# Design of Fractional Order PID Controller Based on Genetic Algorithms

**Dr. Mazin Z. Othman and Dr. Emad A. Al-Sabawi**
**Computer and Information Engineering Department**
**College of Electronics Engineering/ University of Mosul -Iraq**

## Abstract

Fractional Order Proportional-Integral-Derivative (FOPID) controller is based on integration and differentiation of non-integer order. It is usually denoted by $PI^{\mu}D^{\lambda}$ (0 < µ < 1 and 0 < λ < 1) . In this work the gain values of the proportional, Integral, and derivative (i.e $K_p, K_i, K_d$) as well as the values of µ and λ are obtained using Genetic Algorithms (GAs).The tuning procedure is based on the principle of model reference control. Illustrative example is presented in which FOPID controller is designed and compared with Integer Order PID (IOPID) ones. It was shown that FOPID controllers gave more freedom in faithfully following the dynamics of the reference model.

*Key Words*: Fractional order, PID tuning rules, Genetic Algorithms, Model reference control.

# تصميم مسيطر تناسبي تكاملي تفاضلي كسري المرتبة بالاعتماد على الخوارزمية الجينية

## المستخلص

المسيطر التناسبي التكاملي التفاضلي كسري المرتبة (FOPID) أساسه التفاضل والتكامل بمرتبة ذات عدد غير صحيح . هو عادة ما يعرف بالرمز $PI^{\mu}D^{\lambda}$ حيث ( 1> µ >0 و 0 < λ > 1). في هذا البحث تم احتساب قيم معاملات الكسب للجزء التناسبي والتكاملي والتفاضلي بالإضافة إلى قيمتي µ و λ باستخدام الخوارزمية الجينية . إن عملية التنغيم تمت وفق مبدأ سيطرة النموذج المرجعي (model reference control). تم تقديم مثال توضيحي صمم فيه مسيطر نوع FOPID وتم مقارنته مع مسيطر تناسبي تكاملي تفاضلي صحيح المرتبة (IOPID) . هذه المقارنة وضحت أن المسيطر نوع FOPID يعطي حرية تصميمية لزيادة دقة تتبع النظام لديناميكية المرجع النموذج.

**Received: 3 – 4 - 2011** **Accepted: 22 – 8 - 2011**

## 1. Introduction:

Integer Order PID controllers are widely used in industry because they are simple, effective, robust, and easily tuned. Many conventional tuning methods were developed during last 60 or 70 years either using frequency domain approach (*Astrom-Hagglund* tuning algorithm [1]), or time domain approach (*Ziegler-Nichols*). Evolutionary techniques such as Genetic Algorithms are also played tremendous rules in finding optimal tuning gains according to the way that the control-engineering designer was focus on [1].

However, the idea of using fractional-order calculus in developing fractional order controllers for the dynamic system control is well addressed in [2] and [3]. Advantages of using fractional-order PID controller have been introduced in a number of publications. It was claimed that, many specifications can be met by the closed-loop system (increasing design flexibility), because the fractional-order PID controller has five tuning parameters [4].

Different approaches were developed for tuning the FOPID. Experimental tuning rules for fractional PID controllers, similar to the 1st and the 2nd sets of tuning rules proposed by Ziegler and Nichols for integer PIDs were proposed in [5]. Moreover, based on frequency domain,[6] states that the designed fractional order PID controller should meet the stability robustness of the feedback control loop according to the desired gain margin and phase margin. The Quantitative Feedback Theory (QFT) was used as a base to tune the FOPID controller for a time-delayed first order plant [7]. FOPID is also used to enhance the performance of plants having nonlinear characteristics (backlash and saturation) [8]. Even *soft computing* techniques (such as Neural Networks and Fuzzy Logic Systems) are used in adaptively producing the FOPID tuning gains (see for example [9], [10]).

However, many researchers to find the optimal tuning parameters of the FOPID use the Genetic Algorithms (GAs). Different approaches were proposed in the context of the GAs to choose an objective function that really meets the desired specifications and obeys the practical constraints.

In this work, the model reference approach is adopted in order to develop an objective function for the GAs. By choosing a model, all time domain and frequency domain specifications can entirely met by the designer.

## 2. Fractional Order PID Controllers:

Fractional order control systems are described by fractional order differential equations. Fractional calculus allows the derivatives and integrals to be arbitrary number.

