Control and Optimization in Applied Mathematics (COAM)
DOI. 10.30473/COAM.2021.56679.1156
Vol. 5, No. 2, Summer - Autumn 2020 (39-64), ©2016 Payame Noor University, Iran

Research Article

Adjusting the Coefficients of the $PI^{\alpha}D^{\beta}$ Controllers Using Iterative Feedback Tuning (IFT) Algorithm

Mohammad Ali Vali[1*], Mahdi Mashayekhi Esfichar[2], Shahryar Farahmand Rad[3]

[1]Department of Applied Mathematics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran.
[2,3]Department of Mathematics, Payame Noor University (PNU), P.O. BOX 19395-4697, Tehran, Iran.

Received: December 13, 2020; **Accepted:** August 02, 2021.

Abstract. Iterative feedback tuning (IFT) is an algorithm for adjusting the coefficients of the integer-order type proportional-integral-derivative (PID) controllers without needing a system model. The IFT algorithm is performed iteratively with the aim of optimizing the control coefficients at each stage via an objective function. In this research, for the first time, the IFT algorithm is used to adjust all the coefficients of the fractional order PID controllers, i.e., $PI^{\alpha}D^{\beta}$ controllers to have optimal performance. For this purpose, fractional order calculations and the integer-order version of the IFT algorithm are firstly presented, and the novel IFT algorithm is then used to adjust coefficients of the $PI^{\alpha}D^{\beta}$ controller. Finally, the performance of the proposed method is illustrated and verified through some examples.

Keywords. Fractional order PID, Iterative feedback tuning, PID controller, $PI^{\alpha}D^{\beta}$ controller, Fractional order calculus, Fractional order systems.

MSC. 93-XX;93Cxx.

* Corresponding author
mvali@uk.ac.ir, mahdimashayekhi121@yahoo.com, sh_ fmand@pnu.ac.ir
http://mathco.journals.pnu.ac.ir

1 Introduction

Fractional order differential calculus and integral is an essential branch of mathematics that deals with concepts and definitions of fractional order (FO) derivatives and integrals of any desired order $\alpha \in \mathbb{R}^+$. This branch of mathematics offers a wide variety of applications, including electrochemical statistics, rock typology, etc. (see [15]-[38]). Also, it is used in various control systems, e.g., route planning [5], water level control in tanks [28], DC rpm control [42] and fractional order systems with delay [29]. Up until now, many ways of adjusting the coefficients of the conventional PID controllers have been introduced. Also, several methods have been proposed for tuning fractional order controllers. In 2008, Nur Deniz et al. [30] used Fourier series and frequency response of a system with integer order n and fractional order PID (FOPID) controller and analytically found optimal controller gains for some of the definite orders n based on a cost function. Also, in 2019, Gabriel and Grandi [14] first considered the desired frequency response and then tried to minimize the difference between real plant frequency response and that of the desired by tuning the FOPID controller gains. Moreover, Bode's optimal loop shaping technique was employed in [24] to design the controller with desired performance. Some research, like Acharya and Mishra [1], used heuristic methods to tune the parameters of the controller. In that paper, multi-agent-based symbiotic organisms search (MASOS) was proposed to optimize the FOPID controller. Recently, Yang et al. [43] suggested an interactive teaching-learning optimizer (ITLO) to design a robust FOPID controller to control a supercapacitor energy storage system for various operating conditions. Evolutionary optimization (EO) is another meta-heuristic algorithm used by authors in [22] to tune FOPID parameters. In [11], Erol used a combined time/frequency domain analytical method to tune the FOPID parameters. Generally, these methods can be divided into two categories of model-based, and model-free [12]. Model-based methods tune the coefficients based on the system model and the dynamic equations from which they are derived. Optimizing the cost function to perform the necessary analysis and finding the optimal state usually entails a model through which these actions are carried out. However, it is often difficult to find an accurate model of the system, and this requires specialized expertise and knowledge of the plant. Hence, in such cases, the mentioned issue would be unraveled by the methods that can design the controller based on a criterion without a system model. The IFT algorithm is one of the methods employed for such cases. This algorithm has two basic features: first, it applies to systems whose processes are iterative, e.g., industrial and robotic systems, and second, it does not require a system model to tune the controller parameters. Therefore, applying IFT eliminates the need for system identification [23]. In fact, IFT is a gradient-based optimization method, but instead of using the system model gradient directly, it calculates the gradient from the results of a series of experiments performed on the model. The IFT algorithm was presented for the first time in 1994 by Hjalmarsson et al. for adjusting the control loops iteratively [21]. This algorithm has been used in various control systems, including speed and servo drive positioning control [21], inverted pendulum crane model control [32], Hard disk drive (HDD) head positioning servomechanism control [4], Torsional control system [7], control of wastewater recycling systems [26] and robot arm control [41]. In all of the aforementioned cases, IFT is used to adjust the IO controllers. In the FO mode, the IFT algorithm is employed to adjust the FO system's controller coefficients in the $PI^{\alpha}D$ mode [33]. Each of the previously reviewed methods has its advantages and drawbacks.

The purpose of this study is to provide a model that could apply the IFT algorithm for tuning the FO controller coefficients in the $PI^{\alpha}D^{\beta}$ mode. This novel technique has the benefit of simplicity since it requires only a few parameters to adjust the convergence speed of the tuning algorithm. Moreover, it has a strong mathematical basis, unlike many random-based methods. Also, it does not need model identification and has fast convergence. On the other hand, since the technique needs the signals from the last run of the system, it requires a storage device with enough memory to acquire/capture the signals. A proper laptop with appropriate IO cards can be used to achieve this goal, and since tuning of the controller is only needed once in a while (for example, every 6-months), this capture setup is not required to permanently connect the system. Therefore, it does not impose a huge extra cost on the system owner. To present the technique, the steps of the IFT algorithm are first briefly presented to summarize the adjustment of IO coefficients; then, this algorithm is generalized for the first time to adjust the coefficients of FO systems in the $PI^{\alpha}D^{\beta}$ mode.

This paper is organized in five sections. In the next section, we discuss the principles of fractional calculus and FO derivatives and recall some basic theorems and formulas. Section 3 is devoted to the IFT algorithm and the way it works to improve the system performance. In Section 4, we present the IFT method to tune the $PI^{\alpha}D^{\beta}$ controller. Ultimately, conclusions are drawn in Section 5.

2 Preliminaries

Fractional order calculus dates back to more than 300 years ago and most of the theories related to it were developed before the 20^{th} century. In 1695, upon Hopital's response to Leibniz's letter about the question "whether the definition of an IO derivative can be generalized to FO derivatives", fractional order calculus emerged as a new subject in mathematics. In fact, at the time, Hopital asked Leibniz about a derivative with a non-IO of 0.5, and Leibniz responded: "This leads to contradictions, but it will yield useful results in the future". This question led many mathematicians to pursue this topic over the following years. Focusing on this subject, renowned mathematicians such as Liouville, Reimann, and Weyl have conducted extensive studies on the fractional order calculus theory. Note that there are various definitions of FO derivation and integration of FO in the expansion of fractional order calculus. Some of these definitions have been developed directly from fractional order calculus, including Cauchy's integral definition, Grünwald-Letnikov's definition, Riemann-Liouville's definition, and Caputo's definition. For a more comprehensive study on the history of fractional calculus, see reference [35].

2.1 Riemann-Liouville's fractional order derivative

Consider an infinite sequence of n-variable integrals and derivatives of a function in the following form:

$$\cdots, \int_a^t \int_a^{t_2} f(t_1)dt_1dt_2, \int_a^t f(t_1)dt_1, f(t), \frac{df(t)}{dt}, \frac{d^2f(t_1)}{dt_1}, \cdots$$

where a discrete sequence of operators is on the function f. Fractional derivative of a desired order α is defined as an interpolation operator for this sequence and is denoted by ${}_aD_t^\alpha$. If α is negative, then the operator is called fractional integral. The idea of the Riemann-Liouville's fractional integral definition was drived from the same Riemann-Liouville's n-variable integral. In other words, for $n \in \mathbb{N}$, the following formula is generated which is known as Cauchy's formula:

$$\int_a^t \int_a^{t_n} \cdots \int_a^{t_3} \int_a^{t_2} f(t_1)dt_1dt_2 \cdots dt_{n-1}dt_n = \frac{1}{(n-1)!} \int_a^t (t-\tau)^{n-1} f(\tau)d\tau.$$

The above relation can be expanded for defining the FO integrals as follows:

$${}_aD_t^{-\alpha} = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau)d\tau$$

for $t > \alpha$, where α is a positive real number, and $\Gamma(\alpha)$ is Gamma function which is defined over the $\mathbb{R}\backslash\mathbb{Z}^-$ domain as follows:

$$\Gamma(\alpha) = \int_0^\infty (e)^{-t} t^{\alpha-1} dt.$$

This function is the generalization of factorial function for natural numbers and simply concludes: $\Gamma(\alpha+1) = \alpha\Gamma(\alpha)$.

Remark 1. Some authors denote the fractional integral by ${}_aI_t^\alpha$.

Riemann-Liouville fractional derivative from the desired order α is defined as follows:

$${}_a^{RL}D_t^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_a^t (t-\tau)^{n-\alpha-1} f(\tau)d\tau,$$

which can also be written as

$${}_a^{RL}D_t^\alpha f(t) = \frac{d^n}{dt^n}({}_aD_t^{-(n-\alpha)} f(t)).$$

Further details are available in [31].

