Fractional Order PID Controller Design for DC Motor Speed Control System via Flower Pollination Algorithm

Deacha Puangdownreong[1], Non-member

ABSTRACT

Over two decades, the fractional (non-integer) order PID (FOPID or $PI^{\lambda}D^{\mu}$) controller was introduced and demonstrated to perform the better responses in comparison with the conventional integer order PID (IOPID). In this paper, the design of an optimal FOPID controller for a DC motor speed control system by the flower pollination algorithm (FPA), one of the most efficient population-based metaheuristic optimization searching techniques, is proposed. Based on the modern optimization framework, five parameters of the FOPID controller are optimized by the FPA to meet the response specifications of the DC motor speed control system and defined as constraint functions. Results obtained by the FOPID controller are compared with those obtained by the IOPID designed by the FPA. As the simulation results show, the FOPID can provide significantly superior speed responses to the IOPID.

Keywords: Fractional Order PID Controller, Integer Order PID Controller, Flower Pollination Algorithm, Modern Optimization

1. INTRODUCTION

Following the literature, fractional calculus is a more than 300 years old topic [1,2]. It has multifarious applications in science and engineering especially in control systems. The fractional order PID (FOPID) controller (or $PI^{\lambda}D^{\mu}$) was first proposed by Podlubny in 1999 [3], as an extended version of the conventional integer order PID (IOPID). Based on fractional calculus, the FOPID controller is characterized by five parameters: proportional gain (K_p), integration gain (K_i), derivative gain (K_d), integral order (λ) and derivative order (μ). Once the FOPID is compared with the IOPID, there are two extra parameters λ and μ making the FOPID controller more efficient, but more complicated than the IOPID in design and implementation procedures. FOPID has been successfully conducted in many applications, for instance, process control [4], automatic voltage regulator (AVR) [5], DC motor control [6], power electronic control [7], inverted pendulum control [8] and gun control system [9]. Several design and tuning methods for FOPID have been consecutively launched, for example, rule-based methods [10,11] and analytical methods [12,13]. Review and tutorial articles of the FOPID controller providing the state-of-the-art and its backgrounds have been completely reported [1,2].

Recently, control synthesis has been changed from the conventional paradigm to the new framework based on modern optimization using metaheuristics as an optimizer [14,15]. The flower pollination algorithm (FPA), first proposed by Yang in 2012 [16], is one of the most powerful population-based metaheuristic optimization searching techniques. The FPA algorithm mimics the behaviour of pollination of flowering plants in nature. In the FPA algorithm, the Lévy flight distribution is used for efficient movement by pollinators in order to generate the feasible solution within the particular search space. By literature surveys, the performance evaluation of the FPA against many standard test functions was proposed [16,17]. Results obtained by the FPA outperformed those by well-known algorithms including genetic algorithms (GA) and particle swarm optimization (PSO). Moreover, the FPA algorithms were proved for the global convergent property [18] and successfully applied to many real-world engineering problems, for example, pressure vessels design [16], disc break design [17], electrical power system [19], image processing [20], wireless sensor networking [21], clustering [22], global function optimization [23], computer gaming [24], structural engineering [25], control system design [26-29] and model identification [30]. The state-of-the-art and applications of the FPA have been reviewed and reported [31].

In this paper, the FPA is applied to optimally design the FOPID controller for the DC motor speed control system. The rest of the paper is outlined as follows: Fractional calculus, fractional order PID (FOPID) controller and stability analysis of linear, time-invariant (LTI) fractional order system are briefly described in Section 2. Problem formulation of the FPA-based FOPID controller design for the DC motor speed control system is performed in Section 3. Results and discussions are illustrated in Section 4, while conclusions are summarized in Section 5.

Manuscript received on February 18, 2018 ; revised on January 11, 2019.

[1] The author is with Graduate School, Southeast Asia University, 19/1 Petchakasem Rd., Nongkhaem, Bangkok, Thailand, 10160., E-mail: dpdeachap@gmail.com

2. FRACTIONAL CALCULUS

2.1 Fractional Calculus

In fractional calculus, a generalization of integration and differentiation can be represented by the non-integer order fundamental operator ${}_aD_t^\alpha$, where a and t are the limits of the operator. The continuous integro-differential operator is defined as expressed in (1), where $\alpha \in \Re$ stands for the order of operation.

There are three definitions used for the generally fractional differintegral. The first definition is Grunwald-Letnikov (GL) as stated in (2), where $[\cdot]$ is integer part and n is an integer satisfying the condition $n-1 < \alpha < n$. The binomial coefficient is stated in (3), while the Euler's gamma function $\Gamma(\cdot)$ is defined by (4).

$$a^{D_t^\alpha} = \begin{cases} \frac{d^\alpha}{dt^\alpha} & \Re(\alpha) > 0 \\ 1 & \Re(\alpha) = 0 \\ \int_a^t (d\tau)^{-\alpha} & \Re(\alpha) < 0 \end{cases} \quad (1)$$

$$a^{D_t^\alpha} f(t) = \lim_{h\to 0} h^{-\alpha} \sum_{r=0}^{\left[\frac{t-a}{h}\right]} (-1)^r \binom{n}{r} f(t-rh) \quad (2)$$

$$\binom{n}{r} = \frac{\Gamma(n+1)}{\Gamma(r+1)\Gamma(n-r+1)} \quad (3)$$

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt \quad (4)$$

The second definition is Riemann-Liouville (RL) as expressed in (5), for $n-1 < \alpha < n$. The third definition is Caputo definition as shown in (6), where n is an integer and $n-1 < \alpha < n$. Among those, the Caputo definition is most popular in engineering applications [1].

$$a^{D_t^\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt}\right)^n \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (5)$$

$$a^{D_t^\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} d\tau \quad (6)$$

$$\begin{aligned} \mathcal{L}\{a^{D_t^\alpha} f(t)\} &= \int_0^\infty e^{-st} {}_0D_t^\alpha f(t) dt \\ &= s^\alpha F(s) - \sum_{k=0}^{n-1} s^k {}_0D_t^{\alpha-k-1} f(t)\Big|_{t=0} \end{aligned} \quad (7)$$