There are several definitions of fractional derivatives [10]. Grunwald-Letnikov definition is perhaps the best known one due to its most suitable for the realization of discrete control algorithms. The $m_{th}$ order fractional derivative of continuous function f (t) is given by:

$$D^m f(t) = \frac{d^m f(t)}{dt^m} = \lim_{h \to 0} h^{-m} \sum_{j=0}^{[x]} (-1)^{-j} \binom{m}{j} f(t - jh)) , \qquad \text{---------------} (1)$$

where $[x]$ is a truncation and $x = \frac{t-m}{h}$. $\binom{m}{j} = \frac{m(m-1).......(m-1+j}{j!}$ .The general calculus operator, including fractional order and integer order is defined as:

$$aD_t^m = \begin{cases} d^m / dt^m \cdots\cdots \Re(m)\rangle 0 \\ 1 \cdots\cdots\cdots\cdots \Re(m) = 0\,) \\ \int_a^t (d\tau)^{-m} \cdots\cdots \Re(m)\langle 0 \end{cases} \qquad \text{-------------------- (2)}$$

where a and t are the limits related to operation of fractional differentiation, $m$ is the calculus order.

Accordingly, the differential equation of the fractional order controller $PI^{\mu}D^{\lambda}$ is defined as;

$$u(t) = K_p e(t) + K_i D^{-\mu} e(t) + K_d D^{\lambda} e(t) \qquad \text{------------------ (3)}$$

The continuous transfer function of FOPID is obtained through Laplace transform, which is given by:

$$U(s) = (K_p + \frac{K_i}{s^{\mu}} + K_d s^{\lambda})E(s) \qquad \text{----------------------- (4)}$$

## 3. Discretized model of the continuous FOPID controller:

The above FOPID controller (4) can be approximated using discretization methods. In the following s is approximated by using the bilinear transform, the Al-Alaoui transform, and the phase enhanced Al-Alaoui transform [11], represented below by equations (5), (6), and (7) respectively.

$$s \approx \frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})} \qquad \text{---------------------- (5)}$$

$$s \approx \frac{8}{7T}\frac{(1-z^{-1})}{(1+z^{-1}/7)}) \qquad \text{----------------------- (6)}$$

$$s \approx \frac{8}{7T}\frac{z^{0.5}(1-z^{-1})}{(1+z^{-1}/7)} = \frac{8}{7T}\frac{(1-z^{-1})}{z^{-0.5}(1+z^{-1}/7)} \qquad \text{----------------------- (7)}$$

The phase enhanced Al-Alaoui transform is obtained by a half-sample advance of the Al-Alaoui transform, which is achieved by multiplying (2) by $Z^{0.5}$, or dividing it by $Z^{-0.5}$. The half-sample advance, shown in (7), changes the almost linear phase of (6) to almost 90 degrees, while the magnitude remains the same [11].

*Indirect Discretization*

Indirect discretization consists of two steps. In the first step a rational analog transfer function that approximates the irrational transfer function of the fractional controller is obtained. In the second step an analog to digital transformation of the rational analog approximation is done.

1) *The Analog Rational Approximation*

The approximation is carried out in the following two steps:

- Determine a frequency range [ω<sub>A</sub>, ω<sub>B</sub>] where it is needed to approximate the fractional order transfer function.
- Perform an analog integer-order approximation of the fractional-order transfer function

$$s^{-r}_{[\omega_A-\omega_B]} \cong \frac{\omega_B(\omega_A+s)}{(\omega_A)^n(rs^2+\omega_B s+(1-r)\omega_A\omega_B)}\left(\frac{1+\dfrac{s}{\omega_B}}{1+\dfrac{s}{\omega_A}}\right)^r \quad \text{-------------------- (8)}$$

Then equation (8) is expanded using continued fraction expansion CFE, Taylor series expansion, or the method of recursive poles and zeros to obtain an analog approximation. The analog approximation with recursive poles and zeros is summarized below [11]:

$$\left(\frac{1+\dfrac{s}{\frac{d}{b}\omega_A}}{1+\dfrac{s}{\frac{b}{d}\omega_B}}\right)^r = \lim_{N\to\infty}\prod_{k=-N}^{k=N}\frac{1+\dfrac{s}{\omega_k^{'}}}{1+\dfrac{s}{\omega_k}} \quad \text{-------------------- (9)}$$