2.2 Caputo's fractional order derivative

Since the solution of the Riemann-Liouville fractional derivative equations required initial fractional derivative conditions, it could not be sufficiently useful for solving fractional equations. Thus, another definition was needed in order to better model the phenomena and be consistent with the initial conditions of the problems. Hence, Caputo [9] defined a new fractional derivative as follows:

$$ {}_{a}^{C}D_{t}^{\alpha} f(t) = \frac{1}{\Gamma(n-\alpha)} \int_{a}^{t} (t-\tau)^{n-\alpha-1} f^{(n)}(\tau) d\tau, \qquad n-1 \leq \alpha < n. $$

As seen in Caputo's FO derivative, unlike that of Liouville-Riemann, the n order derivative of the function was first obtained followed by obtaining the integral of the FO. The relationship between the fractional derivatives of Riemann-Liouville and Caputo is as follows:

$$ {}_{\alpha}^{RL}D_{t}^{\alpha} f(t) = {}_{\alpha}^{C}D_{t}^{\alpha} f(t) + \sum_{k=0}^{n-1} \frac{f^{(k)}(\alpha)(t-\alpha)^{(k-\alpha)}}{\Gamma(k+1-\alpha)}, \qquad n-1 \leq \alpha < n. \tag{1} $$

It can be concluded from relation (1) that if $f^{(k)}(\alpha) = 0$ for $k = 0, 1, ..., n-1$, then these two definitions are equal.

3 Iterative Feedback Tuning (IFT)

Using reference [33], we wrote this section. A view of a control loop can be seen in Figure 1, where G_P is the plant transfer function, G_C is the controller transfer function, Y_d is the desirable or reference output, Y is output of the process, E is error or difference between the desired output and process output, U is the controlling signal and also, ρ is the controller parameters' vector [21, 18].

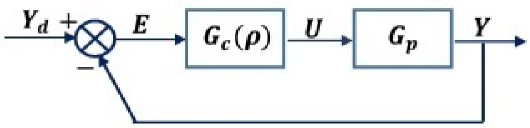

Figure 1: Typical control system.

For the PID controllers, this vector has three coefficients K_P, K_I, and K_D; and $\rho = [K_P \quad K_I \quad K_D]^T$. The aim of the IFT process is to minimize an objective (cost) function such as

$$ J(\rho_i) = \frac{1}{2N} \sum_{k=1}^{N} (y_k(\rho_i) - y_d^k)^2 = \frac{1}{2n} \sum_{k=1}^{N} e_k^2(\rho_i) \quad \text{for} \quad i = 1, 2, ..., \tag{2} $$

where N is the total number of samples, y_k is k-th sample of the system's output, y_d^k is k-th sample of the desired output and ρ_i is the set of the controller parameters in the

i-th iteration. Also, $e_k(\rho_i)$ is k-th sample of the output error. As seen in relation (2), the cost function is depends on ρ_i because if controller parameters change, the output and consequently, the cost function change. Thus, to minimize the cost function, one has to move in the opposite direction of the gradient of the function. Therefore, the descending gradient method is used as follows:

$$\rho_{i+1} = \rho_i - \gamma_i \frac{\partial J(\rho_i)}{\partial \rho} \qquad i = 1, 2, \cdots, \tag{3}$$

where γ_i is a positive real number affecting the convergence rate. The relation (3) is the main relation for updating control parameters in the IFT method. As can be seen in relation (3), in order to calculate ρ_{i+1}, it is necessary to calculate the gradient of the cost function, i.e., $\frac{\partial J(\rho_i)}{\partial \rho}$. Since the cost function is dependent on the output of the system, which in turn depends on the system conversion function, it seems that for calculating this gradient analytically, the system's model is needed. However, IFT provides a solution that does not require a model. To illustrate this problem, the gradient is first calculated as follows:

$$\frac{\partial J(\rho_i)}{\partial \rho} = \frac{1}{N}\left(\sum_{k=1}^{N}(y_k(\rho_i) - y_d^k)\left(\frac{\partial y_k(\rho_i)}{\partial \rho}\right)\right), \qquad i = 1, 2, \cdots. \tag{4}$$

In this relation, y_d^k is determined because the output is the intended parameter. Also, $y_k(\rho_i)$ is a known term since it is obtained from the output sensors. The only unknown term is $\frac{\partial y_k(\rho_i)}{\partial \rho}$ which, at first glance, requires the calculation of a closed form of the system model response. It can be shown that

$$\frac{\partial Y(\rho_i)}{\partial \rho} = \frac{\partial G_C(\rho_i)/\partial \rho}{G_C(\rho_i)} \frac{G_C(\rho_i)G_P}{1 + G_C(\rho_i)G_P} E \qquad \text{(see [33])}.$$

Similarly, the last fraction at the right side is the transfer function of the closed loop system, and E is its input. Replace the output call with Y, and then relation (5) is derived as

$$\frac{\partial Y(\rho_i)}{\partial \rho} = \frac{\partial G_C(\rho_i)/\partial \rho}{G_C(\rho_i)} Y. \tag{5}$$

Thus, to calculate the output gradient in this method, Y_D reference input is first given to the system and E output error is measured. Next, in another step, the previous step's error E is given to the system as the input of the current step, and its output Y is measured. Finally, the output of this step D is given to the system (5) as an input, so the output of this step turns out the same $\frac{\partial Y(\rho_i)}{\partial \rho}$. Furthermore, it is observed that obtaining $\frac{\partial Y(\rho_i)}{\partial \rho}$ does not depend on the knowledge of the (G_P) system model. Now, through relations (3) and (4), the controller coefficients are updated. As a result, applying the IFT algorithm eliminates dependency on the model by performing a series of tests in a smart manner. The algorithm flowchart is shown in Figure 2. It should be noted that the parameters are not tuned in real-time, unlike some gradient-based methods, but they are updated after two runs of the system are completed in each iteration of tuning. As mentioned earlier, this requires enough memory to capture the output error and some calculation period between subsequent iterations. However, this is not normally a problem with modern PCs. As an example, the calculation time between iterations for the case studies in this research were in the order of 5-10 seconds.

As a further remark, in (5), the relation is in the frequency (Laplace) domain. But, since the output signal is captured and available in time domain, e.g.,

$$y(\rho_i) = [y_1(\rho_i), y_2(\rho_i), \cdots, y_N(\rho_i)],$$

the Laplace inverse of relation (5) is in practice used, to calculate $\frac{\partial Y(\rho_i)}{\partial \rho}$ in time domain, i.e., $\frac{\partial y_k(\rho_i)}{\partial \rho}$. This is easily done via discrete approximation of the equation or by commercially available software like Matlab.

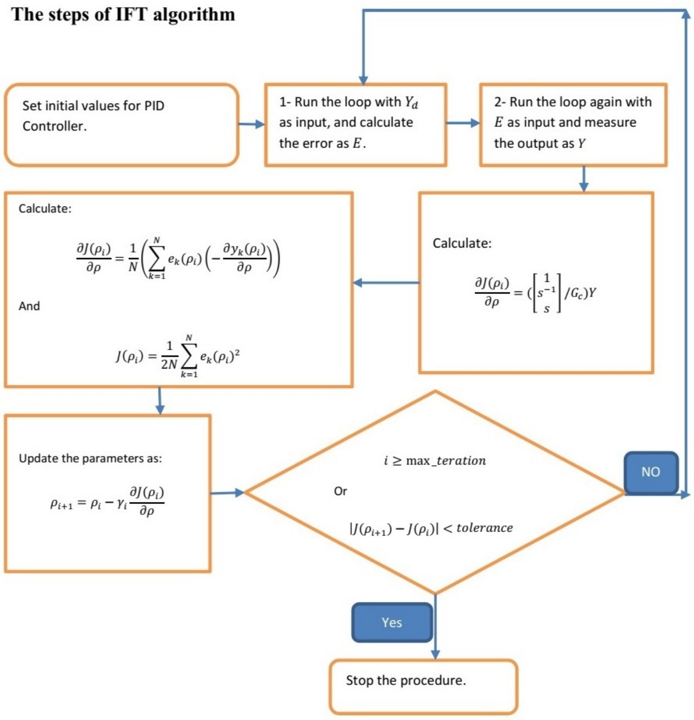

Figure 2: Iterative Feedback Tuning (IFT) Flowchart.

3.1 Algorithm convergence conditions

As illustrated in relation (3), the parameter γ_i affects the speed of convergence and indeed, plays the role of the updating stage. Usually, a big step leads to quick convergence to the optimal point, but there is the risk of divergence and oscillation. Instead, selecting a small γ_i causes the algorithm to converge slowly but with a greater reliance. Also, γ_i can be a stable or variable number in each iteration. In any case, what matters is that it leads to the convergence of the algorithm. The condition of converging the algorithm based on selection γ_i is expressed in the following theorem.

Theorem 1. Consider the updating rule (3). Suppose that γ_i satisfies the following conditions:

$$\sum_{i=1}^{\infty} \gamma_i = \infty \quad , \quad \sum_{i=1}^{\infty} {\gamma_i}^2 < \infty. \tag{6}$$

Then the local algorithm will converge ([18]-[19]).

This theorem has a complicated proof that can be seen in reference [34]. A suitable selection for γ_i in relation (6) is $\frac{\alpha\beta}{i}$. This selection fulfills the condition of convergence, because

$$\sum_{i=1}^{\infty} \frac{\alpha\beta}{i} = \infty \quad , \quad \sum_{i=1}^{\infty} (\frac{\alpha\beta}{i})^2 = \frac{(\alpha\beta)^2\pi^2}{6} < \infty.$$

The second condition comes from the fact $\sum_{i=1}^{\infty} (\frac{1}{i})^2 = \frac{\pi^2}{6}$. As the number of iterations increases, γ_i will gradually shrink. Therefore, initially, the step is larger and the algorithm converges faster, but as it gets closer to the optimal point it gets smaller, and the risk of divergence decreases.

