PID Controller Tuning Using Evolutionary Algorithms

MOHD S. SAAD[1], HISHAMUDDIN JAMALUDDIN[2], INTAN Z. M. DARUS[2]
[1] School of Manufacturing Engineering,
Universiti Malaysia Perlis
MALAYSIA
[2]Department of System Dynamics & Control,
Faculty of Mechanical Engineering,
Universiti Teknologi Malaysia.
MALAYSIA

sazlisaad@unimap.edu.my, hishamj@fkm.utm.my, intan@fkm.utm.my,
http://www.fkm.utm.my/~intan

Abstract: - This paper presents the implementation of PID controller tuning using two sets of evolutionary techniques which are differential evolution (DE) and genetic algorithm (GA). The optimal PID control parameters are applied for a high order system, system with time delay and non-minimum phase system. The performance of the two techniques is evaluated by setting its objective function with mean square error (MSE) and integral absolute error (IAE). Both techniques will compete to achieve the globally minimum value of its objective functions. Meanwhile, reliability between DE and GA in consistently maintaining minimum MSE is also been studied. This paper also compares the performance of the tuned PID controller using GA and DE methods with Ziegler-Nichols method.

Key-Words: - Differential Evolution, Genetic Algorithm, PID Controller, Ziegler-Nichols

1 Introduction

PID is a remarkable control strategy, most widely used in processes industries such as oil and gas, chemical, petrochemical, pulp and paper, food and beverage, etc. PID controller has been proven in terms of reliability and robustness in controlling process variables ranging from temperature, level, pressure, flow, pH etc. Other factors that attracted industries to choose PID could be due to low cost, easy to maintain, as well as simplicity in control structure and easy to understand. However, improper PID parameters tuning could lead to cyclic and slow recovery, poor robustness and the worst case scenario would be the collapse of system operation [1]. This led researchers to explore the best method in searching optimum PID parameters.

Since the introduction of PID controller, many strategies have been proposed to determine the optimum setting of PID parameters. Ziegler-Nichols [2] and Cohen-Coon [3] are amongst the pioneer in PID tuning methods. They have proposed experimental PID tuning methods based on trial and error method, and process reaction curve methods. However, the difficulties may arise to tune the PID controller when the system is complex such as high order, time delay, non-minimum phase and non-linear processes. For example, Ziegler Nichols method may gives high overshoots, highly oscillatory, and longer settling time for a high order system and Cohen-Coon method only valid for the system having S-shaped step response of the plant [4],[5]. To overcome these difficulties, various methods have been used to obtain optimum PID parameters ranging from conventional method such as Refined Ziegler Nichols [6] and pole placement [7] and the implementation of modern heuristic optimization techniques such as genetic algorithms, simulated annealing, population based incremental learning, and particle swarm optimization [8]. Heuristic optimization is a technique of searching good solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is [9].

Recently, GA has been extensively studied by many researchers in searching for optimal PID parameters due to its high potential of escaping being trapped a local minimum. Kim et al. [10] proposed an improved GA method to tune PID controller for optimal control of RO plant with minimum overshoot and fast settling time compared with conventional tuning method. Yin et al. [11] have successfully used GA to tune PID controller for low damping, and slow response plant. Md Zain et al. [12] applied GA for optimization of PID parameters used to control a single-link flexible manipulator in vertical motion. Simulation results revealed that optimum PID parameters enable the system to perform well in reducing vibration at the end-point of the manipulator. Despite of excellence performance by GA is searching globally optimum solution in search space, some researchers have pointed out some deficiencies in GA performance. Those deficiencies are (i) poor premature convergence, (ii) loss of best solution found, (iii) no absolute assurance that a genetic algorithm will find a global optimum [13].

DE has been found to be a promising algorithm in numeric optimization problems. It has been proposed by Storn and Price [14]. DE has been designed to fulfill the requirement for practical minimization technique such as consistent convergence to the global minimum in consecutive independent trials, fast convergence, easy to work with, as well as ability to cope with non-differentiable, non-linear and multimodal cost functions [14]. Therefore, the algorithm has gained a great attention since it was proposed. Ruijun [15] studied the performance of DE and particle swarm optimization (PSO) in optimizing the PID controller for first order process has revealed that DE is generally more robust (with respect to reproducing constant results in different runs) than PSO. Youxin and Xiaoyi [16] has applied DE algorithm in tuning the PID controller for electric-hydraulic servo system of parallel platform. Simulation results show the controlled system has satisfactory response and the proposed parameter optimum method is an effective tuning strategy. Bingul and Zafer [17] has demonstrated the successful of DE in tuning the PID controller for unstable and integrating processes time delay, where it produces smaller settling time with minimum overshoot.

Even though the PID tuning method using GA and DE has been extensively studied by many researchers, the details on how the thing has been implemented still vague. Though, this paper is intended to provide a better understanding of how the PID controller is tune using two popular heuristic approaches by GA and DE. The performance of GA and DE in searching globally optimal PID parameters and its reliability to maintain the optimum value for several independent trials has been investigated for a high order system, system with time delay and non-minimum phase system. This paper also compares the transient performance of the system using GA and DE tuning methods with Ziegler-Nichols method.

2 PID Controller

PID controller parameters consist of three separate parameters: proportionality, integral and derivative values are denoted by k_p, k_i, and k_d. Appropriate setting of these parameters will improve the dynamic response of a system, reduce overshoot, eliminate steady state error and increase stability of the system [7]. The transfer function of a PID controller is:

$$C(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s \tag{1}$$

The fundamental structure of a PID control system is shown in Fig. 1. Once the set point has been changed, the error will be computed between the set point and the actual output. The error signal, $E(s)$, is used to generate the proportional, integral, and derivative actions, with the resulting signals weighted and summed to form the control signal, $U(s)$, applied to the plant model. The new output signal will be obtained. This new actual signal will be sent to the controller, and again the error signal will be computed. New control signal, $U(s)$, will be sent to the plant. This process will run continuously until steady-state error.

