International Journal of Innovative
Computing, Information and Control
Volume 8, Number 11, November 2012

 ISSN 1349-4198
pp. 7761–7779

IMPLEMENTATION OF PID CONTROLLER TUNING USING DIFFERENTIAL EVOLUTION AND GENETIC ALGORITHMS

Mohd Sazli Saad[1], Hishamuddin Jamaluddin[2]
and Intan Zaurah Mat Darus[2]

[1]School of Manufacturing Engineering
Universiti Malaysia Perlis
Ulu Pauh Main Campus, 02600 Arau, Perlis, Malaysia

[2]Faculty of Mechanical Engineering
Universiti Teknologi Malaysia
81310 Skudai, Johor Bahru, Malaysia
intan@fkm.utm.my

Received July 2011; revised February 2012

Abstract. *This paper presents the implementation of PID controller tuning using two modern heuristic 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 these techniques is evaluated by setting their objective functions as mean square error (MSE) and integral absolute error (IAE). The reliability between DE and GA in consistently maintaining minimum MSE is studied. The performance of the PID control systems tuned using GA and DE methods are also compared with Ziegler-Nichols method.*

Keywords: PID controller, Differential evolution, Genetic algorithm, Ziegler-Nichols, Mean square error

1. **Introduction.** PID is a remarkable control strategy, widely used in processes industries such as oil and gas, chemical, petrochemical, pulp and paper, food and beverage. 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 controller 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 and Nichols [2], Cohen and Coon [3] are amongst the pioneers in PID tuning methods. They have proposed experimental PID tuning methods based on trial and error, and process reaction curve. 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 give high overshoots, highly oscillatory, and longer settling time for a high order system and Cohen-Coon method is only valid for systems having S-shaped step response [4,5]. To overcome these difficulties, various methods have been used to obtain optimum PID parameters ranging from conventional methods 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 at a local minimum. Kim et al. [10] proposed an improved GA method to tune PID controller for optimal control of reverse osmosis (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. 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. Even though, GA has proved its capability in searching for the optimum solution from the problem space, there is no absolute assurance that a genetic algorithm will find a global optimum. There are two main drawbacks in GA: poor premature convergence and loss of best solution found [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 developed to fulfill the requirement for practical minimization technique such as fast and consistent convergence to the global minimum in consecutive independent trials, easy to work with, as well as its ability to cope with non-differentiable, non-linear and multimodal cost functions [14]. Therefore, the algorithm has gained great attention since its proposal. Dong [15] studied the performance of DE and particle swarm optimization (PSO) in optimizing PID controller for first order process and found that DE is generally more robust (with respect to reproducing constant results in different runs) than PSO. Luo and Che [16] applied DE algorithm in tuning PID controller for electric-hydraulic servo system of parallel platforms. Simulation results showed satisfactory response of the control system and the proposed parameter optimum method is an effective tuning strategy. Arya et al. [17] applied DE in tuning PID controller for automatic generation control where comparative studied showed that DE has produced better optimal transient response of frequency and tie line power changes compared with particle swarm optimization based gains.

Even though PID tuning methods using GA and DE have been extensively studied by many researchers, the details on how the algorithms are implemented are still vague. This paper is intended to provide a better understanding of how PID controller is tuned 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 have 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. The practical implementation of PID controller using GA and DE cannot be done using direct on-line optimization method due to the needs to compute the fitness of every chromosome/individual in the population. These require a complete response of the system to be obtained for all chromosomes/individuals before new optimal PID parameters are produced. Thus, practical concept of PID controller tuning is explained briefly in Section 7. This concept is based on indirect controller optimization via self-tuning control technique.

This paper is organized as follows. Section 2 briefly explains the fundamental of PID controller. Sections 3 and 4 describe the implementation of GA and DE techniques to optimally tune PID controller. Section 5 describes the implementation of PID tuning using Ziegler-Nichols tuning method. Section 6 shows the case studies of PID tuning for the systems of high order, with time delay and non-minimum phase. Section 7 describes the practical concept for PID tuning using heuristic optimization methods. Finally, the paper is concluded in Section 8.

2. **PID Controller.**

PID controller parameters consist of three separate terms: 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 Figure 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.

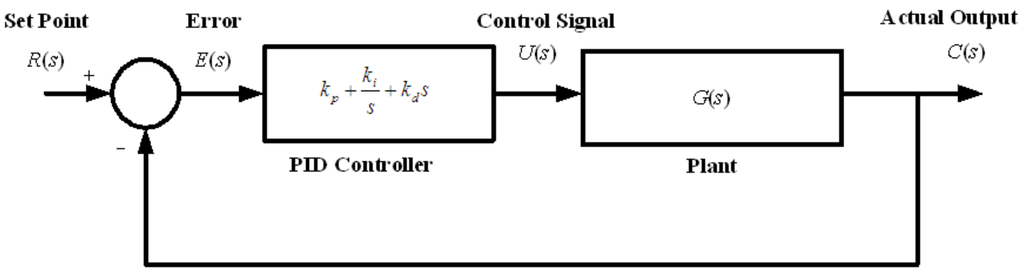


FIGURE 1. PID control structure

3. **Genetic Algorithm for PID Tuning.**

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 will allow 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:

i. **Initialize the setting of GA parameters and generate an initial, random population of individuals**. 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 the 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:

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

where m_j is the number of bits, and b_j and a_j 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 (2) are equal to 15, 15 and 10 bits respectively. The total length chromosome is 40 bits which can be represented as Figure 2.

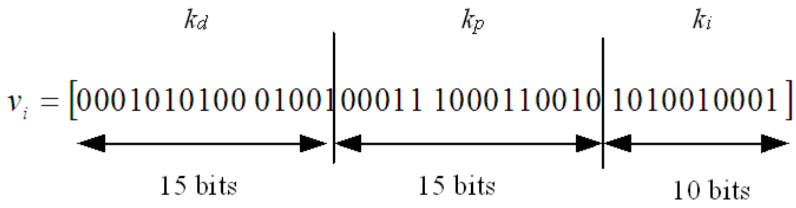


FIGURE 2. A chromosome representing PID parameters

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

ii. Evaluate the fitness of each chromosome. The fitness of each chromosome is evaluated by converting its binary string into a real value which represents the PID parameters. The conversion process of each chromosome is done by encoding into real number as follows:

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

For example, the corresponding values for k_p, k_i and k_d are given below:

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 parameters is passed to PID controller. A complete response of the system for each PID set and its initial fitness value is computed using a defined objective function. In this study, the mean square error (MSE) and integral absolute error (IAE) are chosen as the objective functions. Figure 3 shows the flowchart of the tuning procedure. The goal of GA is to seek for minimum fitness value.

$$MSE = \frac{1}{t} \int_0^{\tau} (e(t))^2 dt \tag{4}$$

$$IAE = \int_0^{\tau} |e(t)|\, dt \tag{5}$$

iii. Perform selection, crossover and mutation. All chromosomes will go through the selection process based on their fitness values. The higher the fitness value, the more chance an individual in the population will be selected. Tournament selection [19] is chosen because this method offers a better selection strategy. It is able to adjust its selective pressure and population diversity to improve GA searching performance, unlike roulette selection which allows weaker chromosomes to be selected frequently and also cause noisy convergence profile.

