ICIC Express Letters
Volume **15**, Number **2**, February **2021**
 ISSN 1881-803X
pp. 165–172

OPTIMAL FRACTIONAL-ORDER PID CONTROLLER DESIGN FOR BLDC MOTOR SPEED CONTROL SYSTEM BY USING PARALLEL FLOWER POLLINATION ALGORITHM

Prapapan Khluabwannarat and Deacha Puangdownreong

Department of Electrical Engineering
Southeast Asia University
19/1 Petchkasem Rd., Nongkhangphlu, Nonghkaem, Bangkok 10160, Thailand
deachap@sau.ac.th

Received July 2020; accepted September 2020

Abstract. *The fractional-order PID (FOPID) controller was introduced over two decades and demonstrated to perform the better responses in comparison with the conventional integer-order PID (IOPID) controller. The optimal FOPID controller can be achieved by the powerful metaheuristic search techniques based on modern optimization context. In this paper, the optimal design of the FOPID controller for the brushless direct current (BLDC) motor speed control system by using the parallel flower pollination algorithm (PFPA) is presented. The PFPA, one of the modified versions of the original flower pollination algorithm (FPA), is introduced based on the time sharing strategy employing the multiple point single strategy (MPSS) method which is efficiently run on a single CPU platform. As results, the optimal FOPID controller can be successfully optimized by the PFPA for the BLDC motor speed control system according to the given design specification. Moreover, the FOPID controller can yield very satisfactory responses superior to the IOPID controller.*
Keywords: Fractional-order PID controller, Parallel flower pollination algorithm, Brushless DC motor speed control system, Modern optimization

1. **Introduction.** Based on the fractional calculus, the fractional-order PID (FOPID) controller was firstly proposed by Podlubny in 1994 [1,2] as an extended version of the conventional integer-order PID (IOPID) controller. The FOPID possesses five tuning parameters, i.e., proportional gain (K_p), integral gain (K_i), derivative gain (K_d), integral order (λ) and derivative order (μ), whereas the conventional IOPID controller consists of only three tuning parameters, i.e., K_p, K_i and K_d. Podlubny proved the superiority of the FOPID to the IOPID when applied for control systems [1,2]. Review and tutorial articles of the FOPID controller providing the state of the art and its backgrounds have been completely reported [3,4]. Control theorists believe that since the IOPID controller dominates the industry, the FOPID controller will gain increasing impact and wide acceptance [3,4].

To date, metaheuristic optimization searching techniques have become potential candidates and widely applied to various real-world engineering problems [5,6]. One of the most efficient metaheuristic optimization techniques is the flower pollination algorithm (FPA) proposed by Yang in 2012 [7]. The FPA performed the superior search performance to the genetic algorithm (GA) and particle swarm optimization (PSO) [7]. The global convergence properties of the FPA algorithm have been proven by Markov chain theory [8]. Moreover, the FPA was successfully applied to optimizing many real-world engineering problems including power systems, signal and image processing, wireless sensor networking, antenna array, structural and mechanical engineering as well as control

DOI: 10.24507/icicel.15.02.165

systems design. The state-of-the-art and various applications of the FPA have been reviewed and reported [9,10]. In addition, many modified versions of the FPA have been developed to improve its search performance, for example, the chaos-based FPA (CFPA) [11] and the hybridization of FPA with the GA [12].

Following the literature, the brushless DC (BLDC) motor is one of the motor types rapidly gaining popularity. The BLDC motor has been used in industries since 1970's such as appliances, automotive, aerospace, consumer, medical, industrial automation equipment and instrumentation [13]. Controlling the BLDC motor can be effectively achieved under the feedback control loop with some potential controllers such as the IOPID controller designed by the PSO [14], the artificial neural network fuzzy inference system (ANFIS) designed by the hybrid PSO [15] and the FOPID controller designed by the original FPA [16]. In 2020, the parallel flower pollination algorithm (PFPA) was firstly proposed [17] as one of the modified versions of the original FPA to enhance the search performance. The PFPA algorithm 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 PFPA performed better search performance than the original FPA for function optimization problems [17]. To extend the performance of the PFPA algorithm for other real-world control applications, the optimal design of the FOPID controller for the BLDC motor speed control system by using the PFPA algorithm is proposed in this paper. Superiority and advantages of the FOPID controller to the conventional IOPID controller designed by the PFPA for BLDC motor speed control system will be compared and reported.