$$\omega_k^{'} = \left(\frac{d}{b}\omega_A\right)^{\frac{r-2k}{2N+1}} \quad \text{------------------- (10)}$$

$$\omega_k = \left(\frac{b}{d}\omega_B\right)^{\frac{r+2k}{2N+1}} \quad \text{------------------- (11)}$$

$$s^r \approx K\left(\frac{ds^2+bs\omega_B}{d(1-r)s^2+bs\omega_B+dr}\right)\prod_{k=-N}^{k=N}\frac{s+\omega_k^{'}}{s+\omega_k} \quad \text{------------------- (12)}$$

$$K = \left(\frac{d}{b}\omega_A\right)^r\prod_{k=-N}^{k=N}\frac{\omega_k}{\omega_k^{'}} \quad \text{------------------- (13)}$$

In this work b = 10 , d = 9 , $\omega_A$=0.01 rad/sec. , $\omega_B$=10000 rad/sec.,and N=7 as in [11].These parameters are selected to ensure the dynamic range of most industrial plants. The procedure for the approximation can be briefly summarized in the following steps:-

a) The frequency range [$\omega_A$, $\omega_B$] and N are given.

b) Based on the fractional order r, calculate $\omega_k'$ and $\omega_k$ according to (10) and (13).

c) Compute K from (13).

d) Obtain the approximate rational transfer function from (12) to replace $s^r$ .

2) *Discretizing the Analog Approximation*

To discretize the analog rational approximation of (12) select suitable s-to-z transforms. The bilinear transformation, [equation (1)] is employed by replacing s in (12) by the right hand side of equations (1).

## 4. The Genetic Algorithm [12], [13]:

The Genetic Algorithms (GAs) is a direct random search technique, which can find the optimal solution in complex multi-dimensional search space. It is a model of natural evolution operators, which manipulate individuals in a population over several generations to improve their fitness gradually. These individuals that, represent a candidate solution to the problem under consideration, often encoded as a bit string. Therefore, GA does not involve much experience about the complexity of the problem rather than deals with codes that represent the key parameters of the problem.

There are two basic issues in the GA, the first one is how to code the problem this is called *coding*. Two common ways for coding, they are either real or binary coding. The GA requires a set of possible solutions called initial populations which are randomly produced. The second issue is how to qualify each individual (string) which is called the *fitness evaluation*. This step depends on the problem nature. It is either mathematic equation or a rule-based procedure or in some cases a combination of both. These two issues are problem dependent.

The remaining part of the GA is problem independent, which consists mainly of the GA operators. Three operators are basically important, that are *selection (reproduction), crossover, and mutation*. The selection procedure is to reproduce more strings whose fitness functions are higher than those whose fitness functions are low. This is important to drive the search towards better and better solution. Reproducing can be implemented by many methods (for example *Roulette Wheel, Tournament…etc.*).The crossover is important to create new individuals from already exist ones. The mutation is an operator that avoids the search process to be trapped at a local area in the hyper search space by exchanging bits of a certain string.

The following algorithm clarifies the main steps in GA:

*Step-1*: Create an initial population of strings.
*Step-2*: Calculate the fitness of each string.
*Step-3*: While ( an acceptable solution is not found) OR ( does not
exceed maximum number of generation)

- Perform reproduction for next generation.
- Perform crossover between parents to create new offspring.
- Apply mutation with certain probability.
- Calculate the fitness of each offspring.

End while

## 5. Genetic-Based Tuning of FOPID Controllers:

The undertaken design control problem is to find the optimal tuning gains for the FOPID given in equation (4) that ensures certain objective function. The objective function is suggested here to force the plant to behave similar to a predefined model (model reference approach).Therefore, each GA solution (string $s_i$) has to minimize the following objective function:

$$J(s_i)=\sum_{i=0}^{T_o} j_{s_i}(i) \qquad \text{------------------ (14)}$$

where,

$$j_{s_i}(t)=\begin{cases} -\left|e_{s_i}(t)\right|-\alpha(s_i) & if\ \left|e_{s_i}(t)\right|>\varepsilon \\ \varepsilon-\left|e_{s_i}(t)\right|-\alpha(s_i) & if\ \left|e_{s_i}(t)\right|\le\varepsilon \end{cases} \qquad \text{------------------ (15)}$$

$$\alpha(s_i)=\begin{cases} 0 & if\ \ s_i\ represents\ a\ closed-loop\ stable\ system \\ M & elsewhere \end{cases} \qquad \text{------------------ (16)}$$

