Prapapan KHLUABWANNARAT, Deacha PUANGDOWNREONG

Department of Electrical Engineering, Southeast Asia University, Bangkok, 10160, Thailand

doi:10.15199/48.2020.11.16

Parallel Flower Pollination Algorithm and Its Application to Fractional-Order PID Controller Design Optimization for BLDC Motor Speed Control System

***Abstract**. This paper proposes the newest modified version of the flower pollination algorithm (FPA) named the parallel flower pollination algorithm (PFPA) and its application to design the optimal fractional-order proportional-integral-derivative (FOPID) controller for the fractional-order brushless direct current (BLDC) motor speed control system. The proposed PFPA is based on the time sharing strategy employing the multiple point single strategy (MPSS) method which is efficiently run on a single CPU platform. The search performance of the PFPA over the original FPA investigated against 10 benchmark functions is presented in this paper. For the fractional-control application based on the modern optimization, results of the FOPID controller designed by the PFPA for the BLDC motor speed control system are compared with those of the integer-order PID (IOPID) controller designed by the PFPA. As results, it was found that the BLDC motor speed controlled system with the FOPID controller designed by the proposed PFPA can provide very satisfactory responses superior to that with the IOPID controller designed by the FPA, significantly.*

Streszczenie.** W artykule opisano nową zmodyfikowana wersję algorytmu zapytania FPA i jego zastosowanie do projektu sterownika do bezszczotkowego silnika BLDC. Zaprezentowano benchmark dziesięciu funkcji i potwierdzono lepsze parametry w porównaniu z konwencjonalnym sterownikiem IOPID.* ***Algorytm FPA I jego zastosowanie do projektu I optymalizacji sterownika do silnika bezszczotkowego silnika BLDC

Keywords: Parallel flower pollination algorithm, Fractional-order PID controller, Brushless DC motor speed control system, Modern optimization.
Słowa kluczowe: algorytm FPA, silnik BLDC, sterownik silnika, sterownik PID ułamkowego rzędu .

Introduction

To date, metaheuristic optimization techniques have become potential candidates and widely applied to various real-world control and automation engineering problems [1, 2, 3]. One of the most efficient metaheuristic optimization techniques is the flower pollination algorithm (FPA) proposed by Yang in 2012 [4]. The original FPA performed the superior search performance to the genetic algorithm (GA) and particle swarm optimization (PSO) [4, 5]. The global convergence properties of the FPA algorithm have been proven by Markov chain theory [6]. Moreover, the FPA was successfully conducted to optimize several real-world problems including power system, signal and image processing, wireless sensor networking, clustering and classification, antenna array, global function optimization, computer gaming, traveling transportation, structural and mechanical engineering, and control system design [7]. The state-of-the-art and various applications of the FPA have been reviewed and reported [7, 8]. In addition, many modified versions of the FPA have been developed to improve its search performance, for example, chaos-based FPA (CFPA) [9] and hybridization of FPA with the GA [10].

Regarding to control system analysis and design, the fractional-order proportional-integral-derivative (FOPID) controller was firstly proposed by Podlubny in 1994 [11, 12]. Based on the fractional calculus, the FOPID controller is an extended version of the conventional integer-order PID (IOPID) controller. Once compared with the IOPID controller, there are two extra parameters making the FOPID controller more efficient than the IOPID controller [13, 14]. Following the literature, the FOPID controller has been successfully conducted in many applications, for instance, process control, automatic voltage regulator, DC motor speed control, power electronic control, inverted pendulum control, gun control system, micro intelligent vehicles and wind turbine systems [15]. The state-of-the-art and backgrounds of the FOPID controller have been completely reported [15, 16].

In this paper, the newest modified version of the FPA named the parallel flower pollination algorithm (PFPA) is proposed to improve the search performance of the original FPA. Based on the time sharing strategy, the proposed PFPA employs the multiple point single strategy (MPSS) method which is efficiently run on a single CPU platform. With the proposed PFPA algorithm, it can be easily adapted to run on multiple platforms or multi-core processor. Then, the proposed PFPA is applied to design the optimal FOPID controller of the fractional-order brushless direct current (BLDC) motor speed control system based on the modern optimization approach.

Flower pollination algorithm (FPA)

Proposed by Yang [4], the original FPA algorithm is based on the pollination of flowering plants in nature which can be divided into two types of pollinator, i.e., biotic and abiotic, and can be divided by two types of pollination, i.e., cross-pollination and self-pollination. The original FPA algorithm is developed by four following rules [4, 5].

Rule-1 Biotic pollinator and cross-pollination are regarded as the global pollination process via Lévy flight random distribution by using (1) and (2), where $\boldsymbol{x}$ is solution, $\boldsymbol{g^*}$ is current best solution, L is Lévy flight random distribution and $\Gamma(\lambda)$ is the standard gamma function.