2. **FOPID Feedback Control Loop.** 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 (1), 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 (1) can be expressed in (2) [1], where $f(t)$ is time-domain function and $F(s)$ is s-domain function.

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

$$ \mathcal{L}\left\{{}_aD_t^{\pm\alpha} f(t)\right\} = s^{\pm\alpha}F(s) \tag{2} $$

The FOPID feedback control loop can be represented by the block diagram as shown in Figure 1, 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 models in time-domain and s-domain are stated in (3) and (4), respectively [1,2], where K_p is the proportional gain, K_i is the integral gain, K_d is the derivative gain, λ is the integral 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 $r(t)$ to reduce $e(t)$ and regulates $d(t)$, simultaneously.

$$ u(t) = K_pe(t) + K_iD_t^{-\lambda}e(t) + K_dD_t^{\mu}e(t) \tag{3} $$

$$ G_c(s) = K_p + \frac{K_i}{s^{\lambda}} + K_ds^{\mu}, \quad 0 < \lambda, \mu < 2 \tag{4} $$

Relationship between the conventional IOPID and FOPID controllers can be represented in Figure 2. In general, the range of fractional orders (λ and μ) is varied from 0 to 2. Referring to Figure 2, it was found that if $\lambda = 0$ and $\mu = 0$, it is the conventional P

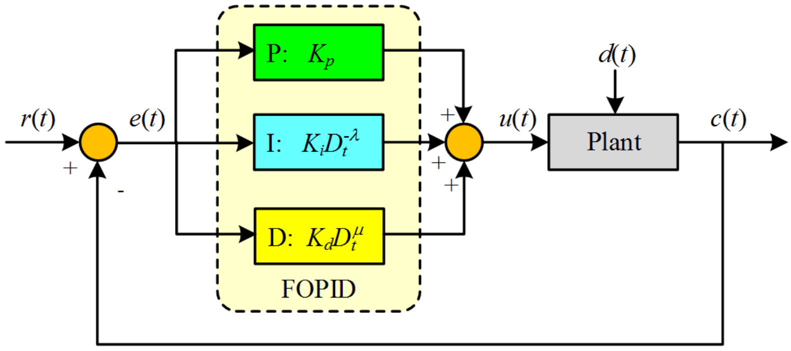

FIGURE 1. FOPID feedback control loop

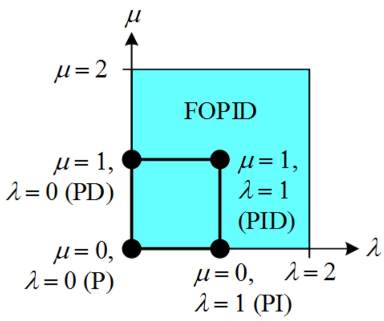

FIGURE 2. Relationship between IOPID and FOPID controllers

(IOP) controller, if $\lambda = 1$ and $\mu = 0$, it is the conventional PI (IOPI) controller, if $\lambda = 0$ and $\mu = 1$, it is the conventional PD (IOPD) controller and if $\lambda = 1$ and $\mu = 1$, it is the conventional PID (IOPID) controller. For FOPID controller, both λ and μ can be varied from 0 to 2 as a fractional value. These make the FOPID more flexible than the IOPID for tuning the optimal parameters.