4 Statement of the Problem

In this section, the tuning of the fractional order IFT method of the $PI^{\alpha}D^{\beta}$ controller (FOIFT) is presented.

4.1 Using the IFT algorithm to optimize the parameters of a $PI^{\alpha}D^{\beta}$ controller

In general, equation (7) represents the structure of the FO controllers of $PI^{\alpha}D^{\beta}$ [27]-[20]:

$$G_C(s) = k_P + k_I s^{-\alpha} + k_D s^{\beta}, \tag{7}$$

where α indicates the degree of integrator and β indicates the differentiator order so that if $\alpha = 1$ and $\beta = 1$, then its structure becomes the same as that of the integer-order (IO) controllers. In the tuning processes of the control systems, if the plant has FO or IO dynamics, its controller can be of FO or IO type as well. In case the system has IO dynamics, selecting the FO controller would lead to a better response. This subject is due to the greater flexibility created by free parameters (of derivative and integral order) of the FO controller as compared to the IO controller [42]. On the other hand, it was shown by Guermah [15] and Zhao et al. [46] that the choice of FO controllers for FO systems could result in a more accurate and robust performance. Therefore, it can be argued that regardless of the type of the dynamic system, i.e., IO or FO, the FO controller would produce a better response.

Several methods have been proposed to tune the controller parameters of $PI^{\alpha}D^{\beta}$ [40]-[2] as most of them are based on a model of the plant. But, in some cases, it is difficult to find such a model or the model may not be accurate enough to design a suitable controller. Under such conditions, using model-independent methods such as IFT would be beneficial. Based on the relations [22], [12] and [32] in order to use the IFT algorithm we need the ratio of controller derivation to its parameters. Therefore,

$$\frac{\partial G_C(\rho_i)}{\partial \rho} = \begin{cases} \frac{\partial G_C}{\partial k_P} = 1, \\ \frac{\partial G_C}{\partial k_I} = s^{-\alpha}, \\ \frac{\partial G_C}{\partial k_D} = s^{\beta}, \\ \frac{\partial G_C}{\partial \alpha} = -s^{-\alpha} \ln s, \\ \frac{\partial G_C}{\partial \beta} = s^{\beta} \ln s. \end{cases}$$

As mentioned before, the IFT algorithm has used for fractional order controller in $PI^{\alpha}D$ in [33]. But due to some restrictions such as fractional order calculations in the frequency domain and also the lack of Laplace transform for the fifth sentence, it has not been extended to $PI^{\alpha}D^{\beta}$. This paper presents a solution that resolves this problem with a suitable approximation. In [13], a good approximation is obtained by s^{α} as follows:

$$s^{\alpha} \approx \frac{\alpha_0 s^2 + \alpha_1 s + \alpha_2}{\alpha_2 s^2 + \alpha_1 s + \alpha_0} \longrightarrow s^{-\alpha} \approx \frac{\alpha_2 s^2 + \alpha_1 s + \alpha_0}{\alpha_0 s^2 + \alpha_1 s + \alpha_2}, \tag{8}$$

where $0 < \alpha < 1$ so that

$$\begin{cases} \alpha_0 = \alpha^2 + 3\alpha + 2, \\ \alpha_1 = \tau(1 - \alpha^2) + 6, \\ \alpha_2 = \alpha^2 - 3\alpha + 2. \end{cases} \tag{9}$$

in which the optimal amount of τ is given in [13]. In frequency domain calculations it suffices to use this approximation to calculate $\frac{\partial G_C}{\partial \alpha}$ and $\frac{\partial G_C}{\partial \beta}$. To do this, first of all the approximation of s^{α} and s^{β} are substituted and then the differentiation of α and β is done. Using (8) and (9), we obtain

$$s^{\beta} \approx \frac{b_0 s^2 + b_1 s + b_2}{b_2 s^2 + b_1 s + b_0} \quad , \quad \begin{cases} b_0 = \beta^2 + 3\beta + 2, \\ b_1 = \tau(1 - \beta^2) + 6, \\ b_2 = \beta^2 - 3\beta + 2, \end{cases} \tag{10}$$

Therefore,

$$G_C(\rho_i) = k_P + k_I s^{-\alpha} + k_D s^{\beta} = \frac{k_D s^{\alpha+\beta} + k_P s^{\alpha} + k_I}{s^{\alpha}}, \tag{11}$$

and

$$G_C(\rho_i) \approx k_P + k_I \frac{\alpha_2 s^2 + \alpha_1 s + \alpha_0}{\alpha_0 s^2 + \alpha_1 s + \alpha_2} + k_D \frac{b_0 s^2 + b_1 s + b_2}{b_2 s^2 + b_1 s + b_0}. \tag{12}$$

Using (8)-(12), we can write

$$\frac{\frac{\partial G_C(\rho_i)}{\partial \rho}}{G_C(\rho_i)} = \begin{cases} \frac{\frac{\partial G_C}{\partial k_P}}{G_C(\rho_i)} = \frac{s^{\alpha}}{k_D s^{\alpha+\beta} + k_P s^{\alpha} k_I}, \\ \frac{\frac{\partial G_C}{\partial k_I}}{G_C(\rho_i)} = \frac{1}{k_D s^{\alpha+\beta} + k_P s^{\alpha} k_I}, \\ \frac{\frac{\partial G_C}{\partial k_D}}{G_C(\rho_i)} = \frac{s^{\alpha+\beta}}{k_D s^{\alpha+\beta} + k_P s^{\alpha} k_I}, \\ \frac{\frac{\partial G_C}{\partial \alpha}}{G_C(\rho_i)} \approx \frac{k_i s^{\alpha}}{k_D s^{\alpha+\beta} + k_P s^{\alpha} k_I} A(s), \\ \frac{\frac{\partial G_C}{\partial \beta}}{G_C(\rho_i)} \approx \frac{k_d s^{\alpha}}{k_D s^{\alpha+\beta} + k_P s^{\alpha} k_I} B(s), \end{cases}, \tag{13}$$

in which $A(s)$ and $B(s)$ will be obtained by relations (14) and (15) as follows:

$$\begin{aligned} A(s) &= \frac{\partial}{\partial \alpha}\left(\frac{\alpha_2 s^2 + \alpha_1 s + \alpha_0}{\alpha_0 s^2 + \alpha_1 s + \alpha_2}\right) \\ &= \frac{\partial}{\partial \alpha}\left(\frac{(\alpha^2 - 3\alpha + 2)s^2 + (\tau(1-\alpha^2) + 6)s + (\alpha^2 + 3\alpha + 2)}{(\alpha^2 + 3\alpha + 2)s^2 + (\tau(1-\alpha^2) + 6)s + (\alpha^2 - 3\alpha + 2)}\right) \\ &= \frac{(6\alpha^2 - 12)s^4 - (12\tau\alpha^2 + 6\alpha_1)s^3 + (12\tau\alpha^2 + 6\alpha_1)s - 6\alpha^2 + 12}{(\alpha_0 s^2 + \alpha_1 s + \alpha_2)^2}, \end{aligned} \tag{14}$$

and

$$\begin{aligned} B(s) &= \frac{\partial}{\partial \beta}\left(\frac{b_0 s^2 + b_1 s + b_2}{b_2 s^2 + b_1 s + b_0}\right) \\ &= \frac{(-6\beta^2 + 12)s^4 + (12\tau\beta^2 + 6b_1)s^3 - (12\tau\beta^2 + 6b_1)s + 6\beta^2 - 12}{(b_2 s^2 + b_1 s + b_0)^2}. \end{aligned} \tag{15}$$

Remark 2. If $\alpha > 1$, then $s^{-\alpha} = s^{-m-\sigma} = s^{-m}(s^{\sigma})^{-1}$, $m \in \mathbb{N}$ and $0 < \sigma < 1$. s^{σ} will be approximated using (8).

Remark 3. If $\beta > 1$, then $s^{\beta} = s^{n+\delta} = s^n s^{\delta}$, $n \in \mathbb{N}$ and $0 < \delta < 1$. s^{δ} will be approximated using (8).

It should be emphasized that Remarks 2 and 3 should be used only in $A(s)$ and $B(s)$ terms.