Rule-2 Abiotic pollinator and self-pollination are regarded as the local pollination process with random walk by using (3) and (4), where $\varepsilon \in [a, b]$ is a random drawn from a uniform distribution.

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

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

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

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

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \varepsilon(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t) \tag{3}$$

$$\varepsilon(\rho) = \begin{cases} 1/(b-a), & a \le \rho \le b \\ 0, & \rho < a \text{ or } \rho > b \end{cases} \tag{4}$$

Parallel flower pollination algorithm (PFPA)

To improve the search performance of the original FPA, the PFPA is proposed in this paper. As mentioned earlier, the proposed PFPA is intended for running on a single CPU platform. Assuming that there are FPA_h, h =1, 2,…,N in the proposed PFPA. The time sharing strategy employing the multiple point single strategy (MPSS) method used in the multipath adaptive tabu search (MATS) [17, 18] is conducted for the proposed PFPA. Once the CPU starts the search at the first generation (or iteration), the FPA_1 begins to running state, while other FPA_h, h = 2,…,N are in waiting state. Once the FPA_1 finishes its running state, it goes to the waiting state. At this time, the FPA_2 begins to running state, while the FPA_1 and other FPA_h, h = 3,…,N are in waiting state. The operation goes on in this manner until the FPA_N finishes its first generation. Afterward, the CPU then returns to start the second generation by running FPA_h, h =1, 2,…,N in sequential manner. The operation is repeated until one of the FPA hits the optimal solution. For the proposed PFPA in this paper, all FPA_h, h =1, 2,…,N will start with different initial solutions by using random process. Then, they will search for the better solution over the same search space. The PFPA algorithm is suitable for running on a single CPU platform. Nonetheless, it can be easily adapted for running on multi-core CPU or parallel platforms.

The proposed PFPA algorithm is represented by the pseudo code as shown in Fig. 1 and the flow diagram in Fig. 2. The proposed PFPA algorithm differs from the parallelized FPA [19] in that the flower population of the proposed PFPA will not be split into several groups like the parallelized FPA in [19]. All FPAs in the proposed PFPA will has the same amount of flower population. This leads the proposed PFPA possible to seek for the global solution faster.

```
Initialize:
- Objective function f(x), x = (x1, x2,…, xd) and search spaces
- For FPAh, h = 1,2…,N
      - Initialize a population of n flowers/pollen gametes with random
solutions
      - Find the best solution g* in the initial population
      - Define a switch probability p ∈ [0, 1]
- Set Max_Gen as the termination criteria (TC) and t = 1 as counter
while (t < Max_Gen)
      for h = 1 : N (all N FPAs in the PFPA)
            for i = 1 : n (all n flowers in the population)
                  if rand > p, (global pollination)
                        - Draw a step vector L via Lévy flight in (2)
                        - Activate global pollination in (1) to generate new solutions
                  else (local pollination)
                        - Draw e from a uniform distribution in [0, 1] in (4)
                        - Randomly choose j and k among all the solutions
                        - Invoke local pollination in (3) to generate new solutions
                  end if
                  - Evaluate new solutions
                  - If the better solutions are found, update g*
            end for
      end for
      - Update t
end while
- Find and report the current best solution g*
```

Fig.1. Pseudo code of proposed PFPA

Performance evaluation

To evaluate its search performance, the proposed PFPA is tested and compared with the original FPA. For comparison, the original FPA and the proposed PFPA are implemented by MATLAB version 2018b (License No.#40637337) run on Intel(R) Core(TM) i5–3470 CPU@3.60GHz, 4.0GB–RAM. Parameters of the original FPA are set according to Yang's recommendations [4, 5] and the preliminary study, i.e. the numbers of flower n = 20 and a switch probability p = 0.2 (20%) for switching between local pollination and global pollination. In this work, the PFPA consisting of five FPAs are assumed. The search parameters of all FPA_h, h =1, 2,…,5 are set as the same values of the original FPA. 100 trials are conducted for both the FPA and PFPA algorithms to find the global optima. Both algorithms will be stopped when these two termination criteria (TC) are satisfied, i.e., (i) the variations of function values are less than a given tolerance $\delta \leq 10^{-5}$ or (ii) the search generation reaches the maximum generation (Max_Gen = 1,000). The former criterion implies that the search is success, while the later means that the search is not success. Both FPA and PFPA are tested against 10 selected benchmark functions [20, 21] as summarized in Table 1, where $\boldsymbol{x}^*$ is the optimal solution, $f(\boldsymbol{x}^*)$ is the optimal function value and D is dimension.