For solving engineering problems, the Laplace transform is routinely conducted. The formula of the Laplace transform of the RL fractional derivative in (5) is stated in (7), for $n-1 < \alpha \le n$, where $s \equiv j\omega$ denotes the Laplace transform (complex) variable. Under zero initial conditions for order $\alpha(0 < \alpha < 1)$, the Laplace transform of the RL fractional derivative defined in (5) can be expressed in (8).

$$\mathcal{L}\{a^{D_t^{\pm\alpha}}\} = s^{\pm\alpha} F(s) \quad (8)$$

2.2 FOPID Controller

The fractional order PID controller (FOPID or PI D) is an extended version of the conventional integer order PID (IOPID). The generalized transfer function of the FOPID is given by the differential equation as stated in (9), where $u(t)$ is the control signal, $e(t)$ is the error signal and λ and $\mu \ge 0$. By taking the Laplace transform, the generalized transfer function of the FOPID is then expressed in (10).

$$u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^\mu e(t) \quad (9)$$

$$G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu \quad (10)$$

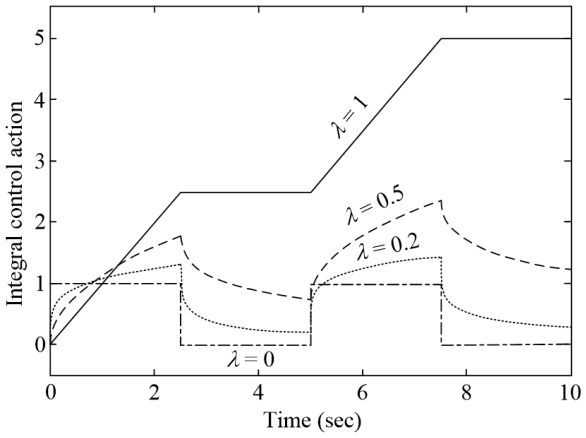


Fig.1: *Integral control action for a square error signal and λ = 0, 0.2, 0.5, 1.*

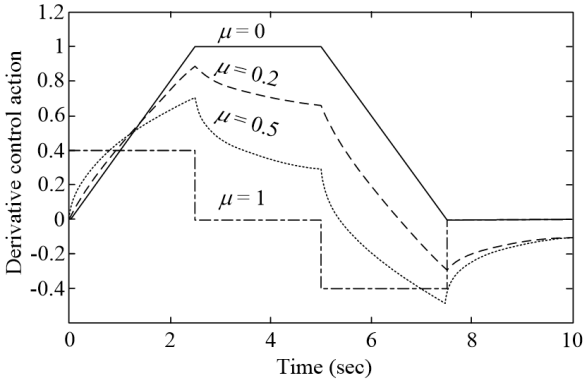


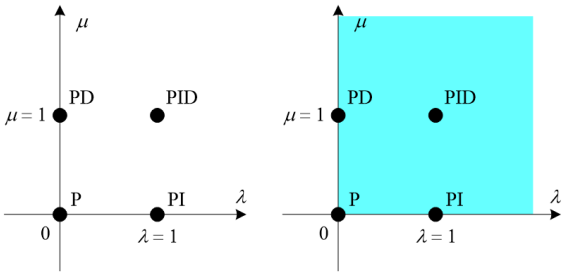
Fig.2: *Derivative control action for a trapezoidal error signal and μ = 0, 0.2, 0.5, 1.*

Referring to (10), λ is the fractional order of the integral element, while μ is the fractional order of the

derivative element. Fig. 1 shows the effects of λ = 0, 0.2, 0.5 and 1 on the control signals [32]. As can be observed, the effects of the control action over the error signal vary between the effects of a proportional action ($\lambda = 0$, square signal) and an integral action ($\lambda = 1$, straight line curve). For intermediate values of , the control action increases for a constant error, which results in the elimination of the steady-state error, and decreases when the error is zero, resulting in a more stable system.

For the derivative control action, the effects of the control action over the error signal are shown in Fig. 2, where μ = 0, 0.2, 0.5 and 1, respectively [32]. As can be observed, they vary between the effects of a proportional action ($\mu = 0$, trapezoidal signal) and a derivative action ($\mu = 1$, square signal). For intermediate values of μ, the control action corresponds to intermediate curves. It must be noted that the derivative action is not zero for a constant error and the growth of the control signal is more damped when a variation in the error signal occurs, which implies a better attenuation of high-frequency noise signals.

Relationship between the IOPID and the FOPID can be represented by a graphical way as visualized in Fig. 3 [2,32]. In general, the range of fractional orders (λ and μ) is varied from 0 to 2. However, in most research works, the range of λ and μ is varied from 0 to 1. Referring to Fig. 3, if $\lambda = \mu = 1$, it is the IOPID controller.



(a) IOPID (b) FOPID

Fig.3: *FOPID and IOPID from points to plane: (a) integer-order and (b) fractional-order.*

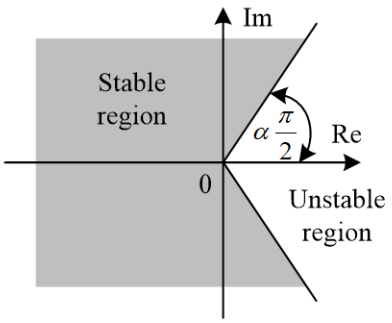
2.3 Stability of LTI Fractional Order Systems

In classical control theory, the LTI system is stable if the roots of the characteristic polynomial (or poles) are negative or have negative real parts if they are complex conjugate. It means that they are located on the left half of the complex plane. For the fractional order LTI case, the stability is different from the integer one. However, Matignon's stability theorem based on the Riemann sheet can be utilized for stability analysis of both integer order LTI and fractional order LTI systems [1-2, 32-33].