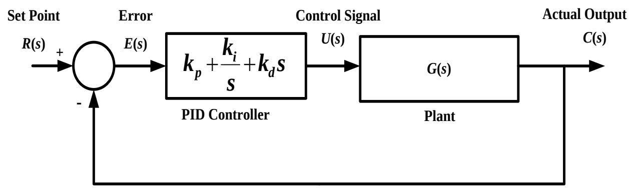


Fig. 1. PID control structure

3 Genetic Algorithm

GA was first introduced by John Holland as reported in [18]. It is a heuristic optimization technique inspired by the mechanisms of natural selection. GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem in which its performance is evaluated based on a fitness function. Based on the fitness of each individual and defined probability, a group of chromosomes is selected to undergo three common stages: selection, crossover and mutation. The application of these three basic operations allows the creation of new individuals to yield better solutions then the parents, leading to the optimal solution. The implementing of genetic algorithm in PID tuning is as follows:

3.1 Initial setting of GA parameters

GA is implemented with small population size. This requirement is important in practice in order to allow the controller to be optimized as fast as possible. In this study, the size of initial populations is set to be 20, crossover rate P_c = 0.9, mutation rate P_m = 0.01, and a number of generation G = 100.

The initial population is set by encoding the PID parameter, k_p, k_i and k_d into binary strings known as chromosome. The length of strings depends on the required precision which is about 4 significant figures. The required bits string is calculated based on the following equation:

$$2^{m_j - 1} < (b_j - a_j) \times 10^4 \le 2^{m_j} - 1 \qquad (2)$$

where m_j is the number of bits, and b_j and a_i are an upper bound and lower bound of PID parameters. For example, if $k_p \in [0,2.0]$, $k_i \in [0,2.5]$, and $k_d \in [0,0.1]$, the required bits calculated based on Eq. (2) are equal to 15, 15 and 10 bits respectively. The total length chromosome is 40 bits which can be represented in Fig. 2.

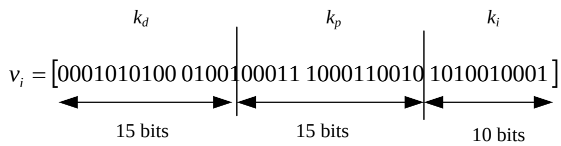


Fig. 2 Chromosome of PID parameters

In this study, the population in each generation is represented by 20 populations × 40 bits chromosome length.

3.2 Evaluate their fitness of each chromosome

The fitness of each chromosome is evaluated by converting its binary strings into real values which refer as PID parameter values and substitute into its objective function (some called as fitness function). The converting process of each chromosome is done by encoding into real number by using the following equation:

$$x_j = a_j + \text{decimal}(\text{substring}_j) \times \frac{(b_j - a_j)}{2^{m_j} - 1} \qquad (3)$$

For example, the corresponding values for k_p, k_i and k_d are given in Table 1.

Table 1
Binary to decimal value

	Binary string	Decimal value
k_p	000101010001001	2697
k_i	000111000110010	3634
k_d	1010010001	657

Therefore, the real number becomes:

$$k_p = 0 + 2697 \times \frac{(2-0)}{2^{15} - 1} = 0.1600$$

$$k_i = 0 + 3634 \times \frac{(2.5-0)}{2^{15} - 1} = 0.2700$$

$$k_d = 0 + 657 \times \frac{(0.1-0)}{2^{10} - 1} = 0.0600$$

Each set of PID parameter is passed to the objective function in order to compute for an initial fitness value. In this study, the mean square error (MSE) and integral absolute error (IAE) are chosen as objective function which is obtained from PID control system (refer to Fig. 3). The goal of GA is to seek for minimum fitness value.

3.3 Selection and reproduce using a probabilistic method (e.g., roulette wheel*)*

Then, all the fitness values and its corresponding chromosome will go for selection process. The higher the fitness value, the more chance an individual in the population has to be selected. Tournament selection is chosen because of this method offered better selection strategy such that it is able to adjust it selective pressure and population diversity to improve GA searching performance. Unlike roulette selection which allows the weaker chromosomes to be selected many times and also cause noisy convergence profile.

3.4 Implement crossover operation on the reproduced chromosomes

After selection process was completed, the crossover will be preceded. For basic GA, single point crossover is chosen. The two mating chromosomes are randomly select one cut-point and exchange the right part of the two parents to generate offspring.

3.5 Execute mutation operation with low probability

Mutation prevents the algorithm to be trapped in local minima and maintain diversity in the population. Commonly, lower mutation rate should be chosen. Higher mutation rate may probably cause the searching process will change into random search.

3.6 Repeat step 2 until the stopping criterion is met

After the selection, crossover and mutation process are done; again its binary strings in each chromosome in the population need to be decoded into real numbers in the next generation. A new set of PID parameter is send to the PID control system to compute for a new fitness value. This process will go through steps 3, 4 and 5 sequentially and repeated until the end of generations where the best fitness is achieved. The flowchart of a GA algorithm is shown in Fig. 2.

4 Differential evolution for PID tuning

Differential Evolution (DE) algorithm is a heuristic optimization algorithm recently introduced. Unlike simple GA that uses binary coding for representing problem parameters, DE uses real coding of floating point numbers. The crucial idea behind DE is a scheme for generating trial parameter vectors. Basically, DE adds the weighted difference between two population vectors to a third vector.

