10.24425/acs.2025.155400

Archives of Control Sciences
Volume 35(LXXI), 2025
No. 2, *pages* 375–424

Design of FOPID controllers by using 3D plots

Mehmet Emir KOKSAL

A three-dimensional (3D) graphical method is developed to tune the gain parameters of the fractional-order proportional-integral-derivative (FOPID) controller knowing fractional orders. The controlled plant itself is also fractional order (FO) and it may have a time delay. For special cases, the method is applicable for integer-order PID controller design for integer and fractional-order systems with or without time delay. Some graphical design tools are used beneficially in literature using 2D plots. Although the use of some minor 3D plots has appeared, they are based on shifting 2D plots to the third dimension, which makes the approach semi- or quasi-3D. This paper presents the mathematical formulations of five design specifications in accordance with the 3D drawing with programming implementations by MATLAB. For designing controllers by using the introduced 3D graphical method, system design specifications such as phase margin (PM), gain margin (GM), phase flatness (PF), low-frequency output disturbance rejection (LFODR) and high-frequency noise rejection (HFNR) are considered, and their important characteristics are shown. The requirements are mapped in the 3D Euclid space $(K_p, K_i, K_d) \in R^3$ by 3D surfaces and/or lines so that the proportional, integral, derivative control coefficients K_p, K_i, K_d can be chosen to meet the given specifications in an optimum way and to allow trade-off or compromise.

Key words: PID controller, FOPID controller, phase margin, gain margin, Iso-damping, phase flatness, noise rejection, 3D plots

1. Introduction

The branch of mathematics that deals with derivatives and integrations of fractional (non-integer) orders is known as fractional calculus. A system model that consists of a fractional differential equation (FDE) and/or a fractional integral equation (FIE) is said to be a fractional-order system (FOS). FOSs have recently become very useful in studying typical behaviors of dynamic systems in many

M.E. Koksal (e-mail: mekoksal@omu.edu.tr) is with Department of Mathematics, Ondokuz Mayis University, 55200 Atakum, Samsun, Turkey.

Author would like to thank Prof. Dr. Hans ZWART for sharing his valuable knowledge, ideas and reviews while writing this paper.

Received 9.11.2024. Revised 27.05.2025.

different branches of science. Hence, FO has become an essential tool that can be used for modelling and controlling systems of fractional nature by using FOPID controllers.

There are three commonly used defining formulas for fractional derivatives. They are the Grunwald-Letnikov (GL) fractional derivative [50], Riemann-Liouville (RL) fractional derivative [41] and Caputo's fractional derivative [13] definition defined as

$$ {}_a^C D_t^\alpha f(t) = \frac{1}{\Gamma(\alpha - n)} \int_a^t \frac{f^{(n)}(\tau)\,\mathrm{d}\tau}{(t-\tau)^{1-n+\alpha}} \quad \text{for } n-1 < \alpha < n, \tag{1} $$

where n is the smallest integer that is greater than or equal to α. If $\alpha = n$ then Caputo's definition turns to a classical form. When it comes to solving differential equations, Caputo's fractional derivatives do not need the initial condition's definition of the fractional derivative. Hence, Caputo's fractional derivative is very popular and most of the systems use Caputo's definition.

Since the study of dynamical systems is mostly done by means of transfer functions, the Laplace transform of the FO derivatives and integrals is essential for optimum training. Considering the Laplace transform of the FO, the most overall formulation is given as

$$ \mathcal{L}\left\{D^\alpha f(t)\right\} = s^\alpha \mathcal{L}\{f(t)\} - \sum_{k=0}^{n-1} s^k \left[\frac{\mathrm{d}^{\alpha-1-k} f(t)}{\mathrm{d}t^{\alpha-1-k}}\right]_{t=0}, \tag{2} $$

where n is an integer number, such that $n-1 < \alpha < n$ and α can be a non-integer number. When the initial (derivative part) condition is zero, this equation converts to

$$ \mathcal{L}\left\{D^\alpha f(t)\right\} = s^\alpha \mathcal{L} f(t). \tag{3} $$

A FO linear time-invariant system with single input and single output is represented by the general form

$$ \left(\sum_{i=0}^{m} a_i D^{\alpha_i}\right) y(t) = \left(\sum_{j=0}^{n} b_j D^{\beta_j}\right) x(t), \tag{4} $$

where D^γ is the fractional derivative operator, $\gamma = \alpha_1, \alpha_2, \ldots, \alpha_m$ and $\beta_1, \beta_2, \ldots, \beta_n$. Taking the Laplace transform, the FO transfer function becomes

$$ H(s) = \frac{Y(s)}{X(s)} = \frac{\sum_{j=0}^{n} b_j\, s^{\beta_j}}{\sum_{i=0}^{m} a_i s^{\alpha_i}}. \tag{5} $$

For the stability of the system, it is required that all the poles of this transfer function, that is the roots of the characteristic equation

$$\sum_{i=0}^{m} a_i s^{\alpha_i} = 0 \tag{6}$$

are in the left half of the complex s-plane. To check this, Routh Hurwitz standards can't be used directly in FOS because of multivalued functions, also known as "pseudo polynomials with rational power". Instead, the linear matrix inequality method and some other algorithms have been used [30,32,52]. However, an exact s-domain stability test like Routh Hurwitz Criterion exists for testing the stability of commensurate FOSs [47]; the term s^{α_i} in the above equation is written as $(s^{\alpha})^i$ where α is the commensurate power. For $0 < \alpha \leqslant 1$ and $1 < \alpha \leqslant 2$, stable and unstable regions on the complex s-plane are separated by the sectors $\pm\alpha\pi/2$ as shown in Fig. 1.

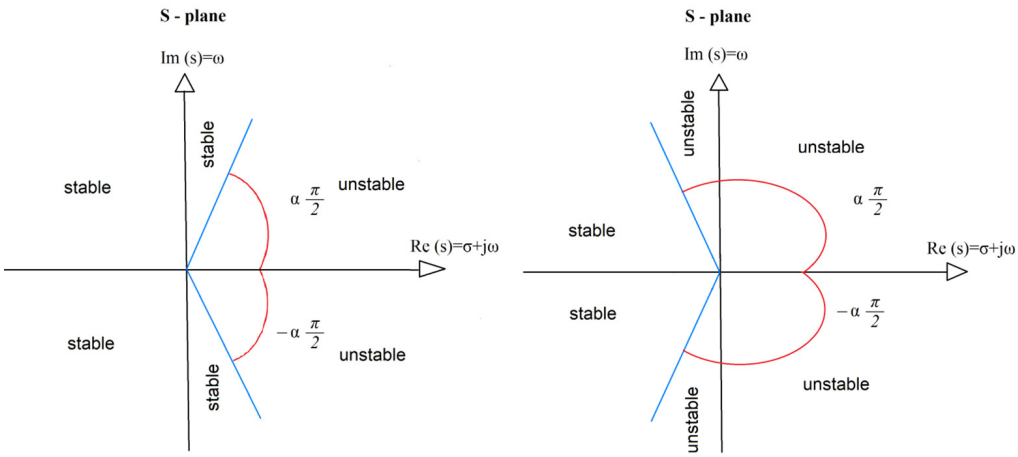

Figure 1: Stability regions of FO LTI system with orders $0 < \alpha \leqslant 1$ (on the left) $1 < \alpha < 2$ (on the right)

Due to its design simplicity and effectiveness, integer-order PID controllers have been used widely in industry since the early 1920s. Nowadays, there is a lot of usage of the PID controller conception in the desire of demanding a precise and enhanced automatic control system. A PID controller continuously computes the error $e(t) = y(t) - r(t)$, considering the preferred set-point, which is the reference point $r(t)$ and the measured values from the plant output $y(t)$; and accordingly, it applies an improvement regarding the P, I and D control terms and tries to lessen the error values by the modification of the controlled output [4]. In

general, PID controller output $u(t)$ is expressed as

$$u(t) = K_P\, e(t) + K_i \int_0^t e(t)\,\mathrm{d}t + K_d \frac{\mathrm{d}}{\mathrm{d}t} e(t), \tag{7}$$

where K_p, K_i and K_d are the proportional, derivative and integral gain constants, respectively.

Over the last few decades, industrial applications have been using systems that contain fractional derivatives and fractional integrals [72]. Different areas that have used FO controllers are electronics [9], documentation of systems [63], robot manipulations [26], canal control for irrigations [25], mechatronic-systems [44, 75], medicine [66], and heat control systems [33].

It is important to mention that the use of FOS theory can compactly regulate and control many physical systems and can easily control these systems with the help of FOPID controllers [58]. Also, FOPID controllers can be used even when the system has time delay behavior or is unstable [31].

The generalization of the traditional PID controller can be modified using FOs. The $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller is as follows

$$u(t) = K_p\, e(t) + K_i\, D^{-\lambda} e(t) + K_d D^{\mu} e(t), \tag{8}$$

where λ is the non-integer integrator-order and μ is the non-integer derivative-order.

The continuous transfer function of the $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller can be represented as

$$G_c(s) = \frac{U(s)}{E(s)} = \left(K_p + K_i \frac{1}{s^{\lambda}} + K_d\, s^{\mu}\right) = K_p \left(1 + \frac{1}{\tau_i s^{\lambda}} + \tau_d\, s^{\mu}\right) \quad (\lambda, \mu \geqslant 0), \tag{9}$$

where τ_i and τ_d are the associated time constants given as

$$\tau_i = \frac{K_p}{K_i} \quad \text{and} \quad \tau_d = \frac{K_d}{K_p}. \tag{10}$$

Different from the standard PID controller, two extra parameters exist in the $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller. Scientists have indicated that these two additional parameters reinforce the system properties, also these parameters increase the flexibility of the system and its stability [12, 76]. When designing FOPID controllers, five parameters (K_p, K_i, K_d, λ, μ) must be considered to achieve design specifications concerning (relative) stability, robustness, disturbance and noise rejection performance characteristics. So, the optimum and the most appropriate control of the system can be reached under the consideration of all the tuning parameters (K_p, K_i, K_d, λ, μ).

$PI^{\lambda}D^{\mu}$ control-design was first studied at the end of the 20th century [58]. Since then and for the sake of their flexibility and reliability because of the five parameters, the research on these controllers has been increasing continuously [7, 11, 22, 28, 36, 37, 39, 40, 42, 54, 56, 65].

It is essential to have a global approach for the automatic optimization of the controller's parameters (K_p, K_i, K_d, λ, μ) to have efficient and effective FOPID controllers. Despite the flexibility of FOPID controllers, there are some difficulties in designing fractional controllers in the time domain. Furthermore, it has been detected that the procedures focused at the "gain" and the "phase" margins presented in the past studies for conventional order systems are not totally appropriate to the FOS.

Numerous evolutions for parameter optimization algorithms have been developed for the last decades, such as genetic algorithms [1, 14, 39, 61]. In [6], it was suggested that differential evolution algorithms have very useful effects on the FOPID controllers. Smart optimization for designing FOPID controllers based on particle swarm optimization (PSO) is proposed in [11, 45, 74]. A PSO algorithm is a progressive research process that has been demonstrated to have actual high effectiveness in facilitating both the time domain and the frequency domain control strategy. Adaptive PSO algorithm accomplishes more rapid convergence-speed and enhanced resolution precision with the smallest incremental computational responsibility [2]. Another application of PSO algorithm for tuning the integral order (λ) and the derivative order (μ) parameters is proposed in [3] for a slide mode FOPID controller and an observer that intends for the non-linear continuous stabilization, so that is guaranteed in synchronous switching closed-loop system stability. PSO algorithm is also used by Dabiri et al. to find the optimum tuning parameters [18], where a method to discretize the FDE with the variable operators has been suggested.

Choosing appropriate tuning parameters is very important for designing an effective control system algorithm. Depending on this selection, the model structure can be modified according to the desired systems. Selecting the best parameters and making the best approximation lead the system to have better progression [62] proposed two different modeling parts. The first one is to design an integer-order system model using "differential evolution algorithm (DEA) and "artificial bee colony" (ABC), and this is done using integer and fractional model constructions with time delays. The second part is done using a new prototype attained with the assistance of MATLAB/Simulink set to design a FOPID controller. Senberger and Bagis have also presented a study that tests the system performance of high order modeling systems using ABC algorithms. Another system model is presented in [49], establishing on the cascade of a linear model and non-linear second-order Volterra model using differential evolution algorithms.

In designing many FOPID controllers, the transfer function of the FO is approximated by integer-order transfer function using different methods [15, 20, 53, 58] so that FOPID controllers have been prolonged to general forms of conventional PID controllers.

For the closed-loop control systems viewpoint, asymptotic tracking of various forms of set-point inputs (for example step and ramp) are some of the noticeable objectives of control systems [68]. To control the system performance, integral performance indices are established on the system error, also time is regarded as a computable measure. A system controller is on its optimal when the system parameters are tuned to values such that the pre-required design restriction is met and when the integral indices touch their lowest points. Some of the important time-domain performance criteria include "Integral Absolute Error (IAE)", "Integral Time Square Error (ITSE)", "Integral Time Absolute Error (ITAE)", "Integral Square Error (ISE)", "Integral Square Time Error (ISTE)" and "Integral Square Time Square Error (ISTES)". The mathematical formulation of these integral performance indices is well-known and can be found in, for example [20].

The fact is that each of the different types of integral performance indices has its advantages in control system design [59]. According to [59], ITSE has the most benefits of punishing the system error further, and because of the presence of the time multiplication term, ITSE restrains out the system oscillation very rapidly. Conversely, when unexpected alterations occur in the reference point, ITAE-based controllers are preferred. The time-multiplication term in ITAE integral indices disciplines the error further at future or later steps and supports reducing the settling-time T_s which is defined as the time that elapses for the output to remain near the reference value with an error less than $\mp 5\%$. Also, the absolute error that takes in the ITAE is for minimizing the overpass. ITAE can produce small overshoot and oscillations compared to the IAE and the ISE performances. Further, the ITAE has the best selectivity and sensitivity compared to the IAE and ISE [64] "In contrast to ITAE, ITSE also has less sensitivity, and it is not contented computationally" [45].

For FOPID optimal design, different smart techniques have been suggested in [21] in order to diminish the integral performance indices. ITAE is used for FOPI control system design in a permanent magnet synchronous motor using Hybrid Differential Artificial Bee Colony Algorithm (HDABCA) in [59]. Optimizing the ITAE with the PSO algorithm established in [45] for FOPID control system design stability. In [8], a very useful application of ITAE using different evolutionary algorithms based on the FO system controllers was proposed.

Recently, few scientists placed metrics concentrated on lessening the control signal. Integral Square Controller Output and Integral Square Deviation Controller Output are the best and most extensively used ones among them. In [10],

a genetic algorithm to improve and find the parameters of the best controller and to optimize the weighted sum of ISE and Integral Square Controller Output have been used. Furthermore, Cao et al. have used PSO for a similar application.

