Jurnal Teknologi, 54 (Sains & Kej.) Keluaran Khas, Jan. 2011: 381–402

INTEGRAL TIME ABSOLUTE ERROR MINIMIZATION FOR PI CONTROLLER ON COUPLED-TANK LIQUID LEVEL CONTROL SYSTEM BASED ON STOCHASTIC SEARCH ECHNIQUES

AMMAR HUSSAM YOUSIF YACOUB[1], SALINDA BUYAMIN[2*], & NORHALIZA ABDUL WAHAB[3]

Abstract. This paper introduces the application of a stochastic search technique, known as Simulated Annealing to the problem of tuning the proportional-integral controller for a linearized coupled tank liquid level control. After describing the basic principles of the Simulated Annealing, the proposed method concentrates on finding the optimal solution of PI controller by optimizing the performance index, the Integral Time Absolute Error, ITAE. The efficiency of Simulated Annealing algorithm for tuning the controller is compared with an evolutionary method, Genetic Algorithm. The comparison is based on the time response performance. The results shows the effectiveness and the capability of the SA to tune the proportional-integral (PI) controller for the coupled tank liquid level control. The proposed method does not depend on the system order and has the ability to tune the controller even there is unknown process parameters. In addition, the technique avoids the requirement for mathematical modeling of the system and the overall results have shown that SA yields better performance as compared to GA, hence, it is recommended for an alternative for optimizing the PI controller.

Keywords: Proportional integral tuning; simulated annealing; genetic algorithm

Abstrak. Kertas ini membincangkan tentang aplikasi teknik pencarian *stochastic* yang dikenali sebagai *Simulated Annealing* bagi mengatasi masalah menala *(tuning)* alat pengawal *Proportional plus Integral* (PI) bagi pengawalan takat cecair di dalam sistem tangki berkembar. Setelah penerangan tentang prinsipal asas kepada Simulated Annealing diberikan , kertas ini mencadangkan tentang pencarian penyelesaian yang optimal bagi alat kawalan PI dengan mengoptimumkan prestasi indek, ITAE. Keberkesanan menggunakan kaedah Simulated Annealing telah dibandingkan dengan satu lagi kaedah iaitu Genetic Algorithm. Perbandingan adalah berdasarkan prestasi *time response*. Hasil keputusan menunjukkan perbandingan antara kaedah Simulated Annealing dan Genetic Algorithm. Kaedah yang di cadangkan tidak bergantung kepada tahap sesuatu sistem dan berupaya untuk menala walaupun tanpa diketahui parameter sesuatu proses . Di samping itu, kaedah yang di cadangkan tidak memerlukan sistem tersebut di modelkan dalam bentuk matematik dan keseluruhan keputusan menunjukkan Simulated Annealing menghasilkan keputusan yang lebih baik dari Genetik Algorithm. Oleh itu, Simulated Annealing bolehlah dicadangkan sebagai salah satu cara bagi mengoptimumkan alat kawalan PI.

Kata kunci: Penalaan proportional integral tuning; imulated annealing; genetic algorithm

[1-3] Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor Darul Ta'azim, Malaysia
* Corresponding author: salinda@fke.utm.my

1.0 INTRODUCTION

Many industrial applications are concerned with level control, may it be a single loop level control or sometimes multi-loop level control. Interacting tanks is an example of such applications. Hence, level control is one of the control system variables which are very important in process industries [1]. There are various numbers of control strategy and methods in controlling the liquid level in the coupled-tank system and some of these researches are: Hybrid control system that consisting of a PID controller and a time optimal controller [2], Nonlinear back stepping liquid level controller [3], Multivariable MIMO controller strategy [4], Sliding mode controller [5], Neuro fuzzy controller and ANFIS controller [6, 7]. Among these methods, there are one method that is widely used which is Proportional plus Integral (PI) controllers which offer for the functionality of liquid level control systems with moderate performance specifications [3].

There are a lot of works related about stochastic search techniques in control systems in particular the Simulated Annealing (SA). SA has been widely used in large combinatorial optimization problems due to its ability of finding the global optimum with a high probability even for ill conditioned functions with numerous local optima [8] such as parameter optimisation [9], controller optimisation [10, 11] and filter tuning [12] although it usually requires a large number of function evaluations to find the optimum solution [8].

In [13], SA is used for the optimal tuning of a Proportional plus Integral plus Derivative (PID) controller to deal with time-varying delay. The PID control system adjusts the command feed to regulate the drilling force through a multipoint interface network using the computational resources of computerized numerical control. The author claimed that the controller parameters obtained on the basis of SA provide better transient response and a better performance than any other methods.

Also in [11], SA is claimed to be an effective way to determining the appropriate parameters of electric drive speed controllers. This stochastic search technique avoids the requirement for mathematical modeling of the drive and its load and is able to deal with the inherent non-linearity of the system. In [14], SA has been used to tune the observer for the induction motor drives and have shown that the SA have the potential to be an alternative method for optimizing the Extended Kalman Filter as compared to GA [15]. While in [16], the GA-based and SA-based optimal tuning techniques are used to optimize the parameter settings of the robot arm PID controllers. The results have shown that the GA-based and SA-based optimal-tuning techniques can work effectively and efficiently and have a great potential to become an optimal-tuning approaches for the robot arm controllers.

This paper is organized in five sections. Following this introduction, a brief overview of the coupled tank system used in this work is provided. Then the stochastic method and the methodology of using SA and GA are discussed. Next, the performance of the proposed method and the comparison with another method is illustrated and summarized and followed by the conclusion of the work.

2.0 COUPLED-TANK SYSTEM

In this work, the KRi Coupled-Tank Control Apparatus PP-100 is used as a low-cost pilot plant. It is designed for laboratory teaching of both introductory and advanced control systems theory. A schematic diagram of the coupled-tank apparatus is shown in Figure 1.

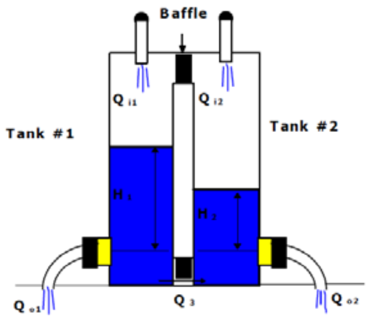

Figure 1 The schematic diagram of coupled tank apparatus

H_1, H_2 = Height of liquid in tank 1 and tank
A_1, A_2 = Cross-sectional area of the tanks.
Q_{i1}, Q_{i2} = Pump flow rate into tank 1 and tank 2.
Q_{o1}, Q_{o2} = Flow rate of liquid out of tank 1 and tank 2
Q_{o3} = Flow rate of liquid between tanks.