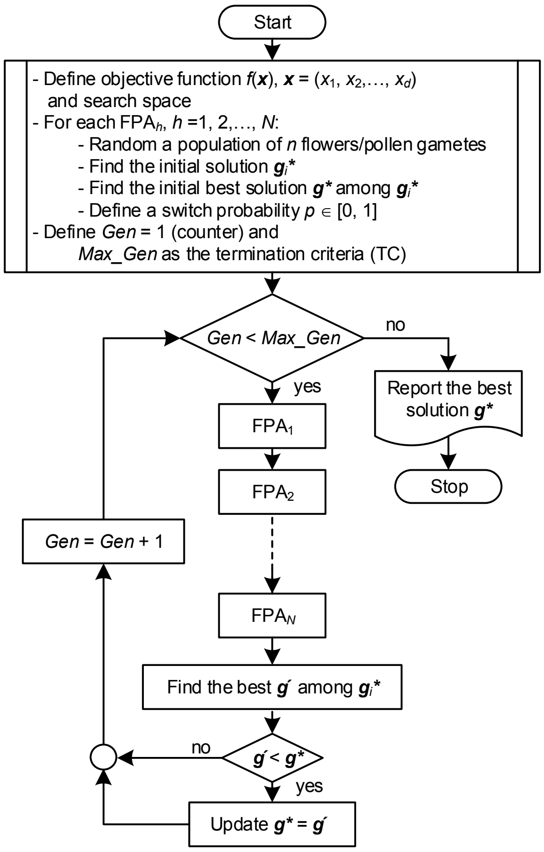


Fig.2. Flow diagram of proposed PFPA

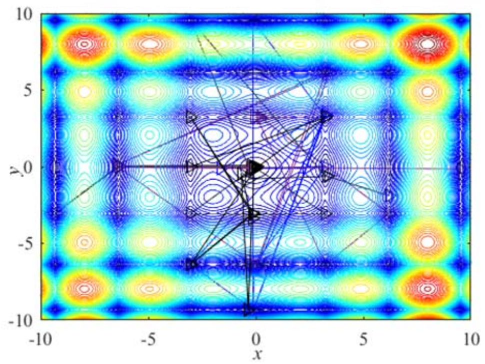
The comparison results between FPA and PFPA are summarized in Table 2. Data in Table 2 are presented by the following format, i.e. AE±SD(SR%), where the AE stands for the average number (mean) of function evaluations, the SD stands for the standard deviation and the SR stands for the success rate. The AE implies the searching time consumed. The less the AE, the less the searching time consumed. The SD implies the robustness of the algorithm. The less the SD, the more the robustness.

Referring to Table 2, the proposed PFPA performs better performance in finding the global optima with SR = 100% of all selected benchmark functions than the original FPA. Although the searching time (varying to the AE) of proposed PFPA is more than that of the original FPA, it can be reduced by using multiple platforms or multi-core processor. Fig. 3 shows the example of the PFPA's search results of global minimum finding of AF function (f_1). Results of other functions are omitted because they have a similar form to that of f_1 as shown in Fig. 3.

Table 1. Benchmark functions

No.	Function names	Functions, Search space, Optimal solution and Optimal function value
1.	Alpine function (AF)	$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1x_i \rvert, -10 \le x_i \le 10, D = 2, \boldsymbol{x^*} = (0,\ldots,0), f_1(\boldsymbol{x^*}) = 0$
2.	Bird function (BF)	$f_2(\boldsymbol{x}) = \sin(x_1)e^{[1-\cos(x_2)]^2} + \cos(x_2)e^{[1-\sin(x_1)]^2} + (x_1 - x_2)^2,$ $-2\pi \le x_i \le 2\pi, \boldsymbol{x^*} = (4.7010, 3.1529), (-1.5821, -3.1302), f_2(\boldsymbol{x^*}) = -106.7645$
3.	Easom function (EF)	$f_3(\boldsymbol{x}) = -\cos(x_1)\cos(x_2)e^{[-(x_1-\pi)^2-(x_2-\pi)^2]}, -100 \le x_i \le 100, \boldsymbol{x^*} = (\pi,\pi), f_3(\boldsymbol{x^*}) = -1$
4.	Holder-Table function (HTF)	$f_4(\boldsymbol{x}) = -\lvert \sin(x_1)\cos(x_2)e^{\lvert 1-(x_1+x_2)^{0.5/\pi} \rvert} \rvert,$ $-10 \le x_i \le 10, \boldsymbol{x^*} = (\pm 8.0550, \pm 9.6646), f_4(\boldsymbol{x^*}) = -19.2085$
5.	Pen-Holder function (PHF)	$f_5(\boldsymbol{x}) = -\exp[\lvert \cos(x_1)\cos(x_2)e^{\lvert 1-[x_1^2+x_2^2]^{0.5/\pi} \rvert} \rvert^{-1}],$ $-11 \le x_i \le 11, \boldsymbol{x^*} = (\pm 9.6462, \pm 9.6462), f_5(\boldsymbol{x^*}) = -0.9635$
6.	Rastrigin function (RF)	$f_6(\boldsymbol{x}) = 10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)], -5.12 \le x_i \le 5.12, D = 2, \boldsymbol{x^*} = (0,0), f_6(\boldsymbol{x^*}) = 0$
7.	Schaffer-1 function (S1F)	$f_7(\boldsymbol{x}) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2)^2 - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}, -100 \le x_i \le 100, \boldsymbol{x^*} = (0,0), f_7(\boldsymbol{x^*}) = 0$
8.	Schaffer-2 function (S2F)	$f_8(\boldsymbol{x}) = 0.5 + \frac{\sin^2(\cos\lvert x_1^2 - x_2^2 \rvert) - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}, -100 \le x_i \le 100,$ $\boldsymbol{x^*} = (0, 1.2531), f_8(\boldsymbol{x^*}) = 0.0016$
9.	Yang-1 function (Y1F)	$f_9(\boldsymbol{x}) = \left(\sum_{i=1}^{D}\lvert x_i \rvert\right)e^{-\sum_{i=1}^{D}\sin(x_i^2)}, -2\pi \le x_i \le 2\pi, D = 2, \boldsymbol{x^*} = (0,0), f_9(\boldsymbol{x^*}) = 0$
10.	Yang-2 function (Y2F)	$f_{10}(\boldsymbol{x}) = e^{-\sum_{i=1}^{D}(x_i/\beta)^{2m}} - 2e^{-\sum_{i=1}^{D}x_i^2} \cdot \prod_{i=1}^{D}\cos^2(x_i),$ $-20 \le x_i \le 20, D = 2, m = 5, \beta = 15, \boldsymbol{x^*} = (0,0), f_{10}(\boldsymbol{x^*}) = -1$