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

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 becomes random search.

iv. Repeat step 2 until end of generations. After selection, crossover and mutation processes have been completed, again the binary string of each chromosome in the population needs to be decoded into real numbers in the next generation. A new set

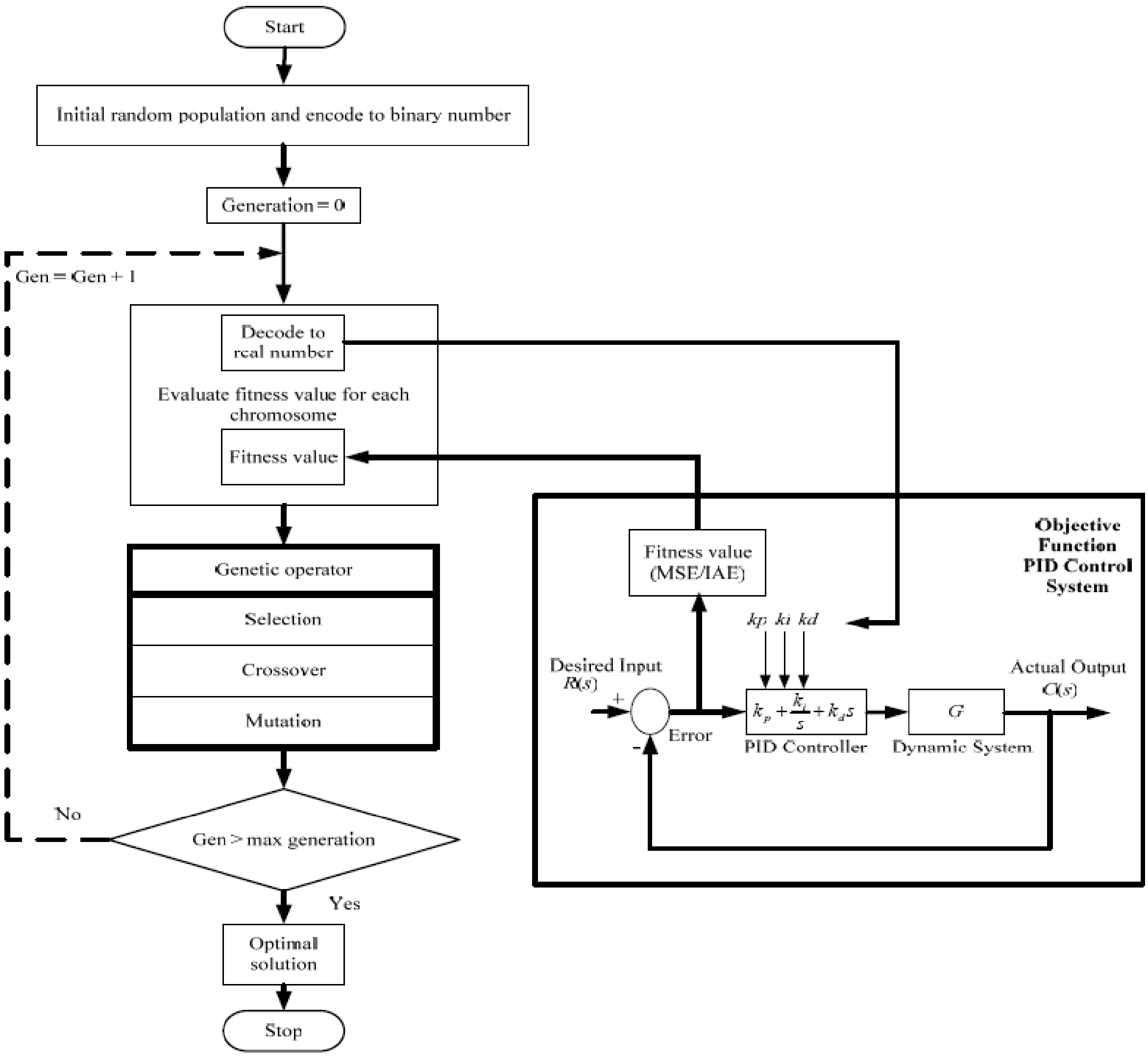


FIGURE 3. Flowchart of genetic algorithm for PID tuning

of PID parameters is sent to the PID control system to compute for a new fitness value. This process will go through steps 2 to 3 sequentially and repeat until the end of generations where the best fitness is achieved. The tuning method using GA can be represented by the flowchart shown in Figure 3.

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 mutation constant 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 [20] has come out with several rules in selecting the control parameters. The rules are listed below:

- 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. The mutation constant, F is usually chosen between $[0.5, 1]$.
- The higher the population size, *NP*, the lower the mutation constant, 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:

i. Setting DE optimization parameters. 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. In this case, problem dimension refers 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, $H = 2.0$.

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

$$Pop_{ij} = L + (H - L).\text{rand}_{ij}(0, 1), \quad i = 1, \ldots, D, \quad j = 1, \ldots, NP \tag{6}$$

$$Fit = f(Pop_j) \tag{7}$$

where

$$f(Pop_j) = \frac{1}{t}\int_0^{\tau} (e(t))^2 dt, \text{ for MSE} \tag{8}$$

$$f(Pop_j) = \int_0^{\tau} |e(t)|\, dt, \text{ for IAE} \tag{9}$$

and rand $ij(0, 1)$ – random number between 0 and 1. Before optimization starts the population needs to be initialized and their fitness values need to be evaluated. The population is initialized randomly within its boundary constraints using (6).

The fitness value (7) which referred as MSE and IAE is computed using (8) and (9) respectively based on the error of the control system. Figure 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 $k_d \in [0, 18]$, and let $\text{rand}_{12}(0, 1) = 1.1118$, $\text{rand}_{13}(0, 1) = 0.8594$, and $\text{rand}_{11}(0, 1) = 1.0471$, then the individuals representing the controller parameters 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 $k_d \in [0, 18]$,

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

The value of k_p, k_i, and k_d in the first individual of population, Pop_{1i} are sent to the PID control system. The output response is simulated and the value of its fitness value is computed (refer to Figure 4).

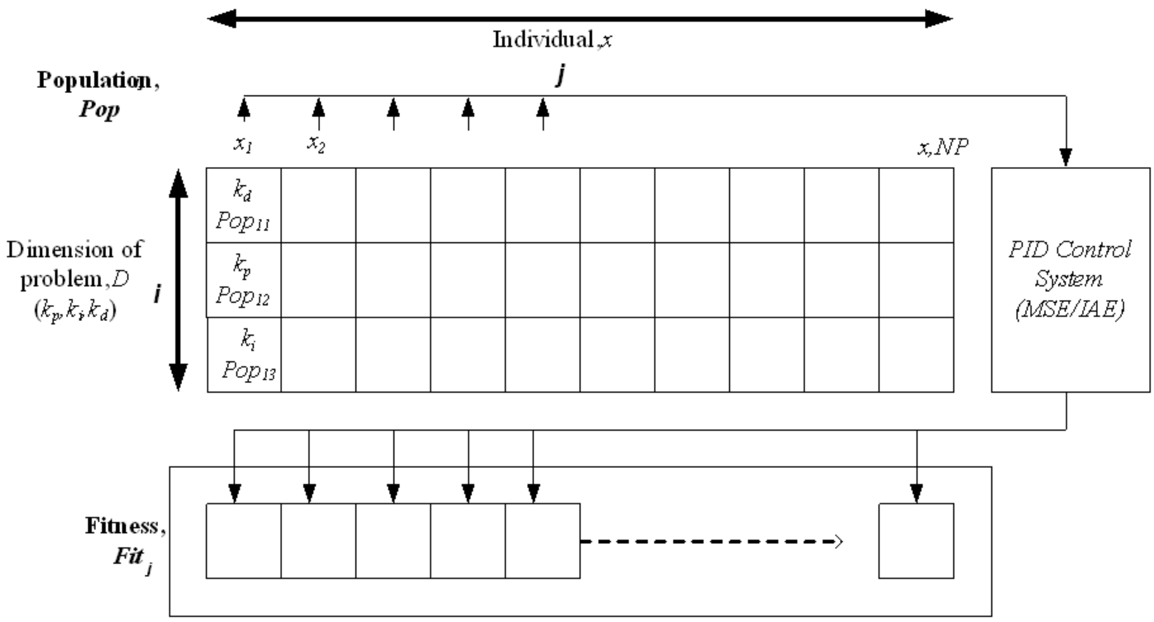


FIGURE 4. The block diagram of population and its corresponding fitness value

iii. Perform mutation and crossover. The optimization process will run iteratively until the end of generations. The first individual fitness value from the current population is set to be the target vector as Figure 5. Then the trial vector is created by selecting three individuals randomly from the current population, mutation using (10) and crossover with the target vector. The fitness value (MSE/IAE) of the trial vector is computed by sending its individuals to the PID controller.