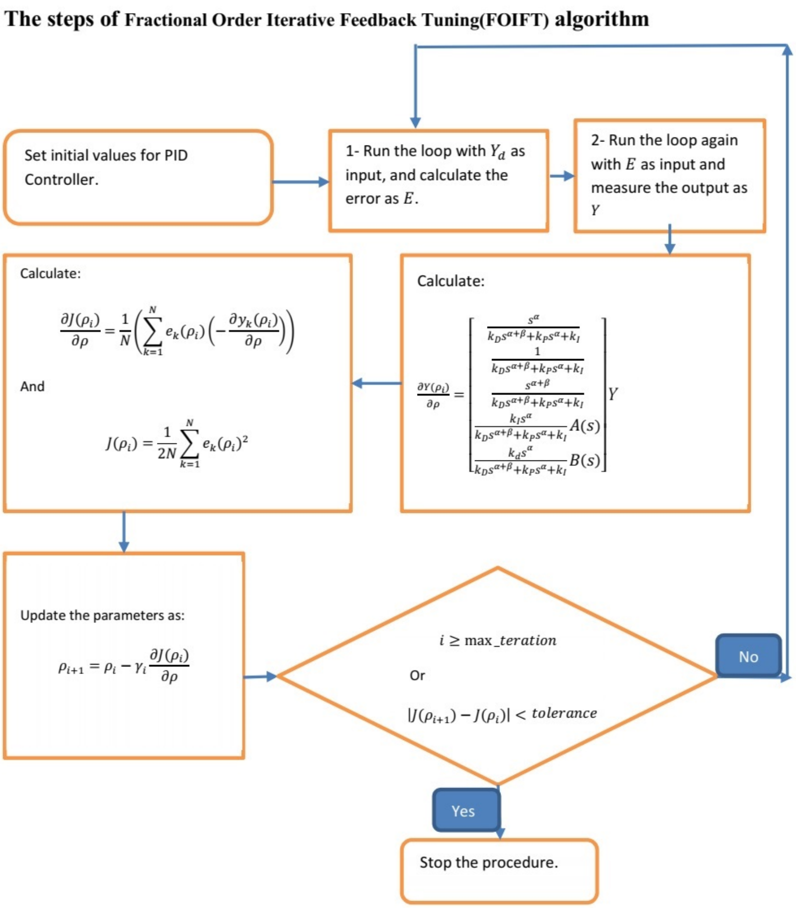

Figure 3: Fractional order iterative feedback tuning (FOIFT) flowchart.

Figure 3 flowchart depicts the steps of optimizing the FO controller using the IFT algorithm. Since the calculation of the terms in the above algorithm is cumbersome, using mathematical software such as Matlab is inevitable. Besides, since the FO calculations are yet to be formally added to the Matlab, it is needed to add some plugins to enable FO calculations before. One of the most complete sets of these pieces of code is the FOMCOM's toolbox [40], which is used for analyzing systems of FO in the Matlab and is available for free at the address indicated in [39].

The mentioned algorithm can be implemented via Simulink Blocks.

To verify the functionality of the proposed algorithm, two exemplary simulations have been performed. These examples are also solved by two methods of integer order IFT for PID controller and fractional order IFT tuning of $PI^{\alpha}D$. As a result, it is observed that the proposed algorithm results in better convergence and output characteristics than the two algorithms of integer order IFT and fractional order IFT method of $PI^{\alpha}D$. This stems from more free parameters of the FOIFT algorithm as compared to the other two algorithms. The results show that the algorithm converges to an optimal point, and the objective function is minimized.

Example 1. Consider the following system (Figure 4):

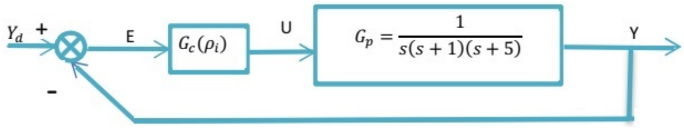

Figure 4: An instance system

In this example, it is emphasized that to optimize $G_C(\rho)$, the transfer function G_P is not used. It is shown that from the output, the error rate is computed for the step function input and placed in the algorithm. Using the proposed IFT algorithm of fractional order, first, the vector ρ_1 with trial and error was selected as:

$$\rho_1 = [k_P \; k_I \; \alpha \; k_D \; \beta]^T = [18 \; 12.815 \; 0.6 \; 9.3098 \; 0.4]^T$$

and $G_C(\rho_1) = 18 + 12.815s^{-0.6} + 9.3098s^{0.4}$. By selecting $G_C(\rho_1)$, the initial system output is obtained as in Figure 5.

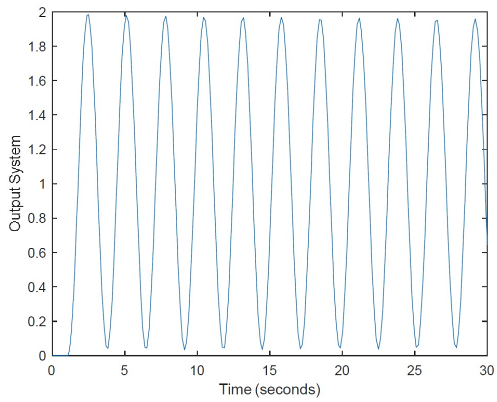

Figure 5: System output for values $\rho_1 = [18 \; 12.815 \; 0.6 \; 9.3098 \; 0.4]^T$.

At this point, (in the time domain, E is denoted by e_{ρ_i} and Y is denoted by y_{ρ_i}) the error $e_{\rho_1} = 0.3352$ was obtained. By giving the value e_{ρ_1} to the system again, the output value $y_{\rho_1} = 31.4245$ was obtained. It should be noted that the calculations are performed using the Simulink Block in the MATLAB program. Also, using Theorem 1, the convergence rate parameters were selected from $\gamma_i = \frac{\alpha\beta}{i}$, and $\gamma_1 = 0.24$. In this step, $N = 293$ and $\sum_{k=1}^{293} e_k^2(\rho_1) = 166.7503$ is calculated by MATLAB. From (2), $J(\rho_1) = 0.2845$. Using (5) and (13), we obtain

$$\frac{\partial y(\rho_1)}{\partial \rho} = [0.9483 \; 0.3604 \; 1.046 \; 13.9112 \; 3.774]^T.$$

Also, using (4), we have

$$\frac{\partial J(\rho_1)}{\partial \rho} = [0.0495 \; 0.0928 \; 0.0712 \; 0.3737 \; 3.3303]^T.$$

The values of ρ_2 were updated using (3):

$$\rho_2 = [17.9882\ 12.7928\ 0.6896\ 9.3268\ 1.1992]^T,$$

and $G_C(\rho_2) = 17.9882+12.7928s^{-0.6896}+9.3268s^{1.1992}$. By selecting $G_C(\rho_2)$, the system output is obtained as in Figure 6.

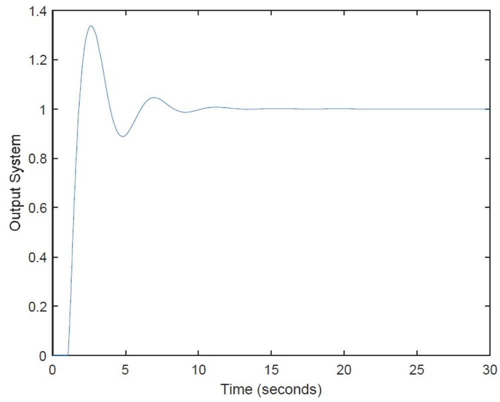

Figure 6: System output for values of $\rho_2 = [17.9882\ 12.7928\ 0.6896\ 9.3268\ 1.1992]^T$.

In this step, $N = 182$ and $\sum_{k=1}^{182} e_k^2(\rho_2) = 70.7659$ is calculated by MATLAB. Using (2), $J(\rho_2) = 0.1944$. The output improvement on the plot is noticeable. At this point, the error $e_{\rho_2} = 0.000020$ was obtained. The algorithm stop condition was $J(\rho_{i+1}) - J(\rho_i) < 0.002$. $J(\rho_2) - J(\rho_1) = 0.0901 > 0.002$. Furthermore, by giving the value e_{ρ_2} to the system with $G_C(\rho_2) = 17.9882+12.7928s^{-0.6896}+9.3268s^{1.1992}$, the output value $y_{\rho_2} = 0.000026$ was obtained. Using Theorem 1, $\gamma_2 = 0.4134$. Using (3), (4), (5) and (13), we obtain

$$\rho_3 = [17.9882\ 12.7927\ 0.6876\ 9.3278\ 1.2109]^T,$$

and $G_C(\rho_3) = 17.9882+12.7927s^{-0.6876}+9.3278s^{1.2109}$. By selecting $G_C(\rho_3)$, the system output is obtained as in Figure 7.

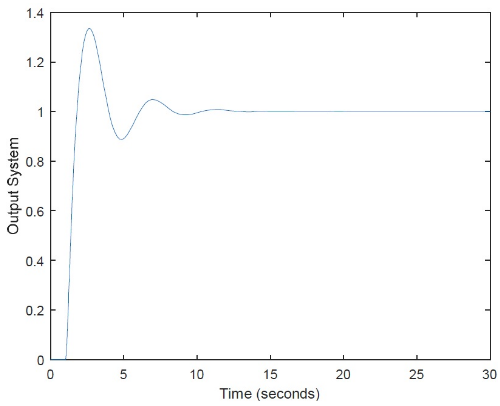

Figure 7: System output for values of $\rho_3 = [17.9882\ 12.7927\ 0.6876\ 9.3278\ 1.2109]^T$.

In this step, $N = 183$ and $\sum_{k=1}^{183} e_k^2(\rho_2) = 70.7659$ is calculated by MATLAB. Using (2), $J(\rho_3) = 0.1930$. $J(\rho_3) - J(\rho_2) = 0.1944 - 0.1930 = 0.0014 < 0.002$ and the algorithm stopped.

Table 1 and Figure 8 include the ρ and $\sum_{k=1}^{n} e_k^2(\rho_i)$ changes.