3. **Parallel Flower Pollination Algorithm.** The parallel flower pollination algorithm (or PFPA) was proposed to enhance the search performance of the original FPA [17]. The PFPA possesses FPA_h, $h = 1, 2, \ldots, N$ for parallel cooperation search manner over the same search space. For the use on a single CPU platform, the time sharing strategy employing the MPSS method [18] is conducted as represented by the simple diagram shown in Figure 3. Once the CPU starts the search at the first generation (or iteration),

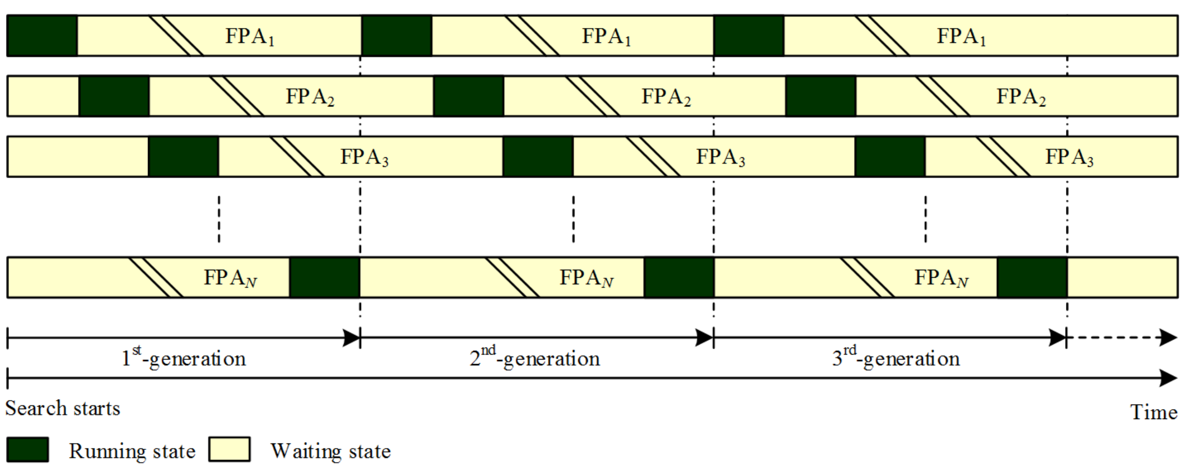

FIGURE 3. Time sharing with multiple point single strategy (MPSS) method

the FPA_1 begins to running state, while other FPA_h, $h = 2, \ldots, N$ 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, $h = 3, \ldots, N$ 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, $h = 1, 2, \ldots, N$ in sequential manner. The operation is repeated until one of the FPAs hits the optimal solution. The PFPA algorithm is suitable for running on a single CPU platform. Nonetheless, it can be easily adapted for use on multi-core CPU or parallel platforms.

For each FPA in the PFPA algorithm, the optimal solution can be obtained by two ways according to two types of flower pollinator in nature, i.e., biotic cross-pollination (global pollination) and abiotic self-pollination (local pollination) [7]. A solution $\boldsymbol{x}_i$ is equivalent to a flower and/or a pollen gamete. For global pollination, flower pollens are carried by the biotic pollinators. With random drawn from a Lévy flight distribution, pollens can travel over a long distance as expressed in (5), 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 (6), while $\Gamma(\lambda)$ is the standard gamma function. For local pollination, flower pollens are carried by the abiotic pollinators. The local pollination can be represented by (7), where $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ are pollens from the different flowers of the same plant species, while ε stands for random drawn from a uniform distribution as stated in (8), where a and b are bounds of random process. A switch probability p is used for switching between local and global pollinations. The PFPA algorithm consisting of FPA_h, $h = 1, 2, \ldots, N$ can be represented by the pseudo code as shown in Figure 4.

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + L\left(\boldsymbol{x}_i^t - \boldsymbol{g}^*\right) \tag{5}$$

$$L \approx \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0) \tag{6}$$

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \varepsilon\left(\boldsymbol{x}_j^t - \boldsymbol{x}_k^t\right) \tag{7}$$

$$\varepsilon(\rho) = \begin{cases} 1/(b-a), & a \le \rho \le b \\ 0, & \rho < a \text{ or } \rho > b \end{cases} \tag{8}$$

4. **Results and Discussions.** The FOPID controller design optimization framework for the BLDC motor speed control system by the PFPA is represented by the block diagram as shown in Figure 5. This framework is adapted from the controller design optimization [19,20] based on the modern optimization. In this work, the BLDC motor of 350 W, 24 VDC, 0.7 A, 300 rpm in laboratory is used as the testing rig. The fractional-order model of such the BLDC motor was identified as stated in (9) [17]. The BLDC motor model $G_p(s)$ in (9) will be used as the plant model in Figure 5.