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

$$v_{j,G+1} = x_{r3,G} + F.(x_{r1,G} - x_{r2,G}) \tag{10}$$

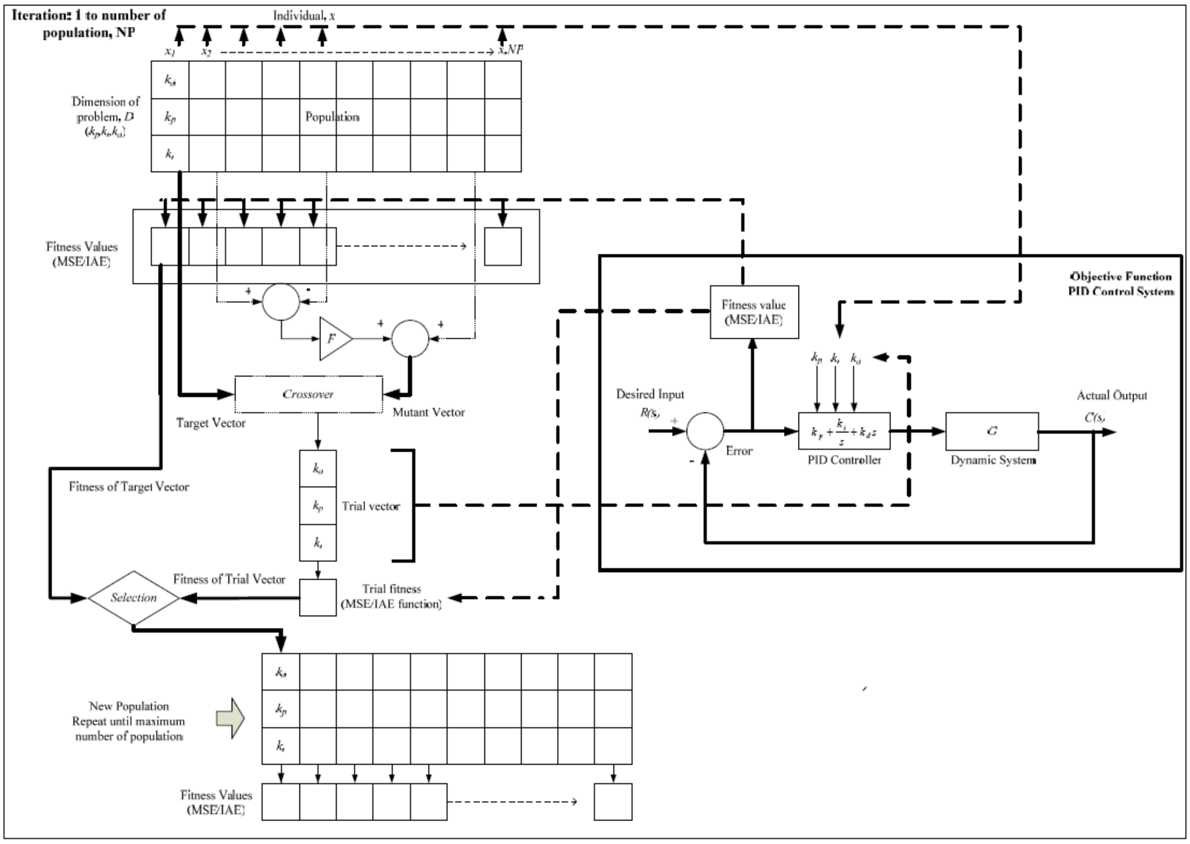


FIGURE 5. Updating for new individual in the population

where the three distinct vectors x_{r1}, x_{r2} and x_{r3} are 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 Figure 6.

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

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

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

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

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 Figure 7.

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

$$\text{if } x_i \notin [L, H], \quad x_i = L + (H - L).\text{rand}_i(0, 1) \tag{11}$$

Equation (11) is purposely used in order to make sure that all the parameter vectors (PID parameters) are within its boundary constraints.

v. 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 the next generation. Figure 8 shows how the selection process is performed.

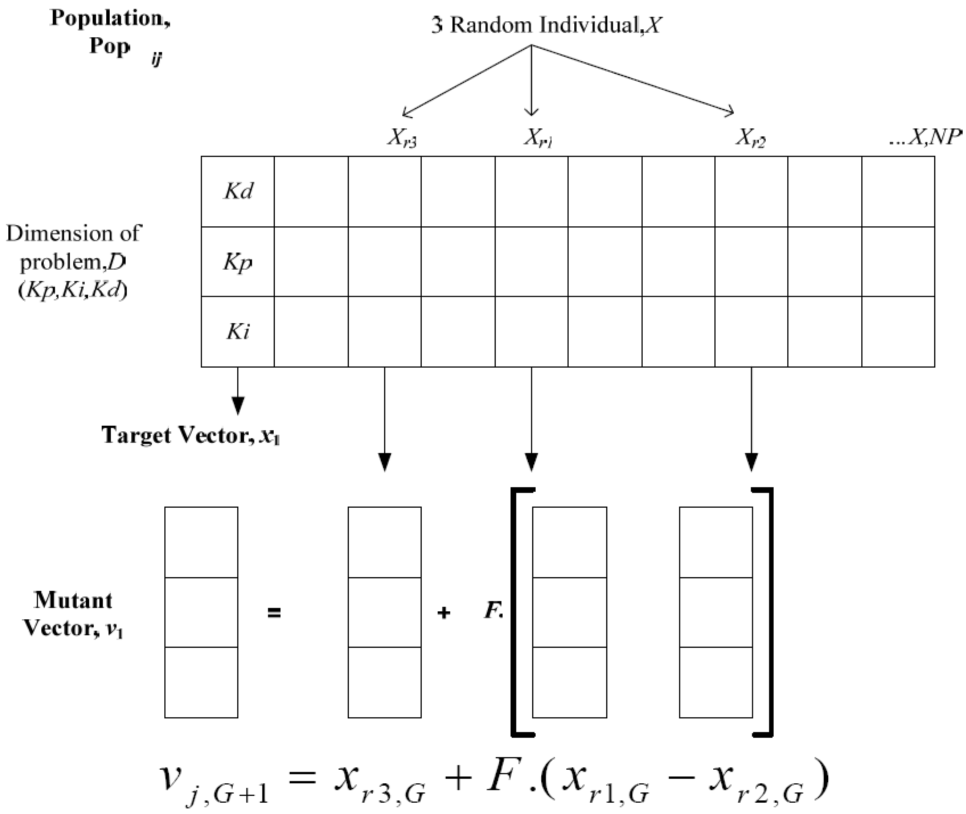


FIGURE 6. Mutation process

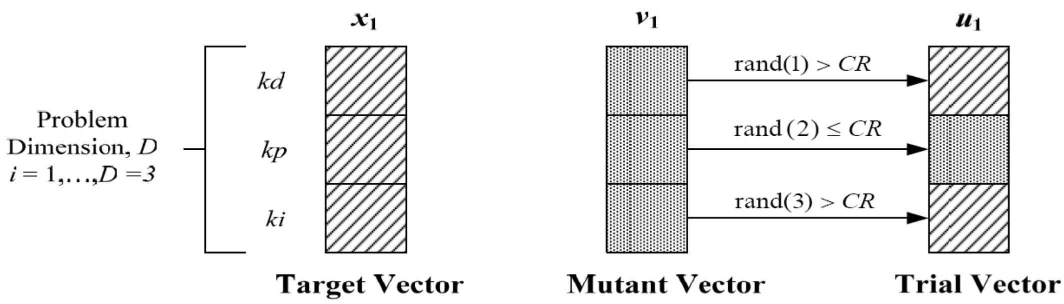


FIGURE 7. Crossover process

vi. Repeat steps 3 to 5 until new population completed. When the first individual in the new population has produced, then the optimization process will repeat for the second individual in population as it now becomes the second target vector in the first generation, $v_{2,1}$. This process will follow steps 3 to 5 until new second individual in the new population is produced. This process will repeat until all the individuals in the new population are updated.

vii. Repeat step 6 until end of generations. The process in step 6 is repeated until the end of generation. At this stage, the optimization process is completed. The global minimum of fitness value is achieved which is referred to optimum parameter of PID controller. Overall optimization process can be seen in Figure 9.