Table 1: Summary of the results of implementing the proposed IFT algorithm of fractional order and change of controller parameters and algorithm error convergence. (Example 1)

iteration	K_P	K_I	α	K_D	β	$\sum e^2$
1	18	12.815	0.6	9.3098	0.4	166.7503
2	17.9882	12.7928	0.6896	9.3268	1.1992	70.7659
3	17.9882	12.7927	0.6876	9.3278	1.2109	70.6575

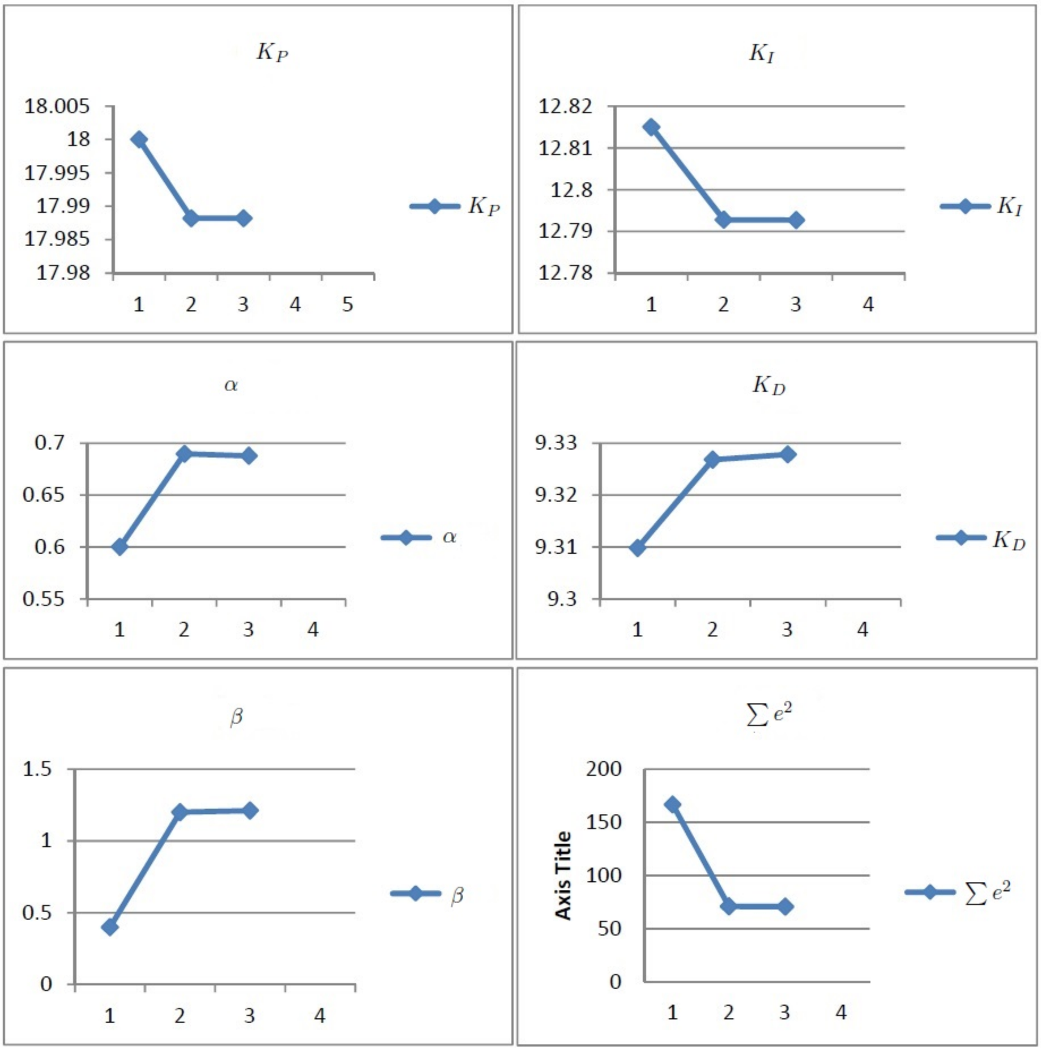

Figure 8: Changes in controller and error parameters as well as their convergence. (Example 1).

Then this example was solved with two algorithms of integer order IFT (Figure 2) and fractional order IFT in $PI^{\alpha}D$ mode [33] in three steps. The reason for choosing only the first three steps is to better compare the convergence speed of these algorithms with the algorithm presented in this paper. A summary of these two methods is given in the tables (Table 2 and Table 3) and figures below (Figures 9-14).

Table 2: Summary of the results of implementing the IFT algorithm of integer order and change of controller parameters and algorithm error convergence. (Example 1)

iteration	K_P	K_I	K_D	$\sum e^2$
1	15	28	9.3098	86.7237
2	15.0203	27.9855	9.3491	85.8082
3	15.0393	27.3491	9.3884	85.0388

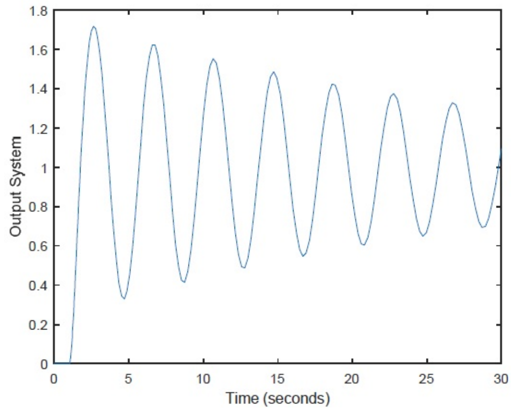

Figure 9: System output diagram for values of $\rho_1 = [15\ 28\ 9.3098]^T$.

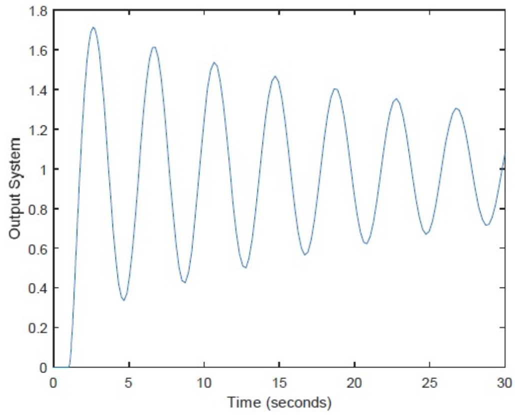

Figure 10: System output diagram for values of $\rho_2 = [15.0203\ 27.9855\ 9.3491]^T$.

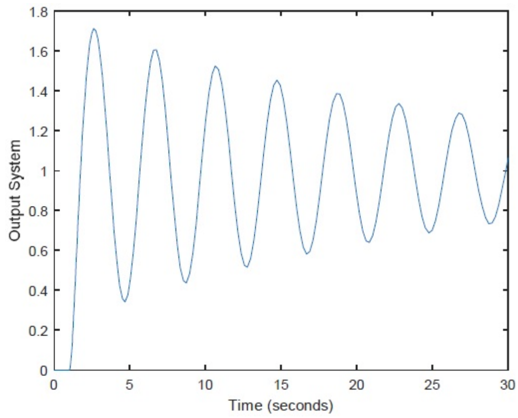

Figure 11: System output diagram for values of $\rho_3 = [15.0391\ 27.9752\ 9.3854]^T$.

Table 3: Summary of the results of implementing the IFT algorithm of fractional order ($PI^{\alpha}D$) and change of controller parameters and algorithm error convergence. (Example 1).

iteration	K_P	K_I	α	K_D	$\sum e^2$
1	30	28	0.6	6.5	207.4036
2	29.977	27.9344	0.9432	6.8271	91.2527
3	29.977	27.9322	1.5533	6.8386	76.7559

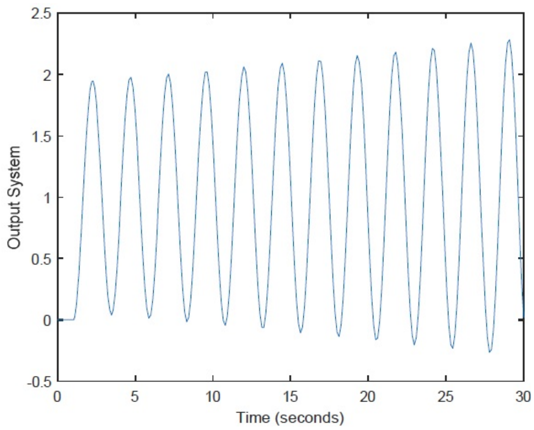

Figure 12: System output for values of $\rho_1 = [30\ 28\ 0.6\ 6.5]^T$.

It was observed that the convergence rate in these methods was less than the fractional order IFT algorithm in $PI^{\alpha}D^{\beta}$ mode. Moreover, as illustrated in Figure 15, the comparison of the results of the three methods shows that the performance is improved in the proposed controller, i.e., less overshoot and shorter settling time is achieved.

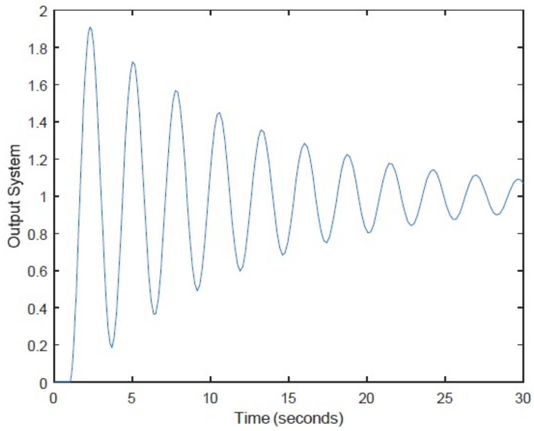

Figure 13: System output for values of $\rho_2 = [29.977\ 27.9344\ 0.9432\ 6.8271]^T$.