$T_o$ is the number of the collected data sample, $\varepsilon$ is the minimum accepted error ( set to be 0.02), and $e_{s_i}(t)$ is the instantaneous error between the plant output and the model output for $s_i$ string as

$$e_{s_i}(t)=y_p(t)-y_m(t) \qquad \text{---------------- (17)}$$

In equation (17), $y_p(t)$ is the instantaneous plant output for $s_i$ string parameters and $y_m(t)$ is the instantaneous model output. In equation (16) $\alpha(s_i)$ represents a penalty function for $s_i$ string that gives an unstable solution (unwanted solution). In this work the unstable solution is that string which have fitness value more than M (M is set to be $10^6$ indicating that instability occurrence).

Binary coding is used to represent the FOPID tuning parameters as shown in Fig. (1)

| $\mu$ | $\lambda$ | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|---|
| 9 bit | 9 bit | 25 bit | 25 bit | 25 bit |

Fig.(1) : FOPID string coding

The whole block diagram representing the proposed design method is shown in Fig. (2).For 0 < μ < 1 and 0 < λ < 1, nine bits representation is used to ensure resolution of $2^{-9} = 0.002$. The ranges of $K_p, K_i, and\ K_d$ are selected to be (0 to 128) with 7 bits represented the integer parts and 18 bits for fractional parts ( i.e with resolution $2^{-18} = 3.8\times10^{-6}$).

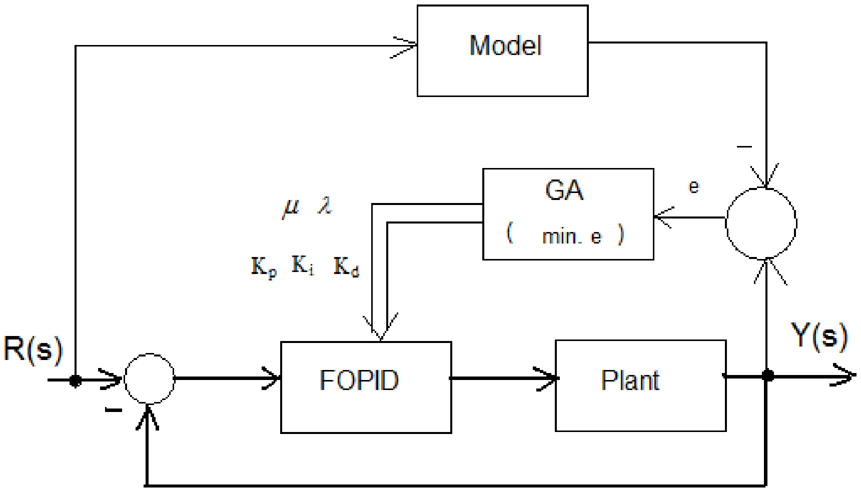

Fig. (2): The block diagram of the proposed design method

## 6. Illustrative Example:

In order to illustrate the power of GAs in finding the optimal tuning parameters of the FOPID controller, and for the comparison purposes, the same example that was considered in [14] will be used here. In this paper, a position PID controller for a shunt D.C. motor was tuned using GAs. The following transfer function described the system under control:-

$$\frac{\Theta(s)}{V(s)} = \frac{K_b}{JL_a s^3 + (R_a J + BL_a)s^2 + (K_b^2 + R_a B)s} \qquad \text{-------------------} \ (18)$$

Where, Θ is the angular displacement of the motor shaft in radians,
V is the armature voltage in volts,
$R_a$ is the armature resistance in ohm,
$L_a$ is the armature inductance in Henry,
$K_b$ is the back emf constant in volt/(rad/sec),
J is the moment of inertia of motor and load in Kg-m2/rad,
B is the frictional constant of motor and load in N-m/(rad/sec).
The motor ratings and parameters are shown in table (1).

