ISSN 2348-375X

Unique Journal of Engineering and Advanced Sciences

Available online: www.ujconline.net

Research Article

# DESIGNING A FRACTIONAL ORDER PID CONTROLLER FOR BIOREACTOR CONTROL

Hamed Banizaman[1], Seyede Mohadese Zarghami Amirsalari[2]

[1]Department of Electrical Engineering, Jahrom Branch, Islamic Azad University, Jahrom, Iran
[2]Departmentof Electrical Engineering, Fasa Branch, Islamic Azad University, Fasa, Iran

Received: 16-06-2014; Revised: 14-07-2014; Accepted: 13-08-2014

*Corresponding Author: **Hamed Banizaman**

## ABSTRACT

In the recent years, Evolutionary Algorithms (EAs) have been the topic of many researches through optimizations.Differential Evolution (DE) is one of the most popular optimization methods for real-valuedproblems and a large number of its variants have been proposed so far. However, bringingtogether different ideas that already led to successful DE versions is rare in the literature In this paper we propose a novel DE based MemeticAlgorithm(DEBMA) which hybridizes the differential evolution algorithm with a Local Search (LS)method to control the convergence rate of the population. In the proposed algorithm, some individuals are chosen for local refinement using a LS method, which leads to a smoother variation and a longer memory effect. The LS demonstrates a potential for interpreting evolution of the algorithm and to control its convergence. In this paper we describe an application of EAs to the design of fractional order proportional-integral-derivative (FOPID) controllers which involve a fractional order integral and a fractional order derivative. Fractional order controllers are more complex to design due to five design parameters. Here we use EAs to design an optimal FOPID controller to control a Bioreactor plant.To show the performance of both the FOPID and the proposed algorithm, a comparison between the designed controller using MA, simple DE and the conventional PID controller is presented.

**Keywords:** Fractional Calculus, Fractional Order PID Controller, Memetic Algorithm, Differential Evolution.

## INTRODUCTION

Fractional-order calculus has a 300-year-old mathematical concept behind[1,2]. Since Podlubny[3,14] proposed fractional-order controllers and then demonstrated their effectiveness in actuating the desired fractional-order system responses, they have attracted considerable attention and been a subject of extensive researches in the last few years[4,5]. Few recent works through using fractional-order controllers for fractional-order systems can be figured at Chen et al[6,7], Nakagawa and Siromachi[8]. Petras[9] proposed a method based on the pole distribution of the characteristic equation in the complex plane. Vinagre et al[10], came up with a frequency domain method based on the desired crossover frequency and phase margin to design the controller. Dorcak et al[11], proposed a design method in state space based on feedback pole placement. Chengbin and Hori[12] showed that by cascading a proper fractional unit to an integer order controller, the fractional-order controller can be synthesized. In fractional-order controller design, the transient response and the frequency response of the non-integer integral and its application to control systems were first introduced in[13]. The Proportional-Integral-Derivative (PID) controller is perhaps the most widely used controller in industries due to the simplicity of designing and implementation for process control applications. Fractional-order $PI^{\lambda}D^{\mu}$ extends the integer order of the traditional PID to the fractional-order both in time domain and in the frequency domain[10].

A simple yet powerful evolutionary algorithm (EA) which was first proposed by Storn and Price[15-17] is the Differential Evolution (DE) algorithm. DE is a population based evolutionary mechanism, which uses simple operators to create new candidate solutions. It has been proved that in real-world applications DE is a reasonably fast, accurate and robust optimizer for many optimization problems. Due to its simple concept and easy implementation, DE has attracted much attention and wide applications in different fields[18,19]. However, DE is easily trapped in local minima and does not guarantee the convergence to the global optimum[20]. Over the years, a class of hybrid EAs called memetic algorithms (MAs) have been widely applied for many complex optimization problems, such as scheduling problems[21], combinatorial optimization problems[22], and multi-objective optimization problems. MAs are population based metaheuristics composed of an evolutionary framework and a set of local search (LS) algorithms which are activated within the generation of cycle of the external framework[23,24]. Recent works show an obvious improvement in the performance of EAs by combining problem-dependent LS. In the framework of MAs, EA operators are used for global search and LS operators are used for local improvement, which can maintain an efficient balance between exploration and exploitation of an algorithm. In the present article, we design a $PI^{\lambda}D^{\mu}$ controller for bioreactor control using EAs expressed above and then compare their performance in designing the controller with the conventional PID controller. The rest of the article is organized as follows. Section 2 provides a brief overview of the fractional calculus and fractional-order control systems. Section 3 demonstrates how EAs can be applied to design the fractional-order PID. Section 4 describes the DE algorithm. In section 5 the proposed MA is described in details. Section 6 describes the Bioreactor control system and finally section 6 presents simulation results and comparisons.