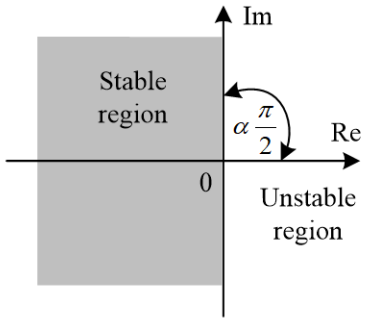
For the case of commensurate-order systems, whose characteristic equation is a polynomial of the complex variable $\lambda = s^{\alpha}$, the stability condition is expressed in (11), where λ_i are the roots of the characteristic polynomial as shown by the Riemann sheet on the complex plane in Fig. 4(a). For the integer order LTI systems, $\alpha = 1$, the stability condition is stated in (12) and confirmed by the Riemann sheet on the complex plane in Fig. 4(b).

$$|\arg(\lambda_i)| > \alpha\frac{\pi}{2}, \quad i = 1, 2, \ldots, n \tag{11}$$

$$|\arg(\lambda_i)| > \frac{\pi}{2}, \quad i = 1, 2, \ldots, n \tag{12}$$



(a) Stability region with fractional order $0 < \alpha \leq 1$



(b) Stability region with integer order $\alpha = 1$

Fig.4: *Stability region of LTI systems: (a) fractional-order LTI systems and (b) integer-order LTI systems.*

3. PROBLEM FORMULATION

The problem formulation consisting of the FOPID control loop, the DC motor model, FPA algorithms and the FPA-based FOPID controller design are consecutively provided in this section.

3.1 FOPID Control Loop

The FOPID control loop is represented by the block diagram as shown in Fig. 5, where $G_p(s)$ and $G_c(s)$ are the plant and the FOPID controller models, respectively. The FOPID receives the error signal $E(s)$ and produces the control signal $U(s)$ to control

the output signal $C(s)$ and regulate the disturbance signal $D(s)$, referring to the reference input $R(s)$.

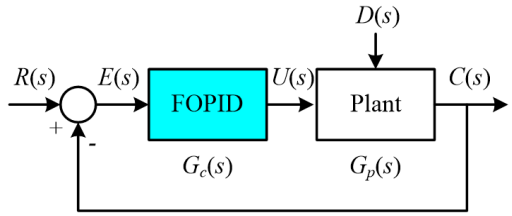


Fig.5: *FOPID control loop.*

3.2 DC Motor Model

The schematic diagram of an armature-controlled DC motor can be represented in Fig. 6 [34], where R_a is an armature-winding resistance, L_a is an armature-winding inductance, R_f is a field-winding resistance, L_f is a field-winding inductance, J is a moment of inertia, B is a viscous-friction coefficient, $e_a(t)$ is an applied armature voltage, $i_a(t)$ is an armature current, $e_f(t)$ is a field voltage, $i_f(t)$ is a field current, $e_b(t)$ is a back emf (electromotive force) voltage, $T(t)$ is a motor torque, $\theta(t)$ is an angular displacement, and $\omega(t)$ is an angular velocity (speed). A mathematical model of an armature-controlled DC motor in terms of the transfer function can be formulated by the differential equations and Laplace transform. The transfer function of the armature-controlled DC motor speed control is stated in (13), where K_t is a torque constant and K_b is a back emf constant.

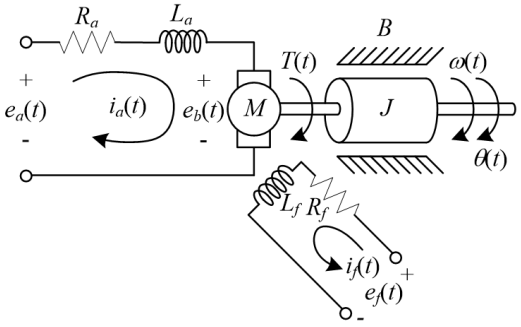


Fig.6: *Schematic diagram of DC motor.*

$$\frac{\Omega(s)}{E_a(s)} = \frac{K_t}{JL_a s^2 + (BL_a + JR_a)s + (BR_a + K_1 K_b)} \quad (13)$$

$$\frac{E_a(s)}{V_{in}(s)} = \frac{K_A}{(\tau_A s + 1)} \quad (14)$$

$$\frac{\Omega(s)}{V_{in}(s)} = \frac{K_A K_t}{(\tau_A s + 1)[JL_a s^2 + (BL_a + JR_a)s + (BR_a + K_1 K_b)]} \quad (15)$$

Commonly, using a DC motor needs a power amplifier as a driver. Due to this, the DC motor plant consists of a driver and a DC motor. The driver model is approximated by the first-order transfer function as stated in (14), where vin(t) is an input voltage. Therefore, the DC motor plant model can be rewritten as expressed in (15). From previous work [35], the DC motor plant consists of a DC motor (LEYBOLD-DIDACTIC GMBH, Type 731-91, 0.3 kW, 220 V, 2.2 A, 2000 rpm) and a driver (SCR full-wave controlled rectifier) as shown in Fig. 7. A speed sensor (tacho-generator LEYBOLD, Type 731-09) and a low-pass filter were also implemented. The parameters of this motor plant have been identified by using MATLAB and System Identification Toolbox [36] and obtained at 1,000 rpm as follows: $R_a = 54.7280\ \Omega$, $L_a = 1.5104$ H, $J = 36.4277$ kg-m^2, $B = 0.0988$ N-m-sec/rad, $K_t =$ N-m/A, $K_b = 1.6046$ V/rpm, $K_A = 3.4449$ and $\tau_A = 0.3350$ sec. Very good agreement between actual speed response and the plant model can be observed in Fig. 8. The DC motor plant in (15) can be written in (16). The model in (16) will be used as the plant $G_p(s)$ in Fig. 5. 2.7761

$$G_p(s) = \frac{\Omega(s)}{V_{in}(s)} = \frac{9.563}{18.43s^3 + 722.9s^2 + 1997s + 9.862} \quad (16)$$

