR fixed (at 9.0, the same as the SNR in the sample figures, above, explaining the algorithms) and increases the number of data points, holding the density of data points constant, so expanding the range the data cover.

3.1 Details of Data

Several choices have to be made when producing sample sets of data for analysis. This section contains notes on these details for the data in this paper. The function used for the signal is a logistic function of the form:

$$y(x) = \frac{R}{1 + \exp\left(x - x_0\right)/w}, \tag{1}$$

In this expression, R is the range of the data, x_0 is the centroid location and w is the width (which can be related to the full width at half maximum, or the half-width and half maximum). Added to this signal is gaussian noise with a standard deviation, σ. The signal to noise ratio (SNR) is given by R/σ. Two thousand runs were preformed and the results averaged to produce the data seen below.

For simplicity, data are evenly spaced, and the x-values are the integers, so that when, say, a centroid location is found at $x = 88$, then 87 data points proceed it.

The algorithms work best if the centroid of the transition occurs near the center of the data set. Data employed in this paper have twice as many data points on one side (always the added data is at high x-value, which have higher y-values) than the other. Adding additional data points to this side doesn't have a significant effect on results, but some improvement will be seen if the signal is shifted to be placed nearer the center of the data window.

3.2 Results Varying SNR

Four results are discussed below: locating the centroid of the signal, range, width, and an estimate of the noise, all as a function of signal to noise ratio. The SNR range from 6.25 to 100.

The true centroid location is 88 in all cases, in this section, but as can be seen in the figure below, the results are systematically higher. This is due to the extra data at higher x-values biasing the results. With an SNR of 6.25, the centroid is off by about 10%, but the true value is just outside the large one sigma uncertainty bar, and well within two sigma.

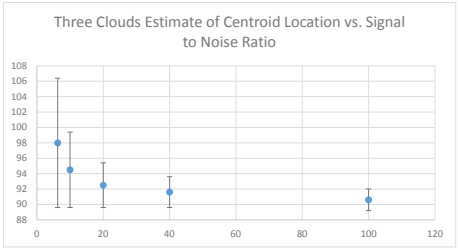


Fig. 5: Estimated centroid location (y-axis) versus SNR (x-axis) is plotted. As the SNR grows, the centroid location becomes more accurate and saturates to a value near, but not on the true value (systematically off by about 2%). Data shown is from the three clouds algorithm, but the two clouds algorithm is only about 0.5% worse, showing a rapid algorithm can give a good estimate of the centroid location, even of a noisy and incomplete signal.

The is a huge improvement in the range estimate by using three clouds instead of two can be seen in Fig. 6. At SNR of 6.25, two clouds is off by 40%, and three clouds only by 6%. The range of data used is 165, which is seen at large SNR in both algorithms, although the two clouds algorithm switches from over- to under-estimation of the range at large SNR. The data were produced from a logistic function with a true range of 200, but only the range of 165 appears in the limited window of data in this example. This is an area where knowledge of the exact function is necessary to finding the true range from a very limited sample of data.

Width of transition calculation versus SNR is plotted in Fig. 7 for the three clouds algorithm. The two clouds results are large overestimates, and not shown. Nonetheless, the crude estimate was necessary, as explain above, to find the improved width displayed. At an SNR of 6.25, width estimate is off by just over 30%, but the uncertainty bar is huge and the true value lies just outside the one sigma uncertainty and as the SNR improves to 10, the width estimate is off by about 20% and the true value is covered by the one sigma uncertainty bar.

In addition to estimated the centroid, height and width of the transition, the three clouds algorithm can give a reasonable estimate of the noise, and since the height is the signal,

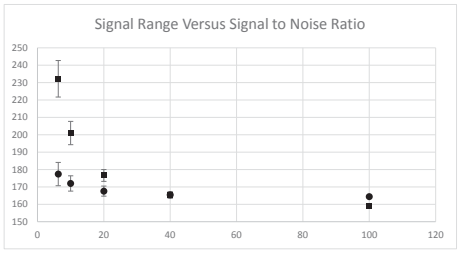


Fig. 6: Estimated range of data, on the y-axis, plotted versus SNR on the x-axis showing the improvement in range estimate as SNR increases. Results from both algorithms are presented: squares for two clouds and circles for three clouds, which show the great improvement in results when moving from the simpler to somewhat more complex algorithms.

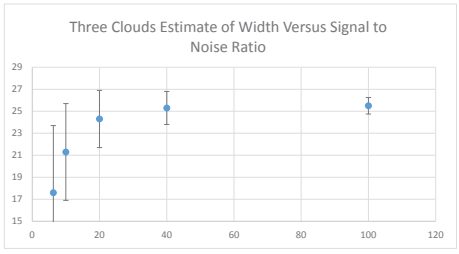


Fig. 7: Estimated width of transition (y-axis) versus SNR (x-axis) for the three clouds algorithm.

the SNR of the data. As seen in Fig. 8, the noise estimate is systematically high. Even when the SNR improves (large SNR, smaller noise) there is still a systematic gap due to the very limited window the data provides. Ideally, there would be a transition between two fixed, stable baselines, but this rarely occurs in experiment as discussed above. Since in this data set the baselines themselves are changing, this is interpreted as noise by the algorithms employed here. This points to a potential further division of the data, whereby smaller regions of data at the extrema of the x-axis (farthest from the centroid) are employed, alone, to find superior estimates of the noise, and the range of the transition, but any such algorithm would have to already have an estimate of SNR to use to feed into it the best range of data to use to minimize error: if the SNR is large enough, few data are needed, but at small SNR, perhaps all should be used.