Different from the evolutionary and PSO-based design, another important way of designing a FOPID controller is for Fuzzy instructions [21].

In the time domain optimization methods of controller design, the error function $e(t) = r(t) - y(t)$ is not the only objective to minimize. There are other performance indices. Different from the time integrals, an optimization method using "Differential Evolutions" has been proposed in [6] for the problem optimizations of FOPID controllers such as "dominant pole-placement". PSO-based optimizations outline has been designed in [35] to minimize the "weighted sum, rise-time, overshoot, settling-time steady-state-errors" of the FOPID control systems. Some other approaches for minimization and system optimization can be found in [19, 55, 60, 74].

2. FOPID control system design

2.1. Basic control structure

For the design of the $PI^{\lambda}D^{\mu}$ controller, the following feedback control structure in Fig. 2 is considered.

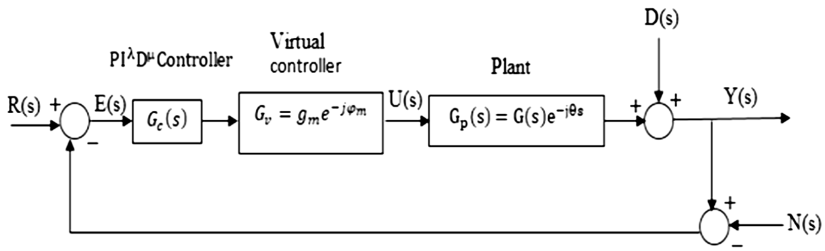

Figure 2: Basic structure of single input and single output FO control system

In this figure, $Y(s)$ represents the output of the system, $U(s)$ is the controller's output signal, $R(s)$ is the reference input, $E(s)$ is the error, $D(s)$ and $N(s)$ represent the output disturbance and measurement noise, respectively. $G_c(s)$ is the FOPID controller's transfer function as defined in Eq. (9) and

$$G_p(s) = \frac{A(s)}{B(s)} e^{-\theta s} = \frac{\sum_{k=0}^{n} b_k s^{\beta_k}}{\sum_{r=0}^{m} a_r s^{\alpha_r}} e^{-\theta s} \tag{11}$$

is the transfer function of the controlled system (FO plant with a delay time θ). The virtual gain G_v (not actually present) is chosen as

$$G_v = g_m\,, \qquad G_v = e^{-j\varphi_m} \quad \text{or} \quad G_v = g_m e^{-j\varphi_m} \tag{12}$$

to gain relative stability requirements GM g_m, PM φ_m or both, respectively, that will be defined in the sequel. It is also called gain-phase margin tester [17, 29]. Therefore, it should be taken as the identity ($G_v = 1$) unless it is stated differently.

The output of the control system is given as

$$\begin{aligned} Y(s) &= \frac{G_c(s)G_vG_p(s)}{1+G_c(s)G_vG_p(s)}R(s) + \frac{G_c(s)G_vG_p(s)}{1+G_c(s)G_vG_p(s)}N(s) \\ &\quad + \frac{1}{1+G_c(s)G_vG_p(s)}D(s) \\ &= T_R(s)R(s) + T_N(s)N(s) + T_D(s)D(s). \end{aligned} \tag{13}$$

Here, $T_R(s)$, $T_N(s)$, $T_D(s)$ are the transfer functions between the input reference, the noise signal, the disturbance signal, respectively, and the controlled output of the plant.

2.2. Relative stability of FOPID control system

For the stability of the basic control system introduced above, all the zeros of the characteristic equation

$$1 + G_c(s)G_vG_p(s) = 0 \tag{14}$$

have to be in the left half-plane. Most times, the characteristic equation does not lead to a commensurate characteristic polynomial (the polynomial of s^α as described in the Introduction) and so the stability test therein cannot be applied. In the general case, to check the system stability in the frequency domain, we first convert the s-plane into the complex plane $G_c(j\omega)G_vG_p(j\omega)$. The Nyquist plot of the open-loop gain $G_c(j\omega)G_vG_p(j\omega)$ is obtained as the plot on the complex $G_c(j\omega)G_vG_p(j\omega)$ plane for $\omega = [0, \infty)$ Through the conversion, Eq. (14) becomes

$$G_c(j\omega)G_vG_p(j\omega) = -1, \tag{15}$$

which points out the absolute stability point (ASP $= (-1 + j0)$) in the complex plane of $G(j\omega) = G_c(j\omega)G_vG_p(j\omega)$ where the system enters into an unstable state [57]. There are two other important stability points on $G(j\omega)$-plane as shown on the typical Nyquit plot of Fig. 3 (drawn for $G_v = 1$); these are the phase crossover point (PCOP) where the phase $\emptyset(\omega)$ of $G(j\omega)$ becomes -180° at the phase crossover frequency ω_{pc}, and the gain crossover point (GCOP) where the

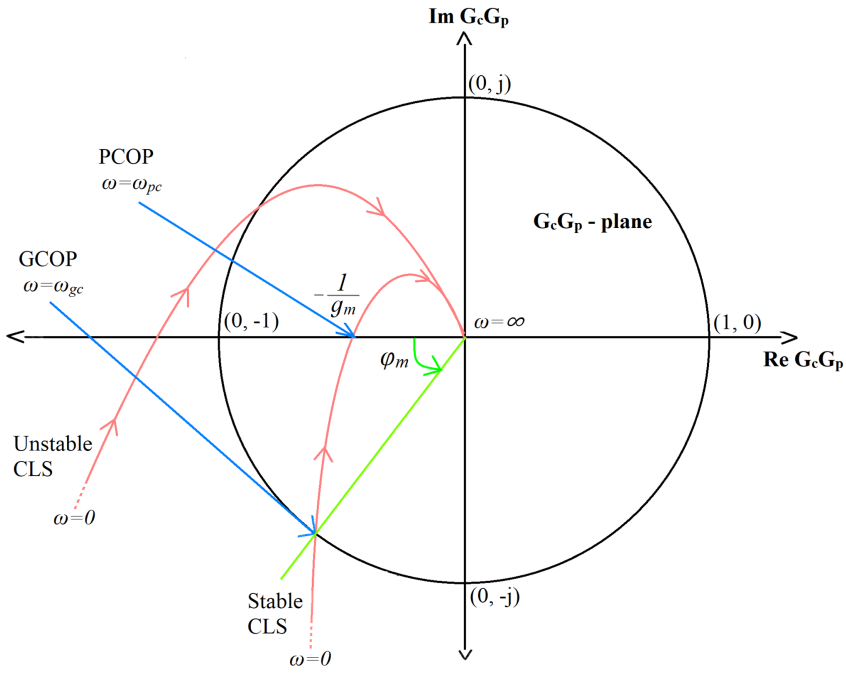

Figure 3: Typical Nyquist plot of the open-loop gain $G_c(j\omega)G_p(j\omega)$

gain $M(\omega)$ of $G(j\omega)$ becomes 1 at the gain crossover frequency ω_{gc}. These are the points of which relative locations with respect to the ASP $= (-1+j0)$ determines the relative stability of the closed loop system. Expressing mathematically:

$$\begin{aligned} \text{PCOP:}\quad \emptyset(\omega_{pc}) &= \operatorname{Arg}\left[G_c(j\omega_{pc})G_p(j\omega_{pc})\right] = -180^\circ; \\ G(j\omega_{pc}) &= \frac{1}{g_m}e^{-j\pi} = -\frac{1}{g_m}, \end{aligned} \tag{16}$$

$$\begin{aligned} \text{GCOP:}\quad M(\omega_{gc}) &= \left|G_c(j\omega_{gc})G_p(j\omega_{gc})\right| = 1; \\ G(j\omega_{gc}) &= 1e^{-j(\pi-\varphi_m)} = 1e^{j(-\pi+\varphi_m)}, \end{aligned} \tag{17}$$

where

$$g_m = -\frac{1}{G_c(j\omega_{pc})G_p(j\omega_{pc})}, \qquad \text{GM} = 20\log(g_m) \quad \text{in dB}, \tag{18}$$

$$\varphi_m = \arg\left[G_c(j\omega_{gc})G_p(j\omega_{gc})\right] + \pi, \qquad \text{PM} = \frac{180^\circ}{\pi}\varphi_m \quad \text{in degrees} \tag{19}$$

are defined as GM and PM, respectively. If φ_m comes out to be greater than π, then $+\pi$ should be replaced by π in Eq. (19). Note that it is usual to express

GM in decibels and PM in degrees. Larger GM results with a PCOP to the very right of the ASP $(-1, 0)$ hence with a more stable CLS. Similarly, greater PM results with a GCOP to the very downright of the ASP point $(-1, 0)$ hence with a more stable CLS. This is because the possibility of the encirclement of the ASP $(-1, 0)$ by the Nyquist plot reduces with larger values of GM and PM and the system becomes relatively more stable according to Nyquist criteria. A gain margin > 20 dB and a phase margin $> 60^\circ$ result with sufficiently stable CLSs. Therefore, GM and PM are very important measures of robustness in designing controllers. It is known that PM is related to the damping of the system and can also serve as a performance measure [27, 51].

2.3. Frequency domain method of FOPID controller design

Traditional PID controller is a standard feedback mechanism control loop commonly used in many industrial applications. This PID controller tries to reduce the error by computing and fixing the error for the corrective achievement that can regulate the procedure accordingly. As seen in Eq. (7), three different parameters K_p, K_i, K_d are involved in the standard PID controllers. K_p is used to determine the present error response, K_i is used to determine the previous latest errors based on the responses, and K_d is used to determine the response of the system to the degree at which the error had been varying.

Though the FOSs are very rare, commonly, the integer-order system's approximation might have some significant variations between the real system and their mathematical styles. So, using FOPID is one of the most important possibilities to progress the traditional PID controllers [24, 34, 58].

In the design of FOPID controllers, there are two surplus parameters λ and μ compared to the conventional PID controller; so, there are five different parameters to consider $(K_p, K_i, K_d, \lambda, \mu)$, which increases the flexibility of succeeding the current preferred requirements. However, these also increase the system's complexity. So, the task is to develop a feasible and reliable FOPID controller that behaves and meets the same system design requirements (necessities) and performance but has less parameter numbers. The most important point here is to find a suitable and achievable estimation for the integral and derivative orders (λ, μ).

Numerous efforts to discover an optimal set for $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controller's five parameters have been published so as to accomplish predefined criteria of the system [6, 23]; especially the integral order (λ) and the derivative order (μ) parameters are either known or can be found by using the existing methods in the literature [3]. Ziegler-Nichols parameters tuning rules were stated in [70]. Also, some other methods have been stated in [6, 23, 46, 71].

In this paper, the non-integer order of integration λ and differentiation μ in $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ are assumed to be determined by one of the above-mentioned methods present in the literature. Then, according to the design specifications stated in the next subsection, the remaining three tuning parameters $K_p,\ K_i,\ K_d$ are determined by using 3D graphical approach in the space $R^3 = (K_p, K_i, K_d)$ as described in Section 4.

2.4. Design specifications

With the plant transfer function $G_p(s)$ in Eq. (11), the objective is to find a controller $G_c(s)$ as in Eq. (9), so that the open-loop system meets the following frequency domain design specifications [51]:

(a) Phase margin specification (φ_m, ω_{gc}): For sufficiently good relative stability, PM defined in Eq. (19) at the gain-crossover frequency ω_{gc} should be larger or equal to some pre-assigned value φ_m. This mean

$$\arg\left[G_c(j\omega_{gc})G_p(j\omega_{gc})\right] \geqslant \varphi_m - \pi.$$

Letting $G_v = e^{-j\varphi_m}$ and subtracting φ_m from both sides, this equation can be written as

$$\arg\left[G_c(j\omega_{gc})G_vG_p(j\omega_{gc})\right] \geqslant -\pi \tag{20}$$

(b) Gain margin specification $(g_m,\ \omega_{pc})$: Sustaining a preferred GM defined in Eq. (18) at the phase crossover frequency ω_{pc} should be larger or equal to some pre-assigned value g_m. This means

$$-\frac{1}{G_c(j\omega_{pc})G_p(j\omega_{pc})} \geqslant g_m\,.$$

Letting $G_v = g_m$ and dividing both sides by g_m this equation can be written as

$$-\frac{1}{G_c(j\omega_{pc})G_vG_p(j\omega_{pc})} \geqslant 1. \tag{21}$$

(c) Robustness to gain variation or Iso-damping $(\psi_{\text{spec}},\ \omega_{gc})$: The phase of the open-loop system must be flat at ω_{gc} [16]. This condition is expressed by

$$\psi_f = -\frac{\mathrm{d}}{\mathrm{d}\omega}\mathrm{Arg}\left[G_c(j\omega)G_p(j\omega)\right]\Big|_{\omega=\omega_{gc}} \leqslant \psi_{\text{spec}} \quad (\text{usually} \geqslant 0). \tag{22}$$

Decreasing ψ_{spec} increases relative stability and the system is more robust to gain changes and the overshoot of the response is almost constant within a gain range, which is known as iso-damping of time response.

(d) Output disturbance rejection $(B_d,\ \omega_d)$: A constraint on the disturbance transfer function $T_D(s)$ in Eq. (13) with $G_v = 1$ can be set to prevent the low-frequency disturbance by setting

$$\left| T_D(j\omega) = \frac{1}{1 + G_c(j\omega)G_p(j\omega)} \right|_{\omega \leqslant \omega_d} \leqslant B_d\,, \tag{23}$$

where B_d is an upper limit for the noise gain for frequencies $\omega \leqslant \omega_d$. B_d is the load disturbance suppression for frequencies $\omega \leqslant \omega_d$ [19, 51].

(e) High-frequency noise rejection $(C_n,\ \omega_n)$: A constraint on the noise transfer function $T_N(s)$ in Eq. (13) with $G_v = 1$ (also known as the complementary sensitivity function or noise attenuation requirement [19]), can be set to prevent the interference of high-frequency noise by setting

$$\left| T_N(j\omega) = \frac{G_c(j\omega)G_p(j\omega)}{1 + G_c(j\omega)G_p(j\omega)} \right|_{\omega \geqslant \omega_n} \leqslant C_n\,, \tag{24}$$

where C_n is the lower bound on the complementary sensitivity function for frequencies $\omega \geqslant \omega_n$ [19, 51].

The condition (b) is not included as a design specification in [51]. Due to its importance from a relative stability point of view and to guarantee a stable control system, it is included in the design requirements in this paper. If both specifications (a) and (b) are given simultaneously, then Eqs. (20) and (21) should be satisfied simultaneously by choosing the virtual gain $G_v = g_m e^{-j\varphi_m}$. That is why Eq. (12) is defined and used as the virtual gain. Furthermore, although steady-state error cancellation is included in [51], the fractional integration $s^{-\lambda}$ in the controller is an efficient term to cancel the steady-state error. So the steady-state error specification is automatically fulfilled, and this specification is null and not included in the above specification list.