(a) Search movement



(b) Convergent rates

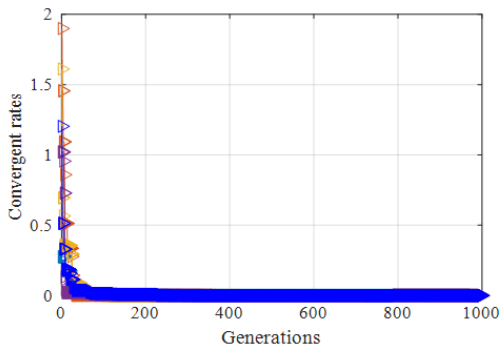


Fig.3. Results of minimum finding of AF by PFPA

Table 2. Results of performance evaluation of original FPA and proposed PFPA

Func-tions	Original FPA	Proposed PFPA
AF	15,287.80±5,127.90(55%)	55,984±3,477.10(100%)
BF	1,555.00±436.96(100%)	7,406±424.69(100%)
EF	14,933.40±1,809.00(100%)	74,801±1,919.94(100%)
HTF	9,038.60±2,986.60(99%)	41,645±2,976.47(100%)
PHF	2,726.60±1,363.63(100%)	14,457±1,236.97(100%)
RF	17,739.00±2,137.09(78%)	85,160±1,813.27(100%)
S1F	13,776.20±3,317.86(97%)	65,299±3,522.93(100%)
S2F	16,669.80±3,553.77(68%)	69,690±3,814.26(100%)
Y1F	18,332.00±1,568.85(77%)	89,681±1,514.60(100%)
Y2F	16,756.40±3,549.08(69%)	75,688±3,512.75(100%)

FOPID controller design by PFPA algorithm

Regarding to the 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 (5), where $\alpha \in \Re$ stands for the order of operation. Under zero initial conditions for order α ($0 < \alpha < 1$), the Laplace transform of the continuous integro-differential operator in (5) can be expressed in (6) [22, 23]. A general fractional-order system can be formulated by a fractional differential equation as stated in (7) or by the corresponding fractional-order transfer function as expressed in (8), where a_i ($i = 0,\ldots,n$) and b_i ($i = 0,\ldots,m$) are constant, and α_i ($i = 0,\ldots,n$) and β_i ($i = 0,\ldots,m$) are real numbers which can be arranged as $\alpha_n > \alpha_{n-1} > \cdots > \alpha_0$ and $\beta_m > \beta_{m-1} > \cdots > \beta_0$.

$$(5) \quad {}_aD_t^{\alpha} = \begin{cases} \dfrac{d^{\alpha}}{dt^{\alpha}} & \Re(\alpha) > 0 \\ 1 & \Re(\alpha) = 0 \\ \int_a^t (d\tau)^{-\alpha} & \Re(\alpha) < 0 \end{cases}$$

$$(6) \quad \mathcal{L}\{{}_aD_t^{\pm\alpha} f(t)\} = s^{\pm\alpha}F(s)$$

(7) $$\begin{cases} a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots \\ \quad + a_1 D^{\alpha_1} y(t) + a_0 D^{\alpha_0} y(t) = \\ \qquad b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + \cdots \\ \qquad\quad + b_1 D^{\beta_1} u(t) + b_0 D^{\beta_0} u(t) \end{cases}$$

(8) $$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \cdots + a_0 s^{\alpha_0}}$$

Following the literature, there are several approaches to model the BLDC motor [24, 25, 26, 27]. In this work, the BLDC motor of 350 W, 24 VDC, 0.7 A, 300 rpm in laboratory as shown in Fig. 4 is used as the testing rig. The fractional-order model of such the BLDC motor was identified by the original FPA as stated in (9) [28]. A good agreement of the fractional-order model $G_p(s)$ in (9) to actual speed dynamics at 305 rpm of BLDC motor is depicted in Fig. 5. The fractional-order model $G_p(s)$ of the BLDC motor in (9) will be used as the plant model of this work.