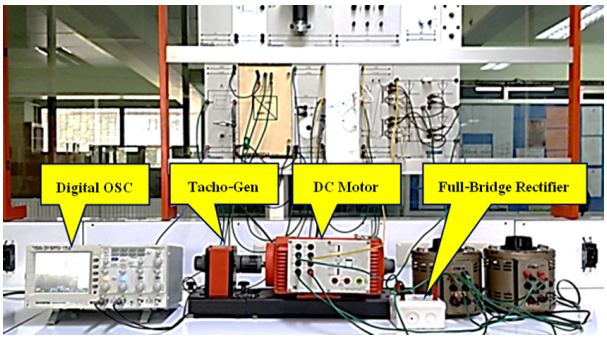


Fig.7: *DC motor plant.*

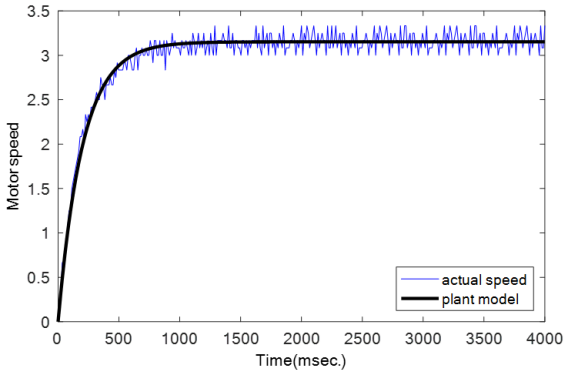


Fig.8: *Plots of actual speed response and plant model.*

3.3 FPA Algorithms

In nature, the objective of flower pollination is the survival of the fittest and optimal reproduction of flowering plants. Pollination in flowering plants can take two major forms, i.e. biotic and abiotic [37]. About 90% of flowering plants belong to biotic pollination. Pollen is transferred by a pollinator such as bees, birds, insects and animals. About 10% of remaining pollination is abiotic such as by wind and diffusion in water. Pollination can be achieved by self-pollination or cross-pollination as visualized in Fig. 9 [38,39]. Self-pollination is the fertilization of one flower from pollen of the same flower (Autogamy) or different flowers of the same plant (Geitonogamy). They occur when a flower contains both male and female gametes. Self-pollination usually occurs at short distance without pollinators. It is regarded as local pollination. Cross-pollination, Allogamy, occurs when pollen grains are moved to a flower from another plant. The process happens with the help of biotic or abiotic agents as pollinators. Biotic, cross-pollination may occur at long distance with biotic pollinators. It is regarded as global pollination. Bees and birds as biotic pollinators behave Lévy flight behaviour [40] with jump or fly distance steps obeying a Lévy distribution.

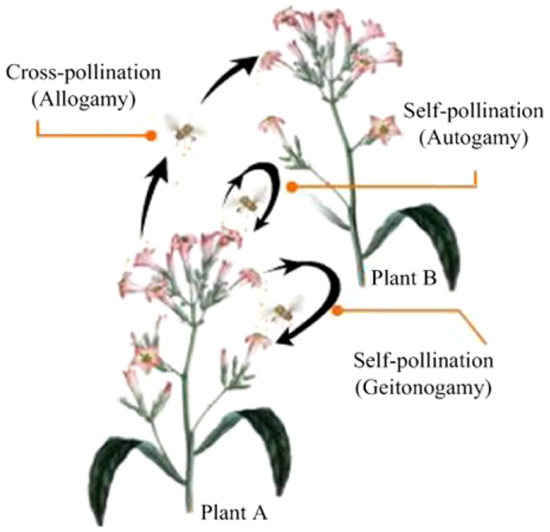


Fig.9: *Flower pollination in nature.*

From the above characteristics of flower pollination, the FPA algorithm proposed by Yang [16] is based on four particular rules as follows:

• Biotic and cross-pollination are global pollination processes via Lévy flight (Rule-1).

• Abiotic and self-pollination are local pollination processes with random walk (Rule-2).

• Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of the two flowers involved (Rule-3).

• Local pollination and global pollination can be controlled by a switch probability $p \in [0,1]$ (Rule-4).

In FPA algorithm, a solution $\boldsymbol{x}_i$ is equivalent to a flower and/or a pollen gamete. For global pollination, flower pollens are carried by pollinators. With Lévy flight, pollens can travel over a long distance. Therefore, Rule-1 and flower constancy in Rule-3 can be expressed in (17), where $\boldsymbol{g}^*$ is the current best solution found among all solutions at the current generation/iteration t, and L stands for the Lévy flight that can be approximated by (18), while $\Gamma(\lambda)$ is the standard gamma function.

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + L(\boldsymbol{x}_i^t - \boldsymbol{g}^*) \tag{17}$$

$$L \approx \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s >> s_0 > 0) \tag{18}$$

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \varepsilon(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t) \tag{19}$$

$$\varepsilon(\rho) = \begin{cases} 1/(b-a), & a \le \rho \le b \\ 0, & \rho < a \text{ or } \rho > b \end{cases} \tag{20}$$

- Objective $f(\boldsymbol{x})$, $\boldsymbol{x} = (x_1, x_2, ..., x_d)$
- Initialize a population of n flowers/pollen gametes with random solutions
- Find the best solution $\boldsymbol{g}^*$ in the initial population
- Define a switch probability $p \in [0, 1]$

while ($t <$ *MaxGeneration*)
  for $i = 1 : n$ (all n flowers in the population)
    if *rand* $< p$,
      - Draw vector L via Lévy flight
      - Activate global pollination
    else
      - Draw ε from uniform distribution in [0,1]
      - Randomly choose j and k solutions
      - Invoke local pollination
    end if
    - Evaluate new solutions
    - If new solutions are better, update solutions
  end for
  - Find the current best solution $\boldsymbol{g}^*$
end while

Fig.10: *Pseudo code of FPA algorithm.*

For local pollination, Rule-2 and Rule-3 can be represented by (19), where $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are pollens from the different flowers of the same plant species, while stands for random walk by using uniform distribution in [0,1]. Once setting $a = 0$ and $b = 1$ in (20), it is called a standard uniform distribution. Flower pollination activities can occur at all scales, both local and global pollination. In this case, a switch probability or proximity probability p in Rule-4 is used to switch between common global pollination to intensive local

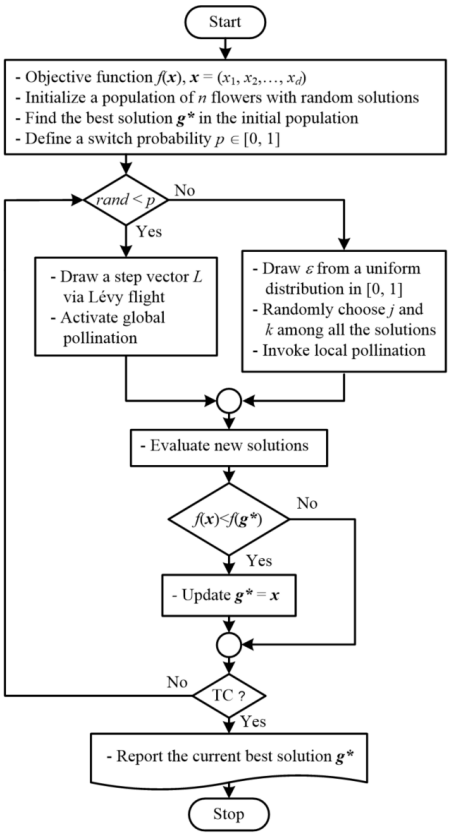


Fig.11: *Flow diagram of FPA algorithm.*

pollination. The FPA algorithm can be summarized by the pseudo code as shown in Fig. 10 and the flow diagram as shown in Fig. 11. From Yang's research reports [16,17], the number of pollens $n = 25$, switch probability $p = 0.8$ and $\lambda = 1.5$ works better for most applications and are the recommended parameter set based on preliminary parametric studies.