For each tank 1 and tank 2, the dynamic equation is developed as follows [1]:

$$A_1 \frac{dH_1}{dt} = Q_{i1} - \alpha_1\sqrt{H_1} - \alpha_3\sqrt{H_1 - H_2} \quad (1)$$
$$A_2 \frac{dH_2}{dt} = Q_{i2} - \alpha_2\sqrt{H_2} + \alpha_3\sqrt{H_1 - H_2} \quad (2)$$

Where parameters α_1, α_2, α_3 are proportionality constants which depend on the coefficients of discharge.

For a set of inflows Q_{i1} and Q_{i2}, the liquid level in the tanks is at some steady state levels H_1 and H_2. Consider small variations in each inflow, q_1 in Q_{i1} and q_2 in Q_{i2}. The resulting perturbation in levels is h_1 and h_2 respectively. From equations (1) and (2), the linearized perturbations equations can be derived [1]:

$$A_1 \frac{dh_1}{dt} = q_1 - q_{o1} - \frac{\alpha_3}{2\sqrt{H_1 - H_2}}(h_1 - h_2) \quad (3)$$

$$A_2 \frac{dh_2}{dt} = q_2 - q_{o2} + \frac{\alpha_3}{2\sqrt{H_1 - H_2}}(h_1 - h_2) \quad (4)$$

The valve/pump actuator can be modeled using the following differential equation describes the valve/pump actuator's dynamics [17], [18].

$$\tau_a \frac{dq_i(t)}{dt} + q_i(t) = Q_c(t) \quad (5)$$

Where, τ_a is the time constant of the valve/pump actuator, $q_i(t)$ is the time-varying input flow rate and $Q_c(t)$ is the computed or the commanded flow rate. The objective of the system is to control the liquid level in Tank 2 by controlling the flow rate of the liquid into Tank 1. For simplification, assuming q_2 and Q_{i2} to be zero.

3.0 STOCHASTIC SEARCH TECHNIQUE

Stochastic search is a method of solving many hard combinatorial problems. Stochastic search can be defined as "a method that makes use of random numbers and is able to find good solutions within reasonable time without guaranteeing the optimum" [19] where near optimal solutions are sufficient for most engineering tasks [20].

3.1 Simulated Annealing

Simulated Annealing (SA) which chooses their path randomly through the design space has been successfully applied to many system optimization problems. SA is viable approaches to finding optimal, or near optimal solutions for large scale problems. The attractive feature of SA is that it is very easy to program and the algorithm typically has few parameters that require tuning.

Simulated Annealing (SA), which has much in common with evolutionary computation, is a derivative-free stochastic search method for determining the optimum solution in an optimization problem. The method was proposed earlier [21] and has since been used extensively to solve large-scale problems of combinatorial optimization, such as the well-known traveling salesman problem (TSP), the design of very large scale integration (VLSI) circuitry and in the design of optimum controllers. The main difference between evolutionary computation and SA is that the latter is inspired by the annealing process for metals during cooling, while the former is based on evolutionary processes.

The principle of annealing is simple: at high temperatures the molecules in a metal move freely but as the metal is cooled gradually this movement is reduced and atoms align to form crystals. This crystal-line form actually constitutes a state of minimum energy. Metals that are cooled gradually reach a state of minimum energy naturally, while if they are forcibly cooled they reach a polycrystalline or amorphous state whose energy level is significantly higher. Metals that are annealed are pliable while the latter are brittle.

However, even at low temperatures there exists a small, but finite probability that the metal will enter a state of higher energy. This implies that it is possible that the metal will leave the state of minimum energy for a new state where the energy is increased. During the cooling process, the intrinsic energy may rise or drop but as the temperature is lowered the probability that the energy level will increase suddenly is reduced.

The probability that a change in the state of the metal at some temperature T and initial energy level E_1 to some other state with energy level E_2 is given by:

$$P = \left\{ \begin{matrix} e^{-\frac{(E_2 - E_1)}{kT}} & if\ E_2 > E_1 \\ 1 & otherwise \end{matrix} \right\} \quad (6)$$

Where κ is Boltzmann's constant.

This thermodynamic principle was adapted to numerical analysis by Metropolis et al. in 1953 giving rise to the terms SA. SA attempts to minimize energy. This is similar to minimizing a Lyapunov function in modern control theory. In implementing the Metropolis algorithm the following must be known:

- The objective function **ϕ** (by analogy with the energy E of the metal) whose minimum is sought and,
- A control parameter **T** (the simulated temperature) whose temporal strategy defines the changes in the simulated temperature at every iteration of the algorithm.

Although the analogy between the physical annealing process and SA is far from perfect, there is clearly much in common. In all the stochastic algorithms of optimization, an attempt is made to guide the solution from an initial random point to an optimum solution as rapidly as possible. This can lead to entrapment in some local optimum from which it may be difficult if not impossible to extricate. SA does not suffer this problem, since the technique is stochastic and searches the solution space randomly.

The flow chart of the SA is shown in Figure 2. For the solution of an optimization problem with SA, the following steps are required:

(1) An initial random solution vector x_1 in the bounded parameter space is selected and its objective function $\varphi(x_1)$ is computed,

(2) An initial temperature $T(0)= T_{init}$ is specified,

(3) Using some stochastic or heuristic strategy, a new solution vector x_2 is selected and the corresponding objective function value is evaluated $\varphi(x_2)$,

(4) The difference of the objective function $\Delta\varphi = \varphi(x_2)$-$\varphi(x_1)$ is computed,

(5) If $\Delta\varphi < 0$, then the solution vector x_2 is accepted, otherwise if $\Delta\varphi > 0$ accept the solution vector according to the *probability of acceptance:*

$$p(k) = e^{-\frac{\Delta\varphi}{t(k)}}$$

otherwise go to step 7,

(6) Set $x_1 = x_2$ and $\phi(x_1)=\phi(x_2)$ and weight the current simulated temperature with the coefficient λ, where $0<\lambda<1$, decreasing the simulated temperature successively at every iteration, so that at the $(k+1)st$ iteration:

$$T(k+1) = \lambda T(k)$$

where k is the iteration index,

(7) If the current simulated temperature is lower or equal to the final temperature, i.e., $T(k) \leq T_{final}$, then accept the current solution vector as being *optimum*, otherwise return to Step 3 and repeat the process.