The key parameters of control are: *NP* - the population size, *CR* - the crossover constant, *F* - the weight applied to random differential (scaling factor). It is worth noting that DE's control variables, *NP*, *F* and *CR*, are not difficult to choose in order to obtain promising results. Storn [19] have come out with several rules in selecting the control parameters. The rules are listed follow:

- The initialized population should be spread as much as possible over the objective function surface.
- Frequently the crossover probability $CR \in [0,1]$ must be considerably lower than one (e.g. 0.3). If no convergence can be achieved, $CR \in [0.8, 1]$ often helps.
- For many applications $NP = 10 \times D$, where *D* is the number of problem dimension. *F* is usually chosen at [0.5, 1].
- The higher the population size, *NP*, the lower the weighting factor *F* should choose.

These rules of thumb for DE's control variables which is easy to work with is one of DE's major contribution [14]. The detailed Differential Evolution algorithm used in tuning the PID controller is presented below:

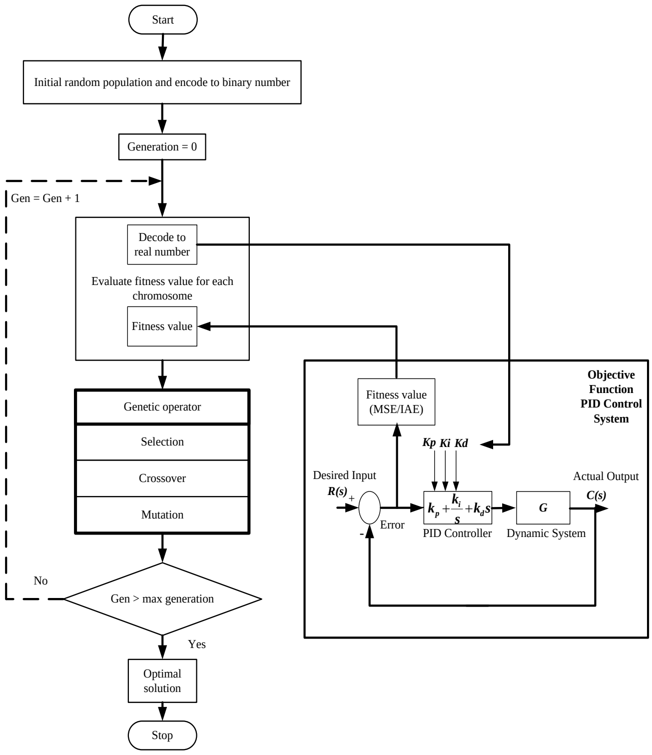


Fig. 3. Flowchart of genetic algorithm for PID tuning

4.1 Setting DE optimization parameters

All the DE optimization parameter required for optimization process is listed below:

- *D* – problem dimension
- *NP*, *CR*, *F* – control parameters
- *G* – Number of generation/stopping condition
- *L*,*H* – boundary constraints

In this study, population size, *NP* = 100, crossover constant, *CR* = 0.9, mutation constant, *F* = 0.6, and a number of generation *G* = 100. The problem dimension, *D* is set based on the number of parameters used in the objective function. The problem dimension is refer to the number of PID parameters k_p, k_i and k_d which is equal to 3. The boundary constraint is set based on the PID parameters range. For example, if $k_p \in [0, 2.0]$, it means that low boundary, *L* = 0 and high boundary, *H* = 2.0.

4.2 Vector population initialization

Initialize all the vector population randomly in the given upper & lower bound and evaluate the fitness of each vector.

$$Pop_{ij} = L + (H - L).\text{rand}_{ij}(0,1) \quad (4)$$

$i = 1,...,D$, $j = 1,...,NP$

$$Fit = f(Pop_j) \quad (5)$$

Before the optimization is launched the population needs to be initialized and its fitness function needs to be evaluated. The population is initialized randomly within its boundary constraints is done using Eq. (4). Each of the individual in the population is used to compute the fitness value which referred as MSE/IAE. The fitness value is computed by the fitness function as in Eq. (5) which is referring to PID control system. Fig. 4 shows the block diagram of population and its corresponding fitness value. For example, if $k_p \in [0, 10]$, $k_i \in [0, 13]$, and $k_d \in [0, 18]$, population and fitness values are calculated as:

For $k_p \in [0, 10]$,

$$Pop_{12} = 0 + (10-0).\text{rand}_{12}(0,1)$$
$$Pop_{12} = 11.1180 = k_p$$

For $k_i \in [0, 13]$,

$$Pop_{13} = 0 + (10-0).\text{rand}_{13}(0,1)$$
$$Pop_{13} = 8.5944 = k_i$$

For $k_d \in [0, 18]$,

$$Pop_{11} = 0 + (18-0).\text{rand}_{11}(0,1)$$
$$Pop_{11} = 10.4714 = k_d$$

Then, the value of k_p, k_i, and k_d are sent to the objective function which is the PID control system to compute for fitness value (refer to Fig. 4).

4.3 Perform mutation & crossover

Whenever initialization process is done, now come to optimization process. The optimization process will run iteratively until the end of generations. By referring to Fig. 8, the first individual fitness value from the current population is set to be the target vector. Then the trial vector is created by selecting three individuals randomly from the current population, mutate using Eq. (6) and crossover with the target vector. The fitness value (MSE/IAE) of the trial vector is computed by sending its individuals to the fitness function.

4.3.1 Mutant vector

For each vector $x_{j,G}$ (target vector), a mutant vector is generated by:

$$v_{j,G+1} = x_{r3,G} + F.(x_{r1,G} - x_{r2,G}) \quad (6)$$

where the three distinct vectors x_{r1}, x_{r2} and x_{r3} randomly chosen from the current population other than target vector $x_{j,G}$. The detail example how the mutant vector is determined is shown in Fig. 5.