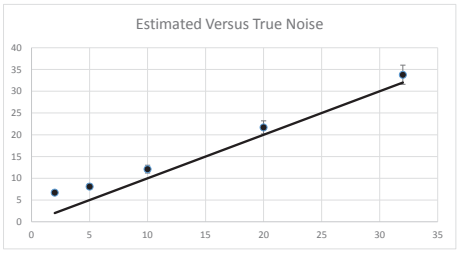


Fig. 8: Calculated noise (y-axis) versus true noise (x-axis) is plotted. Calculations are systematically somewhat high, due to the inability of the alogirthms to separate the long slow change in the baselines from the random noise. Noise is gaussian distributed, and the values computed on this graph are one sigma values.

3.3 Results Varying Number of Data Points

In this section, the signal to noise ratio is held at 9 (the same as the SNR in the initial noisy signal in Fig. 1 above), but the number of data points on each side of the transition is increased. Still, we require the signal to be off-center with double the data on one side of the transition then on the other. Only 88 data points were allowed on one side of the transition in the section above, and the width of the signal was 26.5, thus about 3.3 widths worth of data were employed on each side by the algorithms. In this section, we consider two widths up to 30 widths worth of signal on each side.

Results for width and noise of the signal improve (slightly) as more data is employed, and this is summarized in Table 1. This leaves the range and centroid location to be examined, in figures and text below.

The range of the signal was not accurately found above, even with high SNR. We find even with a relatively poor SNR of 9, with enough data, the range can be found very accurately, as the true baselines are revealed, as seen in Fig. 9. However, the two cloud algorithm systematically overestimates the range, but the three cloud algorithm agrees well when sufficient data is employed.

The centroid location is accurately predicted, especially when a great deal of data is available. The systematic problems found above go away when enough data is available on each baseline. With 55 data points, there is a 18% error (estimate is high, always, due to a larger number of data points at large x, which is due to the centroid intentionally placed off-center), but when 110 data points are used, this drops to under 5%, and under 1% when 250 data points are available for use, as seen in Fig. 10.

Data Points	Width	Noise
55	17.4	24.1
88	21.3	21.7
110	21.5	23.8
172	23	23.3
250	23.7	23.1
310	23.9	23.0
500	24.3	22.6
1000	24.6	22.4

Table 1: Width and noise estimates from three clouds algorithm for various numbers of data points on each side of the transition. True value of the noise was 22 and the width was 25, meaning there is a slight systematic underestimate of the width, and overestimate of noise, both of which improve greatly as the number of data points increase. The one sigma standard deviation of the mean values for width are all between 4.4 and 5.0, and for the noise level are between 2.2 and 0.3 and this latter value decreases as the number of data points increases. The density of data points is held fixed, thus the increase of data occurs outside the transition region, in the baselines. This means that longer baselines help determine all parameters.

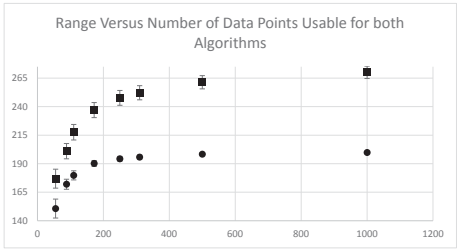


Fig. 9: Estimated range of the data (y-axis) versus number of data points employed (x-axis) for both algorithms. The data from the two clouds algorithm is represented by the squares, and the circles represent three cloud algorithm results.

4. Future Work

Additional fast algorithms will be developed and compared with these to optimize for both run time, accuracy and precision[1]. Ultimately, a crossover could be found from one algorithm to another as a function of, say, noise or number of data points when one algorithm is clearly superior for one type of data only. Other types of noise need to be considered: gaussian noise is not common in experiment. Shot noise, in which the level of noise is proportional to the counts of

[1]By accuracy, we mean the percent difference from average value found to the true value, whereas by precision we mean the size of the uncertainty bars around the values found

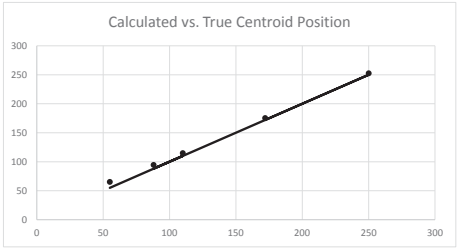


Fig. 10: Estimated centroid position (y-axis) versus true centroid position (x-axis). Data are from the three clouds algorithm, and indistinguishable results are obtained by the two clouds algorithm. The line is where data fall to show agreement, and so has slope one, and y-intercept of zero.

data within a detector, is an alternative example. Another area for work is the subtraction of linear baselines. In the present work, the data smoothly vary from one (unknown, but constant) baseline to another, such as from zero to one, or minus one to one, but often in experiment, there are linear baselines with non-zero slope, and the signal varies from one to another. Thus these linear baselines often have to be subtracted before any analysis can occur. In helix to coil transitions in peptides, employing the circular dichorism technique, various groups use somewhat different equations for the baselines[8].

5. Conclusions

Algorithms for rapid, non-iterative, characterization of sigmoid curves are developed and the effectiveness of these algorithms are measured as a function of signal to noise ratio and amount of data available. The centroid location, width of transition, the height of the transition, and the level of noise are estimated from noisy sigmoid functions.

References

[1] A. M. Legendre. *Nouvelle méthodes pour la détermination des orbites des comètes*. Paris, France: Courcier, 1805.

[2] H. Akima, "A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures", *J. of the Assoc. for Comp. Machinery*, vol. 17, pp. 590-602, Oct. 1970.

[3] R. D. Cook, "Detection of Influential Observation in Linear Regression", *Technometrics*, vol. 19, pp. 15-18, Feb. 1977.

[4] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters", *J. Soc. Indust. Appl. Math.*, vol. 2, pp. 431-444, Jun. 1963.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *J. Royal Stat. Soc. B*, vol. 39, pp. 1-38, Jan. 1977.

[6] H. Yi, and Z. XiaoPing, "An Improved Iris Localization Algorithm Based on Curve Fitting,", *Applied Mechanics and Materials*, vol. 380, pp. 1176-1179, Aug. 2013.