If the SA algorithm is to succeed, it is important that the temporal annealing strategy that is followed, i.e., the simulated temperature profile, be suitable. The rate at which the simulated temperature is decreased depends on the weighting coefficient λ. If it is too high a simulated cooling rate leads to non-minimum energy solutions, while if it is too low a cooling rate leads to excessively long computation times. The closer the value of λ is to unity, the slower simulated temperature decreases. In order to achieve effective exploration of the search space, it is advisable to use $0.95 < \lambda < 0.98$. Finally, as in Evolutionary Computation, the trajectory of an optimization problem is critically dependent on the initial estimates of the optimum solutions that are heuristic or the result of statistical analysis [20], [22], [23], [24], [25].

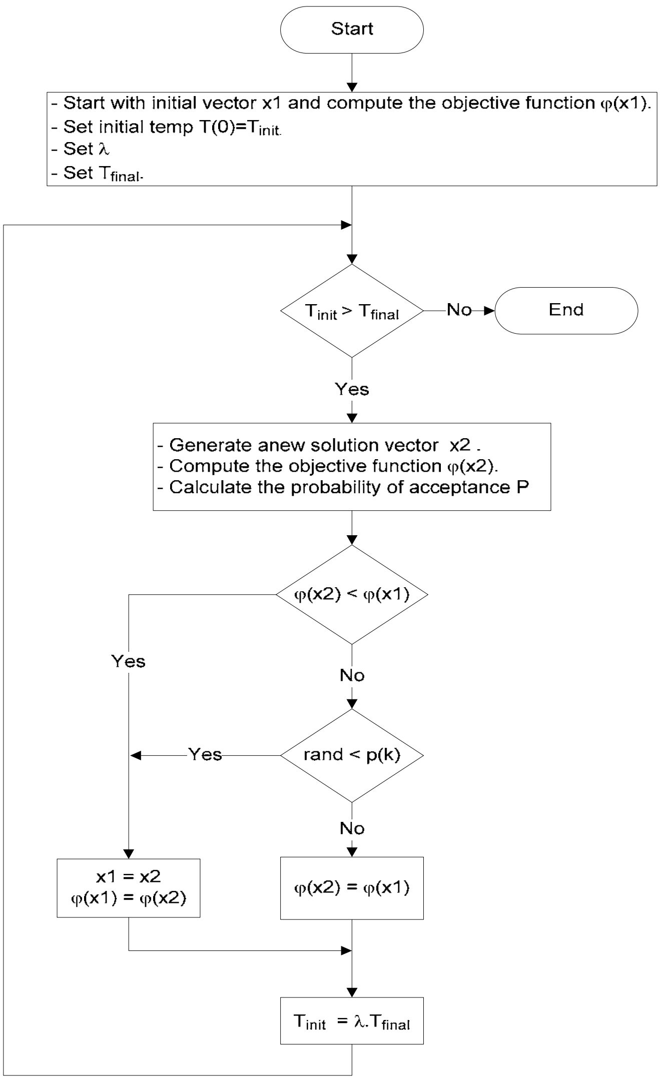

Figure 2 Flow chart of the SA algorithm

3.2 Genetic Algorithm

Genetic algorithm (GA) is a direct random search technique to find a global optimal solution in a complex search space. It was first invented in 1970's by Holland [26]. GA is modeled on the natural biological evolution process. It operates on a population of potential solutions or individuals over several generations to gradually improve on their fitness. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain, and breeding them together using genetic operators.

The potential solution for a problem is an individual known as a chromosome. The chromosomes can be represented by strings of numbers, normally but not necessarily, binary numbers. After decided on the chromosome representation, it is possible to access the performance and filtering the individual members of a population. The individuals are evaluated according to the objective and fitness function. The best individuals are selected to mate and generate offspring. Then, a new generation is created and the best fit individuals are selected to replace the least fit individuals of the previous generation while keeping the same population size. Through an iterative process, the population evolves towards better regions of the search space. The algorithm then converges to the best chromosome which represent the optimal or near optimal solution for the problem.

Given a clearly defined problem to be solved and a binary string representation for candidate solutions, the pseudo code for a GA and the flowchart for the algorithm are presented in a basic as in Figure 3 and 4. GA applies the following major steps:

(1) Represent the problem variable domain as a chromosome of a fixed length, choose the size of a chromosome population N, the crossover probability p_c and the mutation probability p_m.

(2) Define a fitness function to measure the performance, or fitness, of an individual chromosome in the problem domain. The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

(3) Randomly generate an initial population of chromosomes of size N:

$$x_1, x_2, \ldots, x_N$$

(4) Calculate the fitness of each individual chromosome:

$$f(x)_1, f(x)_2, \ldots, f(x_N)$$

(5) Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness. Highly fit chromosomes have a higher probability of being selected for mating than less fit chromosomes.

(6) Create a pair of offspring chromosomes by applying the genetic operators -crossover and mutation.

(7) Place the created offspring chromosomes in the new population.

(8) Repeat Step 5 until the size of the new chromosome population becomes equal to the size of the initial population, N.

(9) Replace the initial (parent) chromosome population with the new (offspring) population.

(10) Go to Step 4, and repeat the process until the termination criterion is satisfied.

```
begin
        t = 0
        initialize P(t)
        evaluate P(t)
        while (t < max # of generations) do
        begin
```

Figure 3 Pseudo code of GA given by Chipperfield [27]

4.0 SIMULATION RESULTS

The PI controller of coupled tank liquid level control is simulated using SA so that the ITAE is minimized. The following parameters had been used for simulation as shown in Table 1 which was tested experimentally in previous work [17].