To achieve the above required performance parameters $(\varphi_m,\ \omega_{gc})$, $(g_m,\ \omega_{pc})$, $(\psi_{\text{spec}},\ \omega_{gc})$, $(B_d,\ \omega_d)$, and $(C_n,\ \omega_n)$, the values that the controller gains $K_p,\ K_i,\ K_d$ can take will be discussed graphically in the next section by using 3D plots in the space $(K_p,\ K_i,\ K_d) \in R^3$.

3. Formulation of the design specifications for the 3D plots

Based on the design specifications set in the previous section, their formulations for the 3D plots are considered in this section. The plots supply an optimum set of design parameters $(K_p,\ K_i,\ K_d) \in R^3$.

For refreshing the mind, the plant transfer function $G_p(s)$ is a FO rational function with a time delay; the FO integrator and differentiator control orders λ

and μ are either known or can be found by using the existing methods mentioned above.

The computations of the loci of acceptable regions in the 3D space satisfying the given specifications in Section 2.4 are obtained in the same order therein.

3.1. Stability boundaries

To find the stability boundaries [6, 17, 29]] for achieving the specified phase and gain margins defined by items (a) and (b) in Section 2.4, consider Eq. (14) together with Eqs. (9), (11) and (12). Note that stability boundaries are computed by considering $G_v = g_m$ to guarantee a GM of g_m, $G_v = e^{-\varphi_m}$ to guarantee a PM of φ_m and $G_v = g_m e^{-\varphi_m}$ to guarantee both. Hence, with $G_v = g_m e^{-j\varphi_m}$, the characteristic equation (14) results; further, the relative stability boundaries are defined by replacing the inequalities in Eqs. (20) and (21) by the equality signs; thus, the stability boundaries satisfy Eq. (15).

Real root boundary (RRB) is found by letting $s = 0$ in Eq. (14), which leads to

$$K_i = 0. \tag{25}$$

Hence, RRB is a plane perpendicular to $K_p - K_i$ plane in $(K_p, K_i, K_d) = R^3$.

Complex root boundary (CRB) is obtained by $s = j\omega$ in Eq. (14); this leads to Eqs. (15). Using the expressions (9), (11), (12) in (15), multiplying by $\omega^{\lambda} e^{j(0.5\pi\lambda-\gamma)}/N$, collecting the terms with K_p, K_i, K_d on the left side, and finally, equating the imaginary and real parts on both sides, we obtain

$$\begin{bmatrix} \omega^{\lambda}\sin\left(\frac{\pi}{2}\lambda\right) & 0 & \omega^{\lambda+\mu}\sin\left(\frac{\pi}{2}(\lambda+\mu)\right) \\ \omega^{\lambda}\cos\left(\frac{\pi}{2}\lambda\right) & 1 & \omega^{\lambda+\mu}\cos\left(\frac{\pi}{2}(\lambda+\mu)\right) \end{bmatrix} \begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = -\frac{\omega^{\lambda}}{N} \begin{bmatrix} \sin\left(\frac{\pi}{2}\lambda-\gamma\right) \\ \cos\left(\frac{\pi}{2}\lambda-\gamma\right) \end{bmatrix} \tag{26}$$

where

$$G_v G_p(j\omega) = N(\omega) e^{j\gamma(\omega)}, \tag{27}$$

$$N(\omega) = g_m \left\{ \frac{\left[\sum_{k=0}^{n} b_k \omega^{\beta_k} \cos\left(0.5\pi\beta_k\right)\right]^2 + \left[\sum_{k=0}^{n} b_k \omega^{\beta_k} \sin\left(0.5\pi\beta_k\right)\right]^2}{\left[\sum_{r=0}^{m} a_r \omega^{\alpha_r} \cos\left(0.5\pi\alpha_r\right)\right]^2 + \left[\sum_{r=0}^{m} a_r \omega^{\alpha_r} \sin\left(0.5\pi\alpha_r\right)\right]^2} \right\}^{\frac{1}{2}}, \tag{28}$$

$$\gamma(\omega) = -\varphi_m + \beta(\omega) - \theta\omega, \tag{29}$$

$$\beta(\omega) = \arctan \frac{\sum_{k=0}^{n} b_k \omega^{\beta_k} \sin\left(0.5\pi\beta_k\right)}{\sum_{k=0}^{n} b_k \omega^{\beta_k} \cos\left(0.5\pi\beta_k\right)} - \arctan \frac{\sum_{r=0}^{m} a_r \omega^{\alpha_r} \sin\left(0.5\pi\alpha_r\right)}{\sum_{r=0}^{m} a_r \omega^{\alpha_r} \cos\left(0.5\pi\alpha_r\right)}. \tag{30}$$

Drawing K_p from the first line of Eq. (26) and inserting the result in the second line of Eq. (26), and similarly Drawing K_d from the first line of Eq. (26) and inserting the result in the second line of Eq. (26), we obtain the following matrix equation:

$$\begin{bmatrix} \omega^{\lambda}\sin\left(\frac{\pi}{2}\lambda\right) & 0 & \omega^{\lambda+\mu}\sin\left(\frac{\pi}{2}(\lambda+\mu)\right) \\ 0 & \sin\left(\frac{\pi}{2}\lambda\right) & -\omega^{\lambda+\mu}\sin\left(\frac{\pi}{2}\mu\right) \\ \omega^{\lambda}\sin\left(\frac{\pi}{2}\mu\right) & \sin\left(\frac{\pi}{2}(\lambda+\mu)\right) & 0 \end{bmatrix} \begin{bmatrix} K_p \\ K_i \\ K_d \end{bmatrix} = -\frac{\omega^{\lambda}}{N} \begin{bmatrix} \sin\left(\frac{\pi}{2}\lambda-\gamma\right) \\ \sin(\gamma) \\ \sin\left(\frac{\pi}{2}\mu+\gamma\right) \end{bmatrix}. \tag{31}$$

In this form, only any 2 of the equations are linearly independent at a time and the remaining is linearly dependent on these two. In Eq. (31), each line contains only two of the variables (K_p, K_i, K_d) unlike the second line of Eq. (26). Therefore, it is more convenient to use than Eq. (26) for the programming implementation of the stability surfaces (for the use of the second and third lines of Eq. (31), see Section 4.1). In 3-dimensional Euclid space $R^3 = (K_p, K_i, K_d)$ Eq. (26) or Eq. (31) defines a surface as the parameter ω changes from to ∞. This surface separates the stable and unstable regions with $G_v = g_m e^{-j\varphi_m}$ present in the control loop; further if $G_v = g_m$ ($G_v = e^{-j\varphi_m}$) the stable region guarantees stability with the required gain (phase) margin. If both of the PM and GM specifications are desired to be satisfied simultaneously, $G_v = g_m e^{-j\varphi_m}$ should be chosen. Due to the presence of g_m and φ_m, the stability boundaries so obtained are called the relative stability surfaces. Otherwise, they are called the (absolute or real) stability surfaces.

Infinite root boundary results when $\omega \to \infty$ does not exist for the control system applications [17, 29].

The illustration of finding stability boundaries RRB, CRB and the programming details will be given in Section 4.

3.2. Phase flatness surface and flatness curve

First, the gain crossover frequency condition in Eq. (17) is used together with Eqs. (9) and (11) defining G_c and G_p to obtain the equation that must be satisfied by $\omega = \omega_{gc}$; $G_p(j\omega)$ is written as in Eq. (27), where N and γ are expressed in Eqs. (28) and (29) with $g_m = 1$, $\varphi_m = 0$. Second, the phase flatness condition in Eq. (22) with the equality sign is used together with Eq. (9) and (11), defining

G_c and G_p. Combining these two conditions, the equations describing constant phase flatness surface in 3-dimensional space are obtained as:

$$K_p^2\omega^{2\lambda} + K_d^2\omega^{2(\lambda+\mu)} + K_i^2 + 2K_pK_d\omega^{2\lambda+\mu}\cos(0.5\pi\mu) + 2K_pK_i\omega^{\lambda}\cos(0.5\pi\lambda) + 2K_dK_i\omega^{\lambda+\mu}\cos\left(0.5\pi(\lambda+\mu)\right) = \frac{\omega^{2\lambda}}{N^2}, \tag{32}$$

$$K_pK_i\lambda\sin(0.5\pi\lambda) + K_pK_d\omega^{\lambda+\mu}\mu\sin(0.5\pi\mu) + K_iK_d(\lambda+\mu)\omega^{\mu}\sin\left[0.5\pi(\lambda+\mu)\right] = \frac{\omega^{\lambda+1}}{N^2}\left(\theta - \psi_{\text{spec}} - \frac{\mathrm{d}\beta}{\mathrm{d}\omega}\right), \tag{33}$$

where N, β, γ are defined before and $\dfrac{\mathrm{d}\beta}{\mathrm{d}\omega}$ is derived to be

$$\frac{\mathrm{d}\beta}{\mathrm{d}\omega} = \frac{\omega^{-1}\sum\limits_{k=1}^{n-1}\left[\sum\limits_{h=k+1}^{n} b_kb_h\omega^{\beta_k+\beta_h}(\beta_k-\beta_h)\sin\left[0.5\pi(\beta_k-\beta_h)\right]\right]}{\sum\limits_{k=1}^{n} b_k^2\omega^{2\beta_k} + 2\sum\limits_{k=1}^{n-1}\left[\sum\limits_{h=k+1}^{n} b_kb_h\omega^{\beta_k+\beta_h}\cos\left[0.5\pi(\beta_k-\beta_h)\right]\right]} - \frac{\omega^{-1}\sum\limits_{r=1}^{m-1}\left[\sum\limits_{h=r+1}^{m} a_ra_h\omega^{\alpha_r+\alpha_h}(\alpha_r-\alpha_h)\sin\left[0.5\pi(\alpha_r-\alpha_h)\right]\right]}{\sum\limits_{r=1}^{m} a_r^2\omega^{2\alpha_r} + 2\sum\limits_{r=1}^{m-1}\left[\sum\limits_{h=r+1}^{m} a_ra_h\omega^{\alpha_r+\alpha_h}\cos\left[0.5\pi(\alpha_r-\alpha_h)\right]\right]}. \tag{34}$$

On the face flatness surface $\psi_f = -\psi_{\text{spec}}$ the value of the face flatness in Eq. (22) is exactly satisfied, that is $\psi_f = \psi_{\text{spec}}$. On one side of this surface, the flatness is smaller than the value specified by this equation, and on the other side, greater. In the design of the controller, any point (K_p, K_i, K_d) chosen on this surface or the greater side will fulfill the requirement

$$\frac{\mathrm{d}}{\mathrm{d}\omega}\text{Arg}\left[G_c(j\omega)G_p(j\omega)\right]\Big|_{\omega=\omega_{gc}} \geqslant -\psi_{\text{spec}} \quad \text{(usually 0)} \tag{35}$$

If the gain cross-over frequency ω_{gc} is given, then Eqs. (32) and (33) define a curve in the 3D (K_p, K_i, K_d) space where Eqs. (28), (29) and (30) are evaluated at $\omega = \omega_{gc}$. This curve is defined as the flatness curve on which Eq. (12) is satisfied with equality.

3.3. Disturbance rejection surface

Using Eqs. (9) and (11) for G_c ($s = j\omega_d$) and G_p ($s = j\omega_d$), respectively, the equation of disturbance rejection surface is obtained from Eq. (23) by taking

the equality sign as

$$\left[P + I\cos(0.5\pi\lambda) + D\cos(0.5\pi\mu) + \frac{\omega_d^\lambda}{N}\cos(\gamma)\right]^2 + \left[-I\sin(0.5\pi\lambda) + D\sin(0.5\pi\mu) + \frac{\omega_d^\lambda}{N}\sin(\gamma)\right]^2 = \frac{\omega_d^{2\lambda}}{N^2 B_d^2}, \tag{36}$$

where $G_p(j\omega_d) = Ne^{j\gamma}$; $N(\omega_d)$, $\gamma(\omega_d)$ are defined in Eqs. (28), (29), (30) with $g_m = 1$, $\varphi_m = 0$; and

$$P = K_p\omega_d^\lambda, \quad I = K_i, \quad D = K_d\omega_d^{\lambda+\mu}. \tag{37}$$

Output disturbance rejection specification is satisfied on the side

$$[\cdot]^2 + [\cdot]^2 \geqslant \frac{\omega_d^{2\lambda}}{N^2 B_d^2} \text{ for all } \omega \leqslant \omega_d, \tag{38}$$

of this surface where the term in the parenthesis represents the left-hand sides of Eqs. (36).

3.4. High-frequency noise rejection surface

High-frequency noise rejection condition in Eq. (24) together with Eqs. (9) and (11) for G_c $(s = j\omega_n)$ and G_p $(s = j\omega_{dn})$, respectively, yield

$$\frac{C_n^2 - 1}{C_n^2}\left[P^2 + I^2 + D^2 + 2PI\cos(0.5\pi\lambda) + 2PD\cos(0.5\pi\mu) + 2ID\cos\{0.5\pi(\lambda+\mu)\}\right] + \frac{2\omega_n^\lambda}{N}\left[P\cos(\gamma) + I\cos(0.5\pi\lambda - \gamma) + D\cos(0.5\pi\mu + \gamma)\right] + \frac{\omega^{2\lambda}}{N^2} \geqslant 0. \tag{39}$$

Or, assuming $|C_n| \leqslant 1$, which is usually the case,

$$\left[P^2 + I^2 + D^2 + 2PI\cos(0.5\pi\lambda) + 2PD\cos(0.5\pi\mu) + 2IDcos\{0.5\pi(\lambda+\mu)\}\right] + \frac{2\omega_n^\lambda C_n^2}{N(C_n^2 - 1)}\left[P\cos(\gamma) + I\cos(0.5\pi\lambda - \gamma) + D\cos(0.5\pi\mu + \gamma)\right] + \frac{\omega_n^{2\lambda} C_n^2}{N^2(C_n^2 - 1)} \leqslant 0. \tag{40}$$

In Eqs. (39) and (40), $G_p(j\omega_n) = Ne^{j\gamma}$; N, γ, P, I, D are defined as in Eqs. (28), (29), (37), all with $g_m = 1$, $\varphi_m = 0$, $\omega = \omega_n$.

Note that the direction of the inequality sign in Eq. (40) should be reversed if $|C_n| < 1$; further, if the equality sign is taken instead of the inequality sign, this equation defines a surface (constant noise rejection surface C_n at frequency ω_n). If $\omega \geqslant \omega_n$, Eq. (24) will be satisfied on the side of this surface, which will be defined in Section 4.