[7] Y. G. Joh, R. Orbach,G. G. Wood, J. Hammann, and E. Vincent, "Extraction of the Spin Glass Correlation Length", *Phys. Rev. Lett*, vol. 82,pp.438-441, Jan. 1999.

[8] G. G. Wood, D. A. Clinkenbeard, and D. J. Jacobs, "Nonadditivity in the alpha-helix to coil transition", *Biopolyers*, vol. 95,pp. 240-253, Apr. 2011.

[9] X. Gao, X. Zhou, and E. Gulari, "Light directed massively parallel on-chip synthesis of peptide arrays with t-Boc chemistry", *Proteomics*, vol. 3, pp. 2135-2141, Nov. 2003.

Travelling wave solutions of the Majda-Biello system

Abdullahi Rashid Adem
International Institute for Symmetry Analysis and Mathematical Modelling,
Department of Mathematical Sciences, North-West University, Mafikeng Campus,
Private Bag X 2046, Mmabatho 2735, Republic of South Africa
Email: Abdullahi.Adem@nwu.ac.za

Abstract— *In this paper we present the exact solutions of the Majda-Biello system. This system describes the nonlinear interaction of long-wavelength equatorial Rossby waves and barotropic Rossby waves with a substantial midlatitude projection in the presence of suitable horizontally and vertically sheared zonal mean flows. The method that is used to construct the exact solutions is the extended tanh method.*

Keywords: Majda-Biello system, extended tanh method, travelling wave solutions

1. Introduction

Real world problems in the fields such as physics, chemistry, biology, fluid dynamics can be modelled by nonlinear partial differential equations (NLPDEs). Thus it is crucial to search for solutions of NLPDEs. However it is almost impossible to find solutions of NLPDEs in closed form. Despite of this fact, in recent years drastic progress has been made in developing methods to find exact solutions of the NLPDEs. Some of the most well known methods are the inverse scattering, the Hirota's bilinear method, the simplest equation method, the homogeneous balance method, the Lie symmetry analysis, etc. [1]–[3], [5]–[7].

In this paper we study the nonlinear system of partial differential equations, namely the Majda-Biello system [8], [9], which is given by

$$u_t - \left(1 - \frac{1}{(2m+1)^2}\right) u_{xxx} + vu_x + uv_x = 0, \tag{1a}$$

$$v_t - v_{xxx} + uu_x = 0. \tag{1b}$$

System (1) describes the nonlinear interaction of long-wavelength equatorial Rossby waves and barotropic Rossby waves with a substantial midlatitude projection in the presence of suitable horizontally and vertically sheared zonal mean flows [8].

2. Exact solutions of (1) using the extended tanh method

In this section we obtain exact solutions of (1). For this purpose we use the extended tanh function method. This method was introduced by Wazwaz [10]. To the best of our knowledge this is the first time that the extended tanh method will be used to find the exact solutions of the system (1). Making use of the wave variable transformation $z = x - \nu t$, system (1) is reduced to the following nonlinear coupled system of ordinary differential equations:

$$-cF'(z) - \left(1 - \frac{1}{(2m+1)^2}\right) F'''(z) + G(z)F'(z) + 2F(z)G'(z) = 0, \tag{2a}$$

$$-cG'(z) + F(z)F'(z) - G'''(z) = 0. \tag{2b}$$

The basic idea in this method is to assume that the solution of (2) can be written in the form

$$F(z) = \sum_{i=-M}^{M} A_i H(z)^i, \tag{3a}$$

$$G(z) = \sum_{i=-M}^{M} B_i H(z)^i, \tag{3b}$$

where $H(z)$ satisfies an auxiliary equation, say for example the Riccati equation

$$H'(z) = 1 - H^2(z), \tag{4}$$

whose solution is given by

$$H(z) = \tanh(z).$$

The positive integer M will be determined by the homogeneous balance method between the highest order derivative and highest order nonlinear term appearing in (2). A_i, B_i are parameters to be determined.

In our case, the balancing procedure gives $M = 2$ and so the solutions of (2) are of the form

$$F(z) = A_{-2}H^{-2} + A_{-1}H^{-1} + A_0 + A_1 H + A_2 H^2, \tag{5a}$$

$$G(z) = B_{-2}H^{-2} + B_{-1}H^{-1} + B_0 + B_1 H + B_2 H^2. \tag{5b}$$

Substituting (5) into (2) and making use of the Riccati equation (4) and then equating the coefficients of the functions H^i to zero, we obtain the following algebraic system

of equations in terms of A_i and B_i $(i=-2,-1,0,1,2)$:

$$
\begin{aligned}
&-4\,B_2A_2-24\,\frac{A_2}{(2\,m+1)^2}+24\,A_2=0,\\
&-24\,\frac{A_{-2}}{(2\,m+1)^2}-4\,B_{-2}A_{-2}+24\,A_{-2}=0,\\
&-3\,B_1A_2-3\,B_2A_1-6\,\frac{A_1}{(2\,m+1)^2}+6\,A_1=0,\\
&-6\,\frac{A_{-1}}{(2\,m+1)^2}-3\,B_{-2}A_{-1}-3\,B_{-1}A_{-2}+6\,A_{-1}=0,\\
&2\,A_0B_{-2}-16\,\frac{A_{-2}}{(2\,m+1)^2}+2\,B_{-1}A_{-1}+2\,B_0A_{-2}\\
&-2\,cA_{-2}+16\,A_{-2}=0,\\
&2\,B_1A_1+2\,A_0B_2-16\,\frac{A_2}{(2\,m+1)^2}+2\,B_0A_2\\
&-2\,cA_2+16\,A_2=0,\\
&-2\,A_0B_{-2}+40\,\frac{A_{-2}}{(2\,m+1)^2}+4\,B_{-2}A_{-2}\\
&-2\,B_{-1}A_{-1}-2\,B_0A_{-2}-40\,A_{-2}+2\,cA_{-2}=0,\\
&-2\,B_1A_1+4\,B_2A_2-2\,A_0B_2+40\,\frac{A_2}{(2\,m+1)^2}\\
&-2\,B_0A_2+2\,cA_2-40\,A_2=0,\\
&-A_0B_{-1}+8\,\frac{A_{-1}}{(2\,m+1)^2}+3\,B_{-2}A_{-1}-B_{-2}A_1\\
&+3\,B_{-1}A_{-2}-B_0A_{-1}-B_1A_{-2}+cA_{-1}-8\,A_{-1}=0,\\
&3\,B_1A_2-B_2A_{-1}+3\,B_2A_1-A_0B_1\\
&+8\,\frac{A_1}{(2\,m+1)^2}-B_{-1}A_2-B_0A_1+cA_1-8\,A_1=0,\\
&B_0A_{-1}+B_0A_1+A_0B_{-1}+A_0B_1+B_2A_{-1}\\
&+B_{-2}A_1+B_{-1}A_2+B_1A_{-2}+2\,A_{-1}+2\,A_1-cA_{-1}\\
&-cA_1-2\,\frac{A_{-1}}{(2\,m+1)^2}-2\,\frac{A_1}{(2\,m+1)^2},\\
&-2\,{A_{-2}}^2+24\,B_{-2}=0,\\
&-2\,{A_2}^2+24\,B_2=0,\\
&-3\,A_{-2}A_{-1}+6\,B_{-1}=0,\\
&-3\,A_1A_2+6\,B_1=0,\\
&-2\,cB_{-2}+2\,A_{-2}A_0+{A_{-1}}^2+16\,B_{-2}=0,\\
&-2\,cB_2+2\,A_0A_2+{A_1}^2+16\,B_2=0,\\
&2\,cB_{-2}+2\,{A_{-2}}^2-2\,A_{-2}A_0-{A_{-1}}^2-40\,B_{-2}=0,\\
&2\,cB_2-2\,A_0A_2-{A_1}^2+2\,{A_2}^2-40\,B_2=0,\\
&cB_{-1}+3\,A_{-2}A_{-1}-A_{-2}A_1-A_0A_{-1}-8\,B_{-1}=0,\\
&cB_1-A_{-1}A_2-A_0A_1+3\,A_1A_2-8\,B_1=0,\\
&-cB_{-1}-cB_1+A_{-2}A_1+A_0A_{-1}+A_{-1}A_2\\
&+A_0A_1+2\,B_{-1}+2\,B_1=0.
\end{aligned}
$$

Solving the above system of algebraic equations, with the aid of Mathematica, one possible set of values of A_i and B_i $(i=-2,-1,0,1,2)$ are

$$
\begin{aligned}
&A_{-2}=48\,\frac{m\,(m+1)}{(2\,m+1)^3},\\
&A_{-1}=0,\\
&A_0=4\,\frac{m\,(m+1)\,(c-8)}{(2\,m+1)^3},\\
&A_1=0,\\
&A_2=48\,\frac{m\,(m+1)}{(2\,m+1)^3},\\
&B_{-2}=24\,\frac{m\,(m+1)}{(2\,m+1)^2},\\
&B_{-1}=0,\\
&B_0=\frac{2\,cm^2+2\,cm-16\,m^2+c-16\,m}{4\,m^2+4\,m+1},\\
&B_1=0,\\
&B_2=24\,\frac{m\,(m+1)}{(2\,m+1)^2},
\end{aligned}
$$

where α is any root of $\alpha^2-2\,m^2-2\,m=0$. As a result, a solution of (1) is

$$
\begin{aligned}
u(t,x) &= A_{-2}\coth^2(z)+A_{-1}\coth(z)+A_0+A_1\tanh(z)\\
&\quad +A_2\tanh^2(z), \qquad (6a)\\
v(t,x) &= B_{-2}\coth^2(z)+B_{-1}\coth(z)+B_0+B_1\tanh(z)\\
&\quad +B_2\tanh^2(z), \qquad (6b)
\end{aligned}
$$

where $z=x-ct$.

3. Conclusions

In this paper exact solutions of the Majda-Biello system were constructed. The method employed to construct the solutions was the extended tanh method. The travelling wave solutions were obtained.

Acknowledgments

The author would like to thank the North-West University, Mafikeng Campus for their financial support.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] R. Hirota, Exact solution of the Korteweg-de Vries equation for multiple collisions of solitons, Phys. Rev. Lett. 27 (1971) 1192-1194.
[2] M.R. Miura, Backlund Transformation, Springer, Berlin, 1978.
[3] J. Weiss, M. Tabor, G. Carnevale, The Painleve property for partial differential equations, J. Math. Phys. 24 (1983) 522-526.
[4] L.D. Moleleki, C.M. Khalique, Symmetries, Traveling Wave Solutions, and Conservation Laws of a (3+1)-Dimensional Boussinesq Equation, Advances in Mathematical Physics Volume 2014, Article ID 672679, 8 pages.
[5] M.L. Wang, Exact solution for a compound KdV-Burgers equations, Phys. Lett. A 213 (1996) 279-287.

[6] S.K. Liu, Z.T. Fu, S.D. Liu, Q. Zhao, Jacobi elliptic function expansion method and periodic wave solutions of nonlinear wave equations, Phys. Lett. A 289 (2001) 69-74.
[7] A.R. Adem, C.M. Khalique, Exact Solutions and Conservation Laws of a Two-Dimensional Integrable Generalization of the Kaup-Kupershmidt Equation,Journal of Applied Mathematics Volume 2013, Article ID 647313,
[8] A.J. Majda, J.A. Biello, The nonlinear interaction of barotropic and equatorial baroclinic Rossby waves, J. Atmospheric Sci. 60 (2003) 1809-1821.
[9] J.V. Jahnova, Symmetries and conservation laws of the Majda-Biello system,Nonlinear Analysis: Real World Applications 22 (2015) 148-154.
[10] A. M. Wazwaz, New solitary wave solutions to the Kuramoto-Sivashinsky and the Kawahara equations, Applied Mathematics and Computation 182 (2006) 1642-1650.