4.3.2 Crossover

Perform crossover for each target vector with its mutant vector to create a trial vector $u_{j,G+1}$.

$$u_{j,G+1} = (u_{1j,G+1}, u_{2j,G+1}, \ldots, u_{Dj,G+1})$$

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1} & \text{if } (\text{rand}_i \le CR) \vee (\text{Rnd} = i) \\ x_{ij,G} & \text{otherwise} \end{cases}$$

$$i = 1, \ldots, D$$

Crossover is done in order to increase the diversity of the perturbed PID parameters for each individual in the population. The block diagram on how this process is done is shown in Fig. 6.

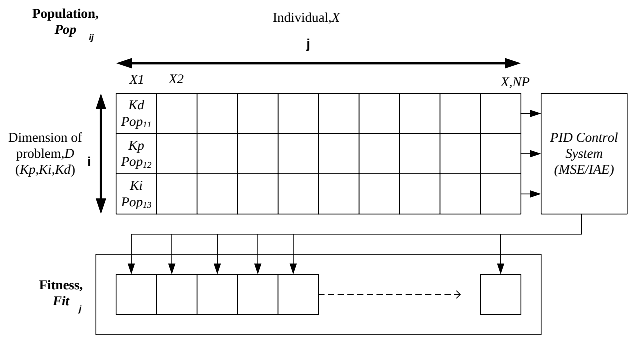


Fig. 4. The block diagram of population and its corresponding fitness value

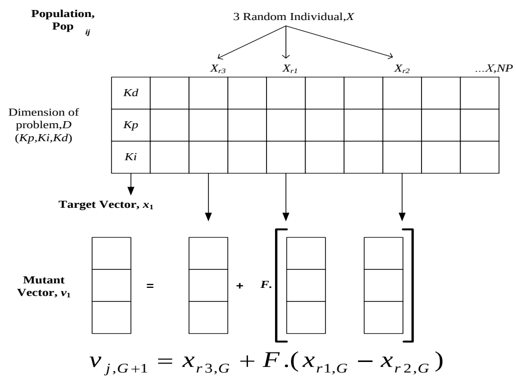


Fig. 5. Mutation process

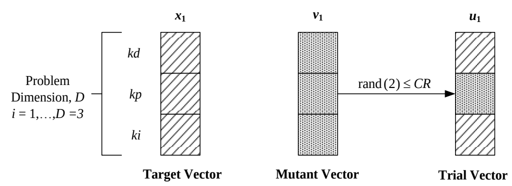


Fig. 6. Crossover process

4.4 Verifying the boundary constraint

If the bound (i.e. lower & upper limit of a variable) is violated then it can be brought in the bound range (i.e. between lower & upper limit) either by forcing it to lower/upper limit (forced bound) or by randomly assigning a value in the bound range (without forcing).

$$\text{if } x_i \notin [L, H], \quad x_i = L + (H-L).\text{rand}_i(0,1) \quad (7)$$

Eq. (7) purposely used in order to make sure that all the parameter vectors (PID parameters) are within its boundary constraints.

4.5 Selection

Selection is performed for each target vector, $x_{j,G}$ by comparing its fitness value with that of the trial vector, $u_{j,G}$. Vector with lower fitness value is selected for next generation. Fig. 7 shows how the selection process is performed.

Process is repeated until a termination criterion is met.

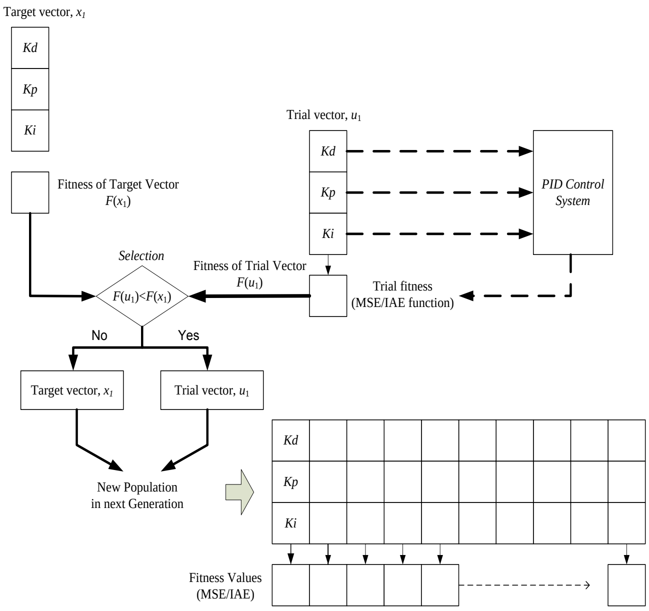


Fig. 7. Selection process

5. Implementation PID Controller Tuning

In this study, the transfer function used to evaluate the performance between DE and GA are a high order system $G_1(s)$, system with time delay $G_2(s)$, and non-minimum phase system $G_3(s)$.These system are:

$$G_1(s) = \frac{25.2s^2 + 21.2s + 3}{s^5 + 16.58s^4 + 25.41s^3 + 17.18s^2 + 11.70s + 1} \quad (8)$$

$$G_2(s) = \frac{10e^{-1.0s}}{(1+8s)(1+2s)} \quad (9)$$

$$G_3(s) = \frac{(1-10s)}{(1+s)^3} \quad (10)$$

The tuning performance of PID controller is evaluated using mean squared error (MSE) and integral absolute error (IAE) which then becomes the objective function that is used to evaluate fitness of each chromosome/individual in GA and DE. GA and DE will heuristically find the optimum value of controller parameters where the smaller the value of objective function the fitter is the chromosome/individual. Finally the transient performance of the system tuned by DE and GA is compared with Ziegler-Nichols method.

5.1 PID Tuning with Ziegler-Nichols method

PID tuning using Ziegler Nichols method is based on the frequency response of the closed-loop system by determining the point of marginal stability under pure proportional control. The proportional gain is increased until the system become marginally stable. At this point, the value of proportional gain known as the ultimate gain, k_u, together with its period of oscillation frequency so called the ultimate period, t_c, are recorded. Based on these values Ziegler and Nichols calculated the tuning parameters shown in Table 2.