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

Referring to Figure 5, the objective function $f(\cdot)$ 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 (10). The objective function $f(\cdot) =$ SSE in (10) 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 PFPA will iteratively search for the optimal values of K_p, K_i, K_d, λ and μ appearing in $G_c(s)$ as stated in (4). The search process needs to meet the inequality constrained functions and the search spaces as stated in (11), 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 (11) are set from the

```
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 (6)
                - Activate global pollination in (5) to generate new solutions
            else (local pollination)
                - Draw ε from a uniform distribution in [0, 1] in (8)
                - Randomly choose j and k among all the solutions
                - Invoke local pollination in (7) 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*
```

FIGURE 4. Pseudo code of PFPA algorithm

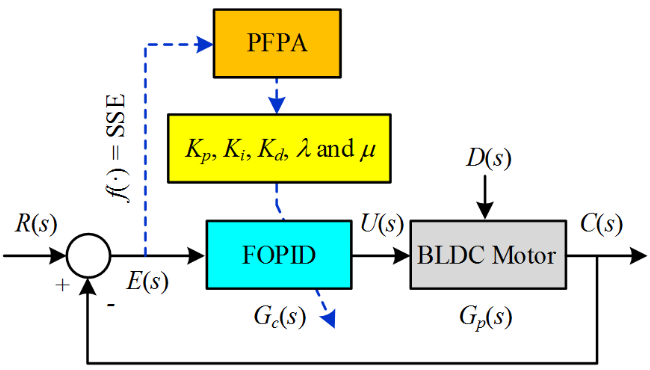

FIGURE 5. FOPID design framework for BLDC motor speed control by PFPA

preliminary study of the considered system.

$$\textbf{Minimize } f(K_p, K_i, K_d, \lambda, \mu) = \sum_{i=1}^{N} [R_i - C_i]^2 \tag{10}$$

$$\text{subject to} \quad \left.\begin{array}{l} t_r \leq 1.0 \text{ s}, \ M_p \leq 10\%, \ t_s \leq 2.0 \text{ s}, \ e_{ss} \leq 0.01\%, \\ 0 < K_p \leq 5, \ 0 < K_i \leq 10, \ 0 < K_d \leq 1.0, \\ 0 < \lambda < 2.0, \ 0 < \mu < 2.0 \end{array}\right\} \tag{11}$$

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 in order to design the FOPID controller for the BLDC motor control system. The FOMCON toolbox [21,22] is utilized for fractional-order control system simulation. In this work, 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 (11) will be set as 1.0 ($\lambda = 1.0$ and $\mu = 1.0$). 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 (12) and (13), respectively.

$$G_c(s)|_{IOPID} = 1.9123 + \frac{4.4996}{s} + 0.2421s \tag{12}$$

$$G_c(s)|_{FOPID} = 2.2986 + \frac{4.7699}{s^{0.9789}} + 0.2341s^{1.2102} \tag{13}$$

The unit-step command-tracking responses in Figure 6 show that the BLDC motor (without controller) gives $t_r = 2.01$ s, $M_p = 0.00\%$, $t_s = 1.43$ s and $e_{ss} = 0.00\%$. The BLDC motor speed control system with the IOPID controller in (12) yields $t_r = 0.51$ s, $M_p = 7.14\%$, $t_s = 1.58$ s and $e_{ss} = 0.00\%$. Also, the control system with the FOPID controller in (13) provides $t_r = 0.52$ s, $M_p = 0.08\%$, $t_s = 0.47$ s and $e_{ss} = 0.00\%$. The unit-step load-regulating responses in Figure 7 show that the BLDC motor (without controller) cannot regulate the load disturbance. The BLDC motor speed control system with the IOPID controller in (12) can completely regulate the load disturbance with the maximum percent overshoot from load regulation (M_{preg}) = 33.02%, the regulating time (t_{reg}) = 1.69 s and $e_{ss} = 0.00\%$. Then, the control system with the FOPID controller in (13) can completely regulate the load disturbance with $M_{preg} = 30.08\%$, $t_{reg} = 1.48$