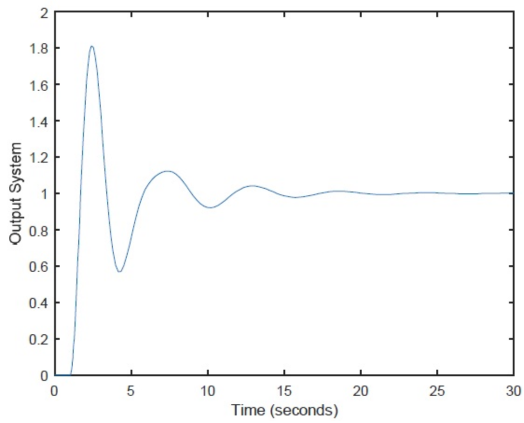

Figure 14: System output for values of $\rho_3 = [29.977\ 27.9322\ 1.5533\ 6.8386]^T$.

Example 2. Consider the following system (Figure 16):

Using the IFT algorithm of fractional order, first, the vector ρ_1 with trial and error was $\rho_1 = [K_P\ K_I\ \alpha\ K_D\ \beta]^T = [9\ 2\ 0.6\ 3\ 4]^T$ and $G_C(\rho_1) = 9 + 2s^{-0.6} + 3s^{0.4}$. By selecting $G_C(\rho_1)$, the initial system output is obtained as is Figure 17.

At this point, the error $e_{\rho_1} = 0.9944$ was obtained. By considering the value e_{ρ_1} to the system again, the output value $y_{\rho_1} = 46.0659$ was obtained. In this step, $N = 351$ and $\sum_{k=1}^{351} e_k^2(\rho_1) = 226.6601$ is calculated by MATLAB. From (2), $J(\rho_1) = 0.3228$. By Theorem 1, $\gamma_1 = 0.24$. Using (5) and (13), we obtain

$$\frac{\partial y(\rho_1)}{\partial \rho} = [3.5695\ 1.4625\ 3.69\ 7.5501\ 8.859]^T.$$

Also, using (4), we can write

$$\frac{\partial J(\rho_1)}{\partial \rho} = [0.3819\ 0.3667\ 0.0615\ 0.794\ 5.0541]^T.$$

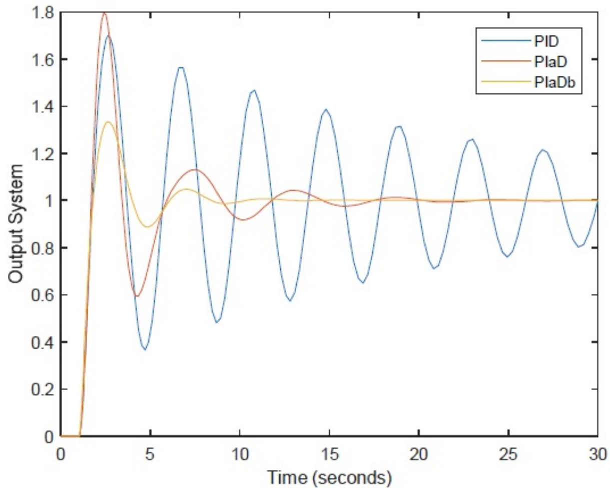

Figure 15: Comparison of the results of the three methods of integer order IFT for PID controller, fractional order IFT tuning of $PI^{\alpha}D$ and fractional order IFT tuning of $PI^{\alpha}D^{\beta}$.

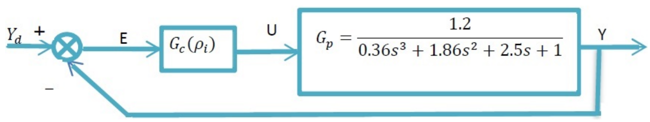

Figure 16

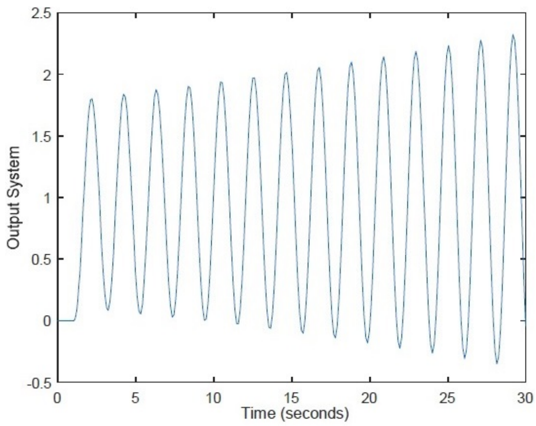

Figure 17: System output for values of $\rho_1 = [9\ 2\ 0.6\ 3\ 4]^T$.

The values of ρ_2 were updated using (3):

$$\rho_2 = [8.9084\ 1.912\ 0.7905\ 3.0147\ 1.6129]^T$$

and $G_C(\rho_2) = 8.9084 + 1.912s^{-0.7905} + 3.0147s^{1.6129}$. By selecting $G_C(\rho_2)$, the system output is obtained as in Figure 18.

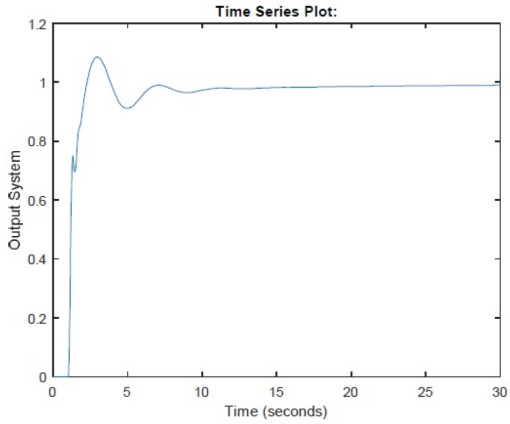

Figure 18: System output for values of $\rho_2 = [8.9084\ 1.912\ 0.7905\ 3.0147\ 1.6129]^T$.

In this step, $N = 226$ and $\sum_{k=1}^{226} e_k^2(\rho_2) = 62.9922$ is calculated by MATLAB. Using (2), $J(\rho_2) = 0.13936$. The output improvement on the plot is noticeable. At this point, the error $e_{\rho_2} = 0.0103$ was obtained. The algorithm stop condition was $J(\rho_{i+1}) - J(\rho_i) < 0.002$. $J(\rho_2) - J(\rho_1) = 0.18344 > 0.002$. Furthermore, by giving the value e_{ρ_2} to the system with $G_C(\rho_2) = 8.9084 + 1.912s^{-0.7905} + 3.0147s^{1.6129}$, the output value $y_{\rho_2} = 0.0105$ was obtained. Using Theorem 1, $\gamma_2 = 0.6374$. Using (3), (4), (5) and (13), we obtain

$$\rho_3 = [8.9087\ 1.912\ 0.7904\ 3.0175\ 1.6056]^T$$

and $G_C(\rho_3) = 8.9075 + 1.912s^{-0.7904} + 3.0175s^{1.6056}$. By selecting $G_C(\rho_3)$, the system output is obtained as in Figure 19.

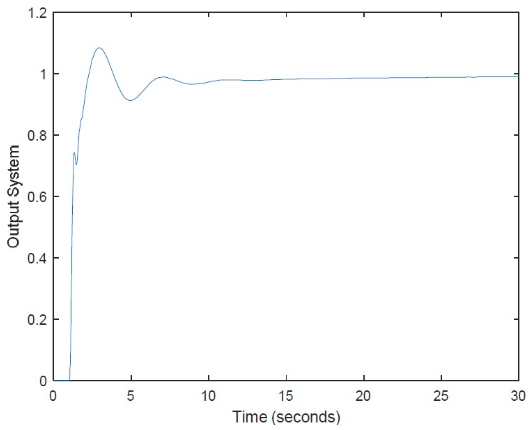

Figure 19: System output for values of $\rho_3 = [8.9087\ 1.912\ 0.7904\ 3.0175\ 1.6056]^T$.

In this step, $N = 226$ and $\sum_{k=1}^{226} e_k^2(\rho_2) = 62.9898$ is calculated by MATLAB. Using (2), $J(\rho_3) = 0.13935$. $J(\rho_3) - J(\rho_2) = 0.00001 < 0.002$ and the algorithm stopped.

Table 4 and Figure 20 include the ρ and $\sum_{k=1}^{n} e_k^2(\rho_i)$ changes.

Table 4: Summary of the results of implementing the proposed IFT algorithm of fractional order and change of controller parameters and algorithm error convergence. (Example 2)

iteration	K_P	K_I	α	K_D	β	$\sum e^2$
1	9	2	0.6	3	0.4	226.6601
2	8.9084	1.912	0.7905	3.0147	1.6129	62.9922
3	8.9087	1.912	0.7904	3.0175	1.6056	62.9898