3.4 FPA-Based FOPID Design

Based on modern optimization framework, the application of the FPA to design optimal FOPID controller for the DC motor speed control system can be represented by the block diagram in Fig. 12. The objective function, J, sum-squared error between the reference speed $R(s)$ and the actual speed $C(s)$ as stated in (21), will be fed to the FPA to be minimized by searching for the appropriate values of the FOPID parameters, i.e. K_p, K_i, K_d, λ and μ subject to inequality constraint functions satisfying the predefined response specifications as stated in (22), where t_r and $t_{r\_\max}$ are rise time and maximum rise time, M_p and $M_{p\_\max}$ are percent overshoot and maximum percent overshoot, t_s and $t_{s\_\max}$ are settling time and maximum settling time, e_{ss} and $e_{ss\_\max}$ are steady-state error and maximum steady-state error, within their particular boundaries or search spaces $K_p \in [K_{p\_\min}, K_{p\_\max}]$, $K_i \in [K_{i\_\min}, K_{i\_\max}]$, $K_d \in [K_{d\_\min}, K_{d\_\max}]$, $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ and $\mu \in [\mu_{\min}, \mu_{\max}]$, respectively.

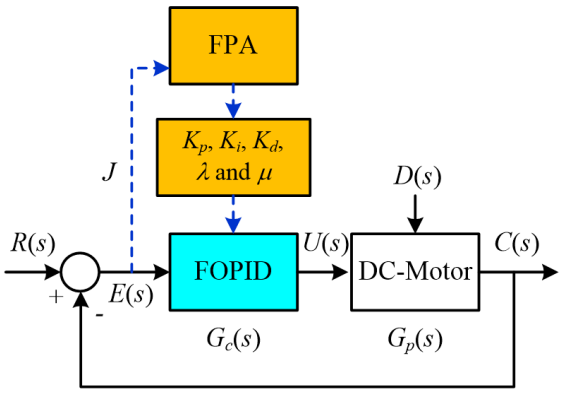


Fig.12: *FPA-based FOPID design framework.*

$$\textbf{Min} \quad J(K_p, K_i, K_d, \lambda, \mu) = \sum_{i=1}^{N} [r_i - c_i]^2 \qquad (21)$$

The FPA algorithm shown in Fig. 10 and Fig 11 will be adapted for the FOPID controller design application as follows:

$$\textbf{Subject to} \quad \left.\begin{array}{l} t_r \le t_{r\_\max}, \\ M_p \le M_{p\_\max}, \\ t_s \le t_{s\_\max}, \\ e_{ss} \le e_{ss\_\max}, \\ K_{p\_\min} \le K_p \le K_{p\_\max}, \\ K_{i\_\min} \le K_i \le K_{i\_\max}, \\ K_{d\_\min} \le K_d \le K_{d\_\max}, \\ \lambda_{\min} \le \lambda \le \lambda_{\max} \text{ and } \mu_{\min} \le \mu \le \mu_{\max} \end{array}\right\} \qquad (22)$$

Step-0 Initialize the objective function J in (21) and constraint functions associated with their search spaces in (22). Randomly generate a population of n flowers. Fine the best solution $\boldsymbol{g}^*$ among initial population. Define a switch probability $p = 0.8$ (or 80%). Set MaxGen as the termination criteria (TC) and Gen = 1 as a generation counter.

Step-1 If Gen MaxGen, go to Step-2. Otherwise go to Step-4.

Step-2 If rand $< p$, draw a step vector L via Lévy flight in (18) and activate global pollination in (17) to generate a new solution x. Otherwise draw a uniform distribution $\varepsilon \in$ [0, 1] in (20). Randomly select j and k among all solutions. Invoke local pollination in (19) to generate a new solution $\boldsymbol{x}$.

Step-3 If $f(\boldsymbol{x}) < f(\boldsymbol{g}^*)$, update solution $\boldsymbol{g}^* = \boldsymbol{x}$ and update Gen = Gen+1. Otherwise update Gen = Gen+1. Go to Step-1 to proceed next generation.

Step-4 Report the best solution found and stop the search process.

4. RESULTS AND DISCUSSIONS

In order to design the optimal FOPID controller for the DC motor speed control system, the FPA algorithms were coded by MATLAB version 2017b (License No.#40637337) run on Intel(R) Core(TM) i5-6500 CPU@3.20GHz, 4.0GB-RAM. The FOPID is implemented by MATLAB with FOMCON toolbox [41-43] where Oustaloup's approximation is realized for fractional order numerical simulation. Number of pollens $n = 25$, switch probability $p = 0.8$ and $\lambda = 1.5$ are set according to the recommendations of Yang [16,17]. Search spaces and constraint functions in (22) are then performed as stated in (23). The maximum generation MaxGen = 200 is then set as the TC. 50 trials are conducted to find the best solution (optimal FOPID controller for the DC motor speed control system). For comparison with the IOPID controller, λ and μ in (23) will be set as one (λ=μ= 1).