Table 1 Coupled-Tank system parameters

Name	Expression	Value
Cross Sectional Area of the coupled tank reservoir	A1&A2	32 cm^2

Continued **Table 1**

		α_1	α_2	α_3
Proportionality α constant that depends on discharge coefficient, orifice cross sectional area and gravitational constant	subscript *i* denotes which tank it refers	14.30 $cm^{3/2}/sec$	14.30 $cm^{3/2}/sec$	20.00 $cm^{3/2}/sec$
Maximum allowable volumetric flow rate pumped by motor	Qi_{max}	300 cm^3/s		
Pump motor(valve) time constant	TC	1 sec (can be adjusted)		

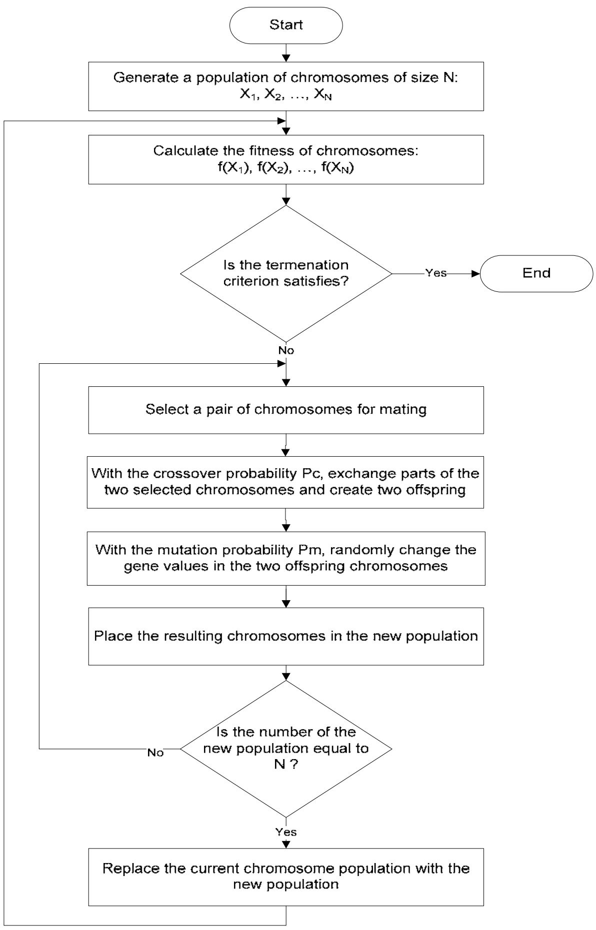

Figure 4 Genetic algorithm architecture

To carry out the simulation, the control system is implemented in Matlab software ver. 2008a. The linearized model has been shown in Figure 5,6 respectively.

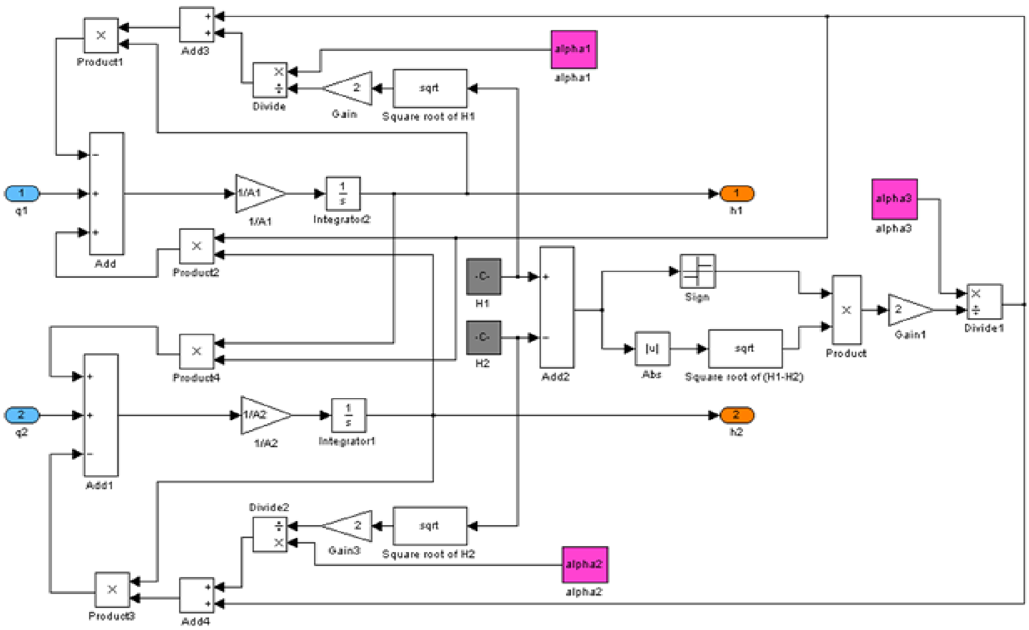

Figure 5 Linearized Model of coupled tank

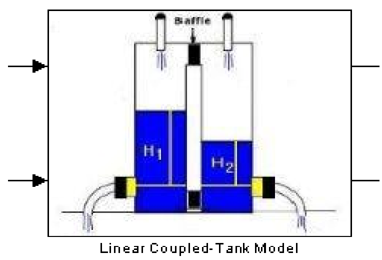

Figure 6 The mask of the model

4.1 Simulated Annealing Results

The SA program was done in Matlab as m-file. In order to run the program we need to initialize some of the program parameters as shown in Table 3.

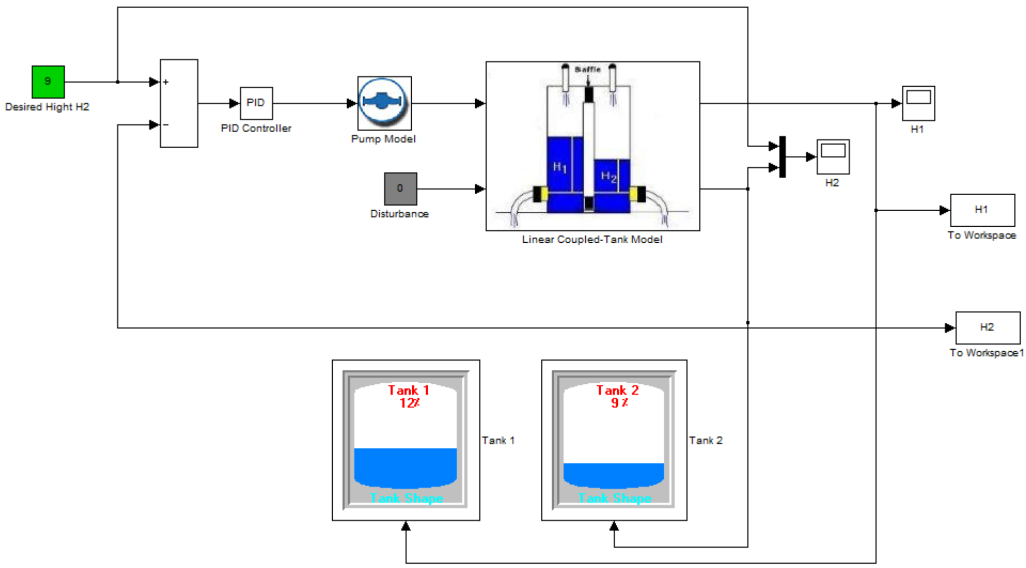

Figure 7 A complete modeling of coupled tank

Table 3 Simulated Annealing initial parameters

T_{init}	10
T_{final}	0.00001
λ	0.98
K_p max limit	30
K_i max limit	5

The SA program used Integral Time Absolute Error **ITAE** as the objective function $\boldsymbol{\phi}$ in order to optimize the PI control parameters K_p and K_i.

$$ITAE = \emptyset(t) = \int_0^\infty t.|e(t)|dt \qquad (6)$$

Where:
e(t) = error signal at time t.