Table 2
Ziegler-Nichols PID Tuning Parameter

Controller	k_p	k_i	k_d
PID	$0.6k_u$	$t_u/2$	$t_u/8$

For mathematical model system, the ultimate gain, k_u, and its ultimate period, t_u, can be determined using root locus technique. When the root locus of the system has been plotted, rlocfind command in Matlab can be used to find the crossing point and gain of the system at real part equal to zero.

In this study, root locus plot for the system given by Eqs. (8)-(10) is shown in Figs. 8(a) - 8(c). The ultimate gain, k_u, ultimate period, t_u, and PID tuning parameters are calculated based on these figures. The details data are listed in Table 3.

Table 3
Ziegler-Nichols PID Tuning Values

	High Order System	System with Time Delay	Non-minimum Phase System
k_u	6.650	1.110	0.258
t_u	1.893	8.344	9.726
k_p	3.990	0.666	0.155
k_i	0.946	4.172	4.863
k_d	0.237	1.043	1.216

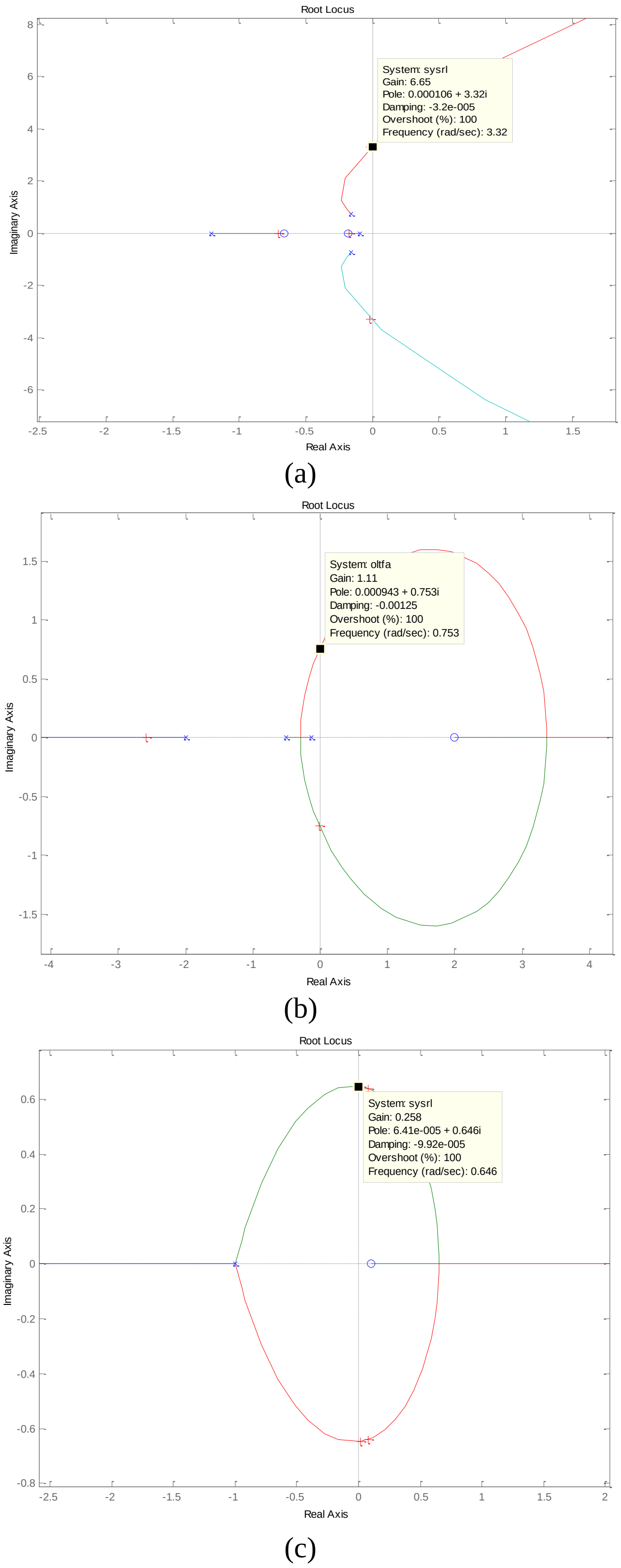


Fig. 8. Root locus plot, (a) high order system, (b) system with delay, and (c) non-minimum phase system

6. Simulation Result

Simulation is carried out in order to study the performance between DE and GA to optimally tune the PID controller for the systems given by Eqs. (3)-(5). The parameter values of DE and GA optimization shown in Table 4 are chosen based on [20]. The parameters range for k_p, k_i and k_d as shown in Table 5 are set based on the previous studies [21], [22], [23]. The performance of both tuning methods is observed in terms of rise time, settling time, maximum overshoot of the response, MSE and IAE. Finally, the convergence rate in achieving the global optimum value of the objective function is investigated.

Table 4
Parameter Setting For DE and GA

DE	GA
Population size = 20	Population size = 20
Crossover Rate = 0.9	Crossover Rate = 0.9
Differentiation constant = 0.6	Mutation rate = 0.01
Gen. number = 100	Gen. number = 100

Table 5
PID Parameter Range

	High Order System		System with Time Delay		Non-minimum Phase	
Parameter Range	min	max	min	max	min	max
k_p	0	10	0	2.0	0	0.5
k_i	0	13	0	0.5	0	0.5
k_d	0	18	0	2.5	0	0.25

The results for closed-loop step response for DE and GA PID tuning method are shown in Figs. 9(a) - 9(c) respectively. For Fig. 9(a), it is clear that the responses from DE and GA tuning methods are almost indistinguishable. The details of these results are shown in Table 6. The values of rise time, settling time, maximum overshoot, MSE and IAE between DE and GA are almost the same. Both tuning methods give better performance compared to Ziegler-Nichols method. As seen from the Table 6, Ziegler-Nichols gives poor rise-time, settling time and highest overshoot. In the case of system with time delay, DE and GA optimized by MSE and IAE give almost the same response as shown in Fig 9(b). DE and GA optimized by MSE offered better rise time compared with DE and GA optimized by IAE.