5. **PID Tuning with Ziegler-Nichols Method.** PID tuning using Ziegler-Nichols method [2] is based on the frequency response of the closed-loop system by determining the point of marginal stability under pure proportional control. The proportional

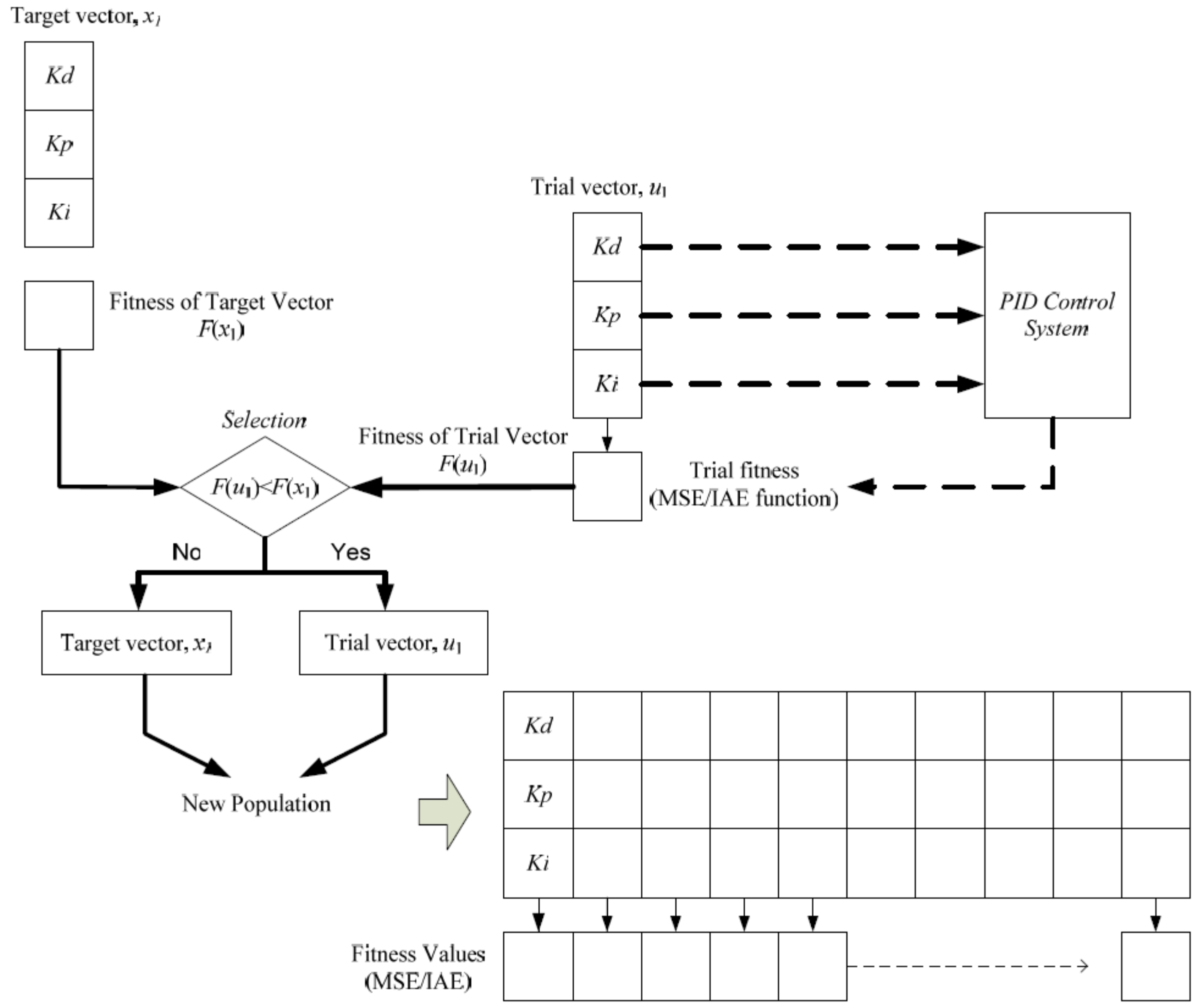


FIGURE 8. Selection process

TABLE 1. Ziegler-Nichols PID tuning parameter

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

gain is increased until the system becomes 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 1.

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.

6. Case Studies.

6.1. Systems considered. In this study, the systems 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

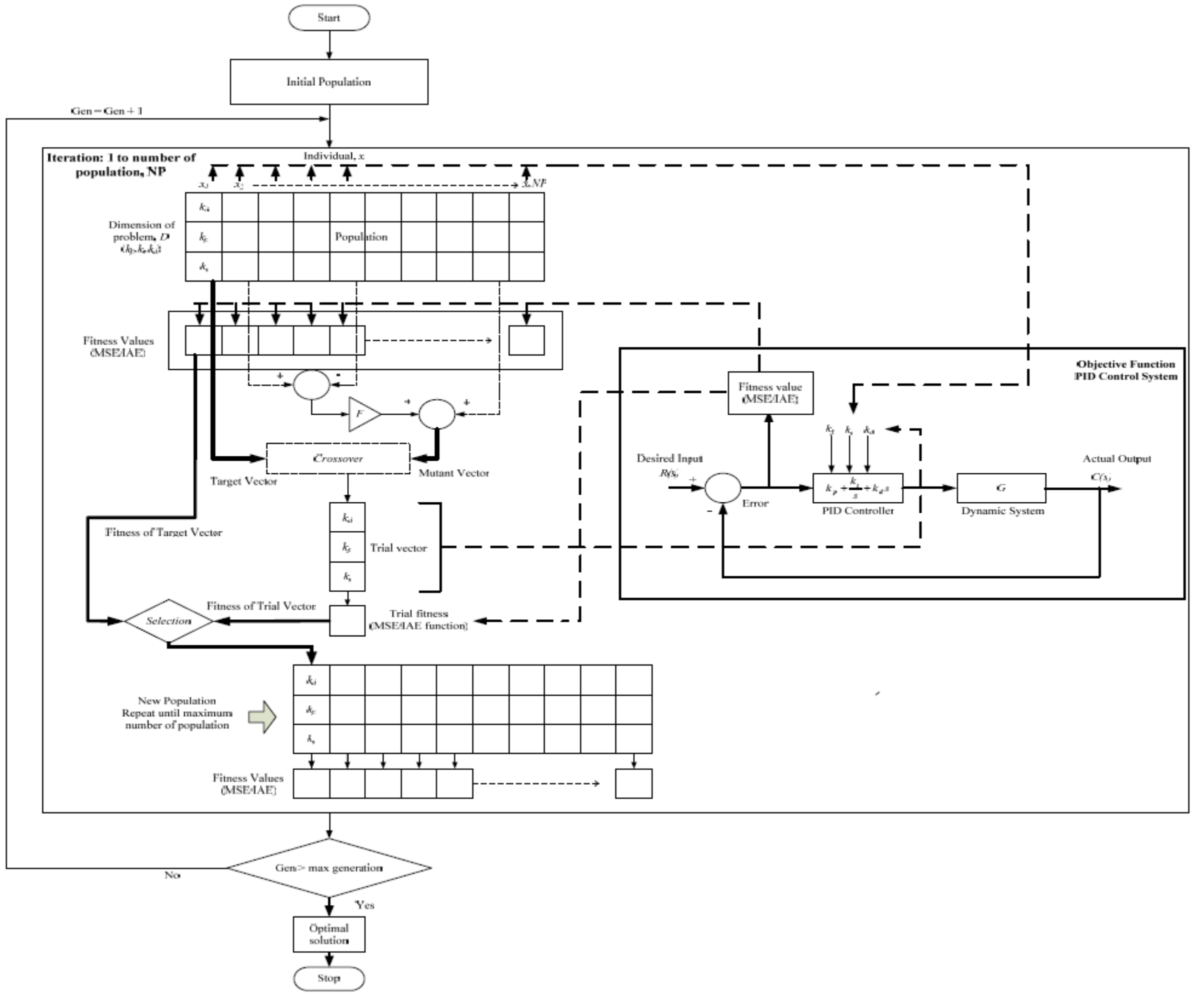


FIGURE 9. Blok diagram of PID tuning using DE algorithms

non-minimum phase system, $G_3(s)$. These systems are:

$$G_1(s) = \frac{25.2s^2 + 21.2s + 3}{s^5 + 16.58s^4 + 25.41s^3 + 17.18s^2 + 11.70s + 1}, \tag{12}$$

$$G_2(s) = \frac{10e^{-1.0s}}{(1 + 8s)(1 + 2s)}, \tag{13}$$

$$G_3(s) = \frac{(1 - 10s)}{(1 + s)^3}. \tag{14}$$

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 as the fitness value of each chromosome/individual in GA and DE. GA and DE will heuristically find the optimum value of the 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.

6.2. **Simulation results.** In this study, PID controller tuning method using heuristic optimization techniques (DE and GA) has to be implemented offline due to the fact that heuristic optimization techniques are stochastic and required computation time spent to compute the fitness of every chromosomes/individuals in the population. Additionally, GA and DE require several iterations to obtain an optimized solution. These require a complete response of the system to be obtained for all chromosomes/individuals before new optimal PID parameters are produced.

Such an offline method requires a plant model in most cases. 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 systems (12)-(14). The parameter values of DE and GA optimization are shown in Table 2, chosen based on [21]. The parameters range for k_p, k_i and k_d as shown in Table 3 are set based on the previous studies [22,23]. The performance of both tuning methods is observed in terms of rise time, settling time, maximum overshoot of the response, and the value of MSE and IAE. Finally, the convergence rate in achieving the global optimum value of the objective function is investigated.

Table 2. 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
Generation number = 100	Generation number = 100

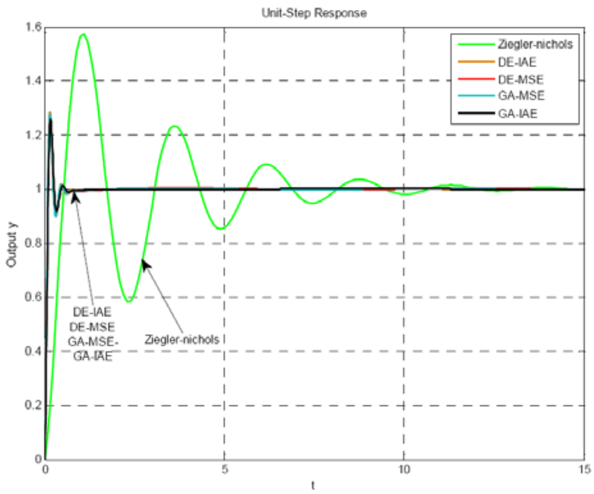
Table 3. PID parameter range

	High Order System		**System with Time Delay**		**Non-minimum Phase System**	
Parameter	**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

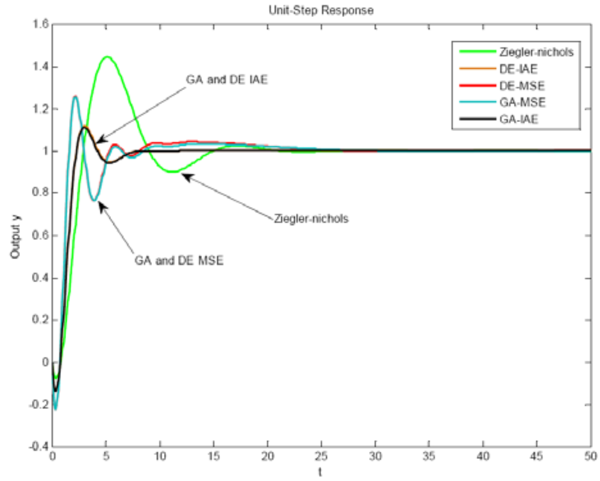
The results for closed-loop step response for DE and GA PID tuning method are shown in Figures 10(a)-10(c) respectively. For Figure 10(a), it is clear that the responses from DE and GA tuning methods are almost indistinguishable. The details of these results are shown in Table 4. The values of rise time, settling time, maximum overshoot, MSE and IAE between DE and GA are almost the same. Both tuning methods give better performance compared with Ziegler-Nichols method. As seen from Table 4, 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 Figure 10(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 5). Even though DE and GA optimized by IAE offered better settling time and overshoot compared with DE and GA optimized by IAE, both methods give superior performance than Ziegler-Nichols method.

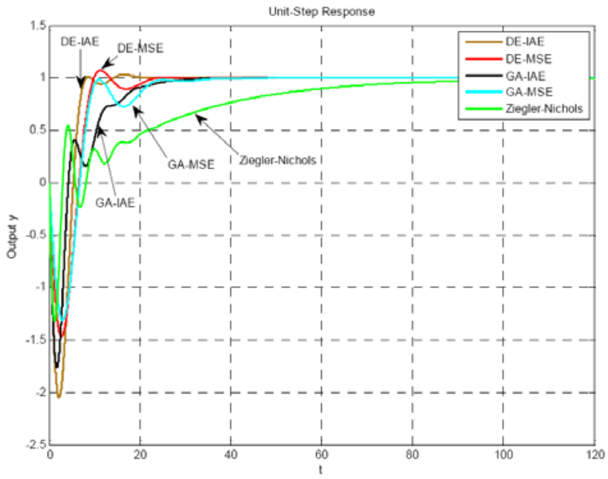
For the case of non-minimum phase system, instead of evaluating in the aspects of rise time, settling time and peak overshoot, the improper undershoot effect should also need to be reduced. All the tuning methods give different responses as shown in Figure 10(c). DE optimized by IAE gives the best values 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 overshoots are produced by GA optimized by IAE, MSE and Ziegler-Nichols except DE optimized by IAE and MSE which



(a) Higher order system



(b) System with time delay



(c) Non-minimum phase system

FIGURE 10. Step response of PID control system

TABLE 4. Comparison performance of ZN, DE and GA for high order system

	High Order System				
	ZN	**DE (MSE)**	**GA (MSE)**	**DE (IAE)**	**GA (IAE)**
MSE	–	0.0011	0.0011	–	–
IAE	–	–	–	11.4246	11.4448
Rise time (s)	0.3904	0.0677	0.0677	0.0617	0.0671
Settling time (s)	9.1802	0.4041	0.4038	0.3987	0.4763
Overshoot (%)	57.5644	27.1927	27.2598	28.4234	28.5395
k_p	3.990	3.5563	3.7500	7.1578	7.5001
k_i	4.218	10.9608	10.9941	11.1596	11.1719
k_d	0.945	18.000	18.000	18.000	18.0000

give small overshoot about 3% and 7% respectively (refer to Table 6). However, in terms of undershoot, Ziegler-Nichols give the lowest undershoot compared with 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, the objective function needs to be modified [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 are used as objective function for tuning the system.

In this study, root locus plot for the system given by systems (12)-(14) is shown in Figures 11(a)-11(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 7.