For the first trial, 200 step responses (according to MaxGen) of the DC motor speed controlled system corresponding to 200 FOPID controllers obtained by the FPA are depicted in Fig. 13 (a black-thick line is the best solution of this trial). The convergent rate of the objective functions in (21) associated with inequality constraint functions in (23) utilized by the FPA for the first trial is depicted in Fig. 14. Once the 50-trials search process stopped, the FPA can successfully provide the optimal parameters of the IOPID and FOPID controllers for the DC motor speed control system as expressed in (24) and (25), respectively. The convergent rates of the objective functions associated with inequality constraint functions utilized by the FPA over 50 trials are plotted in Fig. 15. The step responses of the DC motor speed control system without controller, with IOPID and FOPID controllers designed by the FPA are depicted in Fig. 16.

$$\begin{aligned} \textbf{Subject to } & t_r \leq 150 \text{ msec}, \quad M_p \leq 10.0\% \\ & t_s \leq 350.0 \text{ msec}, \quad e_{ss} \leq 0.01\% \\ & 0 < K_p \leq 5, \quad 0 < K_i \leq 2 \\ & 0 < K_d \leq 5, \quad 0 < \mu, \lambda < 1.0 \end{aligned} \tag{23}$$

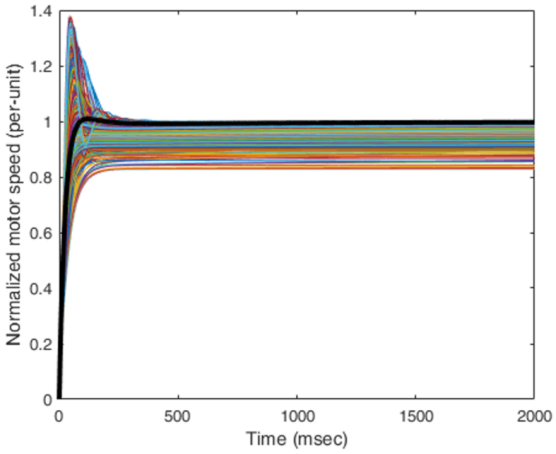


Fig.13: *Step responses with FOPID of the first trial.*

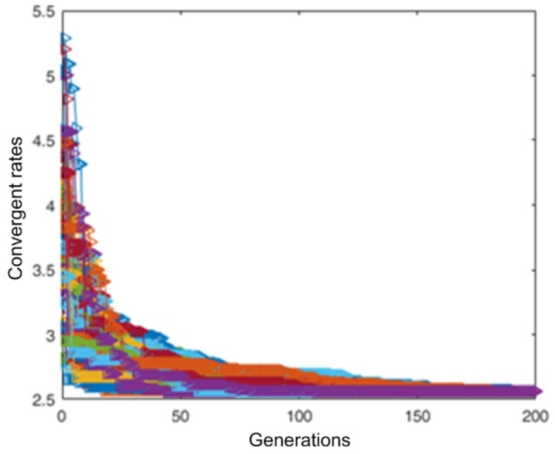


Fig.15: *Convergent rates of FOPID design utilized by FPA over 50 trials.*

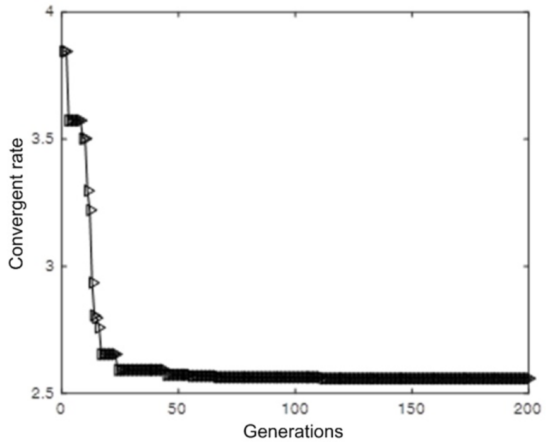


Fig.14: *Convergent rate of the first trial of FOPID design utilized by FPA.*

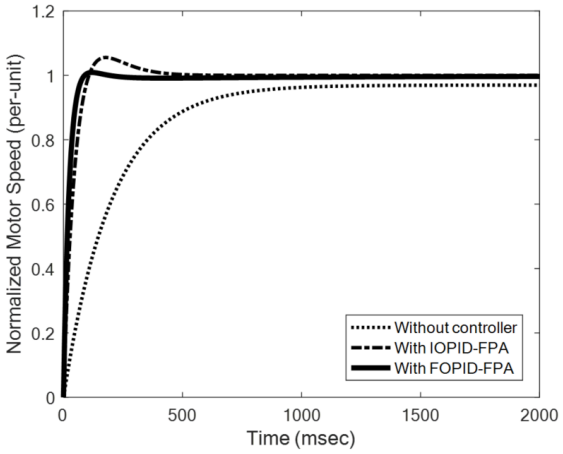


Fig.16: *Step responses of DC motor speed control system without and with IOPID and FOPID.*

$$G_c(s)|_{IOPID} = 4.9124 + \frac{0.0440}{s} + 4.9968s \quad (24)$$

$$G_c(s)|_{FOPID} = 4.9866 + \frac{0.1713}{s^{0.7962}} + 4.9929s^{0.1102} \quad (25)$$

Referring to Fig. 15, the FPA can successfully provide the optimal FOPID controller for the DC motor speed control system within the 190th iteration and with an average search time of 46.8095 sec. Both the IOPID and FOPID optimized by the FPA as stated in (24) and (25) agree with search spaces given in (23). From the step responses in Fig. 16, they can be summarized in Table 1. Results obtained also agree with inequality constraints defined in (23). For comparison, the FOPID controller can provide significantly faster and smoother speed response of the DC motor controlled system than the IOPID controller.

Table 1: *Obtained step responses of the DC motor speed controlled system designed by FPA.*

Controllers	Step responses			
	t_r (ms)	M_p (%)	t_s (ms)	e_{ss} (%)
without	654.24	0.00	825.56	3.284
IOPID	107.09	5.56	326.63	0.002
FOPID	88.93	0.91	145.52	0.001

For stability analysis, it was found that both IOPID and FOPID controllers designed by the FPA can make the DC motor speed controlled system stable according to the stability condition expressed in (11) and (12) as can be observed in Fig. 17 and Fig. 18, respectively.