(9) $$G_p(s) = \frac{1.0}{0.029 s^{2.658} + 0.4784 s^{1.2376} + 1.1075 s^{0.0443}}$$

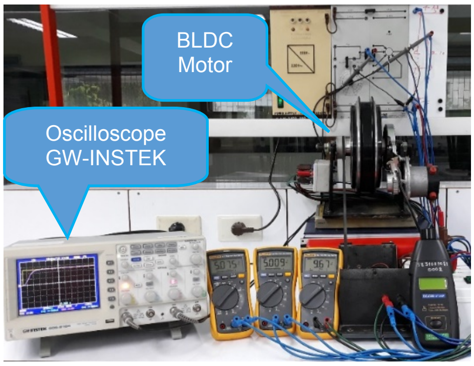


Fig.4. BLDC motor testing rig

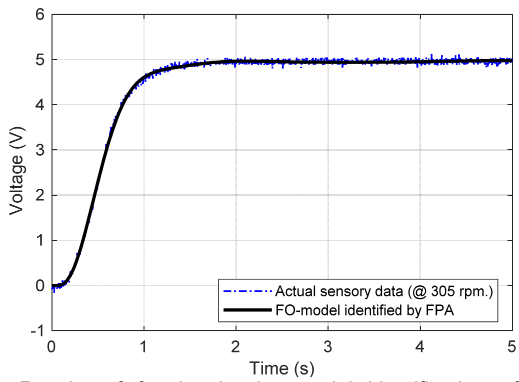


Fig.5. Results of fractional-order model identification of BLDC motor [28]

The FOPID feedback control loop can be represented by the block diagram as shown in Fig. 6, where *r*(*t*) is the reference input signal, *c*(*t*) is the controlled output signal, *e*(*t*) is the error signal between *r*(*t*) and *c*(*t*), *u*(*t*) is the control signal and *d*(*t*) is the disturbance signal. The FOPID controller model in time-domain and *s*-domain are stated in (10) and (11), respectively [11, 12], where K_p is the proportional gain, K_i is the integration gain, K_d is the derivative gain, λ is the integration order and μ is the derivative order. The FOPID will receive *e*(*t*) to be proceeded and generate *u*(*t*) to control plant for giving the satisfactory *c*(*t*) which tracks *r*(*t*) (reduce *e*(*t*)) and regulates *d*(*t*), simultaneously. Once λ and μ are set as one ($\lambda = \mu =$ 1), the FOPID in (11) becomes the conventional IOPID controller [29, 30].

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

(11) $$G_c(s) = K_p + \frac{K_i}{s^{\lambda}} + K_d s^{\mu}, \quad 0 < \lambda, \mu < 2$$

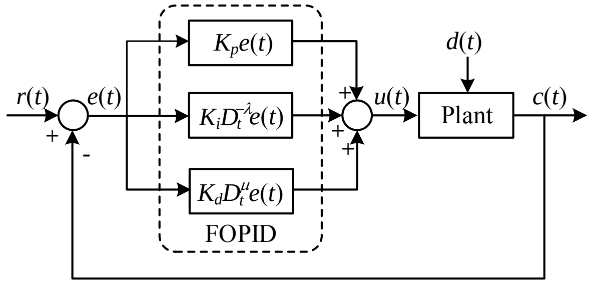


Fig.6. FOPID feedback control loop

The PFPA-based FOPID controller design framework for the fractional-order BLDC motor speed control system is represented by the block diagram as shown in Fig. 7. This framework is adapted from the controller design optimization [1, 31, 32] based on the modern optimization. Referring to Fig. 7, the objective function *f*(•) is set as the sum of squared error (SSE) between the reference input signal *R*(*s*) and the controlled output signal *C*(*s*) of the BLDC motor speed controlled system as stated in (12). The objective function *f*(•) in (12) will be fed to the PFPA to be minimized by searching for the optimal values of the FOPID's parameters (K_p, K_i, K_d, λ and μ). The search process needs to meet the design constrained functions and the search spaces as stated in (13), where t_r is the rise time, M_p is the maximum percent overshoot, t_s is the settling time and e_{ss} is the steady-state error. The constrained functions in (13) is a priory set from the preliminary study of the considered system.