For overall performance, it can be concluded that DE optimized by MSE gives better transient response. The evolution of PID parameter for non-minimum phase system can be seen in Figures 12(a)-12(d). Figures 12(a) and 12(b) show the convergence profile of

Table 5. Comparison performance of ZN, DE and GA for system with time delay

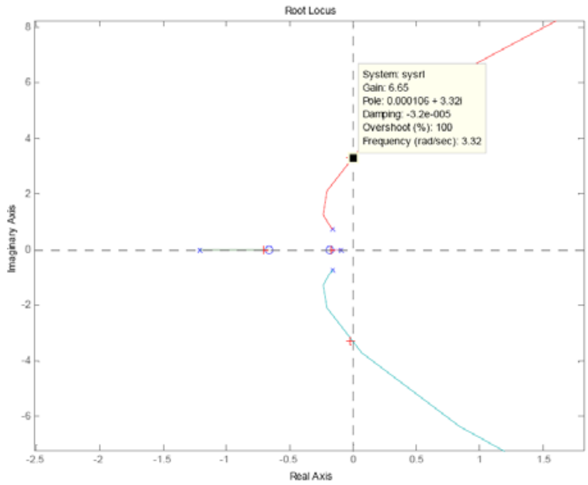
	System with Time Delay				
	ZN	**DE (MSE)**	**GA (MSE)**	**DE (IAE)**	**GA (IAE)**
MSE	–	0.0198	0.0198	–	–
IAE	–	–	–	18.6978	18.7052
Rise time (s)	1.6274	0.6572	0.6646	1.1349	1.1349
Settling time (s)	17.8714	19.2064	18.4438	6.7626	6.7828
Overshoot (%)	44.3219	25.5339	25.2640	11.4190	10.8491
k_p	0.666	0.6099	0.6250	0.6798	0.6718
k_i	0.159	0.1048	0.0937	0.0671	0.0663
k_d	0.694	2.1519	2.1289	1.3419	1.3526

Table 6. Comparison performance of ZN, DE and GA for non-minimum phase system

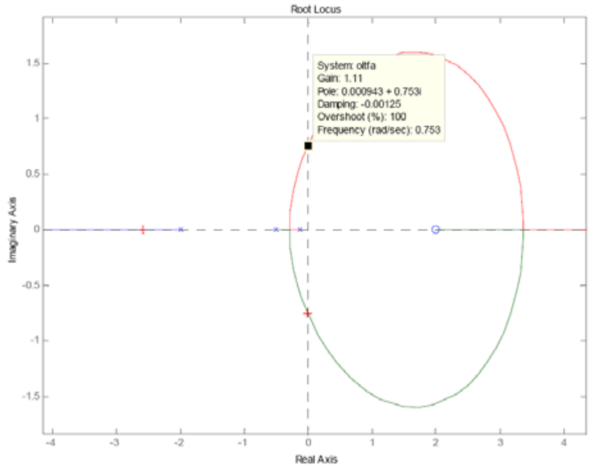
	Non-minimum Phase System				
	ZN	**DE (MSE)**	**GA (MSE)**	**DE (IAE)**	**GA (IAE)**
MSE	–	0.0189	0.0218	–	–
IAE	–	–	–	1333.5	1601.9
Rise time (s)	57.5505	2.4486	2.7176	1.8575	14.9517
Settling time (s)	79.2055	20.5469	22.6840	11.7506	23.5574
Overshoot (%)	–	7.0137	–	2.9414	–
Undershoot (%)	131.8	146.8	157.4	205.7	167.3
k_p	0.1550	0.1903	0.1884	0.2071	0.1875
k_i	0.0320	0.0695	0.0624	0.0765	0.0624
k_d	0.1880	0.1035	0.0662	0.1723	0.1875

Table 7. Ziegler-Nichols PID tuning values

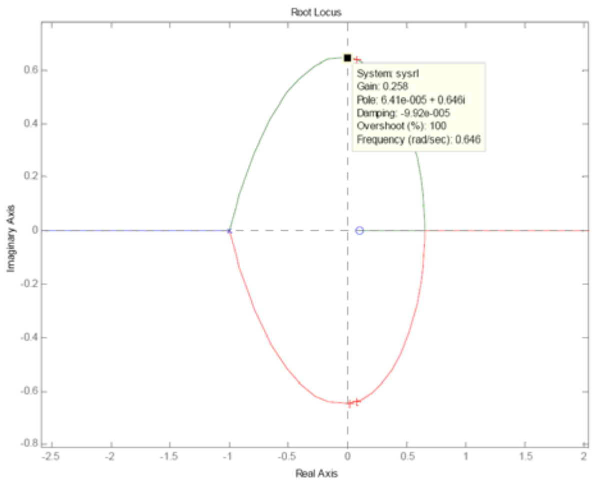
	High Order System	**System with Time Delay**	**Non-minimum Phase System**
Ultimate gain, k_u	6.650	1.110	0.258
Ultimate period, t_u	1.893	8.344	9.726
k_p	3.990	0.666	0.155
k_i	0.946	4.172	4.863
k_d	0.237	1.043	1.216



(a) High order system



(b) System with delay

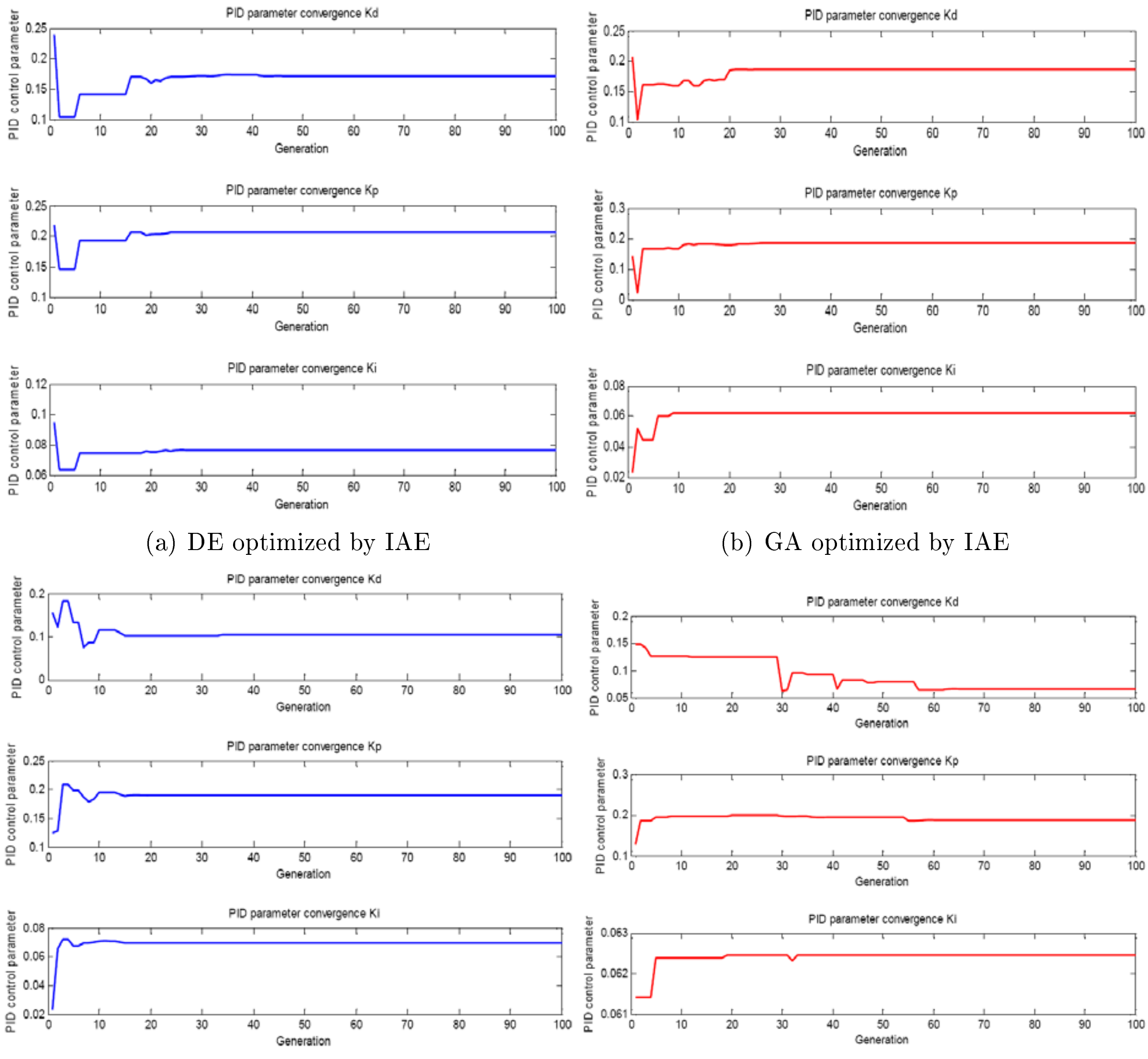


(c) Non-minimum phase system

FIGURE 11. Root locus plot

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 Figures 12(c) and 12(d)). Parameter k_d is 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 the 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 Figures 13-15. From these figures, it is observed that DE is significantly consistent than GA in searching for minimum fitness function. Convergence test done for DE and GA settled almost at the same generation. From the overall results, it is observed that, DE algorithm outperforms GA in terms of its consistency in constantly achieving the globally minimum fitness value.