SA program has work successfully to minimize the objective function in order to control the Tank 2 level for the system as shown in Figure 8 and 9.

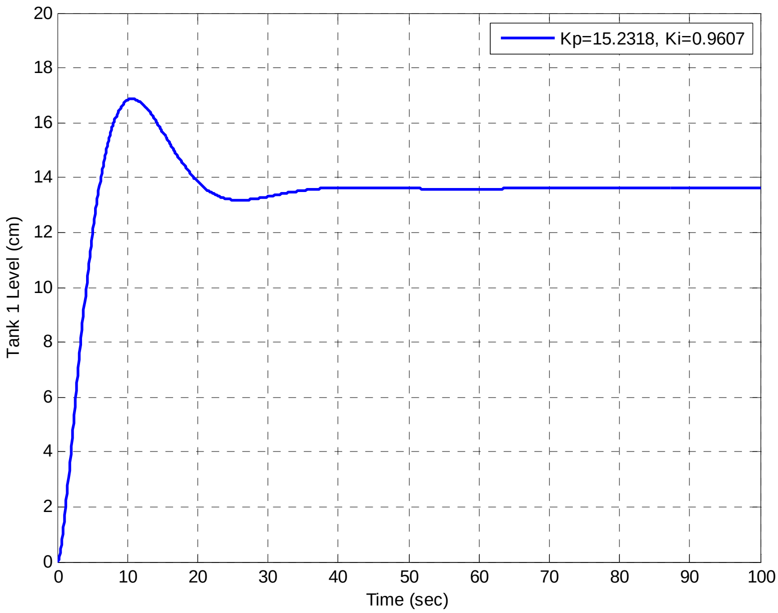

Figure 8 Performance of Tank 1 using SA

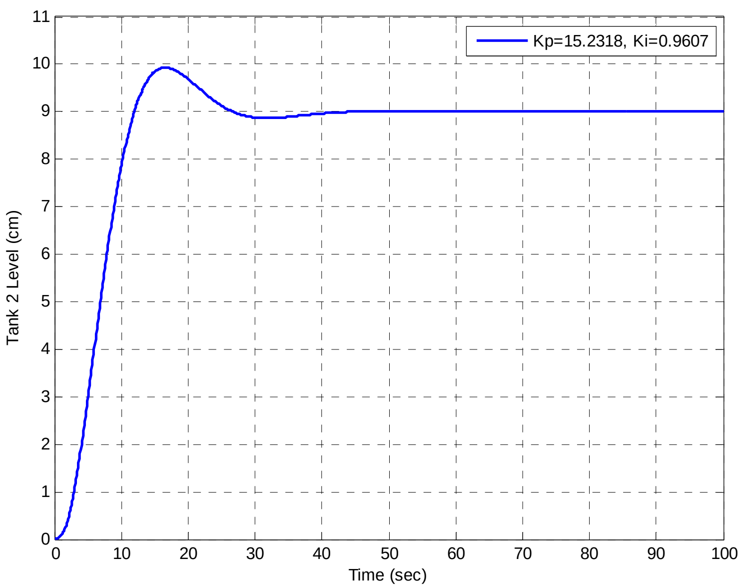

Figure 9 Performance of Tank 2 using SA

SA use random number generators so each time the program run, this algorithm may give different results [28]. The trajectory of the stochastic search as in Figure 10 shows the randomly generated numbers in two dimensions of K_p and K_i. The value for the optimum PI controller parameter is shown as in Figure 11.

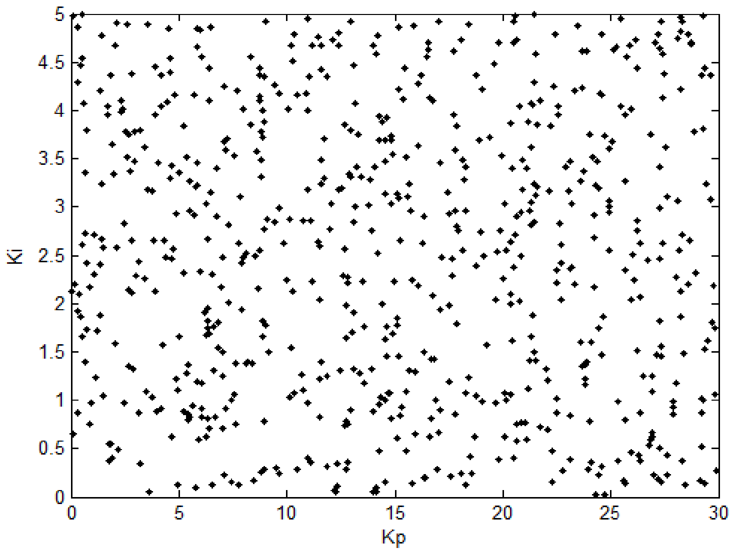

Figure 10 The trajectory of the stochastic search for the system

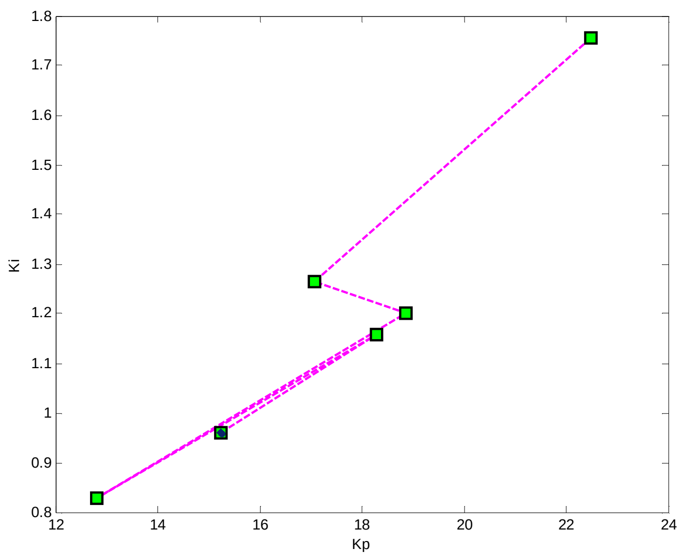

Figure 11 Detail of controller gain, K_p and K_i using SA

For the SA, when the stochastic search iteration increases, it gives lower objective function (ITAE) as shown above in Figure 12. These results prove that the SA worked well to minimize the objective function in order to find near optimum values for both K_p and K_i. Table 4 shows the Tank 2 performance results of the system.