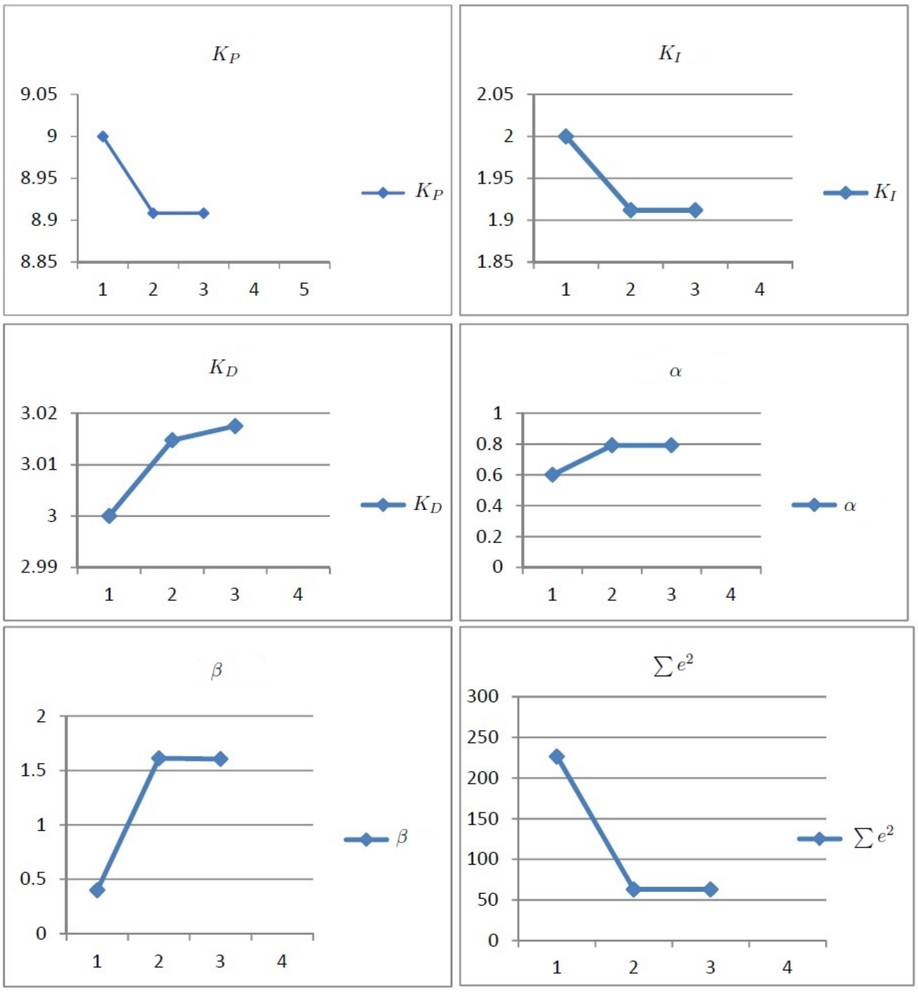

Figure 20: Changes in controller and error parameters as well as their convergence. (Example 2).

Then this example was solved with two algorithms of integer order IFT (Figure 2) and fractional order IFT in $PI^{\alpha}D$ mode [33] in three steps. A summary of these two methods is given in the tables (Table 5 and Table 6) and figures below (Figures 21-26).

Table 5: Summary of the results of implementing the IFT algorithm of integer order and change of controller parameters and algorithm error convergence. (Example 2)

iteration	K_P	K_I	K_D	$\sum e^2$
1	21	15	3.5	110.3648
2	20.962	14.9835	3.7374	86.6785
3	20.9512	14.9791	3.8024	83.9603

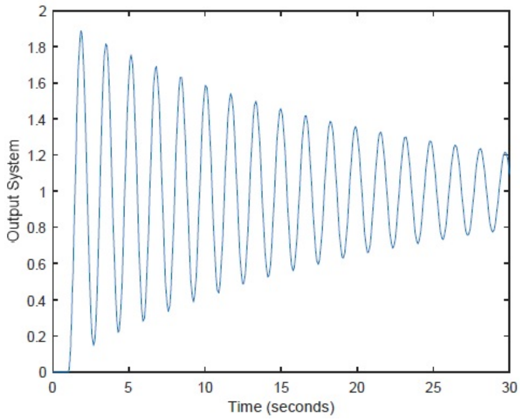

Figure 21: System output for values of $\rho_1 = [21\ 15\ 3.5]^T$.

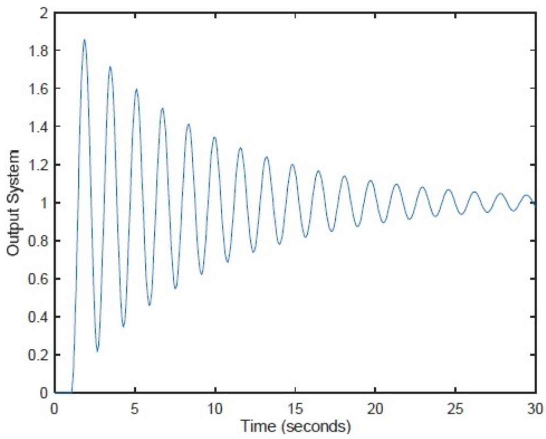

Figure 22: System output for values of $\rho_2 = [20.962\ 14.9835\ 3.7374]^T$.

Table 6: Summary of the results of implementing the IFT algorithm of fractional order ($PI^\alpha D$) and change of controller parameters and algorithm error convergence. (Example 2).

iteration	K_P	K_I	α	K_D	$\sum e^2$
1	18	8	0.6	3	124.9240
2	17.9647	7.97	0.9085	3.2086	78.3536
3	17.963	7.9691	0.9256	3.2186	77.8538

It was observed that the convergence rate in these methods is less than the fractional order IFT algorithm in $PI^\alpha D^\beta$ mode. Moreover, as illustrated in Figure 27, the

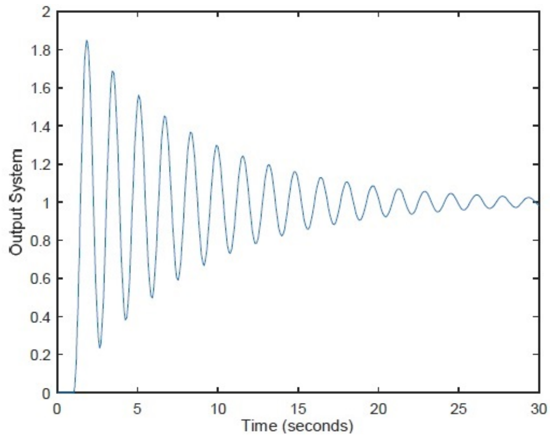

Figure 23: System output for values of $\rho_3 = [20.9512\ 14.9791\ 3.8024]^T$.

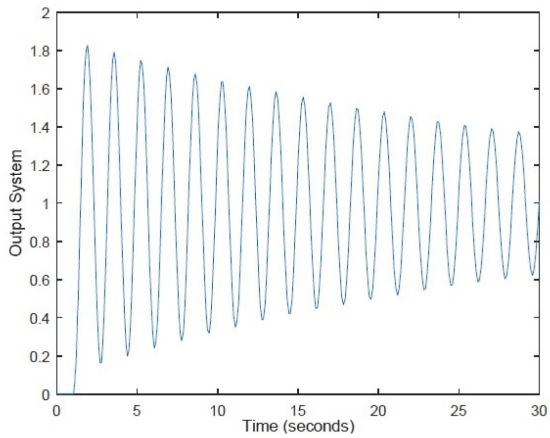

Figure 24: System output for values of $\rho_1 = [18\ 8\ 0.6\ 3]^T$.

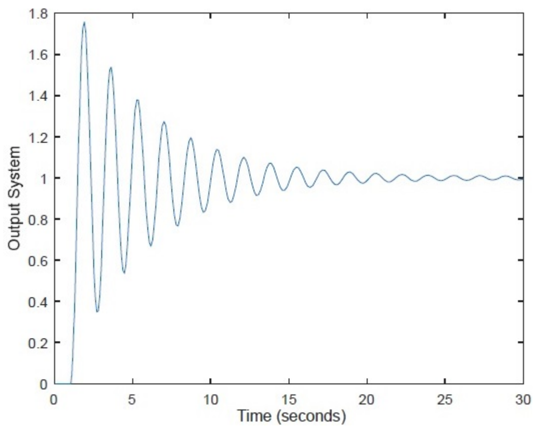

Figure 25: System output for values of $\rho_2 = [17.9647\ 7.97\ 0.9085\ 3.2086]^T$.

comparison of the results of the three methods shows that the performance is improved in the proposed controller, i.e., less overshoot and shorter settling time is achieved.

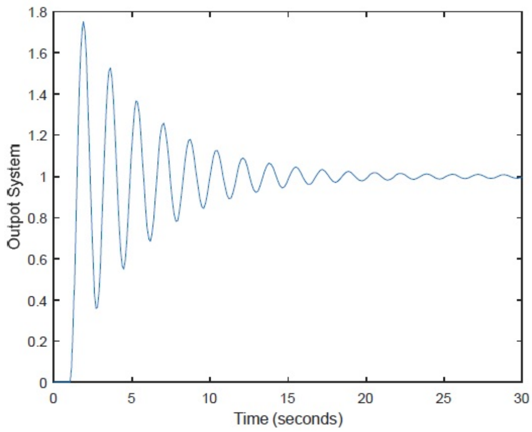

Figure 26: System output for values of $\rho_3 = [17.966\ 7.9691\ 0.9256\ 3.2186]^T$.

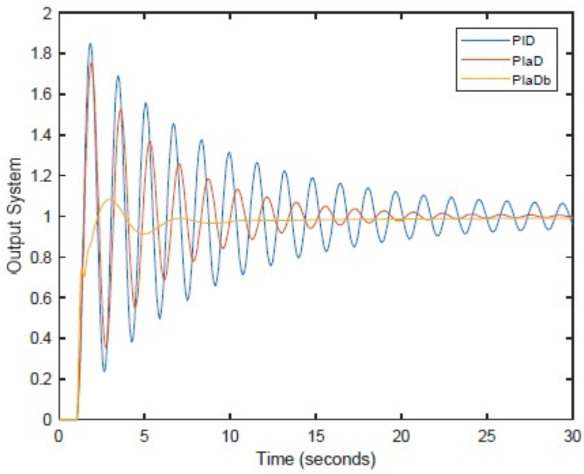

Figure 27: The comparison of the results of the three methods of integer order IFT for PID controller, fractional order IFT tuning of $PI^{\alpha}D$ and fractional order IFT tuning of $PI^{\alpha}D^{\beta}$.