Effect of Cavity Shape on Flow and Heat Transfer Characteristics in Converging Pipes: A Numerical Study

Khalid N. Alammar

Department of Mechanical Engineering, King Saud University
PO Box 800, Riyadh 11421, Saudi Arabia
Tel.: +966 (01) 467-6650; fax: +966 (01) 467-6652
E-mail: alammar@ksu.edu.sa

Abstract

Using the standard *k*-ε turbulence model, a developing, two-dimensional, turbulent converging pipe flow was simulated with round and square cavities. The Reynolds number was fixed at 4.0×10^4. The Prandtl number was 0.74. The pipe inlet diameter was 2 cm and ratio of pipe length to inlet diameter was 40. The pipe was reduced to 1.75 cm at the outlet (convergence ratio). Cavities width and aspect ratio (width to depth) were 0.6 cm and 6, respectively. 34 cavities placed in series were used, resulting in cavity area-to-total area ratio of 0.26. Cavity depth-to-inlet diameter ratio was 0.0167. The simulation revealed circulation within the cavities. Cavity presence was shown to enhance overall heat transfer while increasing pressure drop along the pipe. Compared to a smooth straight pipe, convergence of the pipe resulted in 331-% increase in friction and 25-% enhancement in heat transfer. Adding round cavities, the friction factor increased by 641 % and the overall heat transfer was enhanced by 43 %. With square cavities, the friction factor increased by 572 % and the heat transfer increased by 35 %.

Keywords: Cavity, Friction Factor, Heat Transfer, Pipe Flow, Turbulent.

1. Introduction

Dimples have been shown to reduce drag over spheres and golf balls [1]. They promote turbulence, the fluctuations of which energize the boundary layer, resulting in separation delay and hence drag reduction. Similar effect has been observed when dimples are used on circular cylinders [2]. In addition to drag reduction over blunt bodies, dimples have been found to enhance heat transfer and mixing in both external and internal flows. They enhance mixing by promoting turbulence and by forming vortical flow structures that help entrain mainstream flow into the boundary layer.

Many studies have been conducted to investigate the effect of cavities on flow and heat transfer characters under various conditions. For example, Afanas'yev et al. [3] have demonstrated the merit of dimple arrays in augmenting heat transfer over a flat plate. Enhancements up to 40 % were observed with no friction increases. Chyu et al. [4] and Mahmood et al. [5] conducted similar investigations on channel flows. They reported up to 2.5 heat transfer enhancement, with friction factor increases up to 4. In yet another study, Mahmood et al. [6] studied heat transfer in a channel with dimples on one side and protrusions on the other. Because of additional vortical flows and flow structures induced by the protrusions, it was found that heat transfer was considerably augmented along with a considerable increase in the friction factor.

Bunker et al. [7] have investigated flow and heat transfer characteristics in straight rectangular channels and channels of converging and diverging cross-sectional area with concavities of different shapes, including hemispherical dimples, inverted-truncated cones, shallow cylindrical pits, and a combination of cone and cylindrical pit. Average channel Reynolds numbers of 5000, 12000, and 20000 were tested. Concavity depth-to-diameter ratios of 0.1 to 0.23 were used. Results show that hemispherical sector concavity arrays can achieve heat transfer enhancements of about 50% relative to smooth surfaces, with pressure loss increases of 25% or less. The inverted and truncated cone resulted in equal heat transfer enhancement with similar or less pressure loss. Other shapes of simplified geometries show lower heat transfer enhancements with higher pressure losses.

Nagoga [8] has investigated the effect of cavity geometry on friction and heat transfer characteristics on flow inside turbine blades. He has shown that shallow dimples enhance less, while the friction factor increases with depth. Bunker and Donnellan [9] have investigated the effect of dimple geometry and concentration on heat and flow characteristics in pipes. Dimple aspect ratios ranging from 2.5 to 5 were used with surface area concentrations ranging from 0.3 to 0.7. Deeper dimples were reported to enhance more, while concentration increased both heat transfer and pressure drop. Heat transfer enhancements up to two were reported, with associated friction factor increases up to six times.

Recently, Alammar [10] has numerically investigated the effect of aspect ratio on flow and heat transfer characteristics in pipes. Two aspect ratios (ratio of cavity width-to-depth) were used, namely 3 and 6. He reported that two small vortices formed at the corners of the shallow cavities, while one large vortex formed with the deep cavities. The shallow cavities were reported to enhance heat transfer more while increasing pressure drop more compared to the deep cavities.

Effect of cavity shape on flow and heat transfer characteristics in converging pipes, along with detailed measurements of the flow field has not been reported. Knowledge of the flow field is necessary to explain differences in flow and heat transfer characteristics with different cavities. Numerical simulation, therefore, can be utilized to predict the flow field and help explain the behavior of flow and heat transfer in presence of round and square cavities.

In this work, a steady, turbulent, developing, axisymmetric, converging pipe flow with arrays of round and square cavities was simulated. Effect of cavity shape on flow and heat transfer characteristics was investigated. The Reynolds number, based on inlet diameter, was fixed at 4.0×10^4. The Prandtl number was 0.74. The pipe inlet diameter was 2 cm and ratio of pipe length to inlet diameter was 40. The pipe diameter was reduced from 2.0 cm at the inlet to 1.75 cm at the outlet (convergence ratio). Cavities width and aspect ratio were 0.6 cm and 6, respectively. 34 cavities placed in series were used, resulting in cavity area-to-total area ratio of 0.26. Cavity depth-to-inlet diameter ratio was 0.0167. Cavities were created with sharp edges, and they were confined within the second half of the pipe. Schematic of the pipe and cavities is shown in Fig. 1.