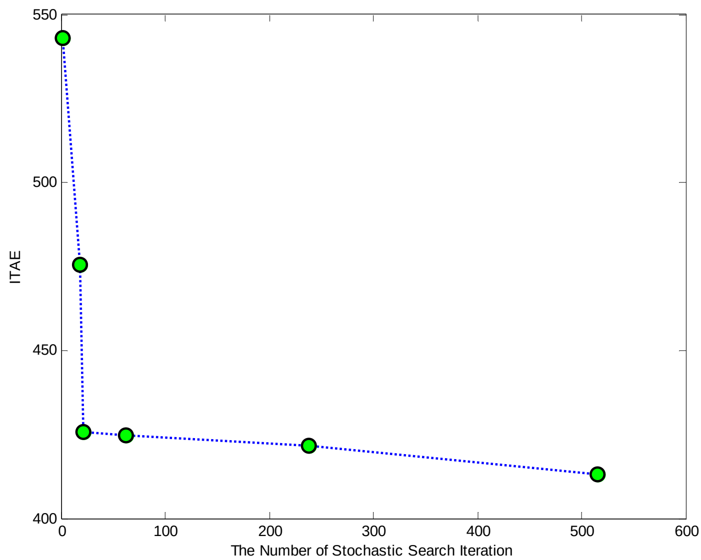

Figure 12 Reduction of ITAE during SA

Table 4 Tank 2 performance results using SA

Tank 2	**Trial 1**	**Trial 2**	**Trial 3**
Kp = Ki =	14.2439 0.9264	15.2318 0.9607	16.0806 0.9859
Rise Time	7.62 sec	7.21 sec	6.92 sec
Peak Amplitude	1.1	1.11	1.12
Setting Time	24.4 sec	31.4 sec	32.4 sec
Steady State Error	0	0	0
ϕ	415.7655	413.0642	415.8450

4.2 Genetic Algorithm Results

GA program has worked successfully to minimize the objective function in order to control the Tank 2 level of the system as shown in Figure 13 and Figure 14. GA use initial random numbers for each solution (chromosome) therefore each time the program run this algorithm may obtain slightly different results; therefore we take three sample runs to analyze. Table 5 shows the Tank 2 performance results of the system.

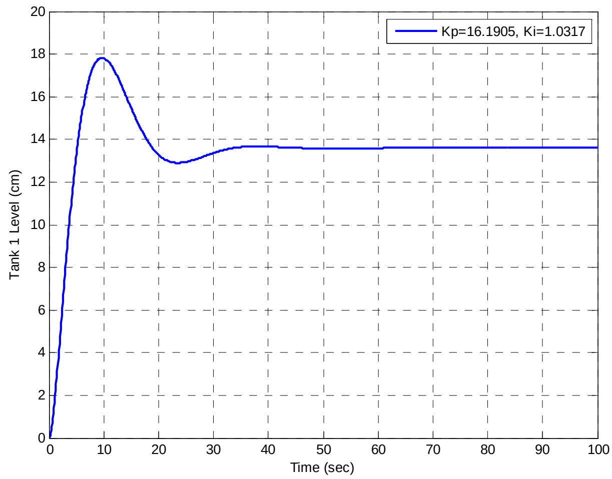

Figure 13 Tank 1 response obtained during GA tuning

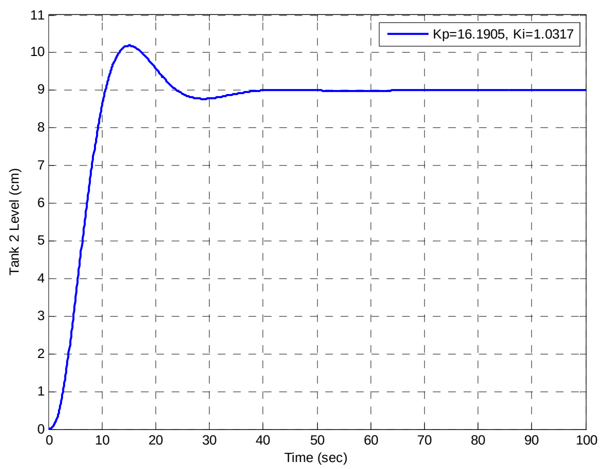

Figure 14 Tank 2 response obtained during GA tuning

Table 5 Tank 2 performance results using GA

Tank 2	**Trial 1**	**Trial 2**	**Trial 3**
K_p = K_i =	16.1905 1.0317	17.619 1.1111	15.2381 0.9524
Rise Time	6.82 sec	6.34 sec	7.23 sec
Peak Amplitude	1.13	1.15	1.11
Setting Time	31.8 sec	31.6 sec	31.8 sec
Steady State Error	0	0	0
Φ	413.9913	417.8844	414.2084

The trajectory of the stochastic search using GA shows the randomly generated numbers in two dimensions of K_p and K_i. The search area and the final value obtained using GA is shown as in Figure 15 and Figure 16.

The GA initializes a random number of chromosomes inside the population and these are decoded to be in the same limit of proportional and integral constrains as SA, then the chromosomes will be changed during generation process depending on crossover and mutation operation. The results show that the objective function (ITAE) reduced during the procedure of new generation (stochastic search) but settled on constant value for many generation and requires more iteration than SA before it reduced again.