In Eqs. (36), (39) and (40), P, I, D are used instead of K_p, K_i, K_d for programming easiness, see Section 4.

4. 3D programming implementations

In this section, programming details for the 3D plots of stability, flatness, disturbance and noise surfaces as well as the flatness curve, are illustrated. The basic ideas for the preparation of the related function subprograms and their use are illustrated. The programs are prepared by using MATLAB language.

4.1. Stability surfaces

Consider Eq. (31) for CRB. For a mesh-grid structure for the variables K_p, K_i. The third line in Eq. (31) is

$$f\left[\left(K_p, K_i\right), \omega\right] = K_p\omega^{\lambda}\sin(0.5\pi\mu) + K_i \sin\left[0.5\pi(\lambda+\mu)\right] + \frac{\omega^{\lambda}}{N}\sin(0.5\pi\lambda+\gamma) = 0 \tag{41}$$

should be satisfied for each grid point (K_p, K_i). When the frequency ω is scanned with convenient steps $\Delta\omega$ over a reasonable range and $f\left[(K_p, K_i), \omega\right]$ is computed for each frequency, the frequencies satisfying Eq. (41) can be found by checking

$$f\left[\left(K_p, K_i\right), \omega_{n-1}\right] \cdot f\left[\left(K_p, K_i\right), \omega_n\right] \leqslant 0. \tag{42}$$

Whenever this equation is satisfied, the solution of Eq. (41) for ω is thus found for the grid point $(K_p,\ K_i)$ within an error $\pm\Delta\omega$. Then the second line of Eq. (31) can be used to find K_d as

$$K_d = \frac{\omega^{-\lambda}K_i\sin(0.5\pi\lambda) + \dfrac{\sin(\gamma)}{N}}{\omega^{\mu}\sin(0.5\pi\mu)} \tag{43}$$

which is the point on the stability surface corresponding to $(K_p,\ K_i)$. A 3D surface plot is obtained by the MATLAB function "mesh".

RRB $K_i = 0$ is easy to plot since it is simply (K_p, K_d) plane.

We note that due to the transcendental nature of the function $f\left[\left(K_p, K_i\right), \omega\right]$, we may have more than one solution. Each solution is considered as a separate

stability surface and checked by some test points to decide which side is stable in the programming.

4.2. Flatness surface

Consider Eqs. (32) and (33). For a mesh-grid structure for the variables K_p, K_i, Eq. (33) is solved for K_d:

$$K_d(K_p, K_i, \omega) = \frac{\dfrac{\omega^{\lambda+\mu}}{N^2}\left(\theta - \psi_{\text{spec}} - \dfrac{\mathrm{d}\beta}{\mathrm{d}\omega}\right) - K_p K_i \lambda \sin(0.5\pi\lambda)}{K_p \omega^{\lambda+\mu}\mu \sin(0.5\pi\mu) + K_i(\lambda+\mu)\omega^{\mu} \sin\left[0.5\pi(\lambda+\mu)\right]}. \quad (44)$$

Substitute this K_d in Eq. (32) to obtain

$$\begin{aligned} f_f\left[\left(K_p, K_i\right), \omega\right] &= K_d^2\left(K_p, K_i, \omega\right)\omega^{2(\lambda+\mu)} \\ &+ K_d\left(K_p, K_i, \omega\right)\left[2K_p\omega^{2\lambda+\mu}\cos(0.5\pi\mu) + 2K_i\omega^{\lambda+\mu}\cos\left(0.5\pi(\lambda+\mu)\right)\right] \\ &+ K_p^2\omega^{2\lambda} + K_i^2 + 2K_pK_i\omega^{\lambda}\cos(0.5\pi\lambda) - \frac{\omega^{2\lambda}}{N^2} = 0. \end{aligned} \quad (45)$$

This equation should be satisfied for each grid point (K_p, K_i). When the frequency ω is scanned with convenient steps $\Delta\omega$ over a reasonable range, the frequencies satisfying Eq. (45) can be found by checking

$$f_f\left[\left(K_p, K_i\right), \omega_{n-1}\right] \cdot f_f\left[\left(K_p, K_i\right), \omega_n\right] \leqslant 0. \quad (46)$$

Whenever this equation is satisfied, the solution of Eq. (45) for ω is thus found for the grid point (K_p, K_i) within an error $\pm\Delta\omega$. Then Eq. (44) can be used to find K_d which is the point on the flatness surface corresponding to (K_p, K_i). 3D Surface plot is obtained by the MATLAB function "mesh". We note that due to the transcendental nature of the function $f_f\left[\left(K_p, K_i\right), \omega\right]$ in Eq. (45), we may have more than one solution. Each solution is considered as a separate flatness surface and at most, four flatness surfaces are considered in the programming.

4.3. Flatness curve

When the GCOF ω_{gc} is given and the flatness is specified at this frequency as in Eq. (22) (with an equality sign), then Eqs. (32) and (33) define the flatness curve in the space (K_p, K_i, K_i) as already indicated in Section 3.2. To find the flatness curve, we draw K_i from Eq. (33) to obtain

$$K_i = \frac{\dfrac{\omega^{\lambda+\mu}}{N^2}\left(\theta - \psi_{\text{spec}} - \dfrac{\mathrm{d}\beta}{\mathrm{d}\omega}\right) - K_d K_p \omega^{\lambda+\mu}\mu \sin(0.5\pi\mu)}{K_p\lambda \sin(0.5\pi\lambda) + K_d(\lambda+\mu)\omega^{\mu} \sin\left[0.5\pi(\lambda+\mu)\right]}. \quad (47)$$

Substituting this in Eq. (32) and rearranging, we obtain the following fourth-order polynomial equation in K_d:

$$q_4 K_d^4 + q_3 K_d^3 + q_2 K_d^2 + q_1 K_d + q_0 = 0, \tag{48}$$

where

$$q_4 = ag^2, \quad q_3 = g(2fa + hc + bg),$$

$$q_2 = af^2 + fhc + 2gfb + eg^2 + h^2 + cHg + dhg,$$

$$q_1 = 2hH + gdH + fcH + fdh + bf^2 + 2gef, \quad q_0 = H^2 + fdH + ef^2,$$

$$a = \omega^{2\lambda+2\mu}, \quad b = 2K_p\omega^{2\lambda+\mu}\cos(0.5\pi\mu), \quad c = 2\omega^{\lambda+\mu}\cos\left[0.5\pi(\lambda+\mu)\right],$$

$$d = 2K_p\omega^{\lambda}\cos(0.5\pi\lambda), \quad e = K_p^2\omega^{2\lambda} - \frac{\omega^{2\lambda}}{N^2}, \quad f = K_p\lambda\sin(0.5\pi\lambda),$$

$$g = \omega^{\mu}(\lambda+\mu)\sin\left[0.5\pi(\lambda+\mu)\right], \quad h = -K_p\omega^{\lambda+\mu}\mu\sin(0.5\pi\mu),$$

$$H = \frac{\omega^{\lambda+\mu}}{N^2}\left(\theta - \psi_{\text{spec}} - \frac{\mathrm{d}\beta}{\mathrm{d}\omega}\right).$$

Choose a set of successive values of K_p, find the roots of Eq. (48) for each value of K_p, and then the corresponding values of K_i by using Eq. (47). Then, obtain 3D plot of (K_p, K_i, K_d) by using MATLAB function "plot3". The function subprogram prepared for computing the flatness curve is done by "FlatnsCurve.m and flatnessC.m" and the results are shown in Section 5 "Examples". In general, Eq. (48) has four roots and only real roots of it are considered. Hence there may be more than one flatness curve, each of which must be justified.

4.4. Disturbance rejection surface

Consider Eq. (36) to find the surface on which the disturbance specification is satisfied with an equality sign. Then the disturbance specification will be satisfied on and on one side of this surface as implied by Eq. (38). Eq. (36) can be arranged as a quadratic equation in the parameter D as follows:

$$D^2 + 2D\left\{P\cos(0.5\pi\mu) + I\cos\left[0.5\pi(\lambda+\mu)\right] + \frac{\omega_d^{\lambda}}{N}\cos(0.5\pi\mu+\gamma)\right\}$$
$$+ \left\{P^2 + I^2 + 2PI\cos(0.5\pi\lambda) + \frac{2\omega_d^{\lambda}}{N}\left[P\cos(\gamma) + I\cos(0.5\pi\lambda-\gamma)\right]\right.$$
$$\left. + \frac{\omega_d^{2\lambda}}{N^2}\left(1 - \frac{1}{B_d^2}\right)\right\} = 0, \tag{49}$$

where N, γ, P, I, D are as defined in Eqs. (28), (29) and (37) with $\omega = \omega_d$. Equation (49) is solved for each grid point of K_p, K_i. Then Eq. (37) is used to find $K_d = D/\omega_d^{(\lambda+\mu)}$. If a real root exists, then there are two solutions for D and hence for K_d, and the permissible region to satisfy Eq. (38), or Eq. (49) with $\geqslant$ is outside of these roots. This corresponds to the outside of the two disturbance surfaces in the 3D implementation obtained by the MATLAB "mesh" function. The evaluation of the disturbance surfaces is conducted by the program "distbound.m" and shown by the examples given in Section 5.

4.5. Noise rejection surface

Noise rejection surface is defined by Eqs. (39) or (40) with the equality sign is taken. Its equation is rewritten below in quadratic form in the parameter D.

$$\begin{aligned} D^2 + 2D\Bigg\{ & P\cos(0.5\pi\mu) + I\cos\left[0.5\pi(\lambda+\mu)\right] \\ & + \frac{\omega_n^{\lambda} C_n^2}{N\left(C_n^2-1\right)}\cos(0.5\pi\mu+\gamma)\Bigg\} + P^2 + I^2 + 2PI\cos(0.5\pi\lambda) \\ & + \frac{\omega_n^{\lambda} C_n^2}{N\left(C_n^2-1\right)}\left[2P\cos(\gamma) + 2I\cos(0.5\pi\lambda-\gamma) + \frac{\omega_n^{\lambda}}{N}\right] = 0, \end{aligned} \tag{50}$$

where N, γP, I, D are as defined in Eqs. (28), (29), (37) with $\omega = \omega_n$. Equation (50) is solved for each grid point of K_p, K_i. Then Eq. (37) is used to find $K_d = D/\omega_n^{(\lambda+\mu)}$. If a real root exists, then there are two solutions for D and hence for K_d, and the permissible region to satisfy Eq. (39) is inside of these roots (the case $|C_n| \leqslant 1$, otherwise outside). This corresponds to the inside of the two noise surfaces in the 3D implementation obtained by the MATLAB "mesh" function. The evaluation of the noise surfaces is conducted by MATLAB program "noisebound.m" and the results are demonstrated in the next section.

4.6. Performance characteristics

For any example with the FOPID controller and a rational FO plant transfer function with time delay, some performance characteristics are calculated and/or plotted to verify the design results. This is done by the prepared function subprogram "performance". The outputs of this program are: noise rejection variation with ω, Disturbance rejection variation with ω, PF variation with ω; further, Nichols chart, polar plot, Nyquist plot, gain and phase plots (Bode diagrams) all for open-loop gain; continuing, controller transfer function G_c, Plant transfer function G_p, open loop transfer function G_cG_p, gain margins GM, g_m (in dB,

and normal), phase margins PM, φ_m (in degrees and radians), phase crossover frequency ω_{pc}, and finally, gain crossover frequency ω_{gc}.

5. Examples

Examples presented in this section are taken as benchmark problems chosen from the literature. Each one of the examples clarifies the theory and design specifications considered in Section 2, demonstrates the validity of formulation aspects in Section 3, and finally, the usability of programming aspects of 3D plots in Section 4.

5.1. Example 1

Consider the experimental platform controlling liquid level in Basic Process Ring 38-100 Feedback Unit treated [51]. The plant (liquid level system) is characterized by the transfer function

$$G_p(s) = \frac{3.13}{433.33s + 1} e^{-50s}. \tag{51}$$

Design specifications are given as: GCOF $\omega_{gc} = 0.008$ rad/s, PM $\varphi_m = \pi/3$, robustness to variations in the gain of the plant must be fulfilled ($\psi_{\text{spec}} = 0$ s at ω_{gc}), sensitivity function $\leqslant B_d = -20$ dB for $\omega \leqslant \omega_d = 0.001$ rad/s noise rejection $\leqslant C_n = -20$ dB for $\omega \geqslant \omega_n = 10$ rad/s. In [51], the function FIMCON in the optimization toolbox of MATLAB is used to find the FOPID controller transfer function

$$G_c = 0.6152 + \frac{0.01}{s^{0.8968}} + 4.3867 s^{0.4773}. \tag{52}$$

Performance characteristics:

The control system performance is analyzed by the subprogram "performance" (described in Section 4.6); the elapsed and CPU times are recorded as 1.6571 s and 1.7188 s, respectively. The realized performance characteristics satisfy the specified ones as seen in Table 1. Especially the ones indicated by '+' sign in the comment column, fulfill the given design specifications. The performance characteristics are illustrated by the plots shown in Figs. 4–10.

Oustaloup continuous approximation [53] is used with order $N = 5$, $\omega_b = 10^{-4}$, $\omega_h = 10^4$ rad/s for evaluation of step response by the FO Simulink blocks which appear in the FOTF Toolbox for FO system analyzer in MATLAB [73]. The function subprograms in this toolbox are also used for programming implementations presented in Section 4. The resulting step response is shown in Fig. 10, and it has an overshoot of 14.3%.