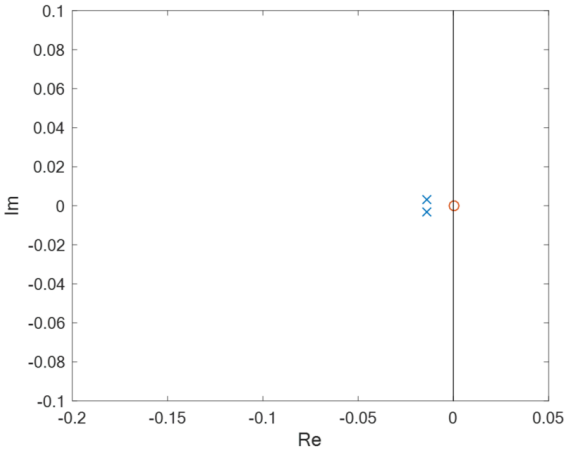


Fig.17: *Stability of DC motor system with IOPID.*

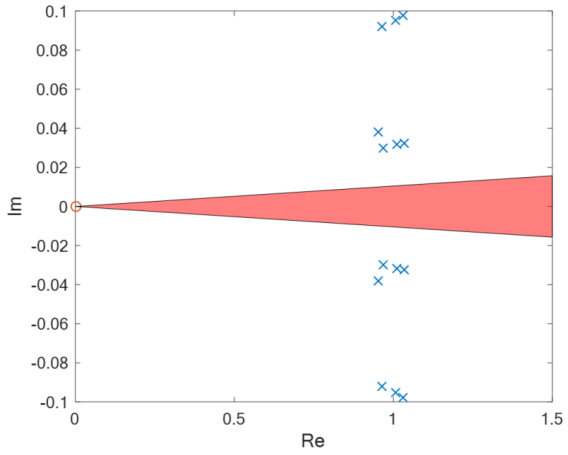


Fig.18: *Stability of DC motor system with FOPID.*

5. CONCLUSION

Design of an optimal FOPID controller for the DC motor speed control system by the FPA has been proposed in this paper. Fractional calculus, FOPID controller and stability analysis of the LTI fractional order system have been briefly illustrated. As one of the most powerful metaheuristic optimization searching techniques, the FPA has been applied for FOPID controller design by searching for its five parameters to meet the response specifications of the DC motor speed control system performed as constraint functions. As results of comparison with the IOPID controller show, it was found that the FPA has provided optimal IOPID and FOPID controllers for the DC motor speed controlled system according to the given constraint functions and search spaces. Both IOPID and FOPID controllers designed by the FPA could make the DC motor speed controlled system stable based on Matignon's stability theorem. The FOPID controller could provide faster and smoother speed response than the IOPID one. For future research, implementation of the FOPID will be conducted to obtain experimental results. In addition, the study of FOPIDA (fractional order of proportional-integral-derivative-accelerated) controller will be extended.

References

[1] Y. Q. Chen, I. Petráš and X. Dingyü, "Fractional Order Control - a Tutorial," *Proceeding of the American Control Conference*, pp.1397-1411, 2009.

[2] P. Shah and A. Agashe, "Review of Fractional PID Controller," *Mechatronics*, Vol. 38, pp.29-41, 2016.

[3] I. Podlubny, "Fractional-Order Systems and $PI^{\lambda}D^{\mu}$ Controllers," *IEEE Transactions on Automatic Control*, Vol. 44, No. 1, pp.208-214, 1999.

[4] C. A. Monje, B. M. Vinagre, V. Feliu and Y. Chen, "Tuning and Auto-Tuning of Fractional Order Controllers for Industry Applications," *Con-*

trol Engineering Practice, Vol. 16, pp.798-812, 2008.

[5] M. Zamani, M. Karimi-Ghartemani, N. Sadati and M. Parniani, "Design of a Fractional Order PID Controller for an AVR using Particle Swarm Optimization," *Control Engineering Practice*, Vol. 17, pp.1380-1387, 2009.

[6] D. Xue, C. Zhao and Y. Q. Chen, "Fractional Order PID Control of A DC-Motor with Elastic Shaft: A Case Study," *Proceedings of the 2006 American Control Conference*, pp.3182-3187, 2006.

[7] A. J. Calderón, B. M. Vinagre and V. Feliu, "Fractional Order Control Strategies for Power Electronic Buck Converters," *Signal Processing*, Vol. 86, No. 10, pp.2803-2819, 2006.

[8] S. K. Mishra and D. Chandra, "Stabilization and Tracking Control of Inverted Pendulum using Fractional Order PID Controllers," *Journal of Engineering*, Vol. 2014, pp.1-9, 2014.

[9] Q. Gao, J. Chen, L. Wang, S. Xu and Y. Hou, "Multiobjective Optimization Design of a Fractional Order PID Controller for a Gun Control System," *The Scientific World Journal*, Vol. 2013, pp.1-8, 2013.

[10] D. Valrio and J. C. Sda, "Tuning of Fractional PID Controllers with Ziegler-Nichols Type Rules," *Signal Processing*, Vol. 86, No. 10, pp.2771-2784, 2006.

[11] Y. Chen, T. Bhaskaran and D. Xue, "Practical Tuning Rule Development for Fractional Order Proportional and Integral Controllers," *Journal of Computational and Nonlinear Dynamics*, Vol. 3, 2008.

[12] R. Caponetto, L. Fortuna and D. Porto, "A New Tuning Strategy for a Non Integer Order PID Controller," *Fractional Differentiation and its Applications*, Bordeaux, 2004.

[13] C. Zhao, D. Xue and Y. Chen, "A Fractional Order PID Tuning Algorithm for a Class of Fractional Order Plants," *Proceedings of the International Conference on Mechatronics & Automation*, Niagara Falls, 2005.

[14] V. Zakian, *Control Systems Design: A New Framework*, Springer-Verlag, 2005.

[15] V. Zakian and U. Al-Naib, "Design of Dynamical and Control Systems by the Method of Inequalities," *Proceedings of the IEE International Conference*, Vol. 120, pp.1421-1427, 1973.

[16] X. S. Yang, "Flower Pollination Algorithm for Global Optimization," *Unconventional Computation and Natural Computation, Lecture Notes in Computer Science*, Vol. 7445, pp.240-249, 2012.

[17] X. S. Yang, M. Karamanoglu and X. He, "Multi-Objective Flower Algorithm for Optimization," *Procedia Computer Science*, Vol. 18, pp.861-868, 2013.