(a) DE optimized by IAE

(b) GA optimized by IAE

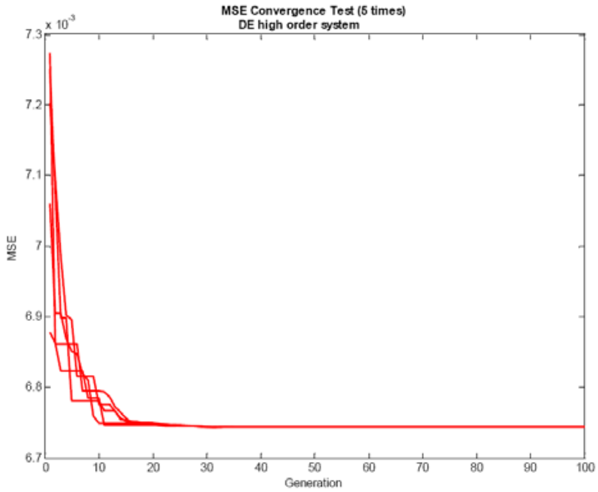
(c) DE optimized by MSE

(d) GA optimized by MSE

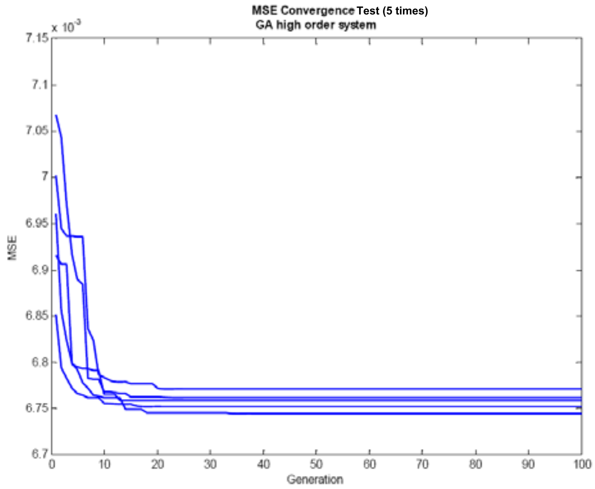
FIGURE 12. Convergence profile of PID parameter

7. Practical Concept for PID Tuning Using Heuristic Optimization Methods. Practical implementation of PID tuning method can be done using concept of self-tuning control technique [25]. Figure 16 shows the block diagram representing the methodology that is generally used for tuning controllers designed for systems with time varying parameters [26]. It is required that the system parameters are identified before the controller can be tuned. When the system has been identified on-line, then optimization process will run to find the optimum control parameter with minimum fitness value such as MSE or IAE. When optimization has been completed, the on-line controller will be updated with the new control parameters.

8. Conclusions. 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

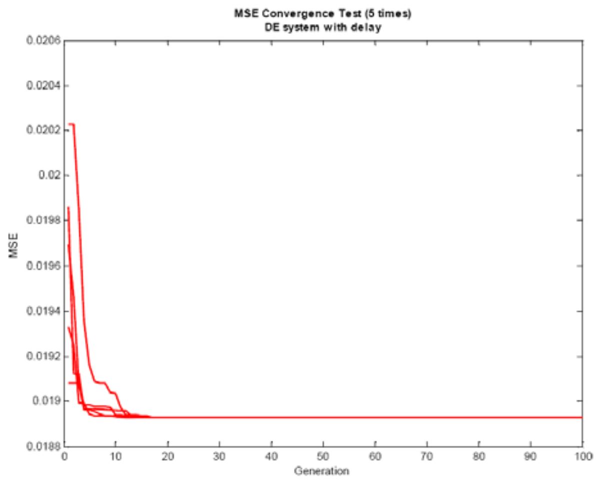


(a) DE

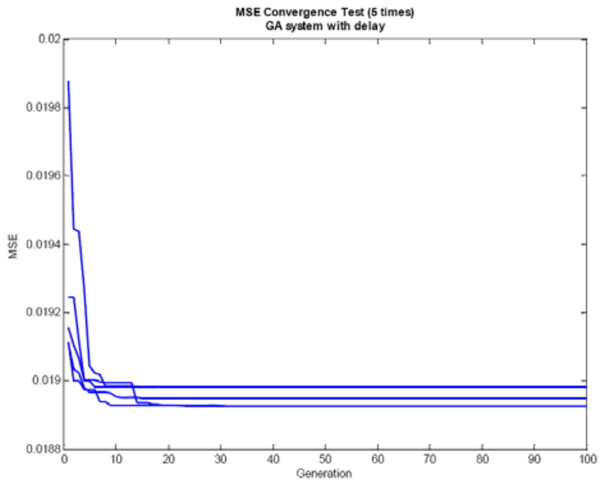


(b) GA

FIGURE 13. Convergence test for a high order system



(a) DE

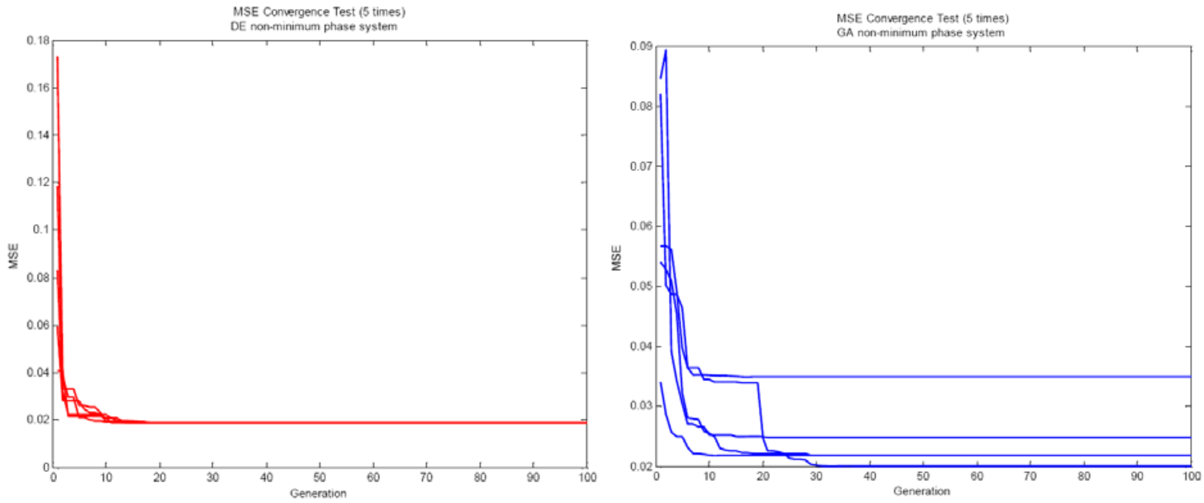


(b) GA

FIGURE 14. Convergence test for system with delay

MSE and IAE respectively. However, for non-minimum phase system, DE optimized by MSE gives better performance with regards to the trade-off between settling time, maximum overshoot and undershoot. In terms of reliability, DE offers consistency in achieving its globally minimum of fitness value. However, the convergence rates for all the trials for higher order system, system with time delays and non-minimum phase system are almost the same. Practical implementation of PID tuning has been explained briefly in the case study.