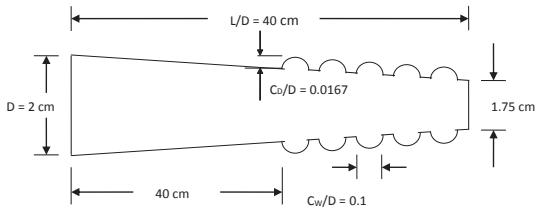


Fig. 1a: Schematic of the pipe and round cavities.

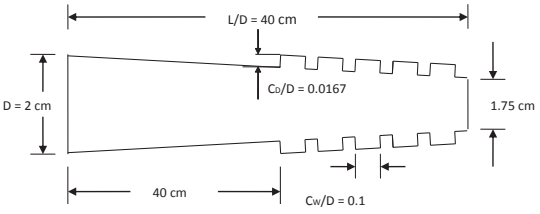


Fig. 1b: Schematic of the pipe and square cavities.

2. The mathematical model

The mathematical model consisted of the following steady Reynolds-averaged conservation equations in Cartesian tensor form:

a. Continuity equation

$$\frac{\partial(\rho U_i)}{\partial x_i} = 0 \tag{1}$$

b. Momentum equations

$$\frac{\partial(\rho U_i U_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial U_i}{\partial x_i}\right)\right] + \frac{\partial}{\partial x_j}\left(-\rho\overline{u_i u_j}\right) \tag{2}$$

For the Reynolds stresses, the Boussinesq hypothesis is invoked [15]:

$$-\rho\overline{u_i u_j} = \mu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \frac{2}{3}\left(\rho k + \mu_t\frac{\partial U_i}{\partial x_i}\right)\delta_{ij} \tag{3}$$

c. Energy equation

$$\frac{\partial[U_i(\rho E + p)]}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(k + \frac{c_p\mu_t}{\mathrm{Pr}_t}\right)\right]\frac{\partial T}{\partial x_j} \tag{4}$$

d. Turbulence model

The standard k-ε turbulence model by Launder and Spalding [16] was used. The model is well suited for high Reynolds number internal flows and has been validated for many industrial flows.

$$\frac{\partial(\rho k U_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + G_k - \rho\varepsilon \tag{5}$$

$$\frac{\partial(\rho\varepsilon U_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial\varepsilon}{\partial x_j}\right] + C_{1\varepsilon}\frac{\varepsilon}{k}G_k - C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} \tag{6}$$

Production of turbulence kinetic energy is given by

$$G_k = -\rho\overline{u_i u_j}\frac{\partial U_j}{\partial x_i}$$

And the Eddy viscosity is

$$\mu_t = \rho C_\mu\frac{k^2}{\varepsilon}$$

e. Wall treatment

The non-equilibrium wall function by Kim and Choudhury [17] was implemented. Reynolds Analogy was invoked for heat transfer. For more details, the reader is referred to Fluent 6.1 (commercial code) user guide.

3. The numerical procedure

Fluent 6.1 (commercial code) was used as the solver. The structured grid was built using Gambit 2.0. The simulation was carried out using SIMPLE [11], and second-order schemes. The linearized equations were solved using Gauss-Seidel method. The mesh for each case consisted of approximately 30,000 quadrilateral cells, Fig 2. Constant velocity of 30 m/s was applied at the inlet and zero gauge pressure was applied at the outlet. Turbulence intensity of 5-% and a hydraulic diameter of 0.02 m were applied at the inlet. A uniform temperature of 300 K was applied at the inlet, and the wall heat flux was fixed at 1000 W/m^2. Due to small overall temperature changes in the fluid, properties were assumed constant throughout.

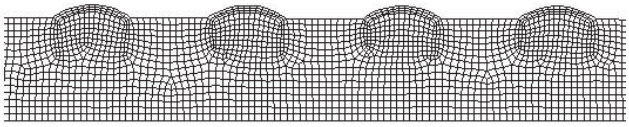


Figure 2a: A snap shot of the numerical grid near (round cavities.)

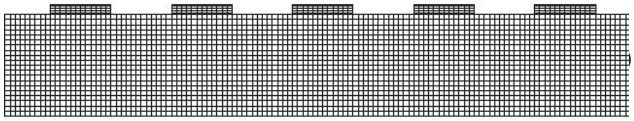


Figure 2b: A snap shot of the numerical grid near (square cavities.)

4. Uncertainty analysis

There are mainly two sources of uncertainty in CFD, namely modeling and numerical [12]. Modeling uncertainty is approximated through experimental

validation. The current code and numerical model was validated using measurements of Baughn et al. [13] at Re = $40x10^4$ and Pr = 0.74. In their experiment, Baughn et al. conducted heat transfer measurements at various Reynolds numbers through a circular pipe with abrupt expansion. Expansion ratio was 2.5. Due to expansion, circulation and separation was present. Similarly, circulation and separation are expected in the present investigation in presence of cavities. As in the present investigation, temperature changes in the experiment were small enough as not to incur significant property variations.

The Nusselt distribution for both measurement and prediction is depicted in Fig. 3. Here, the Nusselt number is normalized by the fully-developed value obtained using Dittus-Boelter correlation [14] and H is the step height. The numerical simulation matches the experiment within the experimental error of ± 5 % reported by Baughn et al. [13]. Hence, modeling uncertainty in this simulation is assumed to be ± 5 %.

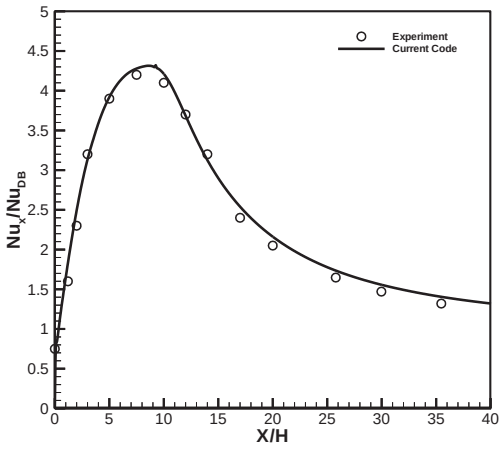


Fig. 3: Normalized Nusselt number at the wall.