Table (1): D.C. motor ratting and specifications

| parameter | value | unit |
|---|---|---|
| Horse power | 2 | hp |
| rated armature voltage | 230 | Volt |
| rated armature current | 8.5 | Amp. |
| rated speed | 1500 | r.p.m. |
| armature resistance $R_a$ | 2.45 | ohm |
| armature inductance $L_a$ | 0.035 | H |
| frictional constant B | $0.5*10^{-3}$ | N-m/(rad/sec) |
| back emf constant $K_b$ | 1.2 | volt/(rad/sec) |
| moment of inertia J | 0.022 | Kg-m2/rad |

The system transfer function becomes [14]:

$$\frac{\Theta(s)}{V(s)}=\frac{1.2}{0.00077s^3+0.0539s^2+1.441s} \qquad \text{--------------------- (19)}$$

It is required to design FOPID controller that gives a closed loop step response as much as close to the step response of the following model:

$$G_{model}(s)=\frac{2500}{(s+50)^2} \qquad \text{--------------------- (20)}$$

The simulation of this system is accomplished under MATLAB environment. The sampling time is chosen to be 0.001 Sec. As indicated in [14], the searching process of the GAs is started from the PID parameters that are obtained using Ziegler-Nichols (Z-N) method and random values for $\mu$ and $\lambda$. Moreover, the GAs parameters are as follows:

The population size is 1200. The maximum number of generation is 70. The number of collected data sample is $T_o = 1000$ sample. The crossover probability, and the mutation probability are chosen by trial and error to be 0.2 and 0.35, respectively (the crossover probability and the mutation probability values depend on the complexity degree of the hyper surface that relates all the parameters to be optimized [13]).

The tuning parameters obtained by the GAs for the FOPID are listed in table (2) .For comparison purposes the tuning parameters obtained in [14] for the genetically tuned PID and the tuning parameters obtained by the Ziegler-Nichols (Z-N) method as initial values for the GAs are also listed.

Table (2): The FOPID and PID tuning parameters

| methods | $\mu$ | $\lambda$ | $K_p$ | $K_i$ | $K_d$ |
|---|---|---|---|---|---|
| Ziegler-Nichols tuning method | 1.0 | 1.0 | 18 | 0.045 | 0.0182 |
| Genetic tuned PID | 1.0 | 1.0 | 19.88 | 0.1376 | 0.5578 |
| Genetic tuned FOPID | 0.60742 | 0.99804 | 25.9428 | 0.01038 | 34.6196 |

The step responses for the three cases outlined in table (2) are illustrated in Fig.(3) together with that of the desired model. Moreover, table (3) compares the rise time, the settling time and the percentage overshoot for the genetically tuned PID and FOPID.

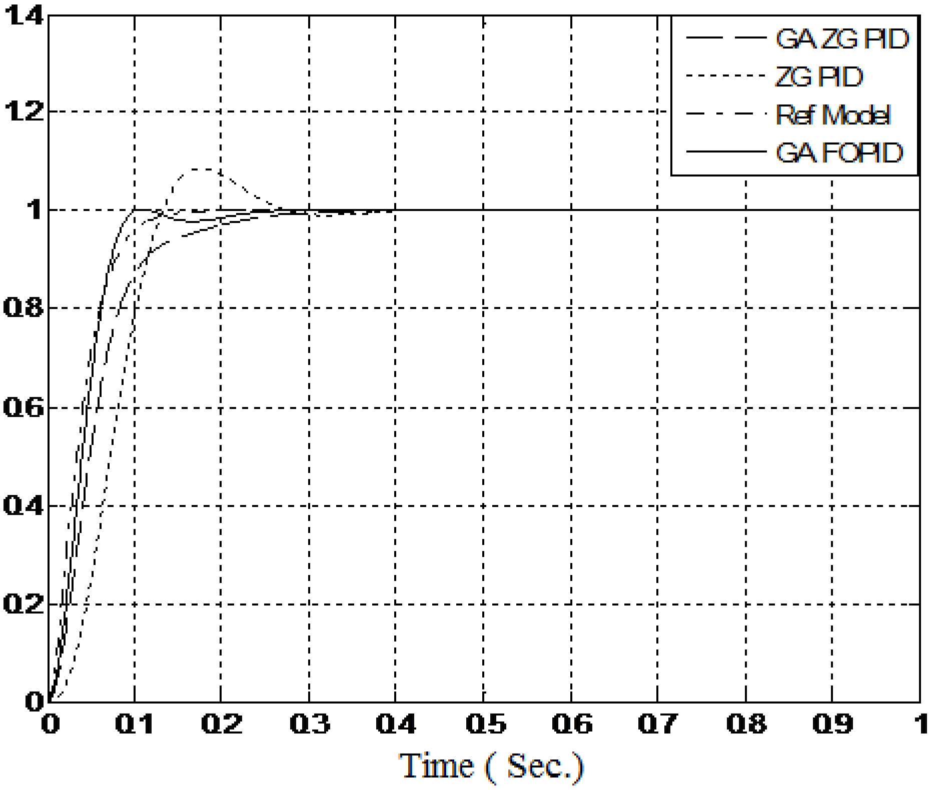

Fig.(3): The position step responses under Z-N tuning ,genetically tuned FOPID and PID controllers.