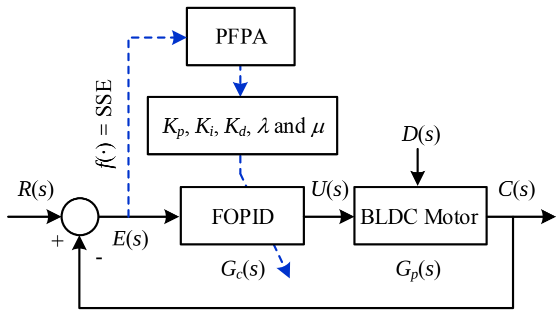


Fig.7. FOPID design framework for BLDC motor speed control by PFPA

(12) $$\textbf{Min} \quad f(K_p, K_i, K_d, \lambda, \mu) = \sum_{i=1}^{N} [R_i - C_i]^2$$

(13) $$\begin{cases} \textit{Subject to} \\ t_r \le 1.0 \text{ s.}, \quad M_p \le 10\%, \\ t_s \le 2.0 \text{ s.}, \quad e_{ss} \le 0.01\%, \\ 0 < K_p \le 5, \quad 0 < K_i \le 10, \\ 0 < K_d \le 1.0, \quad 0 < \lambda, \mu < 2.0 \end{cases}$$

Results and discussions

In order to optimize the FOPID controller for the BLDC motor control system, the PFPA algorithm was coded by MATLAB version 2018b (License No.#40637337) run on

Intel(R) Core(TM) i5-3470 CPU@3.60GHz, 4.0GB-RAM. The FOMCON toolbox [33, 34] is conducted for fractional control system simulation. In this application, the PFPA consisting of five FPAs (N = 5) is assumed. Searching parameters of all five FPAs are set as follows: n = 20 and p = 0.2 (20%). *Max_Gen* = 100 is set as the TC in each trial. 50 trials are processed to find the optimal solution (K_p, K_i, K_d, λ and μ). For comparison with the IOPID controller, λ and μ in (13) will be set as 1.0 (λ = 1.0 and μ = 1.0). After the search process stopped, the IOPID and FOPID controllers optimized by the PFPA for the BLDC motor speed control system are successfully obtained as stated in (14) and (15), respectively. It was found that both IOPID and FOPID controllers are successfully optimized by the PFPA according to the given search spaces in (13).

(14) $$G_c(s)\big|_{IOPID} = 1.9123 + \frac{4.4996}{s} + 0.2421s$$

(15) $$G_c(s)\big|_{FOPID} = 2.2986 + \frac{4.7699}{s^{0.9789}} + 0.2341s^{1.2102}$$

The unit-step command-tracking and unit-step load-regulating responses of the BLDC motor speed control system without and with the IOPID and FOPID controllers in (14) and (15) are depicted in Fig. 8 and Fig. 9, respectively. Results of command-tracking and load-regulating system responses are summarized in Table 3 and Table 4, where M_{p_reg} is the maximum percent overshoot from load regulation, t_{reg} is the regulating time.

From Figs. 8 - 9 and Tables 3 - 4, it was found that both IOPID and FOPID are completely optimized by the PFPA according to the preset design constrains in (13). From Fig. 8 and Table 3, the BLDC motor speed controlled system with the FOPID controller provides smoother command-tracking response than that with the IOPID controller. From Fig. 9 and Table 4, the BLDC motor speed controlled system with the FOPID controller provides faster load-regulating response than that with the IOPID controller. It can be noticed that the BLDC motor speed controlled system with the FOPID controller designed by the proposed PFPA can provide very satisfactory responses according to the given design specifications and superior to the IOPID controller designed by the FPA, significantly.

Table 3. Results of unit-step command-tracking responses of the BLDC motor control systems

Controllers	Responses			
	t_r (s)	M_p (%)	t_s (s)	e_{ss} (%)
without	2.01	0.00	1.43	0.00
IOPID	0.51	7.14	1.58	0.00
FOPID	0.50	0.08	0.47	0.00

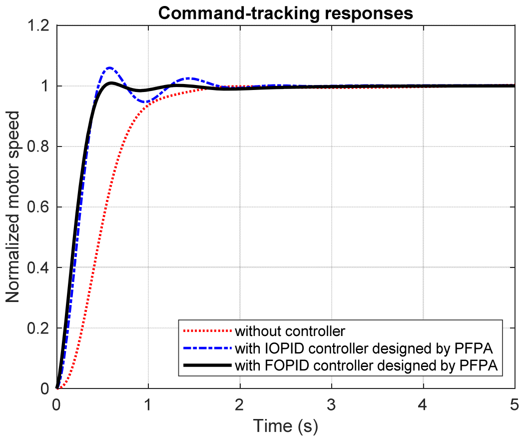


Fig.8. Unit-step command-tracking responses of the BLDC motor speed control system without and with IOPID and FOPID controllers designed by PFPA