**Fractional calculus and fractional-order systems: a brief overview**

To study the fractional-order controllers the starting point is studying a branch of mathematical analysis which is fractional calculus. Fractional calculus is a generation of ordinary differential calculus which considers the possibility of taking real number power of differential and integration operator. There are different ways to define fractional-order derivatives and integrals. The generalized differ-integral operator can be met as follows

$$ {}_aD_t^q f(t) = \frac{d^q f(t)}{[d(t-a)]^q} \quad (1)$$

Where the real order of differ-integral is represented by q, the parameter for which the differ-integral is taken is $t$ and $a$ is the lower limit which in almost all cases considered to be 0 and left out of the notation. Among all different definitions, the Grunwald-Letnikov definition is perhaps the best known[14]. The $q$ order fractional derivative of continuous function $f(t)$ is given by

$$ {}_aD_t^q f(t) = \lim_{h\to 0} \frac{1}{h^q} \sum_{k=0}^{[\frac{t-q}{h}]} (-1)^k \binom{q}{k} f(t-kh) \quad (2)$$

Where $[\frac{t-q}{h}]$ is a truncation, $\binom{q}{k}$ is binomial coefficients, $\binom{q}{0} = 1$, $\binom{q}{k} = \frac{q(q-1)\ldots(q-k+1)}{k!}$ and it can be replaced by Euler's Gamma function as follows

$$ \binom{q}{k} = \frac{\Gamma(q+1)}{k!\,\Gamma(q-k+1)} \quad (3)$$

The general calculus operator including integral order and derivative order is defined as

$$ {}_aD_t^q = \begin{cases} \frac{d^q}{dt^q} & ,Re\{q\} > 0 \\ 1 & ,Re\{q\} = 0. \\ \int_0^t (d\tau)^{-q} & ,Re\{q\} < 0 \end{cases} \quad (4)$$

In time domain, a fractional-order system is described by an n-term inhomogeneous fractional-order differential equation as follows

$$ \begin{aligned} &a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \cdots + a_0 D^{\alpha_0} y(t) \\ &= b_m D^{\beta_m} u(t) + b_{m-1} D^{\beta_{m-1}} u(t) + b_0 D^{\beta_0} u(t). \end{aligned} \quad (5)$$

The fractional-order transfer function of such a system can be obtained using the Laplace transform as follows

$$ G_n(s) = \frac{b_m s^{\beta_m} u(t) + b_{m-1} s^{\beta_{m-1}} u(t) + \cdots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} y(t) + a_{n-1} s^{\alpha_{n-1}} y(t) + \cdots + a_0 s^{\alpha_0}} \quad (6)$$

Where $\beta_k (k = 0,1,\ldots,m)$ and $\alpha_k (k = 0,1,\ldots,n)$ are arbitrary real numbers, $\beta_k > \beta_{k-1} > \cdots > \beta_0 > 0$, $\alpha_k > \alpha_{k-1} > \cdots > \alpha_0 > 0$, $b_k (k = 0,1,\ldots,m)$ and $a_k (k = 0,1,\ldots,n)$ are arbitrary constants.

**The EA-based design of fractional-order PID controllers**

**The FOPID controller**

The $PI^{\lambda}D^{\mu}$ controller is by no mean the most common form of a fractional-order PID controller which involves an integrator of order λ and a differentiator of order μ. The orders λ and μ are not necessarily integer, but any real numbers. The differential equation of fractional-order $PI^{\lambda}D^{\mu}$ controller is described by

$$ u(t) = K_p e(t) + K_i D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \quad (7)$$