However, DE and GA optimized by IAE give better settling time and overshoot which is about 12 seconds faster and 14% reduction with respect to DE and GA optimized by MSE (refer to Table 7). Even though DE and GA optimized by IAE offered better settling time and overshoot compared with DE and GA optimized by IAE, both method give superior performance than Ziegler-Nichols method.

For the case of non-minimum phase system, instead of evaluating in the aspects of rise time, settling time and peak overshoot, the improper undershoot effect should also need to be reduced. All the tuning method gives different responses as shown in Fig. 9(c). DE optimized by IAE give the best values of rise time and settling time followed by DE optimized by MSE, GA optimized by MSE, GA optimized by IAE, and Ziegler-Nichols tuning method. However in terms of overshoot, no overshoot are produced by GA optimized by IAE, MSE and Ziegler-Nichols except DE optimized by IAE and MSE which give small overshoot about 3% and 7% respectively (refer to Table 8). But, in terms of undershoot, Ziegler-Nichols give the lowest undershoot than followed by DE optimized by MSE, GA optimized by MSE, GA optimized by IAE, and DE optimized by IAE. In order to get better results with minimum overshoot and undershoot, modification needs to be done in the objective function [24]. With standard performance criteria, there should be a trade-off between minimum overshoot/undershoot and settling time. In this study, only MSE and IAE used as objective function for tuning the system.

For overall performance, it can be concluded that DE optimized by MSE give better transient response. The evolution of PID parameter for non-minimum phase system can be seen in Figs. 10(a) – 10(d). Figs. 10(a) and 10(b) show the convergence profile of the PID parameter, k_p, k_i and k_d with 100 generations for DE and GA optimized by IAE respectively. From the observation, PID parameter for both techniques almost settles at the same generation. However, for DE and GA optimized by MSE, it shows that PID parameter convergence for DE is faster than GA (refer Figs. 10(c) and 10(d). Parameter k_d seen not very consistent. Its convergence profile fluctuated at the beginning and settled after 50 generation. From these observations, for non-minimum phase system, DE performed better than GA with IAE as a fitness function.

Convergence test is done 5 times for each of the systems in order to investigate the convergence rate and its consistency in searching the globally optimal solution of PID parameters. The lower the value of fitness function the better the closed-loop system response will be.Comparison for the convergence rate of fitness function performance between DE and GA can be seen from Figs. 11, 12 and 13. From these figures, it is observed that DE is significantly consistent than GA in searching for minimum fitness function. Convergence test done in DE and GA settled almost at the same generation. From the overall results we can say that, DE algorithm outperforms GA in terms of its consistency in constantly achieving the globally minimum fitness value.

Table 6
Comparison Performance for High Order System

	High Order System				
	ZN	DE (MSE)	GA (MSE)	DE (IAE)	GA (IAE)
MSE	-	0.001	0.001		
IAE	-	-	-	11.43	11.45
Rise time (s)	0.39	0.07	0.07	0.062	0.067
Settling time (s)	9.18	0.40	0.40	0.40	0.48
Over shoot (%)	57.56	27.19	27.26	28.42	28.54
k_p	3.99	3.56	3.75	7.16	7.50
k_i	4.22	10.96	10.99	11.16	11.17
k_d	0.94	18.00	18.00	18.00	18.00

Table 7
Comparison Performance for System with delay

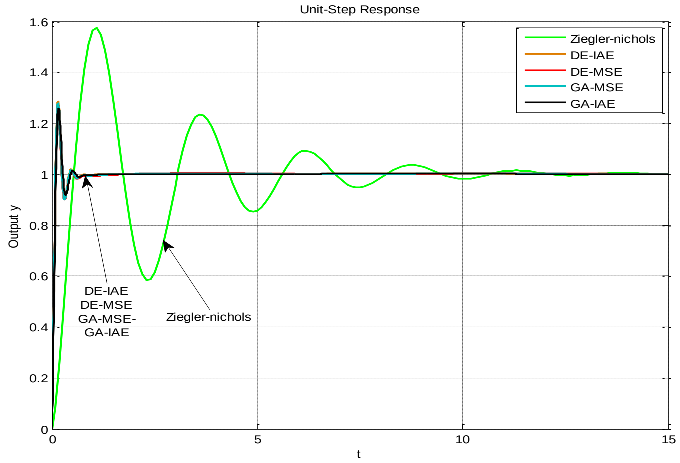
	System with Time Delay				
	ZN	DE (MSE)	GA (MSE)	DE (IAE)	GA (IAE)
MSE	-	0.02	0.0198	-	-
IAE	-	-	-	18.70	18.70
Rise time (s)	1.627	0.66	0.66	1.13	1.13
Settling time (s)	17.88	19.21	18.44	6.76	6.78
Over shoot (%)	44.33	25.54	25.26	11.42	10.85
k_p	0.66	0.61	0.63	0.68	0.67
k_i	0.16	0.10	0.09	0.07	0.07
k_d	0.69	2.15	2.13	1.34	1.35