Acknowledgements. The authors wish to thank the Universiti Malaysia Perlis (UNIM AP) and Universiti Teknologi Malaysia (UTM) for providing the fund and facilities to conduct this research. This research is funded by Universiti Teknologi Malaysia Research University Grant Vote 00J20.



(a) DE (b) GA

FIGURE 15. Convergence test for non-minimum phase system

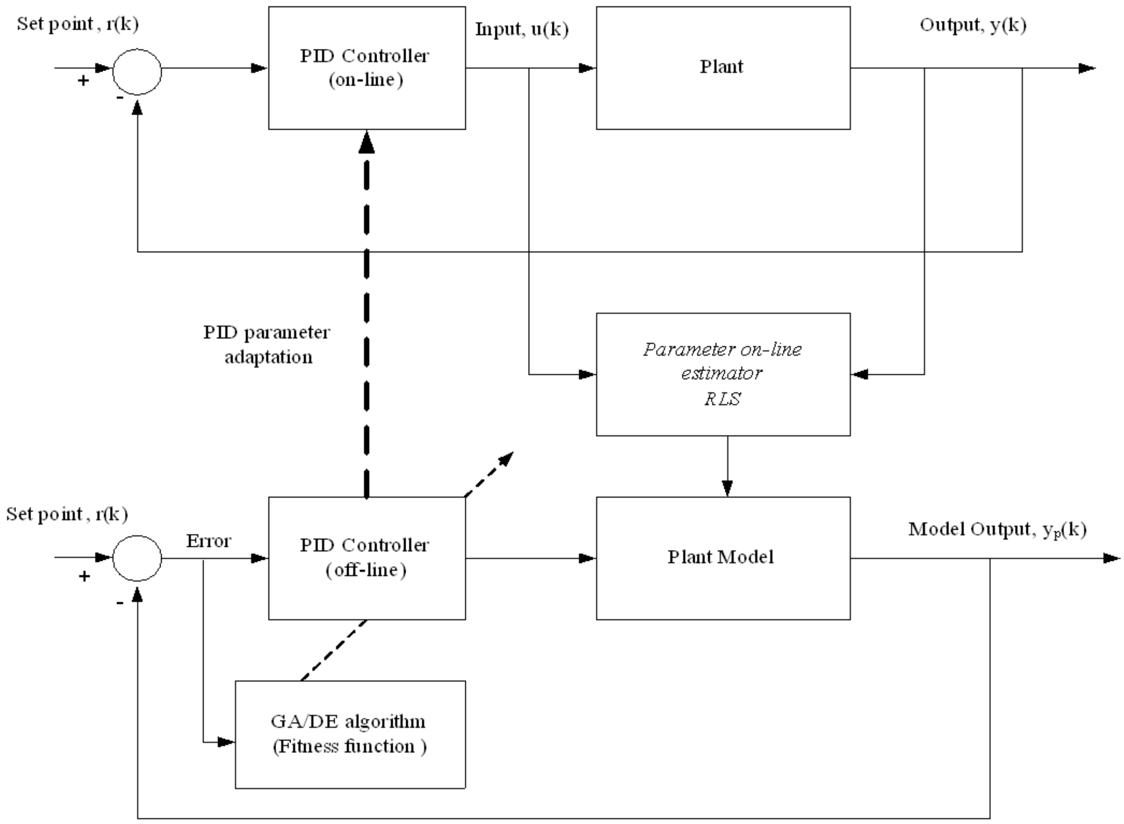


FIGURE 16. Block diagram for practical PID online tuning method

REFERENCES

[1] K. J. Astrom and T. Hagglund, *Automatic Tuning of PID Controllers: Instrument Society of America*, 1988.
[2] J. G. Ziegler and N. B. Nichols, Optimum settings for automatic controllers, *Trans. ASME*, vol.64, no.8, pp.759-768, 1942.
[3] G. H. Cohen and G. A. Coon, Theoretical consideration of retarded control, *Trans. ASME*, vol.75, pp.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*, vol.6, no.3, pp.41-46, 2006.
[5] T. Jain and M. J. Nigam, Optimisation of PD-PI controller using swarm intelligence, *Journal of Theoretical and Applied Information Technology*, vol.4, no.11, pp.1013-1018, 2008.
[6] K. J. Astrom, T. Hagglund, C. C. Hang and W. K. Ho, Automatic tuning and adaptation for PID controllers – A survey, *Control Engineering Practice*, vol.1, no.4, pp.699-714, 1993.
[7] K. J. Astrom and T. Hagglund, *PID Controllers: Theory, Design, and Tuning: Instruments Society of America*, 1995.
[8] P. B. de Moura Oliveira, Modern heuristics review for PID control optimization: A teaching experiment, *Proc. of International Conference on Control and Automation*, Budapest, Hungary, pp.828-833, 2005.
[9] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*, McGraw-Hill, London, 1995.
[10] J. S. Kim, J. H. Kim, J. M. Park, S. M. Park, M. Y. Choe and H. Heo, Auto tuning PID controller based on improved genetic algorithm for reverse osmosis plant, *International Journal of Intelligent Systems and Technologies*, vol.3, no.4, pp.232-237, 2008.
[11] F. Yin, J. Wang and C. Guo, *Design of PID Controllers Using Genetic Algorithms Approach for Low Damping Slow Response Plants*, Springer-Verlag, Berlin Heidelberg, 2004.
[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 optimization, *Proc. of the 3rd UKSim European Symposium on Computer Modeling and Simulation*, Athens, Greece, pp.355-360, 2009.
[13] J. Andre, P. Siarry and T. Dognon, An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization, *Advances in Engineering Software*, vol.32, no.1, pp.49-60, 2001.
[14] R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, no.4, pp.341-359, 1997.
[15] R. Dong, Differential evolution versus particle swarm optimization for PID controller design, *Proc. of the 5th International Conference on Natural Computation*, Tianjin, pp.236-240, 2009.
[16] Y. Luo and X. Che, Tuning PID control parameters on hydraulic servo control system based on differential evolution algorithm, *Proc. of the 2nd International Conference on Advanced Computer Control*, Shenyang, pp.348-351, 2010.
[17] L. D. Arya, H. K. Verma and C. Jain, Differential evolution for optimization of PID gains in automatic generation control, *International Journal on Computer Science and Engineering*, vol.3, no.5, pp.1848-1856, 2011.
[18] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co., Inc., 1989.
[19] J. Zhong, X. Hu, M. Gu and J. Zhang, Comparison of performance between different selection strategies on simple genetic algorithms, *International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, Vienna, pp.1115-1121, 2005.
[20] R. Storn, On the usage of differential evolution for function optimization, *Proc. of the Fuzzy Information Processing Society*, Berkeley, CA, USA, pp.519-523, 1996.
[21] N. Karaboga, Digital IIR filter design using differential evolution algorithm, *EURASIP Journal on Applied Signal Processing*, vol.2005, no.8, pp.1269-1276, 2005.
[22] G. M. de Almeida, V. V. R. e Silva, E. G. Nepomuceno and R. Yokoyama, Application of genetic programming for fine tuning PID controller parameters designed through Ziegler-Nichols technique, *LNCS*, vol.3612, 2005.
[23] P. Chou and T. Hwang, Design of PID controllers using genetic algorithms approach for low damping, slow response plants, *LNCS*, vol.3174, 2004.
[24] W. Jie-sheng, Z. Yong and W. Wei, Optimal design of PI/PD controller for non-minimum phase system, *Transactions of the Institute of Measurement & Control*, vol.28, no.1, 2006.
[25] W. Lin, C. Ou, P. X. Liu and M. Q.-H. Meng, Self-tuning regulator for order-varying systems based on particle swarm optimization, *International Conference on Mechatronics and Automation*, Harbin, China, pp.1120-1124, 2007.
[26] M. Ahmad, L. Zhang and J. C. Readle, Online genetic algorithm tuning of a PI controller for a heating system, *Proc. of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, Glasgow, UK, pp.510-515, 1997.