The continuous form of fractional-order $PI^{\lambda}D^{\mu}$ can be obtained using the Laplace transform as follows

$$ G_c(s) = K_p + K_i s^{-\lambda} + K_d s^{\mu} \quad (8)$$

Obviously, by selecting $\lambda = 1$ and $\mu = 1$, a conventional PID controller can be obtained. The selections of $\lambda = 1$, $\mu = 0$ and $\lambda = 0$, $\mu = 1$ corresponds the classical $PI$ and $PD$ controllers respectively. So it is clear that all these classical types of $PID$ controllers are the special cases of the fractional-order $PI^{\lambda}D^{\mu}$ controller given by[8]. As shown in Riemann-Liouvilledefinition[25], fractional-order systems have an infinite dimension. So all the past input must be memorized in order to realize fractional-order controllers perfectly. Fractional-order $PI^{\lambda}D^{\mu}$ controllers generalize the integer-order $PID$ controllers and expand them from point to plane, Resulting in one of the most advantages of the $PI^{\lambda}D^{\mu}$ controllers which is more flexibility to controller design, due to adding two more degree of freedom to adjust the dynamical properties of the controller and so better control of dynamical systems, which are described by fractional-order mathematical models.

**Formulation of the objective function**

Based on the dominant roots method (root locus method) of synthesizing integral $PID$controllers[26], a design approach is presented. As in the classical root locus method, the damping ratio $\zeta$ and the un-damped natural frequency $\omega_0$ of the closed loop system are found out based on the user desired specifications of rise time $T_r$ and peak overshoot $M_p$. The dominant poles will be

$$ P_{1,2} = -\zeta\omega_0 \pm j\omega_0\sqrt{1-\zeta^2} = -x \pm jy \quad (9)$$

Let the closed loop transfer function be

$$ \frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (10)$$

Where $G_p(s)$ is the transfer function of the process to be controlled, $G_c(s) = \frac{U(s)}{E(s)}$ is that of the controller and $G(s) = G_c(s)G_p(s)$. The feedback gain is assumed as unity, i.e.

$H(s) = 1$. According to (10) the characteristics equation of the closed loop system will be

$$1 + G(s)H(s) = 0 \rightarrow 1 + G_c(s)G_p(s)\,.1 = 0 \quad (11)$$

Clearly, the poles of the system are the roots of this characteristics equation, so they will obviously satisfy the equation. Thus from (11) we have

$$1 + \left[K_p + K_i(s)^{-\lambda} + K_d(s)^{\mu}\right]G_p(s) = 0 \quad (12)$$

Where $s = -x + jy$. The equation has five unknown parameters to be designed. Let $R$ be the real part, $I$ the imaginary part and $\psi = tan^{-1}\left(\frac{I}{R}\right)$ the phase angle of the complex expression (12). Thus, the objective function will be as follows

$$J(K_p, K_i, \lambda, K_d, \mu) = |R|^2 + |I|^2 + |\psi|^2 \quad (13)$$

Our aim is to find out an appropriate optimal solution set $(K_p, K_i, \lambda, K_d, \mu)$ for which $J = 0$. To find the desired set of parameters that leads to optimal value of $J$, a five dimensional search space is selected. So each parameter vector in EAs has five components.

**The Differential Evolution Algorithm**

The differential evolution algorithm involves two phases: initialization and evolution. In the initialization phase, if nothing is known about the problem the DE population is generated randomly. In the evolution phase, individuals from the population go through mutation, crossover, and selection process repeatedly until the termination criterion is met[16]. Without loss of generality, a minimization problem $f(x)$ is discussed here.

$$\min f(x), \qquad x = [x_1, x_2, \ldots, x_3] \quad (14)$$

where $f(x)$ is the objective function, and $x$ is the decision vector consisting of $n$ variables.

In the standard DE model, each individual represents a candidate solution to $f(x)$ within the search space. at each generation,the objective function is evaluated for each individual. The obtained value, or fitness, is used to assess the quality of the individuals and the best member is noted in order to keep track of the progress that is made during the optimization process. The basic procedure of DE is summarized as follows.

1. Randomly initialize the population of individual for DE.
2. Evaluate the objective values of all individuals, and determine the best individual which has the best objective value.
3. Perform mutation operation for each individual in order to obtain each individual's corresponding mutant vector.
4. Perform crossover operation between each individual and its corresponding mutant vector in order to obtain each individual's trial vector.
5. Evaluate the objective values of the trial vectors.
6. Perform selection operation between each individual and its corresponding trial vector so as to generate the new individual for the next generation.
7. Determine the best individual of the current new population with the best objective value. If the objective value of the current best individual is better than that of best individual, then update best individual and its objective value.
8. If a stopping criterion is met, then output best individual and its objective value; otherwise go back to step 3.