Numerical uncertainty has two main sources, namely truncation and round-off errors. Higher order schemes have less truncation error, and as was outlined earlier, the discretization schemes invoked were second-order. In explicit schemes, round-off error increases with number of iterations, and is reduced by increasing significant digits (machine precision). However, having used Gauss-Seidel iterative procedure in a steady-state simulation renders the calculation insensitive to round-off error.

Numerical uncertainty can be approximated through grid independence. The Nusselt number surface distribution for a smooth converging pipe is shown in Fig. 4. The profiles for 5,000 and 30,000 cells overlap except for a narrow region near the entrance. Having used 30,000 cells in the current simulation, we conclude that the overall uncertainty is determined by the modeling uncertainty of ± 5 %.

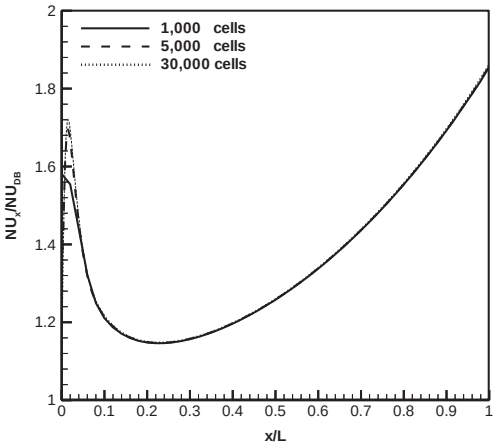


Fig. 4: Nusselt number distribution at the wall (Baseline).

5. Results and discussion

The Nusselt number distribution near the cavities is shown in Fig. 5. Overall enhancement of heat transfer with cavities is predicted. Due to two circulation regions within the square cavities, the Nusselt number dips twice. Compared to baseline, local heat transfer increases up to 2 times in both cases. The profiles are similar to those of turbulence production, suggesting that turbulence is the major mechanism in enhancing heat transfer.

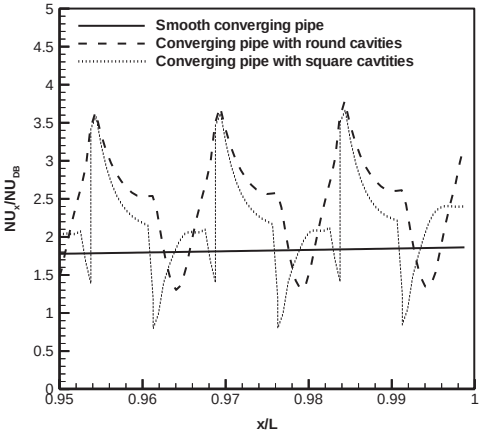


Fig. 5: Nusselt number distribution near cavities.

Velocity vectors within the cavities are depicted in Fig. 6. Circulation is predicted within the cavities. Two circulations are predicted within the square cavity, whereas only one vortex was predicted in the round cavity. Circulation, in general, can enhance heat transfer by entraining mainstream flow into the boundary layer. However, the circulation shown in Figure 6 is confined within the cavities and does not effectively entrain with the mainstream. This

is why turbulence was the major contributor to heat transfer enhancement.

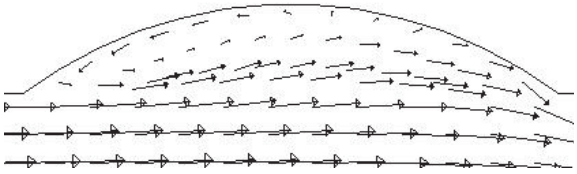


Fig. 6a: Circulation within a round cavity.

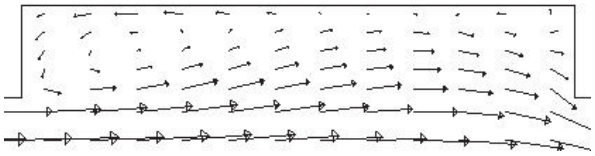


Fig. 6b: Circulation within a square cavity.

Effect of cavities on average friction factor and Nusselt number is shown in Table 1. For comparison, the table includes results for a smooth straight pipe. Convergence of the pipe resulted in 331-% increase in friction and 25-% enhancement in heat transfer, compared to a smooth straight pipe. Adding round cavities, the friction factor increased by 641 % and the overall heat transfer was enhanced by 43 %. With square cavities, the friction factor increased by 572 % and the heat transfer increased by 35 %. Finally, while convergence can contribute to heat transfer enhancement, adding cavities can enhance heat transfer further.

Table 1: Effect of cavity shape on average friction factor and Nusselt number.

case	f	change	Nu	change
Straight pipe	0.0058	NA	108	NA
Converging pipe	0.025	331 %	136	26 %
Round cavities	0.043	641 %	154	43 %
Square cavities	0.039	572 %	146	35 %

Conclusions

A developing, axisymmetric, steady turbulent converging pipe flow with round and square cavities was simulated. Effect of cavity shape on flow and heat transfer characteristics was investigated. The simulation revealed circulation within the cavities. It was shown that cavities can augment heat transfer through converging pipes while increasing pressure drop. Turbulence was shown to be the major contributor to heat transfer enhancement. Round cavities were shown to enhance heat transfer more than square cavities. However, pressure drop was more in the case of round cavities. Pipe convergence alone can enhance heat transfer and increase pressure drop compared to a straight pipe. This simulation shows that the standard k-ε turbulence model is capable of resolving average flow and heat transfer characteristics in converging pipes with cavities.