Table 8
Comparison Performance for Non minimum phase system

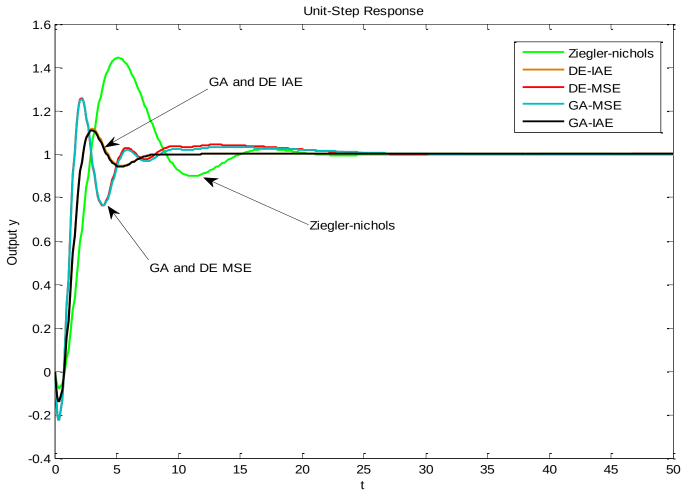
	Non-minimum Phase System				
	ZN	DE (MSE)	GA (MSE)	DE (IAE)	GA (IAE)
MSE	-	0.02	0.02	-	-
IAE	-	-	-	1333.5	1601.9
Rise time (s)	57.55	2.45	2.72	1.858	14.95
Settling time (s)	79.21	20.55	22.68	11.75	23.56
Over shoot (%)	-	7.01	-	2.94	-
Unders hoot (%)	131.80	146.80	157.40	205.70	167.30
k_p	0.16	0.19	0.19	0.21	0.19
k_i	0.03	0.07	0.06	0.08	0.06
k_d	0.19	0.11	0.06	0.17	0.19

7. Conclusion
PID controller has been tuned using Ziegler-Nichols method and modern heuristic algorithms, DE and GA for a high order system, system with time delay and non-minimum phase system. For the same population, crossover rate and number of generation, both tuning methods demonstrated the same performance in searching the best value of MSE and IAE. It is worth noting that for high order system and system with delay, DE and GA give almost the same transient performance for the objective function MSE and IAE. DE optimized by MSE gives better performance with the regards to the trade-off between settling time, maximum overshoot and undershoot for non-minimum phase system. In terms of reliability, DE offer consistency in achieving its globally minimum of fitness value. However the convergence rate for all the trials for higher order system, system with time delays and non-minimum phase system are almost the same.

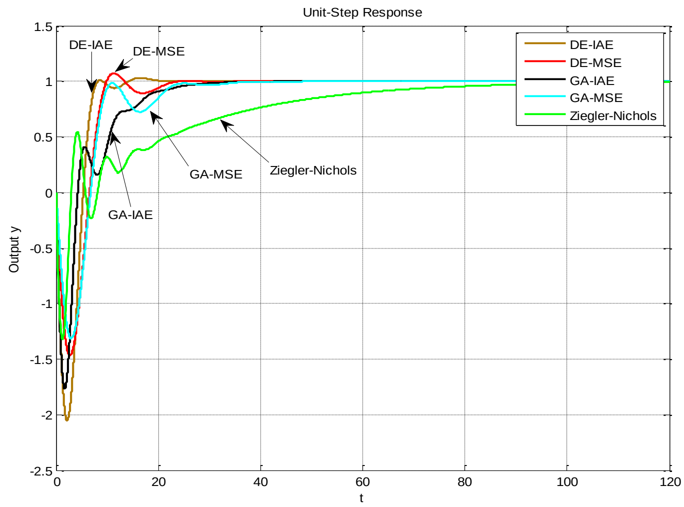
Acknowledgement
Authors would like to express their gratitude to Universiti Teknologi Malaysia (UTM) and Universiti Malaysia Perlis (UNIMAP) for the funding of the research and facilities to conduct this research. This research is funded by Universiti Teknologi Malaysia (UTM) using Research University Grant Vote No. 00H11