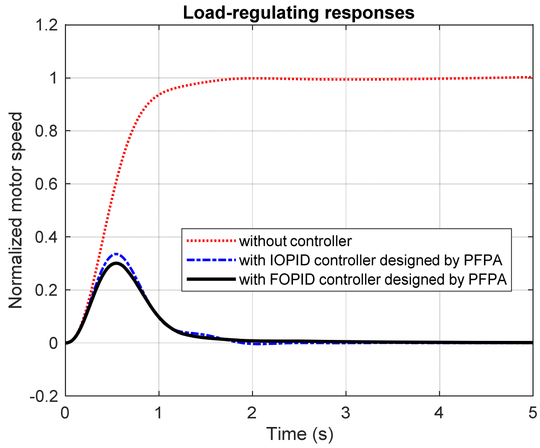


Fig.9. Unit-step load-regulating responses of the BLDC motor speed control system without and with IOPID and FOPID controllers designed by PFPA

Table 4. Results of unit-step load-regulating responses of the BLDC motor control systems

Controllers	Responses		
	M_{p_reg} (s)	t_{reg} (s)	e_{ss} (%)
without	-----cannot be regulated-----		
IOPID	33.02	1.69	0.00
FOPID	30.08	1.48	0.00

Conclusions

The parallel flower pollination algorithm (PFPA) and its control system design application have been proposed in this paper. The proposed PFPA, one of the newest modifed versions of the original FPA, has been developed to improve its search performance by using the time sharing strategy for running on a single CPU platform. This strategy employing the MPSS method is efficiently run on a single CPU platform and easily adapted for running on multiple platforms. The PFPA has been tested against 10 standard benchmark functions to perform its effectiveness. Once compared with the original FPA, the proposed PFPA has shown more efficient in global minimum finding of all functions. Certainly, the PFPA run on single CPU platform will spend more searching time than the original FPA. This can be reduced by implementation on multiple platforms or multi-core processor. Afterwards, the proposed PFPA has been applied to the FOPID controller design optimization for the fractional-order BLDC motor speed control system based on the modern optimization. Results have shown that the PFPA can provide optimal IOPID and FOPID controllers for the BLDC motor speed control system. Moreover, the command-tracking and load-regulating responses of the BLDC motor system controlled by the FOPID are faster and smoother than those by the IOPID controller, significantly. For future work, the fractional-order PIDA (FOPIDA) controller designed by the PFPA (or other promising metaheuristics) will be alternatively studied.

***Authors**: Asst. Prof. Prapapan Khluabwannarat, Department of Electrical Engineering, Faculty of Engineering, Southeast Asia University (SAU), 19/1 Petchkasem Rd, Nonghkaem, Bangkok, 10160, Thailand, Email: prapapank@sau.ac.th; Assoc. Prof. Dr. Deacha Puangdownreong, Department of Electrical Engineering, Faculty of Engineering, Southeast Asia University (SAU), 19/1 Petchkasem Rd, Nonghkaem, Bangkok, 10160, Thailand, Email: deachap@sau.ac.th.*

REFERENCES

[1] Zakian V., *Control Systems Design: A New Framework*, Springer-Verlag, (2005)
[2] Yang X.S., *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons, (2010)