Nomenclature

A=surface area

C_p=specific heat at constant pressure, $kJ/(kg \cdot K)$

$C_{1\varepsilon}$=empirical constant = 1.44

$C_{2\varepsilon}$=empirical constant = 1.92

C_μ=empirical constant = 0.09

D=pipe inlet diameter, m

C_{AR}=cavity aspect ratio = $\frac{C_W}{C_D}$

C_W=cavity width, m

C_D=cavity depth, m

E=empirical constant = 9.8

E=total energy, J

f=friction factor = $\frac{\Delta P}{2(L/D)\rho U_{in}^2}$

h=heat transfer coefficient = $\frac{q}{T_w - T_{bulk}}$, $W/(m^2 \cdot K)$

k=turbulence kinetic energy, m^2/s^2

k=fluid thermal conductivity, $W/(m \cdot K)$

L=pipe length, m

Nu_x=local Nusselt number = $\frac{hD}{k}$

Nu=area-averaged Nusselt number = $\frac{1}{A}\int Nu_x dA$

Nu_{DB}=Nusselt number obtained by the Dittus-Boelter correlation

p=mean pressure of the fluid, Pa

Pr_t=turbulent Prandtl number

q=heat flux through the wall, W/m^2

Re=Reynolds number = $\frac{\rho U_{in} D}{\mu}$

T_w=wall temperature, K

T_{in}=inlet temperature, K

U=mean velocity of the fluid, m/s

U_{in}=inlet velocity, m/s

x_i=Cartesian coordinates

Greek Symbols

δ_{ij}=Kronecker delta

δ_k=empirical constant = 1.0

δ_ε=empirical constant = 1.3

ΔP=pressure drop through the pipe, Pa

ε=turbulence dissipation rate, m^2/s^3

κ=von Karman constant = 0.42

μ=dynamic viscosity of the fluid, $kg/(m \cdot s)$

μ_t=eddy viscosity, $kg/(m \cdot s)$

ρ=mean density of the fluid, kg/m^3

$\rho\overline{u_i u_j}$=Reynolds stresses, Pa

References

1 Bearman, P. W. and Harvey, J. K. Golf ball aerodynamics, Aeronautical Quarterly, Vol. 27, pp. 112-122, 1976.

2 Bearman, P. W. and Harvey, J. K. Control of circular cylinder flow by the use of dimples, AIAA Journal, vol.31 no.10, pp. 1753-1756, 1993.

3 Afnas'yev, V. N., Chudnovskiy, Ya. P., Leont'ev, A. I., and Roganov, P. S. Turbulent Flow Friction and Heat Transfer Characteristics for Spherical Cavities on A Flat Plate, Experimental and Thermal and Fluid Science, Vol. 7, pp. 1-8, 1993.

4 Chyu, M. K., Yu, Y., Ding, H., Downs, J. P., and Soechting, F. O. Concavity Enhanced Heat Transfer in an Internal Cooling Passage, IGTI Turbo Expo, Paper No. 97-GT-437, Orlando, 1997.

5 Mahmood, G. I., Hill, M. L., Nelson, D. L., Ligrani, P. M., Moon, H. K., and Glezer, B. Local Heat Transfer and Flow Structure on and above a Dimpled Surface in a Channel, IGTI Turbo Expo, Paper No. 2000-GT-230, Munich, 2000.

6 Mahmood, G. I., Sabbagh, M. Z., Ligrani, P. M. Heat Transfer in a Channel with Dimples and Protrusions on Opposite Walls, *J. Thermophysics and Heat Transfer*, Vol. 15, No. 3, pp. 275-283, 2001.

7 Bunker, R. S., Gotovskii, M., Belen'kiy, M., and Fokin, B. Heat Transfer and Pressure Loss for Flows Inside converging and Diverging Channels with Surface Concavity Shape Effects, Proceedings of the 4th International Conference on Compact Heat Exchangers and Enhancement Technology, Sep. 29-Oct. 3, Crete Island, Greece, 2003.

8 Nagoga, G. P. Effective Methods of Cooling of Blades of High Temperature Gas Turbines, Publishing House of Moscow, Aerospace Institute, 1996.

9 Bunker, R. S. and Donnellan, K. F. Heat Transfer and Friction Factors for Flows Inside Circular Tubes with Concavity Surfaces, GE Global Research, Technical Information Series, Paper No. 2002GRC351, 2002.

10 Alammar, K. N. Effect of cavity aspect ratio on flow and heat transfer characteristics in pipes: a numerical study, Heat and Mass Transfer, Online First, DOI: 10.1007/s00231-005-0054-x, 2005.

11 Patankar, S.V. and Spalding, D.B. A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three-Dimensional Parabolic Flows, International Journal of Heat and Mass Transfer, Vol. 15, p. 1787, 1972.

12 Stern, F., Wilson, R., Coleman, H., and Paterson, E. Verification and Validation of CFD Simulations, IIHR Report No. 407, Iowa Institute of Hydraulic Research, College of Engineering, The University of Iowa, Iowa City, IA, p. 3, 1999.

13 Baughn, J.W., Hoffman, M.A., Takahashi, R.K., and Launder, B.E. Local Heat Transfer Downstream of an Abrupt Expansion in a Circular Channel with Constant Wall Heat Flux, Journal of Heat Transfer, Vol. 106, pp. 789-796, 1984.

14 Kakac, S. and Yener, Y. Convective Heat Transfer, Second Edition, CRC Press, p. 292, 1995.

15 Hinze, J.O. Turbulence, McGraw-Hill Publishing Co., New York , pp. 403-415, 1975.

16 Launder, B.E. and Spalding, D.B. Lectures in Mathematical Models of Turbulence, Academic Press, London, England, pp. 90-110, 1972.

17 Kim, S. and Choudhury, D. A Near-Wall Treatment Using Wall Functions Sensitized to Pressure Gradient, ASME FED Vol. 217, Separated and Complex Flows, 1995.