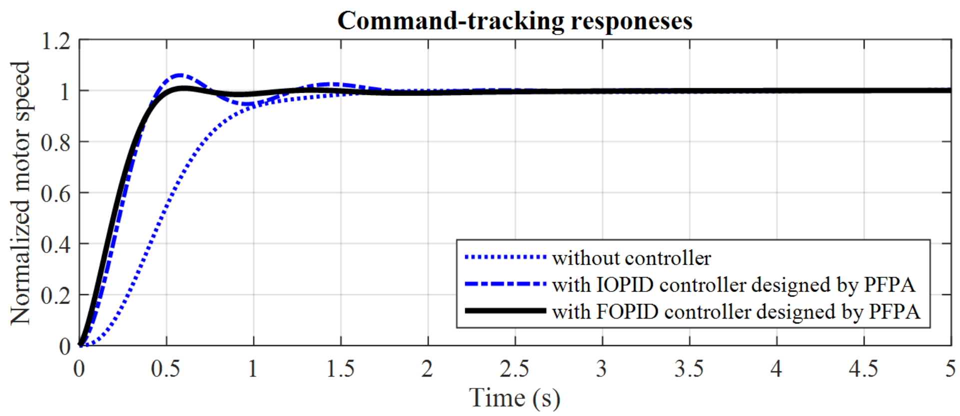

FIGURE 6. Unit-step command-tracking responses of BLDC motor speed control system

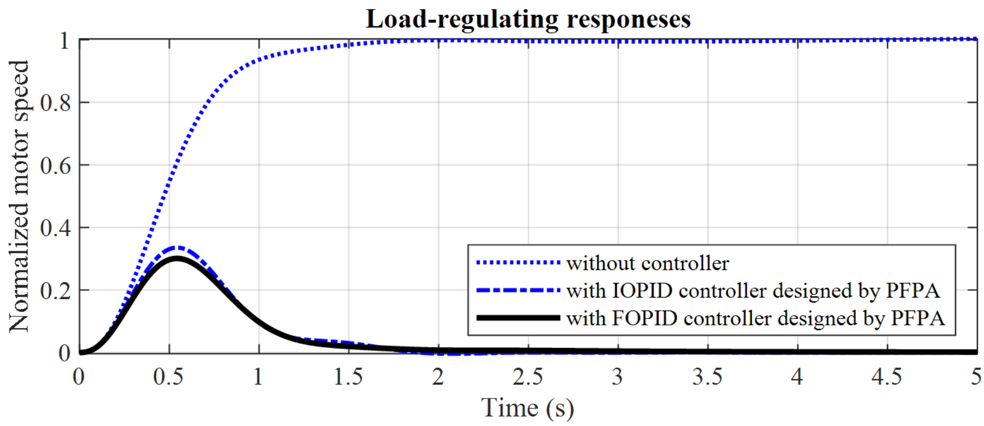

FIGURE 7. Unit-step load-regulating responses of BLDC motor speed control system

s and $e_{ss} = 0.00\%$. For all obtained results, the PFPA can successfully provide the optimal IOPID and FOPID controllers. In addition, the FOPID controller can yield very satisfactory results superior to the IOPID controller with faster and smoother responses. Thus, superiority and advantages of the FOPID controller to the conventional IOPID controller are confirmed.

5. **Conclusions.** The optimal design of the FOPID controller for BLDC motor speed control system using the PFPA algorithm has been proposed in this paper. The PFPA was developed to improve the search performance of the original FPA for running on single CPU platform. In this paper, the PFPA has been applied to the FOPID controller design optimization for the BLDC motor speed control system. As results, it was found that the PFPA can optimally provide both IOPID and FOPID controllers for the BLDC motor speed control system. By comparison, it was evidenced that the FOPID controller yields faster and smoother command-tracking and load-regulating responses of the BLDC motor speed control system than the IOPID controller. For future work, the fractional-order PIDA (FOPIDA) controller designed by the PFPA (or other promising metaheuristics) will be alternatively studied to extend the fractional-order control systems.