9. **The Proposed Memetic Algorithm**

Memetic DE algorithms combine global search DE method with some local search procedures. To propose a novel Memetic DE algorithm, a number of important questions must be addressed: which local search methods to choose, which individuals within DE population should follow the local search, how frequently local search should be commenced and how many fitness calls may it use. Various local search methods are used by Memetic DE algorithms, from a very simple random walk [28] or chaotic search [29] to heuristics like Adaptive hill descending with simplex crossover [30], Simulated Annealing, Hooke Jeeves Algorithm [31], Rosenbrock method or Nelder–Mead simplex [32]. In DEBMA an intermediate approach is proposed, which benefit from the idea of Local mutation model – only individuals that are better than all other individuals within p% neighborhood have a chance to follow the local search (here p% is set to 20%). This way the number of individuals that have a chance to commence local search may vary from 0 to almost 10% of the population, depending on relation in fitness among neighboring individuals. The basic procedure of DE-based MA is summarized as follows..

1. Randomly initialize the population of individuals.
2. Evaluate the objective values of all individuals, and determine the best individual which has the best objective value.
3. Select individuals from current population for local refinement, and apply local search upon each of selected individuals.
4. If the objective value of the selected individuals is better than that of its best, then update its position and its objective value.
5. While (stopping criterion not met)
6. Update the position for each individual and determine the objective value for each individual.
7. If the objective value of the $i$thindividual is better than that of its best, then update its position and its objective value.
8. Select individuals from current population for local refinement, and apply local search upon each of selected individuals.
9. If the objective value of the best individual is better than that of best of all individuals, then update the best individual and its objective value.
10. If a stopping criterion is met, then output best individual and its objective value; otherwise go back to step 6.

**Bioreactor Control System**

Biochemical reactors are cylindrical culture vessels used for the fermentation process in which anaerobic breakdown of complex organic materials by the action of anaerobic micro-organism or free enzymes takes place. Materials such as carbon, nitrogen, oxygen, which are called substrate, and other nutrients are brought with the cell into the culture vessel (bioreactor) and converted within the cell via hundreds of reactions to the various constituents of the cell as well as to biochemical product. Bioreactors provide a controlled environment that is necessary to bring the better growth of microbes, and also maintain constant temperature according to the need of microbes.

The transfer function relating the dilution-rate to the biomass concentration is[28]:

$$G(s) = \frac{-1.39s^2 - 1.99s - 0.2577}{s^3 + 1.408s^2 + 0.656s + 0.1013} \qquad (15)$$

**Experimental Results**

To check the performance of the proposed algorithm in designing fractional $PI^{\lambda}D^{\mu}$ controllers, we applied it on the Bioreactor system. Table I. summarizes the test problem with the corresponding user specifications including the desired maximum overshoot and rise time to design the controllers and the parameter settings used for the EAs. In order to compare both, the performance of $PI^{\lambda}D^{\mu}$ on controlling and the performance of the proposed MA in tuning the controller, we designed two $PI^{\lambda}D^{\mu}$ controllers using DE and the proposed MA and then compared the results with the conventional PIDcontroller.

The simulation results for the Bioreactor control using the conventional PID controller (the green line), the FOPID controller tuned by DE (the red line) and the FOPID controller tuned by the proposed MA (the light blue line) are shown in "Fig.1". As is obvious, FOPIDs enhance the performance and lead to better responses than conventional PID controllers.The conventional PID and the DE-FOPID controller bear with undershoot, but not the proposed MA-FOPID. The performance of MA-FOPID in achieving the desired maximum overshoot is shown in Table II. Also, the controller parameters achieved by EAs are shown in Table II.

## CONCOLUSION

In this article, performance comparison of conventional PID controller and that of fractional order PID controller has been presented. A novel memetic algorithm was proposed and its superb performance on designing a fractional order PID controller in the face of the differential evolution algorithm was investigated and observed. Comparing the responses obtained by the conventional PID with fractional PID controllers, the better performance of system with the fractional order PID controllers was observed.