5 Conclusion

In this paper, the IFT algorithm was proposed to optimize the fractional order PID (FOPID) controller. First, the IFT algorithm along with relevant equations was presented as a suitable practical tuning method in case a model of the plant was not available or was hard to obtain. Next, the structure of a FOPID was introduced. Then, a novel IFT algorithm for FOPID controllers was proposed, and the relevant equations were derived. Moreover, the convergence condition was discussed, and the algorithm's performance was verified by a numerical simulation on a FO system in the Matlab software. The simulation results showed that the algorithm converged to an optimal point without needing a plant model, and the output error diminished quickly over iterations. Thus, even though the algorithm may converge to a local minimum, it would optimize the performance of the controller independent of the knowledge of the plant, and this proves the algorithm's effectiveness.

References

[1] Acharya D. S., Mishra S. K. 2020. "A multi-agent based symbiotic organisms search algorithm for tuning fractional order PID controller", Measurement, 155, 107559.

[2] Ahmadi Dastjerdi A., Saikumar N., Hossein Nia S. H. 2018. "Tuning guidelines for fractional order PID controllers", Mechatronics, 56, 26-36.

[3] Ahmadi Dastjerdi A., Vinagre B. M., Chen Y. Q., Hossein Nia S. H. 2019. "Linear fractional order controllers", Annual Reviews in Control, 47, 51-70.

[4] Al Mamun A., Ho W. Y., Wang W. E., Lee T. H. 2007. "Iterative feedback tuning (IFT) of hard disk drive head positioning servomechanism", IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society, 33, 769-774.

[5] Bingul Z., Karahan O. 2011. "Tuning of fractional PID controllers using PSO algorithm for robot trajectory control", Mechatronics (ICM), IEEE International Conference on Mechatronics, 955-960.

[6] Birs I., Muresan C., Nascu I., Ionescu C. 2019. "A survey of recent advances in fractional order control for time delay systems", IEEE Access, 7, 30951-30965.

[7] Bunin G. A., Francois G., Bonvin D. 2013. "Iterative controller tuning by real-time optimization", IFAC Proceedings Volumes, 46(32), 21-26.

[8] Caponetto R., Dongola G., Fortuna L., Petrás I. 2010. "Fractional order systems modeling and control applications", World Scientific, Series on Nonlinear Science, Series A, 72.

[9] Caputo M. 1967. "Linear model of dissipation whose Q is almost frequency independent-II", Geophysical Journal International, 13, 529-539.

[10] Chang L. Y., Chen H. C. (2009). "Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system", World Scientific and Engineering Academy and Society (WSEAS), 8(1), 158-167.

[11] Erol H. 2021. "Stability analysis of pitch angle control of large wind turbines with fractional order PID controller", Sustainable Energy, Grids and Networks, 26, 100430.

[12] Formentin S., Heusden K. V., Karimi A. 2014. "A comparison of model-based and data-driven controller tuning". International Journal of Adaptive Control and Signal Processing, 28(10), 882-897.

[13] Freeborn T. J., Maundy B., Elwakil A. 2010. "Second-order approximation of the fractional Laplacian operator for equal-ripple response", I2010 53rd IEEE International Midwest Symposium on Circuits and Systems, 1173-1176.

[14] Gabriel G. L., Grandi J. O. 2019. "Tuning of fractional order PID controllers based on the frequency response approximation method", IFAC-Papers OnLine, 52(1), 982-987.

[15] Guermah S. 2011. "CRONE control, applications to a pilot laboratory process", Ph.D. Thesis, University of Mouloud Mammeri De Tizi-Quzou.

[16] Goharimanesh M., Lashkaripour A., Abouei Mehrizi A. 2015. "Fractional order PID controller for diabetes patients", Journal of Computational Applied Mechanics, 46(1), 69-76.

[17] Hjalmarsson H. 1998. "Control of nonlinear systems using iterative feedback tuning", Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207), 4, 2083-2087.

[18] Huusom J. K., Hjalmarsson H., Poulsen N. K., Jørgensen S. B. 2008. "Improving convergence of iterative feedback tuning using optimal external perturbations", 2008 47th IEEE Conference on Decision and Control, 2618-2623.

[19] Huusom J. K., Hjalmarsson H., Poulsen N. K., Jørgensen S. B. 2011. "A design algorithm using external perturbation to improve iterative feedback tuning convergence", Automatica, 47(12), 2665-2670.

[20] Jin Y., Chen Y. Q., Wang C., Luo Y. 2009. "Fractional order proportional derivative (FOPD) and FO [PD] controller design for networked position servo systems", International Design Engineering Technical Conference and Computers and Information in Engineering Conference, DETC2009-87662, 1341-1349.

[21] Kissling S., Blanc Ph., Myszkorowski P., Vaclavik, I. 2009. "Application of iterative feedback tuning (IFT) to speed and position control of a servo drive", Control Engineering Practice, 17(7), 834-840.

[22] Kumar L. Kumar P., Prawendra K., Satyajeet D. N. 2018. "Tuning of fractional order PIλ Dμ controllers using evolutionary optimization for PID tuned synchronous generator excitation system", IFAC-Papers OnLine, 51, 859-864.

[23] Lequin O., Gevers M., Mossberg M., Bosmans E., Triest L. 2003. "Iterative feedback tuning of PID parameters: comparison with classical tuning rules", Control Engineering Practice, 11(9), 1023-1033.

[24] Liu L., Zhang S. 2018. "Robust fractional-order PID controller tuning based on Bode's optimal loop shaping", Complexity, 2018, 6570560.

[25] Luo Y., Chen Y. Q. 2013. "Fractional order motion controls", Chichester, John Wiley & Sons.

[26] Mahathanakiet K., Johnson M., Sanchez A., Wade M. 2002. "Iterative feedback tuning and an application to a wastewater treatment plant", 4th Asian Control Conference, Singapore.

[27] Monje C. A., Chen Y. Q., Vinagre B. M., Xue D., Feliu-Batlle V. 2010. "Fractional order systems and controls: fundamentals and applications", Advances in Industrial Control, Springer-Verlag, London.

[28] Monje C. A., Vinagre B. M., Feliu, V., Chen Y. 2008. "Tuning and auto-tuning of fractional order controllers for industry applications", Control Engineering Practice, 16(7), 798-812.

[29] Nemati A., Alizadeh A., Soltanian F. 2019. "Solution of fractional optimal control problems with noise function using the Bernstein functions", Control and Optimization in Applied Mathematics, 4(1), 37-51.

[30] Nur Deniz F., Yüce A., Tan N., Atherton D. P. 2017. "Tuning of fractional order PID controllers based on integral performance criteria using Fourier series method", IFAC-Papers Online, 50(1), 8561-8566.

[31] Podlubny I. 1999. "Fractional differential equations", Academic Press, San Diego & London.

[32] Precup R., Preitl S., Fodor J., Ursache I. B., Clep P. A., Kilyeni S. 2008. "Experimental validation of iterative feedback tuning solutions for inverted pendulum crane mode control", 2008 Conference on Human System Interactions, 536-541.

[33] Rezaei M., Gharaveisi A., Vali M. 2017. "Fractional order proportional-integral-derivative controller parameter selection based on iterative feedback tuning; case study: Ball Levitation System", Transactions of the Institute of Measurement and Control, 40(6), 1776-1787.

[34] Robbins H., Monro S. 1951. "A stochastic approximation method", Annals of Mathematical Statistics, 22(3), 400-407.

[35] Ross B. 1977. "The development to fractional calculus 1695-1900", Historia Mathematica, 14(1), 75-89.

[36] Padula F., Visioli A. 2011. "Tuning rules for optimal PID and fractional-order PID controllers", Journal of Process Control, 21(1), 69-81.

[37] Petrás I. 2019. "Modified versions of the fractional-order PID controller", Applications in Control (De Gruyter), 6, 57-72.

[38] Sen M. 2014. "Introduction to fractional-order operators and their engineering applications", University of Notre Dame, Netherland.

[39] Tepljakov A. 2015. "FOMCON Toolbox", Available at: http://fomcon.net/fomcontoolbox/overview/.

[40] Tepljakov A., Petlenkov E., Belikov J., Finajev J. 2013. "Fractional-order controller design and digital implementation using FOMCON toolbox for MATLAB", 2013 IEEE Conference on Computer Aided Control System Design (CACSD), 340-345.

[41] Xinghe M., Rui W., Xuhui B. 2013. "Iterative feedback tuning for robotic manipulator trajectory tracking", International Journal of Advancements in Computing Technology, 5, 688-695.

[42] Xue D., Zhao C., Chen Y. Q. 2006. "Fractional order PID control of a DC motor with elastic shaft: a case study", 2006 American Control Conference.

[43] Yang B., Wang J., Wang J., Shu H., Li D., Zeng C., Chen Y., Zhang X., Yu T. 2021. "Robust fractional-order PID control of supercapacitor energy storage systems for distribution network applications: A perturbation compensation based approach", Journal of Cleaner Production, 279, 123362.

[44] Zamani M., Karimi-Ghartemani M., Sadati N., Parniani M. 2009. "Design of a fractional order PID controller for an AVR using particle swarm optimization", Control Engineering Practice, 17(12), 1380-1387.

[45] Zhang Y., Li J. 2011. "Fractional-order PID controller tuning based on genetic algorithm", 2011 International Conference on Business Management and Electronic Information, 764-767.

[46] Zhao C., Xue D., Chen Y. Q. 2005. "A fractional order PID tuning algorithm for a class of fractional order plants", Proceedings of the IEEE International Conference on Mechatronics & Automation, 1, 216-221.