Table 1: Performance characteristics of PI$^{\lambda}$D$^{\mu}$ controlled liquid level system

Characteristics	Specification (*Not specified)	Realized	Comments: –: No comment =: Exact value, +: Satisfied value
ω_{gc} (rad/s)	0.008	0.008	=
PM (degrees) φ_m (rad)	60 1.0472	60.0808 1.0486	=
ψ_{spec} (s)	0	−0.0124	+
B_d B_d (dB)	0.1 −20	0.0706 −23.0178	+
ω_d (rad/s)	0.001	0.001	=
C_n C_n (dB)	0.1 −20	0.0099 −40.1217	+
ω_n (rad/s)	10	10	=
ω_{pc} (rad/s)	*	0.0392	–-
GM (dB) g_m	*	11.7541 3.8699	–-
Overshoot (%)	*	14.3	(found by Simulink)
Rise time (s)	*	50 + 138	(from 10 % to 90%)
Settling time (s)	*	50 + 587	(95–105% of the final value)
Elapsed time	*	1.6571 s	–-
CPU time	*	1.7188 s	–-

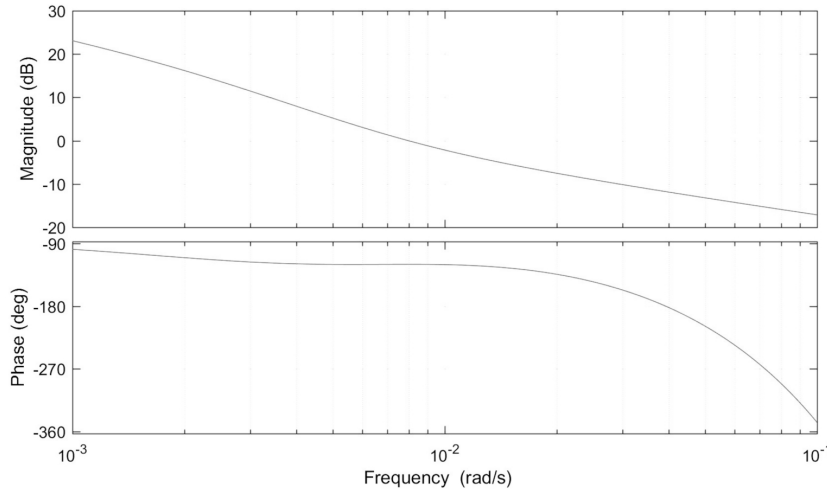

Figure 4: Bode diagram of Example 1

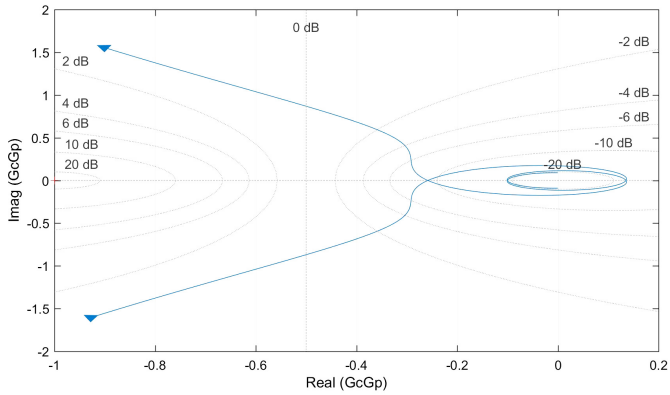

Figure 5: Nyquist plot of Example 1

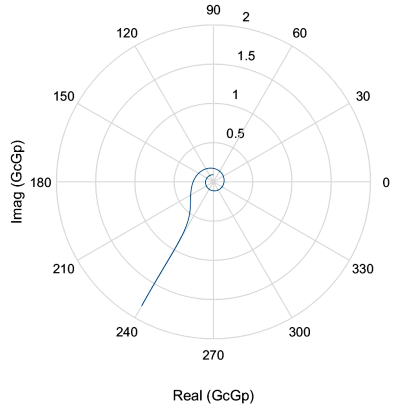

Figure 6: Polar plot of Example 1

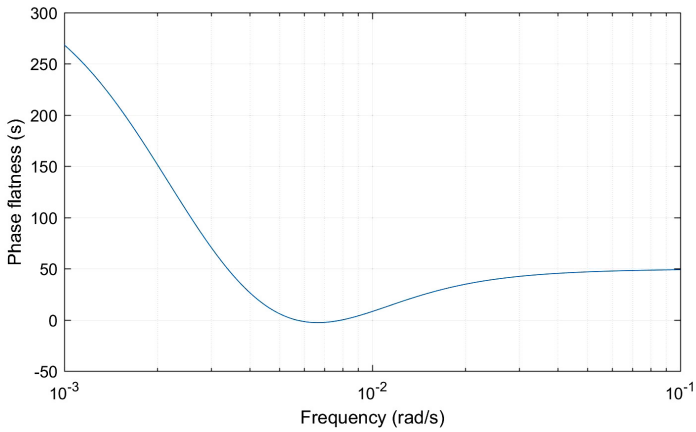

Figure 7: Phase flatness curve of Example 1

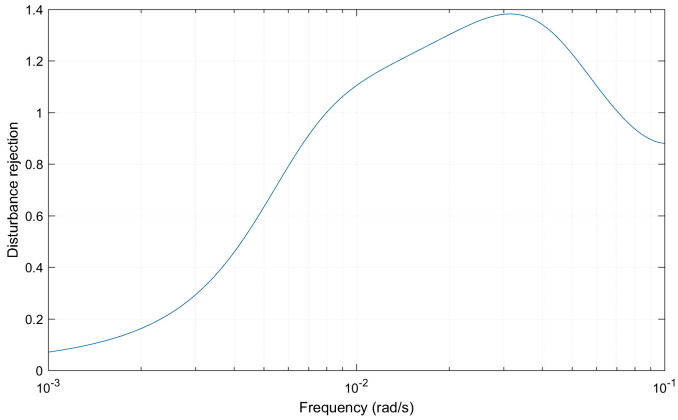

Figure 8: Disturbance rejection curve of Example 1

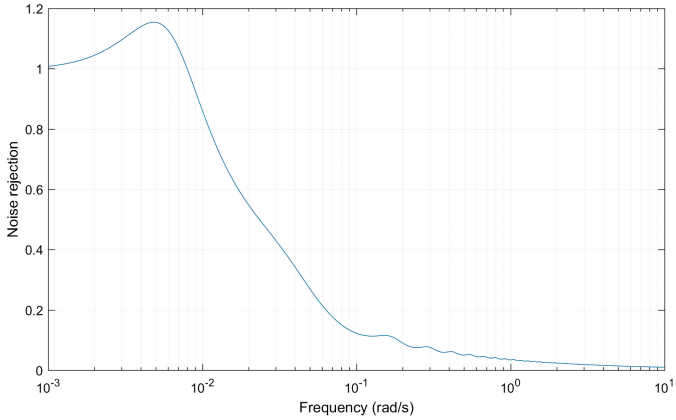

Figure 9: Noise rejection curve of Example 1

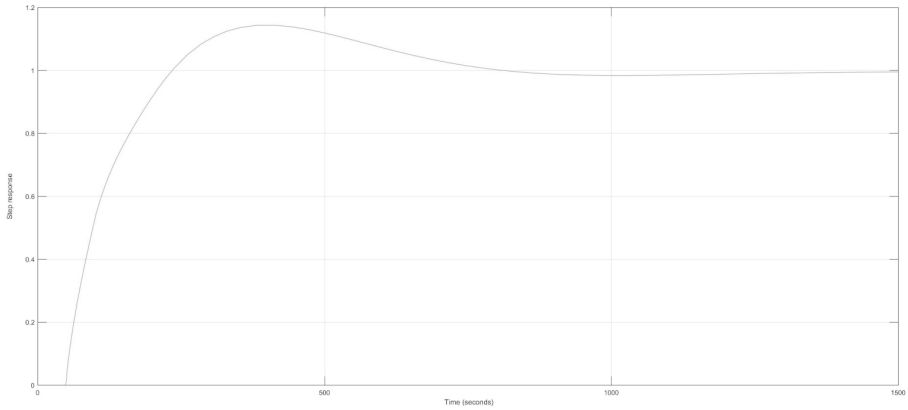

Figure 10: Step response of the closed-loop system of Example 1

Disturbance and noise rejection boundaries:

The disturbance rejection requirement is illustrated by the two boundary surfaces obtained by using MATLAB programing and the result of the 3D plot is shown in Fig. 11. It is observed in this figure that the point TP1 = (K_p, K_i, K_d) with the designed parameters of PI$^{\lambda}$D$^{\mu}$ controller (K_p = 0.6152, K_i = 0.01, K_d = 4.3867) is in the permissible region. Hence the disturbance rejection specification is satisfied. In fact, in Table 1, the transfer function $|T_D|$ = 0.0706, which is less than the specified limit B_d = 0.1 at ω_d = 0.001 rad/s, and $|T_D| \leqslant 0.1$ for all $\omega \leqslant \omega_d$. In Fig. 11, TP1 is the designed point. When the test point TP2 is taken between the boundary surfaces DBU and DBL as TP2 = (K_p = 0.6, K_i = 0.002, K_d = 25), then the disturbance rejection specification will not be satisfied as expected; in fact, it is calculated that $|T_D| = 0.2464 \geqslant 0.1$ at ω_d = 0.001 rad/s.

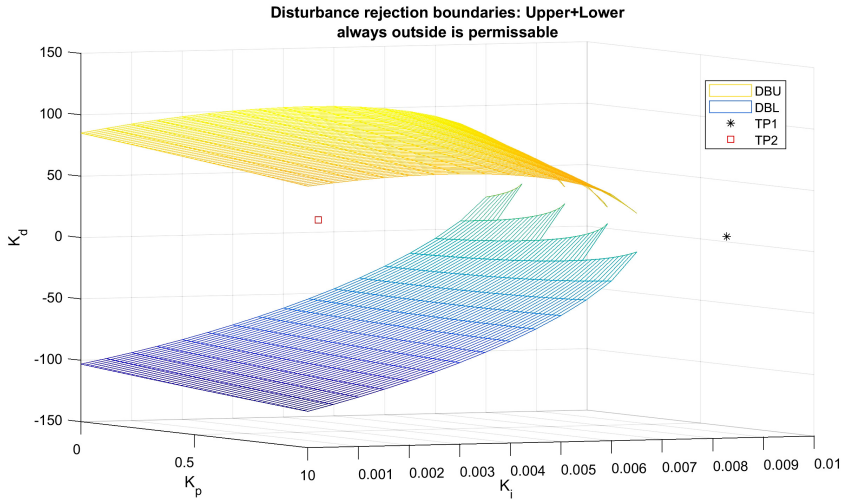

Figure 11: Disturbance rejection boundaries and the test points of Example 1

When the noise specification is considered, Table 1 shows that noise rejection is well satisfied with the chosen controller parameters. In Fig. 12, noise rejection boundaries in 3-dimensional (K_p, K_i, K_d) space, which are obtained by using the prepared MATLAB function and is shown with the test point TP1 of the designed controller. When the controller parameters are changed to K_p = 0.9, K_i = 0.01, K_d = 55, which corresponds to the test point TP2, this point is outside of the permissible region as seen in Fig. 12; then, the noise transfer function takes the value $|T_N(j\omega_n)|$ = 0.1226 (0.1144), which is obviously greater than C_n = 0.1, so the noise rejection specification is violated at TP2.

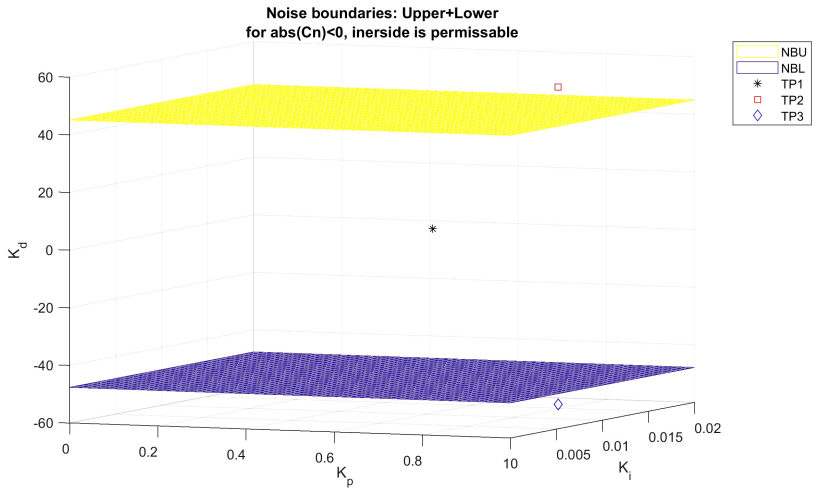

Figure 12: Noise rejection boundaries and the test points of Example 1

A similar result is observed for TP3 = ($K_p = 0.9$, $K_i = 0.01$, $K_d = -55$) where $|T_N(j\omega_n)| = 0.1144 > C_n = 0.1$.

Phase flatness curves and surfaces:

Robustness to variations in the gain of the plant is satisfied by the phase flatness curve $\psi_f = 0$ at ω_{gc}. By choosing $\psi_{\text{spec}} = 0$ at $\omega_{gc} = 0.008$ rad/s in Eq. (22), the flatness curves are obtained as shown in Fig. 13. Obviously, the designed parameters fulfill this requirement (see the test point TP1). In fact $\psi_f = -0.0124 \cong 0$ at $\omega_{gc} = 0.008$ rad/s and TP1 is almost on this curve. In the same figure, the phase flatness surface $\psi_f = 0$ without restricting the gain crossover frequency ω_{gc} are also shown. It is naturally true that the phase flatness curve is on this surface, as seen in the figure. Consider the two test points TP2 = ($K_p = 0.2$, $K_i = 0.018$, $K_d = 10$) and TP3 = ($K_p = 0.5$, $K_i = 0.01$, $K_d = 18$), which are inside and outside of the phase flatness boundaries. At TP3, it is found that $\psi_f = 41.9616$ s $\gg 0$ at $\omega_{gc} = 0.0093$, which is far away from $\psi_{\text{spec}} = 0$. At TP2, it is found that $\psi_f = -48.8777$ s $\ll 0$ at $\omega_{gc} = 0.0281$, which is far away from $\psi_{\text{spec}} = 0$ and being much negative, the relative stability is poor, and the system may go to instability easily with uncertain plant parameters. So, the outer region of the phase flatness boundary is hardly desired design area. Note that the flatness curve data (FlatC1) is not in the visible range of the axis, and the phase flatness surfaces F3 and F4 do not exist for this example.

Complex root boundaries are plotted for a $GM = 11.7541$ dB ($g_m = 3.8699$) as realized by design in [51] (see Table 1). The virtual phase margin φ_m is

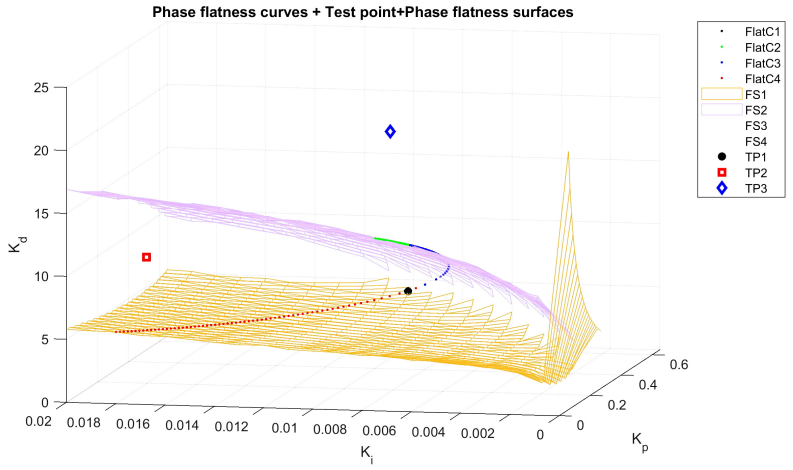

Figure 13: Phase flatness curves and surfaces, and the test points

assumed zero. As expected, due to the delay term of the plant transfer function and the transcendental behavior of the CRB equations treated in Section 3.1 and the possibility of the existence of multiple solutions of some mathematical operation, there are multiple solutions for the boundary surface. Note that the designed system becomes unstable when the open-loop gain is multiplied by the GM that is by $g_m = 3.8699$, which causes the Nyquist plot to pass from the critical point $(-1, 0)$. Hence, the test point TP1 should be on the CRB; this is in fact, so as seen on the red surface CRB2 in Fig. 14.