## REFERENCES


1. Oldham, KB., Spanier J.: The Fractional Calculus: Integrations and Differentiations of Arbitrary Order. Academic Press, New York, 1974.
2. Lubich, CH.: Discretized fractional calculus. SIAM J. Math. Anal. 1986; 17(3): 704–719.
3. Podlubny I.: Fractional-order systems and PID controllers. IEEE Transactions on Automatic Control, 1999; 44(1): 208–214.
4. Oustaloup A.: Fractional order sinusoidal oscillators: Optimization and their use in highly linear FM modulation. IEEE Transactions on Circuits and Systems, 2009; 28(10): 1007–1009.
5. Chengbin M., Hori, Y.: The application of fractional order PID controller for robust twoinertia speed control. In: Proceedings of the 4th International Power Electronics and Motion Control Conference,
6. Chen YQ., Xue D, Dou H.: Fractional calculus and biomimetic control. In: Proceedings of the First IEEE International Conference on Robotics and Biomimetics (RoBio 2004). IEEE, Shengyang, 2004.
7. Chen YQ., Ahn H., Xue D.: Robust controllability of interval fractional order linear time invariant systems. In: Proceedings of the ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Long Beach, CA, 2005; 24-28.
8. Nakagawa, M., Sorimachi, K.: Basic characteristics of a fractance device. IEICE Transactions on Fundamentals E75-A (12), 1992; 1814–1819.
9. Petras, I.: The fractional order controllers: methods for their synthesis and application. Journal of electrical Engineering 50(9-10), 284–288 (1999)
10. Vinagre BM, Podlubny I., Dorcak L. and Feliu V., "On Fractional PID Controllers: A Frequency Domain Approach", Proceedings of IFAC Workshop on Digital Control - PID'00, Terrassa, Spain, 2000
11. Dorcak, L., Petras, I., Kostial, I. and Terpak, J., "State-space controller design for the fractional-order regulated system", Proceedings of the ICCC'2001, Krynica, pp. 15 - 20. May, 2001.
12. Chengbin, M., Hori, Y.: The application of fractional order PID controller for robust twoinertial speed control. In: Proceedings of the 4th International Power Electronics and Motion Control Conference (2004)
13. Manable S, "The non- integer integral and its application to control systems," Electro technical Journal of Japan, vol. 6, no. 3/4, pp.: 83- 87,1961.
14. Podlubny I, Fractional Differential Equations. London. UK, Academic Press, 1999.
15. Price K, Storn R, Lampinen J, Differential Evolution: A Practical Approach to Global Optimization, first ed., Springer-Verlag, New York, 2005.
16. Storn R, K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (4) (1997) 341–359.
17. R. Storn, K. Price, Minimizing the real functions of the ICEC'96 contest by differential evolution, in: Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 1996, pp. 842–844.
18. Das S, Sil S, Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm, Information Sciences 180 (8) (2010) 1237–1256.
19. Wang Y, Li B, Weise T, Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems, Information Sciences, 2010; 180 (12): 2405–2420.
20. Das S, Suganthan PN, Differential evolution – a survey of the state-of-the-art, IEEE Transactions on Evolutionary Computation, 2011; 15 (1): 4–31.

21. Burke EK, Smith AJ, Hybrid evolutionary techniques for the maintenance scheduling problem. IEEE T Power Syst, 2000; 15:122–128 38.
22. Cheng RW, Gen M, Parallel machine scheduling problems using memetic algorithms. Comput Ind Eng., 1997; 33:761–764.
23. Merz P, Memetic algorithms for combinatorial optimization problems: fitness landscapes and effective search strategies. PhD dissertation, University of Siegen, Germany, 2000.
24. Dongli Jia ,Guoxin Zheng, Muhammad Khurram Khan , An effective memetic differential evolution algorithm based on chaotic local search , Information Sciences, 2011; 3175-3187
25. Merz P, Freisleben B, Fitness landscapes, memetic algorithms and greedy operators for graph bipartitioning. Evol Comput, 2000; 8: 61–91.
26. Podlubny I, Fractional-order Systems and Fractional-order Controllers, The Academy of Sciences Institute of Experimental Physics, UEF-03-94, Kosice, Slovak Republic, 1994.
27. Karl J. Astrom, Tore Hagglund, PID controllers, theory, design and tuning, 1995.
28. Karad S. and Chatterji S.: Performance Analysis of Fractional Order PID Controller with the Conventional PID Controller for Bioreactor Control, International Journal of Scientific & Engineering Research, Vol 13, Iss 6, 2012.
29. Liao TW, Two hybrid differential evolution algorithms for engineering design optimization, Applied Soft Computing, 2010; 10 (4): 1188–1199.
30. Jia DL, Zheng GX, Khan MK, An effective memetic differential evolution algorithm based on chaotic local search, Information Sciences, 2011; 181 (15): 3175–3187.
31. Noman N, Iba H, Accelerating differential evolution using an adaptive local search, IEEE Transactions on Evolutionary Computation, 2008; 12 (1): 107–125.
32. Tirronen V, Neri F, Karkkainen T, Majava K, Rossi T, An enhanced memetic differential evolution in filter design for defect detection in paper production, Evolutionary Computation, 2008; 16 (4): 529–555.
33. Caponio A, Neri F, Tirronen V, Super-fit control adaptation in memetic differential evolution frameworks, Soft Computing, 2009; 13(8–9): 811–831.