Table (3): The FOPID and PID step response parameters

| methods | rise time sec. | settling time sec. | overshoot % |
|---|---|---|---|
| Genetic tuned PID | 0.14 | 0.3 | 9 % |
| Genetic tuned FOPID | 0.1 | 0.2 | 0.05 % |

## 7. Conclusions

It has been shown that the FOPID controller tuned by genetic algorithms increases the flexibility area of choosing its parameters in order to have more faithful closed loop model matching characteristics. This was illustrated clearly by designing FOPID and PID controllers for the same most widely used plant in industry, which was the position control of DC motor. It has been concluded that the summation of absolute error for 1000 samples between the system response and that of the desired model in the case of FOPID was (6.9485), while it was (20.4535) for the case of classical PID. This indicates that the model-following characteristic is reached more precisely in the case of FOPID in comparison to that of PID controller.

## References:

[1] Astrom K. J. and Hagglund T., "PID Controllers: Theory, Design and Tuning", Research Triangle Park, Instrument Society of America, 1995.

[2] Oustaloup A, Mathieu B, Lanusse P "The CRONE control of resonant plants: application to a flexible transmission", European Journal of Control 1(2), 1995.

[3] Podlubny I, Dorcak L, Kostial I "On fractional derivatives, fractional-order dynamic systems and $PI^{\lambda}D^{\mu}$-controllers", In Proc. of the 36the Conference on Decision and Control, San Diego, CA 4985–4990, 1997.

[4] Hyo-Sung Ahn, Varsha Bhambhani and Yangquan Chen," Fractional-order integral and derivative controller for temperature profile tracking",*Sadhana* Vol. 34, Part 5, October 2009, pp. 833–850, *Indian Academy of Sciences.*

[5] Duarte Valerio Jose Sa da Costa," Tuning Rules for fractional PID controllers", Technical report submitted to the Department of Mechanical Engineering Technical Univ. of Lisbon, Portogal, 2010.

[6] Chunna Zhao , Dingyu Xue, and YangQuan Chen , " A Fractional Order PID Tuning Algorithm for A Class of Fractional Order Plants", Proceedings of the IEEE International Conference on Mechatronics & Automation Niagara Falls, Canada , July, 2005.

[7] Joaquin Cervera, Alfonso Banios, Concha A. Monje, Blas M. Vinagre," Tuning of Fractional PID Controllers by Using QFT", IEEE industrial electronics, IECON (2006),the 32nd Annual Conference in digital object identifier,2006.

[8] Barbosa R. , Tenreiro J.A., Galhano A.M.," Performance of fractional PID algorithms controlling non-linear systems with saturation and backlash phenomena", Journal of Vibration and Control,13(9-10),1407-1418,2007.

[9] Ruikun G., Youliang Y., Yansong T., Guangxiang Z., Jie,s. ,and Lei,C.," Design and Implement of Neural Network Based Fractiona1 order $PI^{\alpha}$ Controller", Sixth International Conference on Natural Computation (ICNC 2010),2010.

[10] Xiaomin Tian, Yourui Huang, and Canming Zhang," The tuning principle of adaptive fuzzy fractional-order PID controller parameters", Symposium on Security Detection and Information, Procedia Engineering 7, 251–255, 2010.

[11] Al-Alaoui M. A.," Discretization Methods of Fractional Parallel PID Controllers", IEEE international conference on digital object identifier, ICECS (2009).

[12] Mitchell M."An introduction to Genetic Algorithms", MIT press,1996.

[13] Goldberg D.E."Genetic Algorithms in Search, Optimization, and Machine learning:" Adison Wesley, New York.1989.

[14] Thomas N., Poongodi P." Position Control of DC Motor Using Genetic Algorithm Based PID Controller", Proceedings of the World Congress on Engineering 2009 Vol. II, WCE 2009, July 1 - 3, 2009, London, U.K.

**The work was carried out at the college of Electronics Engineering. University of Mosul**