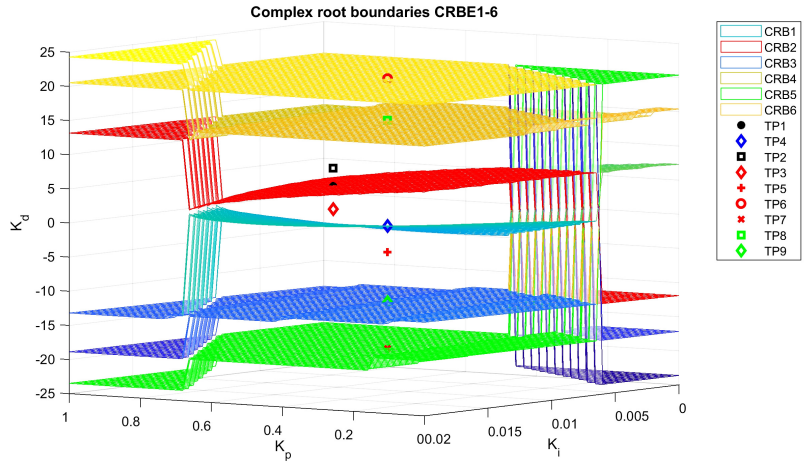

Figure 14: Complex root boundaries and the test points of Example 1

In this figure, some other test points are indicated:

TP2 (0.6152, 0.01, 7) and TP3 (0.6152, 0.01, 1) are chosen above and below of CRB2; it is checked by performance.m that the GMs in dB at these test points are 9.3433 < 11.7541 and 14.5091 > 11.7541, respectively. So, CRB2 is the true stability boundary for achieving the relative stability of GM (11.7541 dB).

Considering CRB1, the test points TP4 (0.48, 0.0095, 1.1102) chosen on this surface and the associated GM is computed as 11.7471 ≅ 11.7541, hence CRB1 is another true CRB; in upperside and lower side of this surface GMs at TP3 and TP5 (0.48, 0.0095, −5), GMs are computed as 14.5091 > 11.7541 and 1.9726 < 11.7541 dB, respectively, which supports the trueness of CRB1.

It deserves to note that the test points TP6 (0.48, 0.0095, 20.43) on CRB6, TP7 (0.48, 0.0095, −19.0772) on CRB5, TP8 (0.48, 0.0095, 14.2996) on CRB4, TP9 (0.48, 0.0095, −12.2779) on CRB3 do not yield the correct $GM = 11.7541$ dB nor its approximate value. Hence the CRBs CRB2-CRB6 are not true relative stability surfaces associated with this GM. The reason is mentioned before, and one must check the trueness of the computed CRBs by using some test points. The use of test points to eliminate unwanted results originating from multiple solutions of some mathematical problems is a common trend in studying FOS [29].

5.2. Example 2

Consider the benchmark problem treated as Example 2 in [75], FO plant model is given as [58]

$$G_p = \frac{1}{0.8s^{2.2} + 0.5s^{0.9} + 1}. \tag{53}$$

According to the proposed method in [75], it is selected that $\lambda = 0.1$ and $\mu = 1.15$ to achieve a GM $g_m = 1.3$ (2.2789 dB) and PM $\varphi_m = \pi/3\,(60^\circ)$, then the controller parameters are determined as $K_p = 233.4234$, $K_i = 22.3972$, $K_d = 18.5274$. Thus, the controller transfer function is

$$\begin{aligned} G_c &= 233.4234 + \frac{22.3972}{s^{0.1}} + 18.5274s^{1.15} \\ &= \frac{18.5274s^{1.25} + 233.4234s^{0.1} + 22.3972}{s^{0.1}}. \end{aligned} \tag{54}$$

Performance characteristics:

The performance characteristics of the designed control system are shown in Table 2. The elapsed and CPU times for running the subprogram “performance” are recorded as 3.1323 and 4.1719 s, respectively. The realized performance

characteristics are satisfying the specified PM and GM as indicated by '+' in the last column of Table 2.

Table 2: Performance characteristics of $PI^{\lambda}D^{\mu}$ controlled system of Example 2

Characteristics	Specification (*Not specified)	Realized	Comments: –: No comment =: Exact value, +: Satisfied value !: Can be improved
ω_{gc} (rad/s)	*	19.8601	–-
PM (degrees) φ_m (rad)	60° 1.0472	60.9404 1.0636	+
ψ_f (s)	*	−0.0244	!
B_d B_d (dB)	*	0.3455 −9.2306	!
ω_d (rad/s)	*	8	–-
C_n C_n (dB)	*	0.2657 −11.5114	!
ω_n (rad/s)	*	70	–-
ω_{pc} (rad/s)	*	not defined	–-
GM (dB) g_m	2.2789 1.3	∞ ∞	+
Overshoot (%)	*	25.2	(found by Simulink)
Rise time (s)	*	0.0707	(from 10 % to 90%)
Settling time (s)	*	0.4860	(95–105% of the final value)
Elapsed time (s)	*	3.1323	–-
CPU time (s)	*	4.1719	–-

Let us first check the realized parameters φ_m, ψ_f, B_d, C_n, g_m shown in Table 2 by verifying the 3D plots. The stability boundaries, flatness curves and surfaces, disturbance and noise boundaries together with the test point TP0 = (K_p = 233.4234, K_i = 22.3972, K_d = 18.5274) obtained with g_m = 1, φ_m = 60.9404°, ω_{gc} = 19.8601, ψ_{spec} = −0.0244 s, B_d = 0.3455 at ω_d = 8, C_n = 0.2657 at ω_n = 70 are shown in Figs. 15, 16, 17, 18, respectively.

Table 3 shows the PMs and GMs calculated at some test points. It is seen that complex root boundary CRBE1, satisfies the property of PM crossing of 60.9404°, that is the test point TP0 is on this surface; note that, in Table 3, the phase at TP2 is above 60.9404° (64.2633°) whilst TP9 and TP10 result with PMs below 60.9404° (31.6260°, 58.2803°).

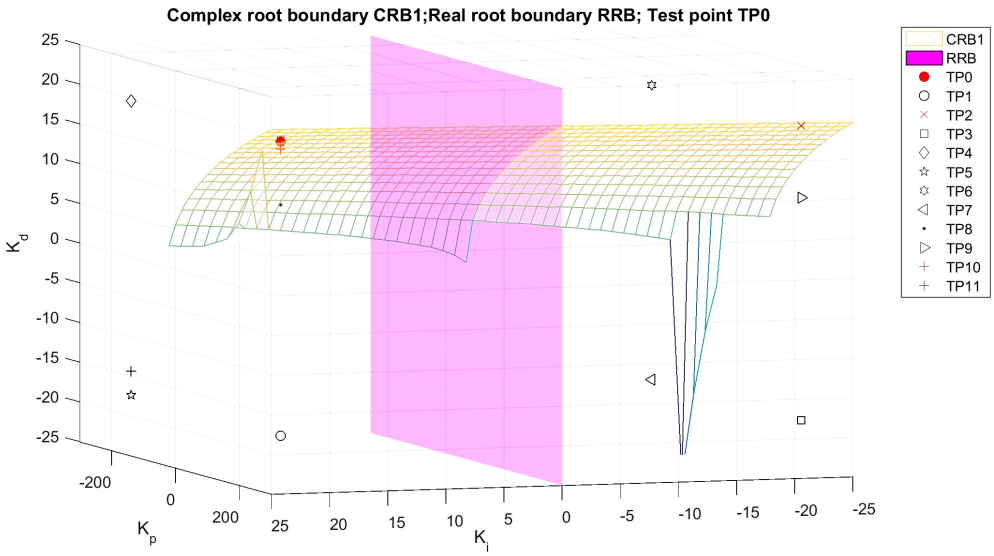

Figure 15: Stability boundaries and the test points for Example 2

Table 3: Stability characteristics of test points for Example 2

Test point	K_p	K_i	K_d	Marker	PM (degrees)	GM (dB)	Stability
TP0	K_{p0} = 233.4234	K_{i0} = 22.3972	K_{d0} = 18.5274	*	60.9404	∞	S
TP1	K_{p0}	K_{i0}	$-K_{d0}$	o	−75.8054	−50.3260	U
TP2	K_{p0}	$-K_{i0}$	K_{d0}	x	64.2633	∞	S
TP3	K_{p0}	$-K_{i0}$	$-K_{d0}$	$\sqcap$	−77.3033	−48.6125	U
TP4	$-K_{p0}$	K_{i0}	K_{d0}	Diamond	102.6967	−46.0955	U
TP5	K_{p0}	K_{i0}	$-K_{d0}$	Pentagram	−115.7367	−46.1052	U
TP6	K_{p0}	K_{i0}	K_{d0}	Hexagram	104.1946	∞	U
TP7	K_{p0}	K_{i0}	$-K_{d0}$	< |	−119.0596	∞	U
TP8	K_{p0}	K_{i0}	10.5274	.	31.8576	−19.9537	U
TP9	K_{p0}	K_{i0}	9.5274	>	31.6260	−23.1595	U
TP10	K_{p0}	K_{i0}	17.5274	+	58.2803	∞	S
TP11	K_{p0}	K_{i0}	−15.5274	+	−123.8282	−46.1044	U

Having the positive GMs and PMs, test points TP0, TP2, TP10 correspond to a stable system, which is also verified by obtaining the step responses shown in

Fig. 16. All the other test points correspond to unstable systems. Since PM for TP10 $< 60°$, the only test points shown in Table 3 remaining in the feasible stable region (stable region satisfying the given PM specification) are TP0 and TP2. The overshoots are measured as 26.9, 24.5% and 28.5% for TP0, TP2, TP10, respectively. This similarity between the overshoots is expected due to near PMs of values 60.9404°, 64.2633°, 58.2803°, respectively.

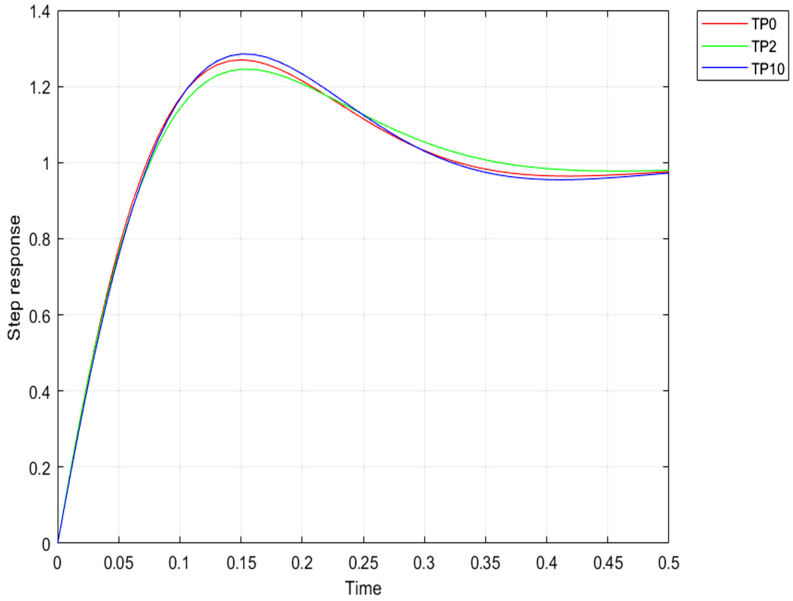

Figure 16: Step responses for the test points TP0, TP2, TP10 of Example 2

Phase flatness curves and surfaces:

Flatness curves and surfaces are shown in Figs. 17 and 18, which include some critical test points at which the relevant performance characteristics are shown in Table 4. The time properties are elapsed time = 29.7287 s, CPU time = 48.6250 s.

The following observations are pointed out about the flatness plots:

i. On all the flatness curves FlatC1–FlatC4, $\psi_f = 0.0244$, which is the case it should be (see TP0-TP12).

ii. On all the flatness surfaces FS1–FS2, $\psi_f = 0.0244$ within three significant digits (see TP13, TP14). Since these points are not on the flatness curves $\omega_{gc} \neq 19.86$, it is 12.8061 (24.2674) at TP13 (TP14).