REFERENCES

[1] I. Podlubny, *Fractional-Order Systems and Fractional-Order Controllers*, UEF-03-94, Slovak Academy of Sciences, Kosice, 1994.

[2] I. Podlubny, Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers, *IEEE Transactions on Automatic Control*, vol.44, no.1, pp.208-214, 1999.

[3] Y. Q. Chen, I. Petráš and X. Dingyü, Fractional order control – A tutorial, *Proc. of the American Control Conference*, pp.1397-1411, 2009.

[4] P. Shah and A. Agashe, Review of fractional PID controller, *Mechatronics*, vol.38, pp.29-41, 2016.

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

[6] X. S. Yang, *Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley & Sons, 2010.

[7] 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.

[8] X. He, X. S. Yang, M. Karamanoglu and Y. Zhao, Global convergence analysis of the flower pollination algorithm: A discretetime Markov chain approach, *Proc. of the International Conference on Computational Science (ICCS2017)*, pp.1354-1363, 2017.

[9] H. Chiroma, N. L. M. Shuib, S. A. Muaz, A. I. Abubakar, L. B. Ila and J. Z. Maitama, A review of the applications of bio-inspired flower pollination algorithm, *Procedia Computer Science*, vol.62, pp.435-441, 2015.

[10] 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.

[11] M. A. B. Metwalli, I. Hezam, D. Yardım, I. A. Ozkan, I. Saritas and D. M. Aslam, A modified flower pollination algorithm for fractional programming problems, *International Journal of Intelligent Systems and Applications in Engineering*, vol.3, no.3, 2015.

[12] M. Abdel-Baset and I. M. Hezam, An effective hybrid flower pollination and genetic algorithm for constrained optimization problems, *Advanced Engineering Technology and Application*, vol.4, p.27, 2015.

[13] V. K. S. Patel and A. K. Pandey, Modeling and simulation of brushless DC motor using PWM control technique, *International Journal of Engineering Research and Applications (IJERA)*, vol.3, no.3, pp.612-620, 2013.

[14] E. H. E. Bayoumi and M. H. Soliman, PID/PI tuning for minimal overshoot of permanent-magnet brushless DC motor drive using particle swarm optimization, *Electromotion Scientific Journal*, vol.14, pp.198-208, 2017.

[15] H. Suryoatmojo, M. Ridwan, D. C. Riawan, E. Setijadi and R. Mardiyanto, Hybrid particle swarm optimization and recursive least square estimation based ANFIS multioutput for BLDC motor speed controller, *International Journal of Innovative Computing, Information and Control*, vol.15, no.3, pp.939-954, 2019.

[16] P. Khluabwannarat and D. Puangdownreong, Design of fractional-order PID controller for fractional-order system via flower pollination algorithm, *Proc. of the 2nd Artificial Intelligence and Cloud Computing Conference (AICCC2019)*, pp.191-196, 2019.
[17] P. Khluabwannarat, A. Nawikavatan and D. Puangdownreong, Application of parallel flower pollination algorithm to fractional-order model identification of BLDC motor, *Proc. of the 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2020)*, pp.131-134, 2020.
[18] J. Kluabwang, D. Puangdownreong and S. Sujitjorn, Multi-path adaptive tabu search for a vehicle control problem, *Journal of Applied Mathematics*, pp.1-20, 2012.
[19] C. Thammarat and D. Puangdownreong, CS-based optimal $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller design for induction motor speed control system, *International Journal on Electrical Engineering and Informatics*, vol.11, no.4, pp.638-661, 2019.
[20] T. Jitwang and D. Puangdownreong, Application of cuckoo search to robust PIDA controller design for liquid-level system, *International Journal of Innovative Computing, Information and Control*, vol.16, no.1, pp.189-205, 2020.
[21] A. Tepljakov, E. Petlenkov and J. Belikov, FOMCON: Fractional-order modeling and control toolbox for MATLAB, *Proc. of the 18th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES'11)*, pp.684-689, 2011.
[22] 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.