[18] X. He, X. S. Yang, M. Karamanoglu and Y. Zhao, "Global Convergence Analysis of the Flower Pollination Algorithm: A Discrete-Time Markov Chain Approach," *Procedia Computer Science*, Vol. 108C, pp.1354-1363, 2017.

[19] A. Abdelaziz, E. Ali and S. A. Elazim, "Combined Economic and Emission Dispatch Solution using Flower Pollination Algorithm," *International Journal Electrical Power Energy Systems*, Vol. 80, pp. 264-274, 2016.

[20] S. Ouadfel and A. Taleb-Ahmed, "Social Spiders Optimization and Flower Pollination Algorithm for Multilevel Image Thresholding: A Performance Study," *Expert Systems*, Vol. 55, pp. 566-584, 2016.

[21] M. Sharawi, E. Emary, I. A. Saroit and H. El-Mahdy, "Flower Pollination Optimization Algorithm for Wireless Sensor Network Lifetime Global Optimization," *International Journal of Soft Computing Engineering*, Vol. 4, No. 3, pp. 54-59, 2014.

[22] P. Agarwal and S. Mehta, "Enhanced Flower Pollination Algorithm on Data Clustering," *International Journal of Computer Applications*, Vol. 38, No. 2-3, pp. 144-155, 2016.

[23] E. Nabil, "A Modified Flower Pollination Algorithm for Global Optimization," *Expert System Application*, Vol. 57, pp. 192-203, 2016.

[24] O. Abdel-Raouf, I. El-Henawy and M. Abdel-Baset, "A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles," *International Journal of Modern Education and Computer Science*, Vol. 6, No. 3, pp.38, 2014.

[25] S. M. Nigdeli, G. Bekdaş and X. S. Yang, "Application of the Flower Pollination Algorithm in Structural Engineering," *Metaheuristics and Optimization in Civil Engineering*, Springer, pp.25-42, 2016.

[26] D. Puangdownreong, "Optimal State-Feedback Design for Inverted Pendulum System by Flower Pollination Algorithm," *International Review of Automatic Control (IREACO)*, Vol. 9, No. 5, pp.289-297, 2016.

[27] S. Hlungnamtip, C. Thammarat and D. Puangdownreong, "Obtaining Optimal PID Controller for DC Motor Speed Control System via Flower Pollination Algorithm," *Proceedings of the 9th International Conference on Sciences, Technology and Innovation for Sustainable Well-Being (STIWB)*, pp.52-56, 2017.

[28] C. Thammarat, A. Nawikavatan and D. Puangdownreong, "Application of Flower Pollination Algorithm to PID Controller Design for Three-Tank Liquid-Level Control System," *Proceedings of the 9th International Conference on Sciences, Technology and Innovation for Sustainable Well-Being (STIWB)*, pp.42-46, 2017.

[29] S. Hlungnamthip, N. Pringsakul, A. Nawika-

vatan and D. Puangdownreong, "FPA-Based PID Controller Design for Temperature Control of Electric Furnace System," *Proceedings of the 2018 International Conference on Engineering and Natural Science (ICENS 2018)*, pp.60-68, 2018.

[30] D. Puangdownreong, C. Thammarat, S. Hlungnamtip and A. Nawikavatan, "Application of Flower Pollination Algorithm to Parameter Identification of DC Motor Model," *Proceedings of the 2017 International Electrical Engineering Congress (iEECON-2017)*, Vol. 2, pp.711-714, 2017.

[31] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-Betar, M. A. Awadallah and X. S. Yang, "Variants of the Flower Pollination Algorithm: A Review," *Nature-Inspired Algorithms and Applied Optimization, Studies in Computational Intelligence*, Vol. 744, pp.91-118, 2018.

[32] C. A. Monje, Y. Q. Chen, B. M. Vinagre, D. Xue and V. Feliu, *Fractional-order Systems and Controls Fundamentals and Applications*, Springer-Verlag, 2010.

[33] D. Matignon, "Generalized Fractional Differential and Difference Equations: Stability Properties and Modelling Issues," *Proceedings of the Math. Theory of Networks and Systems Symposium*, 1998.

[34] K. Ogata, *Modern Control Engineering*, 5th edition, Prentice Hall, 2010.

[35] D. Puangdownreong, "Optimal PID Controller Design for DC Motor Speed Control System with Tracking and Regulating Constrained Optimization via Cuckoo Search," *Journal of Electrical Engineering & Technology*, Vol. 13, No. 1, pp.460-467, 2018.

[36] MathWorks, *System Identification Toolbox™*, MATLAB-R2017b, 2017.

[37] B. J. Glover, *Understanding Flowers and Flowering: An Integrated Approach*, Oxford University Press, 2007.

[38] P. Willmer, *Pollination and Floral Ecology*, Princeton University Press, 2011.

[39] K. Balasubramani and K. Marcus, "A Study on Flower Pollination Algorithm and Its Applications," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Vol. 3, pp.320-325, 2014.

[40] I. Pavlyukevich, "Lévy Flights, Non-Local Search and Simulated Annealing," *Journal of Computational Physics*, Vol. 226, pp.1830-1844, 2007.

[41] A. Tepljakov, E. Petlenkov and J. Belikov, "FOMCON: Fractional-Order Modeling and Control Toolbox for MATLAB," *Proceedings of the 18th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES '11)*, pp.684-689, 2011.

[42] A. Tepljakov, E. Petlenkov and J. Belikov, "FOMCON: a MATLAB Toolbox for Fractional-Order System Identification and Control," *International Journal of Microelectronics and Computer Science*, Vol. 2, No. 2, pp.51-62, 2011.

[43] A. Tepljakov, E. Petlenkov and J. Belikov, FOMCON toolbox, [Available Online: http://www.fomcon.net], 2011.

Deacha Puangdownreong received the B.Eng. degree in electrical engineering from Southeast Asia University (SAU), Bangkok, Thailand, in 1993, the M.Eng. degree in control engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand, in 1996, and the Ph.D. degree in electrical engineering from Suranaree University of Technology (SUT), Nakhon Ratchasima, Thailand, in 2005, respectively. Since 1994, he has been with the department of electrical engineering, Southeast Asia University, where he is currently an associated professor of electrical engineering. His research interests include intelligent control, system identification, optimal control, modern optimization and applications of metaheuristics to control engineering problems.