**Table I: Description of the Test Problems, Desired Specifications and Parameter Settings For EAS**

| Process plant transfer function | Maximum overshoot (%) | Maximum Rise time (sec) | Parameter Settings for EAs | | |
|---|---|---|---|---|---|
| | | | No. of Variables | Initial Population | No. of Iterations |
| $\frac{-1.39s^2 - 1.99s - 0.2577}{s^3 + 1.408s^2 + 0.656s + 0.1013}$ | 10 | 0.3 | 5 | 50 | 50 |

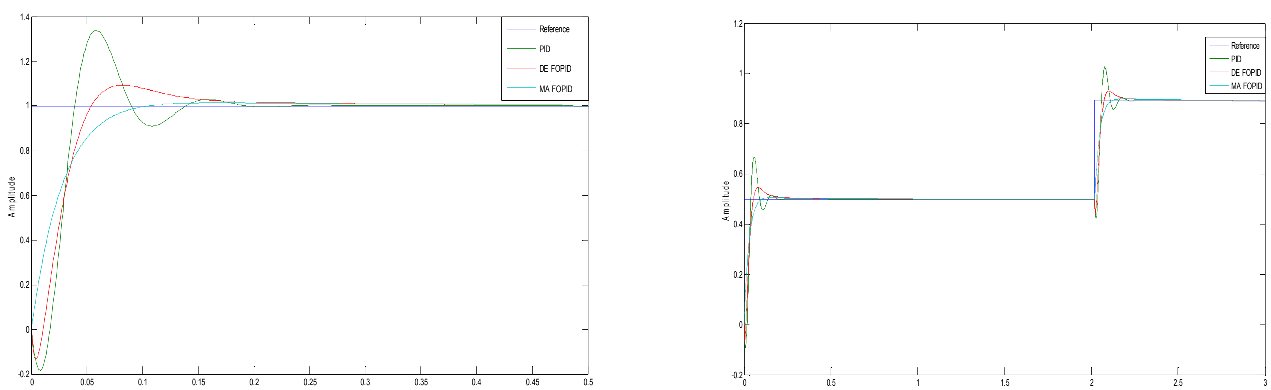


**Figure 1.Comparative simulation results of Bioreactor control (green line: conventional PID controller, red line: FOPID controller by DE, and light blue line: FOPID controller by proposed MA)**

**Table II: Unit Step Response of Comparative Results of Designed Controllers Applied on the Plant**

| Controller Used | Parameters | | | | | Unit Step Response | | |
|---|---|---|---|---|---|---|---|---|
| | *Kp* | *Ki* | *λ* | *Kd* | *μ* | *Over Shoot (%)* | *Rise Time (sec)* | *Steady State Error* |
| Conventional PID | -32.6219 | -5.1297 | - | 0.6811 | - | 33.712 | 0.03144 | 5.3346e-03 |
| FOPID Tuned by DE | -25.5371 | -5.0373 | 0.4337 | 1.3238 | 0.6616 | 9.281 | 0.05523 | 3.4441e-03 |
| FOPID Tuned by MA | -28 | -10 | 0.7335 | 1 | 0.1609 | 1.338 | 0.05681 | 2.5271e-03 |

**Source of support: Nil, Conflict of interest: None Declared**