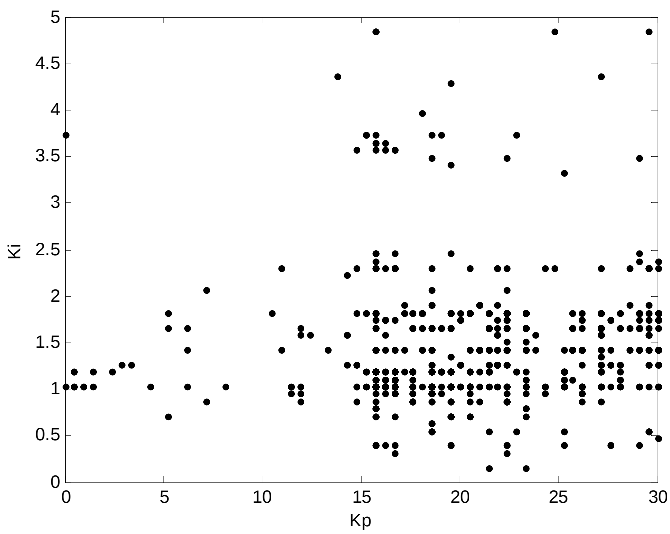

Figure 15 The trajectory of the search using GA

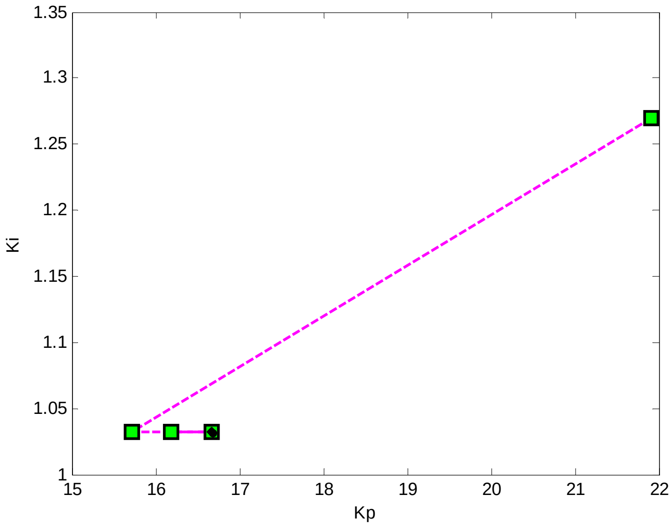

Figure 16 The controller gain, K_p and K_i using GA

The reduction of objective function during each success for GA can be seen in Figure 17.

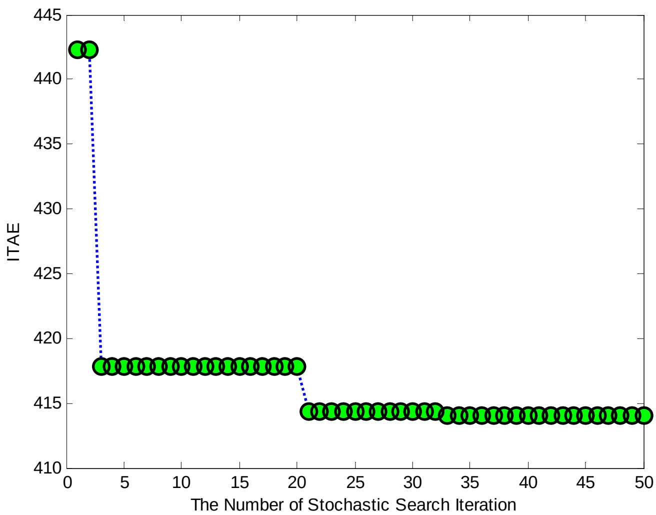

Figure 17 ITAE reduction using GA

These results demonstrate that the GA worked as well as SA to minimize the objective function in order to find near optimum values for both K_p and K_i but with longer computational time.

4.3 Comparison Between Simulated Annealing and Genetic Algorithm Tuning Methods

For SA, it is clear that the PI controller parameter for each alteration will be assumed randomly inside the limit of proportional and integral constraints. On the other hand , GA initializes a random number of chromosomes inside the population and these are decoded to be in the same limit of proportional and integral constraints as SA, then the chromosomes will be changed during generation process depending on crossover and mutation operation. In summary, Figure 18 show Tank 2 response for PI controller parameters obtained from SA and GA and the result is tabulated as in Table 6.

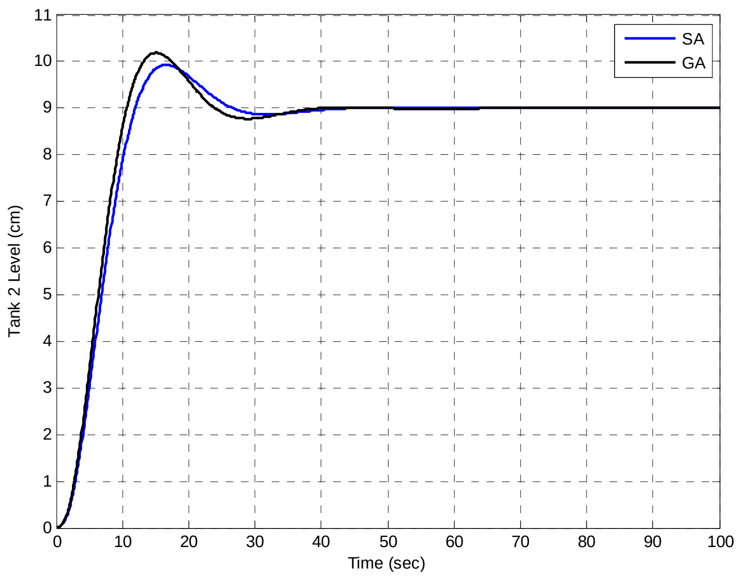

Figure 18 Tank 2 response of using SA and GA

From the result presented, it is shown that both SA and GA capable of providing good K_p and K_i value for the controller. The gain tuned by SA however gives better results in terms of settling time and rise time. SA used shorter iteration time and therefore reduces the overall simulation time. Although the ITAE provided by SA tuning is only slight different with GA, it gives better response in the time response performance.

Table 6 Comparison result using SA and GA

	SA	**GA**
K_p =	15.2318	16.1905
K_i =	0.9607	1.0317
Rise Time	7.21 sec	6.82 sec
Peak Amplitude	1.11	1.13
Settling Time	31.4 sec	31.8 sec
Steady state error	0	0
Iteration	685	1000
ϕ (ITAE)	413.0642	413.9913
Simulation Time	46.19 sec	69.98 sec

5.0 CONCLUSION

SA provides near optimum values and might give different values at each program run due to the algorithm structure of SA depends on generation of random values. The technique avoids the requirement for mathematical modeling of the system. For this approach, the Simulated Annealing has maintained the same values for its parameter for each test run. In dealing with the searching space, the bigger the search area for SA, the longer time is required for the algorithm to converge to optimum solution. The offline tuning done by simulation also gives the possibility of smaller area need to be identified which contribute to reduce computation time. The result obtained during offline tuning can be used for a set guide in searching space for the online tuning.