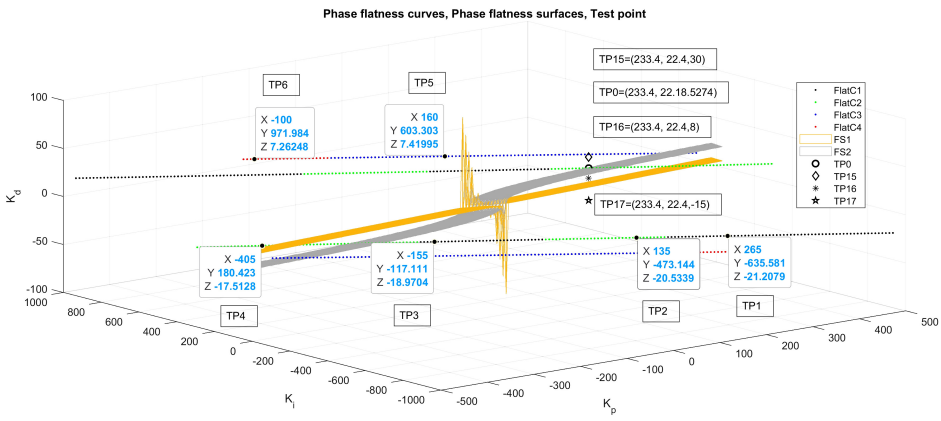

Figure 17: Phase flatness curves and surfaces, and the test points TP0–TP6, TP15–TP17 of Example 2

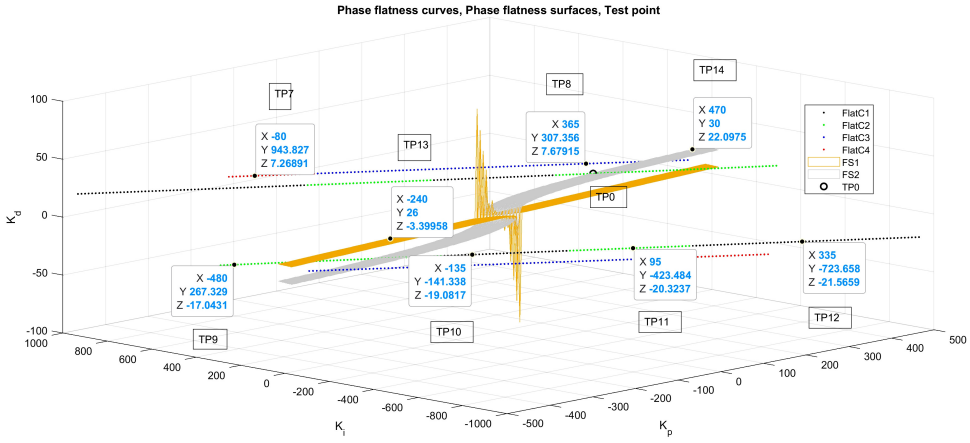

Figure 18: Phase Flatness curves and surfaces, and the test points TP0 and TP7–TP14 of Example 2

iii. In particular, the test point TP0 lies on the flatness curve (FlatC2) and the flatness surface FS2; further the flatness curve FC2 is on the flatness surface FS2. Since the curves and surfaces are not drawn for the same range of the controller parameters, the complete coincidence cannot be observed on the figures.

iv. When the test point is out of the flatness curves and because of this on the flatness surfaces, the phase flatness deviates from $\psi_f = 0.0244$; for example,

Table 4: Phase flatness characteristics of some test points of Example 2

Test point	K_p	K_i	K_d	ω_{gc}	$-\psi_f$	GM (dB)	PM (degrees)	Stability
TP0	233.4234	22.3972	18.5274	19.8601	0.0244	∞	60.9404	S
TP1	265	−635.6	−21.21	19.8625	0.0244	−13.5713	−112.0665	U
TP2	135	−473.1	−20.53	19.8563	0.0244	−13.6652	−113.6843	U
TP3	−155	−117.1	−18.97	19.8600	0.0244	∞	−117.8024	U
TP4	−405	180.4	−17.51	19.8583	0.0244	−46.6953	−122.2076	U
TP5	160	603.3	7.42	19.8602	0.0244	1.1178	−1.6409	U
TP6	−100	972	7.262	19.8604	0.0244	4.1339	−6.5587	U
TP7	−80	943.8	7.269	19.8599	0.0244	3.9307	−6.2016	U
TP8	365	307.4	7.679	19.8607	0.0244	−2.0696	2.8195	S
TP9	−480	267.3	−17.04	19.8582	0.0244	−47.4766	−123.7497	U
TP10	−135	−141.3	−19.08	19.8588	0.0244	∞	−117.4902	U
TP11	95	−423.5	−20.32	19.8565	0.0244	−151.3584	−114.2098	U
TP12	335	−723.7	−21.57	19.8647	0.0244	−109.6292	−111.2481	U
TP13	−240	26	−3.4	12.8061	0.0244	−46.1883	−179.2739	U
TP14	470	30	22.1	24.2674	0.0245	−27.8106	53.4259	S
TP15	233.4	22.4	30	20.1367	−0.0198	−49.7082	−69.5048	U
TP16	233.4	22.4	8	14.0624	0.0461	−14.0027	20.0876	S
TP17	233.4	22.4	−15	20.1367	−0.0198	−49.7082	−69.5048	U

see the test points TP15, TP16, TP17, at which $\psi_f = -0.0198 < 0.0244$, $= 0.0461 > 0.0244$, $= -0.0198 < 0.0244$, respectively. That is face flatness requirement is satisfied between the face flatness surfaces FS1 and FS2, and it is not satisfied outside.

Disturbance rejection boundaries:

Disturbance rejection boundaries including the test points tabulated in Table 5 are shown in Fig. 19. It is seen that the test point TP0 at which the disturbance is $B_d = 0.3455$ (−9.2306 dB) at $\omega_d = 8$ rad/s is on the disturbance boundary as expected. The disturbances at test points TP1, TP2, TP7, which are outside the boundary defined by the disturbance boundary surfaces DBU and DBL are computed as −12.8050, −11.1968, and −11.5875 dB, which are all less than −9.2306 dB. On the other hand, the disturbances at the test points TP3, TP4, TP5, TP6, which are all on the inner side of the disturbance boundary surfaces

Table 5: Disturbance characteristics of some test points for Example 2

Test point	K_p	K_i	K_d	B_d (dB) at ω_d = 8 r/s	C_n (dB) at ω_n = 70 r/s	$-\psi_f$	PM (degrees)	GM (dB)	Stability
TP0	233.423	22.397	18.527	−9.2336	−11.5114	0.0244	60.3404	∞	S
TP1	230.1	31.88	26.84	−11.5581	−8.5031	0.0086	73.9958	∞	S
TP2	236.7	12.92	10.22	−7.1768	−16.6134	0.0486	31.0285	−19.9884	S
TP3	240	3.44	1.918	−6.8526	−29.2536	0.0120	−8.1255	11.1198	U
TP4	243.4	−6.04	−6.388	−8.6061	−19.7476	−0.0287	−44.1608	−46.8469	U
TP5	246.7	−15.52	−14.69	−10.9464	−13.2978	−0.0205	−69.3573	−48.4513	U
TP6	250	−25	−23	−13.0966	−9.8731	−0.0089	−81.8201	−49.7436	U
TP7	120	−4	−19.01	−10.1111	−11.5104	−0.0089	−84.4180	−46.3027	U

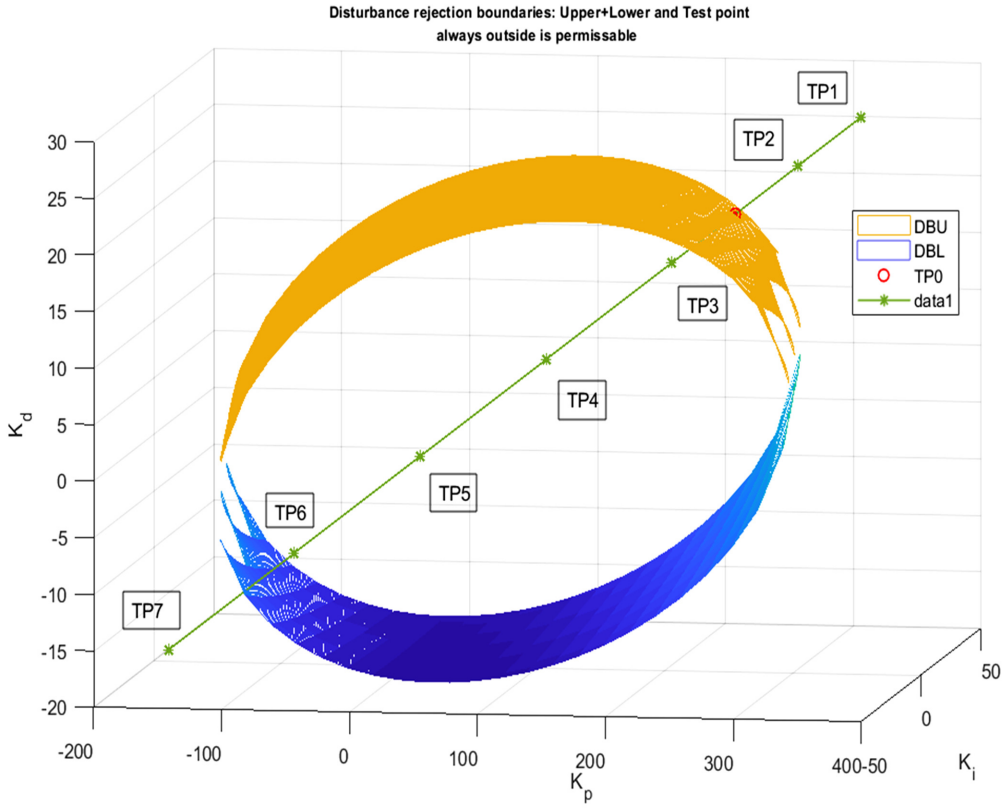

Figure 19: Disturbance rejection surface and test points of Example 2

DBU and DBL are −6.6959, 2.6749, 1.0445 and −7.3311 dB which are greater than −9.2306 dB; this means disturbance specification is not satisfied at these points. So, the theoretical results of Section 4.4 and the 3D plot of disturbance rejection surfaces are well verified.

Noise rejection boundaries:

Noise rejection boundaries including the test points tabulated in Table 6 are shown in Fig. 20. It is seen that the test point TP0 at which the noise gain is $C_n = 0.2657$ (−11.5114 dB) at $\omega = 70$ rad/s is on the noise boundary (NBU) as expected. The same property is valid for the test point TP7, which is on the lower boundary NBL. Being between the noise boundary surfaces NBU and NBL, the test points TP2, TP3, TP4, and TP5 satisfy the noise rejection requirement since the noise gains at these points are all smaller than the specified value −11.5114 dB at $\omega_n = 70$ rad/s. The test point TP1, which is above the noise boundary NBU does not satisfy the noise rejection requirement since it yields a noise gain −8.5031 dB which is greater than the specified value −11.5114 dB. Similarly, the test point TP6, which is below the noise boundary NBL does not satisfy the noise rejection requirement since it yields a noise gain −9.8731 dB which is greater than the specified value −11.5114 dB. As a result, the points remaining between the noise boundary surfaces NBU and NBL are the feasible points from the noise rejection point of view, and the points outside of these boundaries are not permissible.

So, the theoretical results of Section 3.4 and their computer implementation in Section 4.5 for 3D plots of noise rejection surfaces are well verified.

Table 6: Noise characteristics of some test points for Example 2

Test point	K_p	K_i	K_d	B_d (dB) at ω_d = 8 r/s	C_n (dB) at ω_n = 70 r/s	$-\psi_f$	PM (degrees)	GM (dB)	Stability
TP0	233.423	22.397	18.527	−9.2336	−11.5114	0.0244	60.3404	∞	S
TP1	230.1	31.88	26.84	−11.5581	−8.5031	0.0086	73.9958	∞	S
TP2	236.7	12.92	10.22	−7.1768	−16.6134	0.0486	31.0285	−19.9884	S
TP3	240	3.44	1.918	−6.8526	−29.2536	0.0120	−8.1255	11.1198	U
TP4	243.4	−6.04	−6.388	−8.6061	−19.7476	−0.0287	−44.1608	−46.8469	U
TP5	246.7	−15.52	−14.69	−10.9464	−13.2978	−0.0205	−69.3573	−48.4513	U
TP6	250	−25	−23	−13.0966	−9.8731	−0.0089	−81.8201	−49.7436	U
TP7	120	−4	−19.01	−10.1111	−11.5104	−0.0089	−84.4180	−46.3027	U

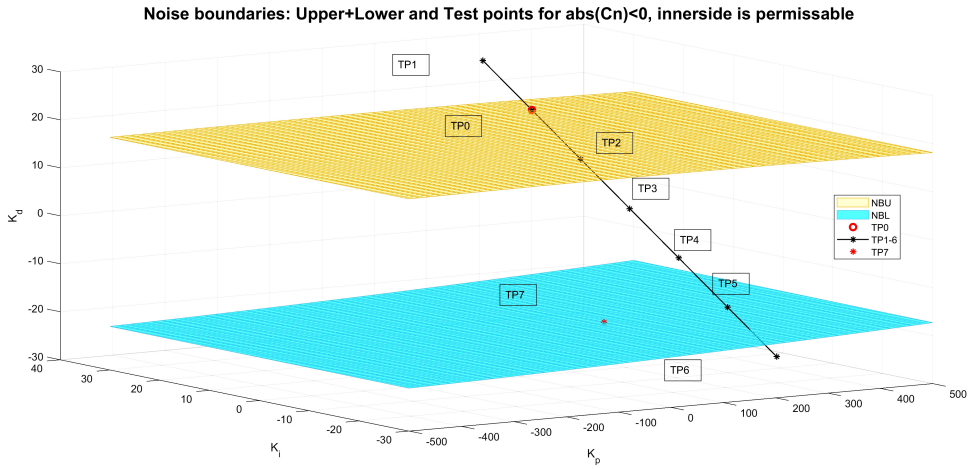

Figure 20: Noise rejection boundaries, and the test points for Example 2

Composite 3D plottings:

When the controller parameters are to be changed to obtain better performances than the specified ones, the new set of parameters should be chosen in the permissible regions defined by the individual plots. Therefore, a collection of the related boundaries is preferred on the same sD coordinate axis. For example, flatness, disturbance and noise are of concern, then the related boundaries are redrawn altogether as in Fig. 21 (flatness curves FC1, FC3, FC4 cannot be observed

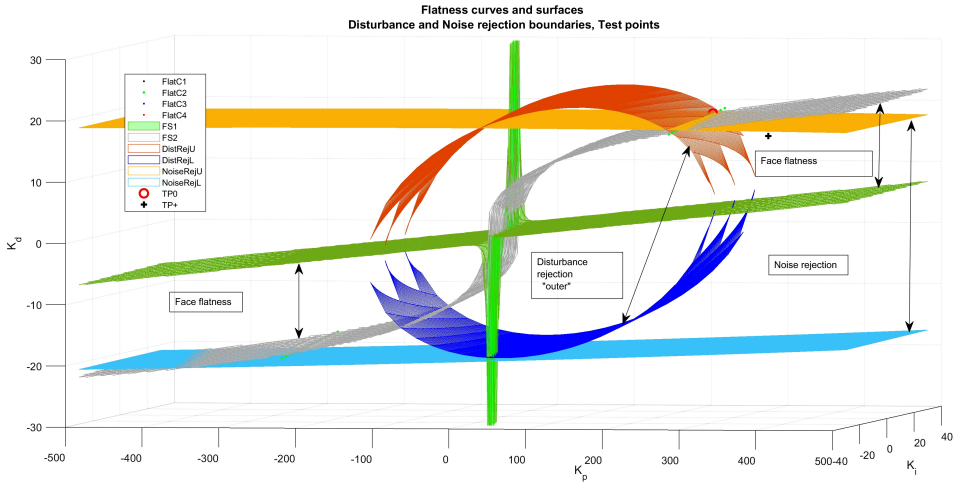

Figure 21: Flatness, disturbance and noise boundaries on a single 3D plot of Example 2

since they are out of the used scale) for the given example. Hence, a better design point (K_p, K_i, K_d) can be chosen by the use of MATLAB 3D plotting facilities together with the subprogram "performance" to test the results. For example, the test point TP+ shown in this figure has design parameters $K_p = 310$, $K_i = 20$, $K_d = 15$ and it is in the permissible regions defined by the flatness, disturbance and noise boundaries. Therefore, it is expected to give performance results better than the test point TP0; in fact, flatness, disturbance rejection and noise rejection values of TP+ are 0.0382 s, 10.5988 dB, 13.2581 dB, which are all greater than those 0.0244 s, 9.2336 dB, 11.5114 dB of TP0 representing the design point of the benchmark example considered.