(a) Higher order system



(b) System with time delay



(c) Non-minimum phase system

Fig. 9. Step response of PID control system

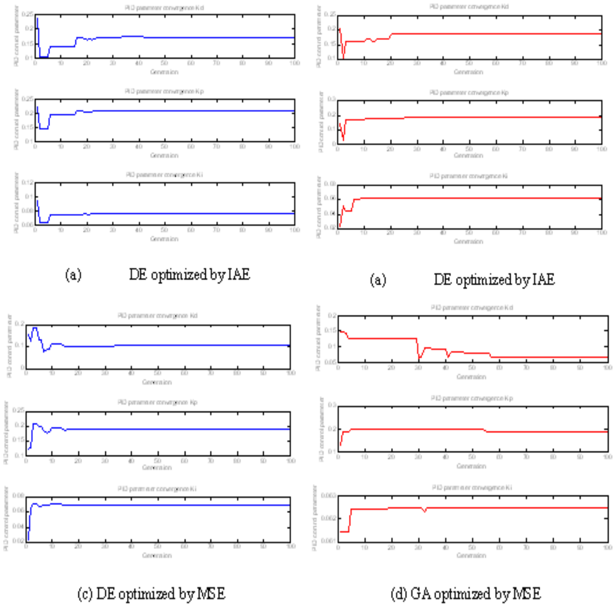


Fig. 10. Convergence profile of PID parameter

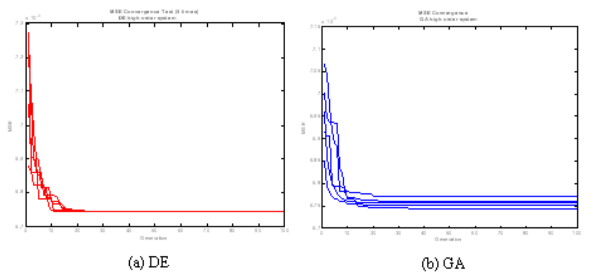


Fig. 11. Convergence test for a high order system

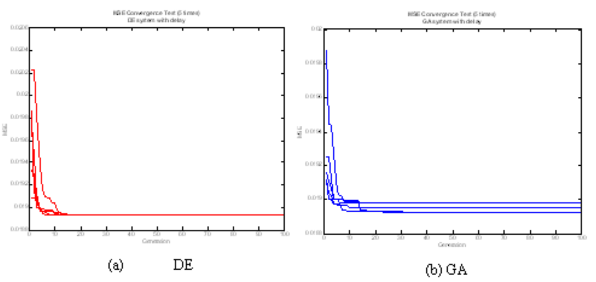


Fig. 12. Convergence test for system with delay

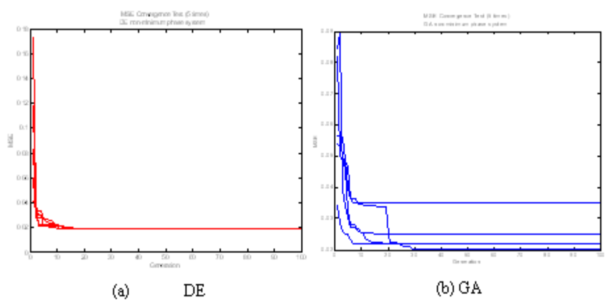


Fig. 13. Convergence test for non-minimum phase system

References:

[1] K. J. Astrom and T. Hagglund, *Automatic Tuning of PID Controllers*. 1988: Instrument Society of America.

[2] J. G. Ziegler and N.B. Nichols, Optimum Settings for Automatic Controllers. *Trans. ASME*. **64**(8): p. 759–768, 1942.

[3] G. H. Cohen and G. A. Coon, Theoretical Consideration of Retarded Control. *Trans. ASME*. **75**: p. 827–834, 1953.

[4] G. Saravanakumar, R.S.D. Wahidhabanu, and V.I. George, Robustness and Performance of Modified Smith Predictors for Processes with Longer Dead-Times. *International Journal on Automatic Control and System Engineering*. **6**(3), 2006.

[5] T. Jain and M.J. Nigam, Optimisation of PD-PI Controller Using Swarm Intelligence. *Journal of Theoretical and Applied Information Technology*, **4**(11): p. 1013-1018, 2008.

[6] K. J. Astrom, Automatic Tuning and Adaptation for PID Controllers - a Survey. *Control Engineering Practice*. **1**(4): p. 699-714, 1993.

[7] K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning*. 1995: Instruments Society of America.

[8] P. B. De Moura Oliveira, *Modern Heuristics Review for PID Control Optimization: a Teaching Experiment*. in Proceeding of International Conference on Control and Automation. 2005. Budapest, Hungary.

[9] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*. 1995, London: McGraw-Hill.

[10] J. S. Kim, *Auto Tuning PID Controller Based on Improved Genetic Algorithm for Reverse Osmosis Plant*. in Proceedings of the World Academy of Science and Engineering. 2008.

[11] F. Yin, J. Wang, and C. Guo, *Design of PID Controllers Using Genetic Algorithms Approach for Low Damping Slow Response Plants*,. Advances in Neural Networks, ed. L.N.i.C. Science. 2004: Springer-Verlag Berlin Heidelberg. 219-220.

[12] B. A. M. Zain, M.O. Tokhi, and S.F. Toha, *PID-based Control of a Single-Link Flexible Manipulator in Vertical Motion with Genetic Optimisation*, in *Third UKSim European Symposium on Computer Modeling and Simulation*. 2009: Athens, Greece.

[13] M. Vanitha and K. Thanushkodi, Solution to Economic Dispatch Problem by Differential Evolution Algorithm Considering Linear Equality and Inequality Constrains. *International Journal of Research and Reviews in Electrical and Computer Engineering*. **1**(1): p. 21-26, 2011.

[14] R. Storn and K. Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*. **11**(4): p. 341–359, 1997.

[15] D. Ruijun, *Differential Evolution Versus Particle Swarm Optimization for PID Controller Design*. in

Fifth International Conference on Natural Computation, ICNC '09. 2009.

[16] L. Youxin and C. Xiaoyi. *Tuning PID control parameters on hydraulic servo control system based on differential evolution algorithm*. in 2nd International Conference on Advanced Computer Control (ICACC). 2010.

[17] Bingul and Zafer, *A New PID Tuning Technique Using Differential Evolution for Unstable and Integrating Processes with Time Delay*, in *Neural Information Processing*, Pal, et al., Editors. 2004, Springer Berlin / Heidelberg. p. 254-260.

[18] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. 1989: Addison-Wesley Publishing Co., Inc.

[19] R. Storn, *On the Usage of Differential Evolution for Function Optimization*. in Proceedings of the Fuzzy Information Processing Society. 1996. Berkeley, CA, USA.

[20] N. Karaboga, Digital IIR Filter Design Using Differential Evolution Algorithm. *EURASIP Journal on Applied Signal Processing*. **2005**: p. 1269-1276, 2005.

[21] G. De Almeida, *Application of Genetic Programming for Fine Tuning PID Controller Parameters Designed Through Ziegler-Nichols Technique*, in *Advances in Natural Computation*, Wang, et al., Editors. 2005, Springer Berlin / Heidelberg. p. 434-434.

[22] A. Oi, *PID Optimal Tuning Method by Particle Swarm Optimization*, in *SICE Annual Conference*. 2008: Japan. p. 3470-3473.

[23] P. Chou and T. Hwang, *Design of PID Controllers Using Genetic Algorithms Approach for Low Damping, Slow Response Plants*, in *Advances in Neural Networks - ISNN 2004*, Yin, et al., Editors. 2004, Springer Berlin / Heidelberg. p. 219-220.

[24] W. Jie-sheng, Z. Yong, and W. Wei, Optimal Design 0f PI/PD Controller for Non-Minimum Phase System *Transactions of the Institute of Measurement & Control*. **28**(1), 2006.