The paper has demonstrated the effectiveness of the SA to tune the proportional-integral (PI) controller for the coupled tank liquid level control. Since GA is one of the powerful tools in optimization, it is chosen for comparison purposes to verify the effectiveness of the proposed method. SA and GA, exhibit the capability to tune the system successfully. Both methods have the ability to deal with time delay systems, because its objective is to minimize the objective function, Integral Time Absolute Error (ITAE). Another advantage of SA and GA is that the tuning does not depend on the system order and has the ability to tune the controller even there is unknown process parameters.

In contrast, GA is more complex to construct than the SA. In addition, GA convergence depends on initial population and will require more generation in order to find the optimum value. The overall results shown that SA yields better performance as compared to GA, hence, it is recommended for an alternative for optimizing the PI controller. The hybrid algorithm of SA and GA can be applied to the system in order to overcome the drawbacks of these two methods. Another suggestion is to used the simplified SA which has claimed to performed a lot better and of reducing the computational complexity.

REFERENCES

[1] U. Centre for Artificial Intelligence and Robotics (CAIRO). *Coupled tank Liquid Level Computer Controlled, Laboratory Teaching Package, Experimental and Operations Service Manual.* Augmented Innovation Sdn. Bhd.

[2] J. Malmborg and J. Eker. 1997. Hybrid Control of a Double Tank System. In Control Applications. Proceedings of the 1997 IEEE International Conference. 133-138.

[3] H. Pan, H. Wong, V. Kapila, and M. S. de Queiroz. 2005. Experimental Validation of a Nonlinear Backstepping Liquid Level Controller for a State Coupled Two Tank System. *Control Engineering Practice.* 13: 27-40.

[4] B. Stenlund and A. Medvedev. 2002. Level Control of Cascade Coupled Flotation Tanks. *Control Engineering Practice.* 10: 443-448.

[5] N. B. Almutairi and M. Zribi. 2006. Sliding Mode Control of Coupled Tanks. *Mechatronics.* 16: 427-441.

[6] S. T. Lian, K. Marzuki, and Y. Rubiyah. 1998. Tuning of a Neuro-Fuzzy Controller by Genetic Algorithms with an Application to a Coupled-Tank Liquid-Level Control System. *Engineering Applications of Artificial Intelligence.* 11: 517-529.

[7] S. N. Engin, J. Kuvulmaz, and V. E. Ömürlü. 2004. Modeling of a Coupled Industrial Tank System with ANFIS. In MICAI 2004: Advances in Artificial Intelligenc. 804-812.

[8] R. Bhuvaneswari and S. Subramanian. 2005. Optimization of Three-Phase Induction Motor Design Using Simulated Annealing Algorithm. *Electric Power Components and Systems.* 33: 947-956.

[9] F. Ares, S. R. Rengarajan, E. Villanueva, E. Skochinski, and E. Moreno. 1996. Application of Genetic Algorithms and Simulated Annealing Technique in Optimising the Aperture Distributions of Antenna Array Patterns. *Electronics Letters.* 32: 148-149.

[10] A. H. Mantawy and M. M. Negm. 2002. A simulated Annealing-Based Optimal Controller for a Three Phase Induction Moto. In Power System Technology. Proceedings. PowerCon 2002. International Conference. 2: 50-755.

[11] P. Acarnley and Y. Al-Sadiq. 2002. Tuning PI Speed Controllers for Electric Drives Using Simulated Annealing. In Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium. 4: 1131-1135.

[12] M. Thompson and J. K. Fidler, 2001. Application of the Genetic Algorithm And Simulated Annealing to LC Filter Tuning. Circuits, Devices and Systems, IEE Proceedings [see also IEE Proceedings G- Circuits, Devices and Systems]. 148: 177-182.

[13] R. Haber, R. Haber-Haber, R. del Toro, and J. Alique. 2007. Using Simulated Annealing for Optimal Tuning of a PID Controller for Time-Delay Systems. An Application to a High-Performance Drilling Process. In Computational and Ambient Intelligence. 1155-1162.

[14] S. Buyamin, J. W., Finch, 2007. Tuning Extended Kalman Filter for Induction Motor Drives using Simulated Annealing. In Proceedings of the 26th IASTED International Conference, Modelling, Identification and Control Innsbruck, Austria.

[15] S. Buyamin and J. W. Finch. 2007. Comparative Study on Optimising the EKF for Speed Estimation of an Induction Motor using Simulated Annealing and Genetic Algorithm. In Electric Machines & Drives Conference. IEMDC '07. IEEE International. 1689-1694.

[16] D. P. Kwok and F. Sheng. 1994. Genetic Algorithm and Simulated Annealing for Optimal Robot Arm PID Control. In Evolutionary Computation. IEEE World Congress on Computational Intelligence. Proceedings of the First IEEE Conference. 2: 707-713.

[17] M. N. b. Mahyuddin. 2005. Direct Model Reference Adaptive Control Of Coupled Tank Liquid Level Control System. In Mechatronics and Automatic Control. Master of Electrical Engineering: Universiti Teknologi Malaysia.

[18] P. C. Chau. 2002. *Process Control: A First Course with MATLAB*: Cambridge University Press.

[19] C. R. Reeves, and Beasly, J. E, 1995. *Introduction Modern Heuristic Techniques for Combinatorial Problems.* C. Reeves, Ed. New York: McGraw-Hill.

[20] B. R. Domer, B., Shea, K., Smith, I. F. C. 2003. A Study of Two Stochastic Search Methods for Structural Control. *Journal of Computing in Civil Engineering.* 17(3:) 132-141.

[21] S. Kirkpatrick, Gelatt, C.D., Jr., Vecchi, and M.P. 1983. Optimization by Simulated Annealing. *Science.* 220: 671-680.

[22] R. E. King. 1999. *Computational Intelligence in Control Engineering*: NY: Marcel Decker.

[23] S. S. Rao. 1996. *Engineering Optimization: Theory and Practice*. 3rd Edition. John Wiley & Sons, Inc,

[24] K. Y. L. a. M. A. El-Sharkawi. 2008. *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*. Wiley-IEEE Press.

[25] R. T. H. a. Z. Gürdal, 1992. *Elements of Structural Optimization*. Springer.

[26] J. H. Holland. 1975. *Adaptation in Natural and Artificial Systems*: University of Mitchigan Press.

[27] A. J. Chipperfield and P. J. Fleming. 1995. The MATLAB Genetic Algorithm Toolbox. In Applied Control Techniques Using MATLAB, IEE Colloquium. 10/1-10/4.

[28] MathWorks. 2008. *Genetic Algorithm and Direct Search Toolbox 2 User's Guide*. The MathWorks, Inc.