[3] Talbi E.G., *Metaheuristics form Design to Implementation*, John Wiley & Sons, Hoboken, (2009)
[4] Yang X.S., Flower Pollination Algorithm for Global Optimization, *Unconventional Computation and Natural Computation, Lecture Notes in Computer Science*, 7445 (2012), 240-249
[5] Yang X.S., Karamanoglua M., He X., Multi-Objective Flower Algorithm for Optimization, *Procedia Computer Science*, 18 (2013), 861-868
[6] He X., Yang X.S., Karamanoglu M., Zhao Y., Global Convergence Analysis of the Flower Pollination Algorithm: a Discretetime Markov Chain Approach, *Proceedings of the International Conference on Computational Science (ICCS2017)*, (2017), 1354-1363
[7] Chiroma H., Shuib N.L.M., Muaz S.A., Abubakar A.I., Ila L.B., Maitama J.Z., A Review of the Applications of Bio-Inspired Flower Pollination Algorithm, *Procedia Computer Science*, 62 (2015), 435-441
[8] Alyasseri Z.A.A., Khader A.T., Al-Betar M.A., Awadallah M.A., Yang X.S., Variants of the Flower Pollination Algorithm: a Review, *Nature-Inspired Algorithms and Applied Optimization: Studies in Computational Intelligence*, 744 (2018), 91-118
[9] Metwalli M.A.B., Hezam I., Yardım D., Ozkan I.A., Saritas I., Aslam D.M., A Modified Fower Pollination Algorithm for Fractional Programming Problems, *International Journal of Intelligent Systems and Applications in Engineering*, 3 (2015)
[10] Abdel-Baset M., Hezam I.M., An E□ective Hybrid Fower Pollination and Genetic Algorithm for Constrained Optimization Problems, *Advanced Engineering Technology and Application*, 4 (2015), 27-34
[11] Podlubny I., *Fractional-Order Systems and Fractional-Order Controllers*, UEF-03-94, Slovak Academy of Sciences, Kosice, (1994)
[12] Podlubny I., Fractional-Order Systems and $PI^{\lambda}D^{\mu}$-Controllers, *IEEE Transactions on Automatic Control*, 44 (1999), 208-214
[13] Monje C.A., Chen Y.Q., Vinagre B.M., Xue D., Feliu V., *Fractional-order Systems and Controls: Fundamentals and Applications*, Springer-Verlag London, (2010)
[14] Valério D., Costa J., A Review of Tuning Methods for Fractional PIDs, *Proceedings of the 4th IFAC Workshop on Fractional Differentiation and Its Applications*, (2010)
[15] Shah P., Agashe A., Review of Fractional PID Controller. *Mechatronics*, 38 (2016), 29-41
[16] Chen Y. Q., Petráš I., Dingyü X., Fractional Order Control - A Tutorial, *Proceedings of the American Control Conference*, (2009), 1397-1411
[17] Kluabwang J., Puangdownreong D., Sujitjorn S., Multi-Path Aaptive Tabu Search for a Vehicle Control Problem, *Journal of Applied Mathematics*, (2012), 1-20
[18] Puangdownreong D., Kluabwang J., Sujitjorn S., Multipath Adaptive Tabu Search: its Convergence and Application to Identification Problem, *Journal of Physical Sciences*, 7 (2012), 5288-5296
[19] Nguyen T.T., Shieh C.S., Horng M.F., Dao T.K., Ngo T.G., Parallelized Flower Pollination Algorithm with a Communication Strategy, *Proceedings of the 7th International Conference on Knowledge and Systems Engineering (KSE)*, (2015), 103-107
[20] Ali M.M., Khompatraporn C., Zabinsky Z.B., A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems, *Journal of Global Optimization*, 31 (2005), 635-672
[21] Jamil M., Yang X.S., A Literature Survey of Benchmark Functions for Global Optimization Problems, *International Journal of Mathematical Modelling and Numerical Optimisation*, 4 (2013), 150-194
[22] Miller K.S., Ross B., *An Introduction to the Fractional Calculus and Fractional Differential Equations*, Willey, New York, (1993)
[23] Podlubny I., *Fractional Differential Equations*, Academic Press. San Diego, (1999)
[24] Popenda A., A Control Strategy of a BLDC Motor, *Przeglad Elektrotechniczny*, ISSN 0033-2097, R. 89 NR 12/2013, (2013), 188-191
[25] Popenda A., Modelling of Multi-Phase BLDC Motor, *Przeglad Elektrotechniczny*, ISSN 0033-2097, R. 94 NR 1/2018, (2018), 85-88
[26] Popenda A., Modelling of BLDC Motor Energized by Different Converter Systems, *Przeglad Elektrotechniczny*, ISSN 0033-2097, R. 94 NR 1/2018, (2018), 81-84
[27] Popenda A., Nowak M., Modelling of BLDC Motor with Different Fashions of Winding Connection, *Przeglad Elektrotechniczny*, ISSN 0033-2097, R. 95 NR 2/2019, (2019), 92-95
[28] Khluabwannarat P., Nawikavatan A. Puangdownreong D., Fractional-Order Model Parameter Identification of BLDC Motor by Flower Pollination Algorithm, *WSEAS Transactions on Systems and Control*, 13 (2018), 573-579
[29] Pourhossein H., Zare A., Monfared M., Hybrid Modeling and PID-PSO Control of Buck-Boost Chopper, *Przeglad Elektrotechniczny*, ISSN 0033-2097, R. 88 NR 8/2012, (2012), 187-191
[30] Azzouz S., Messalti S., Harrag A., Innovative PID-GA MPPT Controller for Extraction of Maximum Power From Variable Wind Turbine, *Przeglad Elektrotechniczny*, ISSN 0033-2097, R. 95 NR 8/2019, (2019), 115-120
[31] Thammarat C., Puangdownreong D., CS-Based Optimal $PI^{\lambda}D^{\mu}$ Controller Design for Induction Motor Speed Control System, *International Journal on Electrical Engineering and Informatics*, 11 (2019), 638-661
[32] Jitwang T., Puangdownreong D., Application of Cuckoo Search to Robust PIDA Controller Design for Liquid-Level System, *International Journal of Innovative Computing, Information and Control*, 16 (2020), 189-205
[33] Tepljakov A., Petlenkov E., Belikov J., 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)*, (2011), 684-689
[34] Tepljakov A., Petlenkov E., Belikov J., FOMCON: a MATLAB Toolbox for Fractional-Order System Identification and Control, *International Journal of Microelectronics and Computer Science*, 2 (2011), 51-62