The composition is not restricted to the flatness, disturbance and noise plots only. Other 3D plots (stability boundaries, specified phase and gain margin boundaries) can also be combined among themselves or together with the above plots in a similar manner.

5.3. Example 3

This example is taken from [5], which presents a FOPI controller design for stabilizing the power control of a pressurized heavy water reactor. The reactor model is given by the transfer function

$$G_p(s) = \frac{1522.8947}{s^{2.0971} + 8.1944s^{1.0036} + 7.7684} e^{-2.0043 \cdot 10^{-12} s}. \tag{55}$$

Design specifications are PM $\varphi_m = \pi/2$ (90°), GCOF $w_{gc} = 0.3$ rad/s at which a flat phase satisfying $\psi_f = 0s$ is required. For this non-integer order plus time delay (NOPTD) plant, Bhase and Patre have obtained the following FOPI controller,

$$G_c(s) = 0.0016323 + \frac{0.001506}{s^{1.004}}. \tag{56}$$

They claimed that their controller design approach provides better robust performance than other methods [20, 39].

The purpose of this example is to show how a PID controller design better than [5] can be achieved by using 3D methods presented in this paper.

Performance characteristics:

The performance characteristics of the control system designed by Bhase and Patre are shown in Table 7. The elapsed and CPU times for running the subprogram "performance" are recorded as 9.4474 s and 7.2355 s, respectively. The realized performance characteristics are satisfying the specified PM as indicated specified PM and PF as indicated by ≅ in Table 7.

Table 7: Performance characteristics of PI$^{\lambda}$ controlled system of Example 3

Characteristics	Specification (*Not specified)	Realized	Comments: –: No comment =: Almost exact,+: Satisfied !: Can be improved
ω_{gc} (rad/s)	0.3	0.3003	=
PM (degrees) φ_m (rad)	90	90.0006 1.5708	=
ψ_f (s)	0	−0.0022	+
B_d B_d (dB)	*	0.1651 −15.6433	!
ω_d (rad/s)	*	0.05	–
C_n C_n (dB)	*	0.1159 −18.7212	!
ω_n (rad/s)	*	3	–
ω_{pc} (rad/s)	*	32.999	–
GM (dB) g_m	*	55.6438 605.6060	!
Overshoot (%)	*	0%	(found by Simulink)
Rise time (s)	*	7.2355	(from 10% to 90%)
Settling time (s)	*	10.1088	(95–105% of the final value)
Elapsed time (s)	*	9.4474	–
CPU time (s)	*	9.2656	–

Complex root boundaries:

For the modification of the designed PI$^{\lambda}$ controller to obtain a proper PID controller, consider PI$^{\lambda=1.004}$ D$^{\mu=1}$. The CRBs are drawn first with $\lambda = 1.00$, and performance characteristics are tested while keeping the same values $K_p = 0.0016323$, $K_i = 0.001506$ and for different K_d values tabulated in Table 8.

Complex root boundaries and test points in the table are shown in Fig. 22.

The following observations are detected from the results of Fig. 22 and Table 8.

i. Only 2 CRBs are observed in the 3D plots, CRB1, and CRB2. Since PM is measured as 90° at TP2 and TP0, these are the actual feasible points between which the PM specification is satisfied. As a result, the feasible region is bounded by the CRB2 on the top and by the CRB1 at the bottom.

ii. The controlled system is unstable for $K_d < -0.0049$, hence the region below $K_d = -0.0049$ (so are the test points TP6-TP9) is out of interest. In facts, these points yield both negative GM and PM.

Table 8: Test points and the associated performance characteristics of Example 3

Test point	K_d	C_n	B_d	$-\psi_f$ (s)	PM (°)	GM (dB)	ω_{gc} (rad/s)	Stability
TP1	0.050	0.9077	0.1793	−0.0023	87.4079	∞	52.4630	S
TP2	0.0351	0.8734	0.1749	−0.0044	89.9963	∞	38.0901	S
TP3	0.02	0.7967	0.1705	−0.0129	96.5714	∞	22.8036	S
TP0	0	0.1159	0.1651	0.0022	90.0006	55.6438	0.3003	S
TP4	−0.0048	44.3871	0.1639	−0.8276	77.4509	0.1772	0.5168	S
TP5	−0.0049	109.1646	0.1639	−0.2993	0.2673	0.0041	3.0149	S, U
TP6	−0.0050	48.0528	0.1638	−0.2598	−7.9166	−0.1627	3.5207	U
TP7	−0.020	1.3362	0.1601	−0.0132	−83.0231	−11.9591	22.8096	U
TP8	−0.0356	1.1653	0.1563	−0.0044	−89.9897	−16.9233	38.5833	U
TP9	−0.06	1.0920	0.1508	−0.0016	−93.5597	−21.4333	61.8681	U

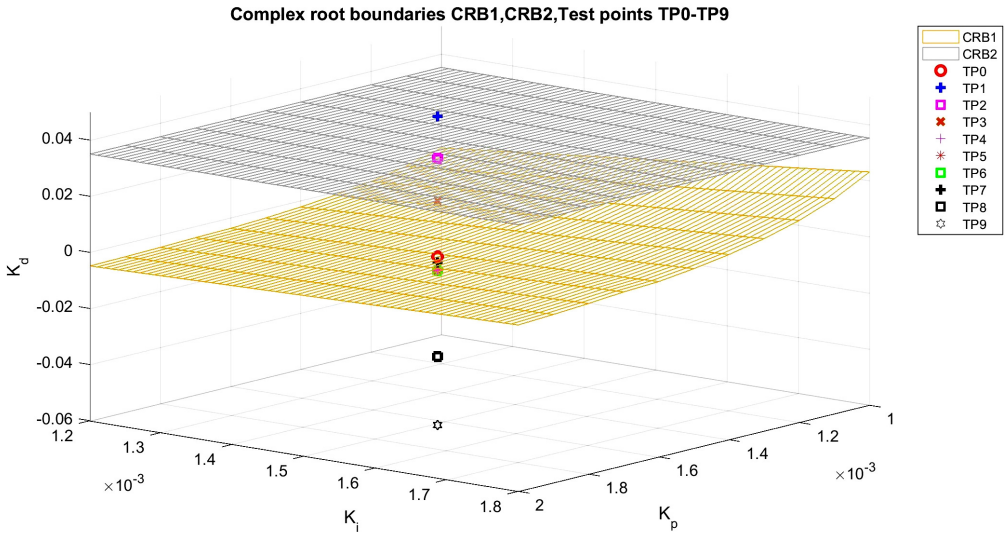

Figure 22: Flatness, disturbance and noise boundaries on a single 3D plot of Example 2

iii. In the sense of phase flatness TP3 with $K_d = 0.02$ yields a better PM ($= 96° > 90°$) then TP0 with $K_d = 0$; further, it yields a flatness over a larger range of frequencies around $\omega_{gc} = 22.8036$ rad/s (though the flatness is slightly less than that of TP0, it is still almost 0). Disturbance rejection

properties are almost the same. Although TP0 has higher noise suppression, this characteristic is not a design specification.

iv. Result: K_d should satisfy $0 \leqslant K_d \leqslant 0.0351$.

Phase flatness curves and surfaces

For discussing the phase flatness boundaries, the test points shown in Table 9 are considered. TP2 is on the flatness surface FS2 and TP0 is on the FS1 whilst TP3 is between these surfaces. See Fig. 23, where the flatness curves FC1 and FC2 do not exist for this example, that is there are only two flatness curves. Hence, the phase requirement specification $\psi_f \geqslant 0.0022$ is satisfied for these

Table 9: Test points for discussing the phase flatness boundaries and the associated performance characteristics of Example 3

Test point	K_d	C_n	B_d	$-\psi_f$ (s)	PM (°)	GM (dB)	ω_{gc} (rad/s)	Stability
TP1*	0.02	0.7967	0.1705	−0.0129	96.5714	∞	22.8036	S
TP2	0.013	0.7175	0.1686	2.0175	95.8821	∞	0.2008	S
TP3	0.007	0.5759	0.1670	1.1140	94.1412	∞	0.2304	S
TP0	0	0.1159	0.1651	0.0022	90.0006	55.6438	0.3003	S
TP4	−0.010	1.9972	0.1626	−0.0509	−65.9491	−6.0306	11.6598	U

*Some of the data for TP1 is not reliable since there are three different ω_{gc}.

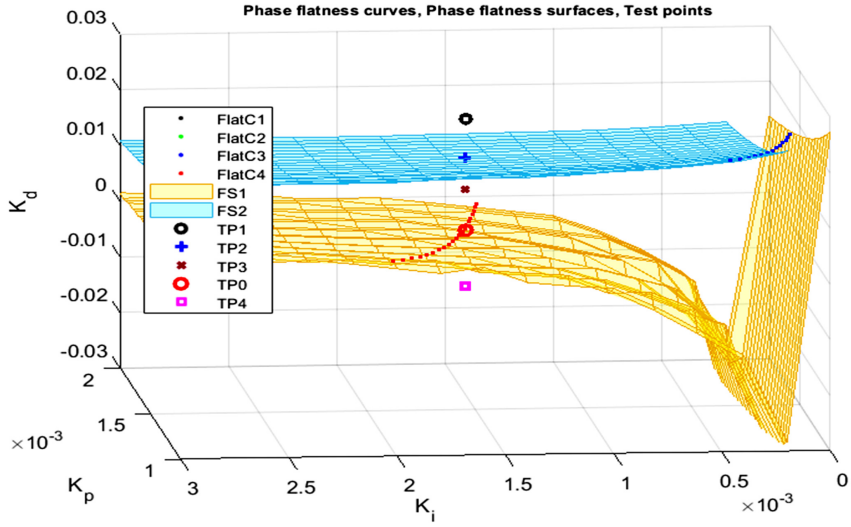

Figure 23: Phase flatness curves and surfaces of Example 3

test points. Note that the original design gives a phase flatness 0.0022, therefore the "0" phase flatness requirement is made more stringent by choosing 0.0022 instead of 0. TP1 and TP4 do not satisfy the phase flatness requirement for having $\psi_f < 0.0022$ s. Note that TP1 is above FS2 and TP4 is below FS1. Note also that TP0 is on the flatness curve (FlatC4) as well, which should be the case.

Noise boundaries:

Noise boundaries are plotted in Fig. 24. The permissible region is between the boundary surfaces NBU and NBL. The designed system is at the test point TP0, which is on the surface NBL. The data for the test points in the figure are given in Table 10. It is observed that TP2, TP3 and TP0 satisfy the noise rejection

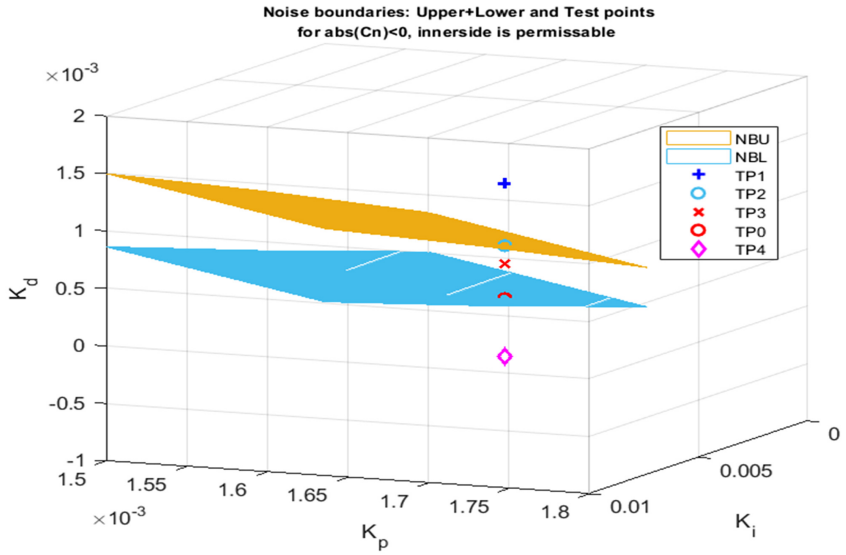

Figure 24: Noise rejection boundary surfaces for Example 3

Table 10: Test points for discussing the noise rejection boundaries and the associated performance characteristics for Example 3

Test point	K_d	C_n	B_d	$-\psi_f$ (s)	PM (°)	GM (dB)	ω_{gc} (rad/s)	Stability
TP1	0.0010	0.1685	0.1654	0.1667	90.8924	∞	0.2852	S
TP2	0.00045	0.1144	0.1653	0.0765	90.4232	∞	0.2931	S
TP3	0.0003	0.1075	0.1652	0.0518	90.2866	∞	0.2954	S
TP0	0	0.1159	0.1651	0.0022	90.0006	55.6438	0.3003	S
TP4	−0.0005	0.1953	0.1650	−0.0810	89.4818	18.7161	0.3090	U

requirement, and the test points TP1 and TP4 do not. The mentioned requirement is not a design specification. It is the limit by considering the PI$^\lambda$ design in [5]. Therefore, the following result can be waived.

Result: K_d should satisfy $0 \leqslant K_d \leqslant 0.00045$.

Disturbance boundaries:

There is no disturbance specification given in the design problem. In fact, the disturbance rejection is hardly affected by the change in the parameter K_d. This is observed from the plots (not included here since they are also observed in Tables 4–6) of the relevant boundaries, which give a large region of permissible values for K_d. See also the results in Tables 4–6 where B_d changes in the limits $\cong$ [0.16–0.18] only.

Step responses:

The step responses of the original design with PI$^\lambda$ controller and the suggested PID controller are shown in Fig. 25. The step response with a combination of PID and PI Delay controller is also shown on the same figure. The critical time-domain characteristics are indicated in Table 11.

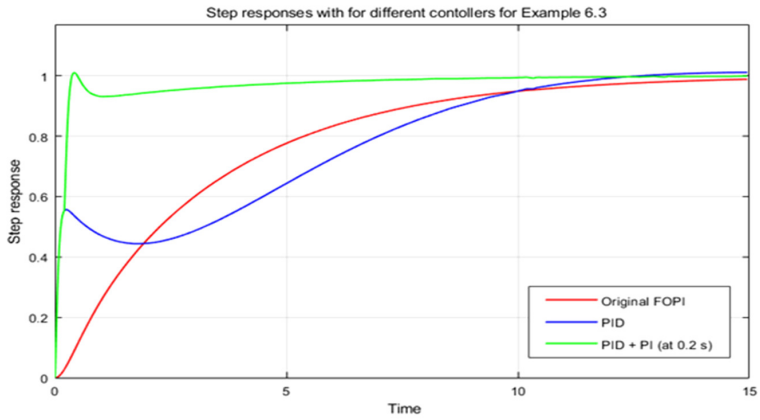

Figure 25: Step responses of the system of Example 3 with different controllers

Table 11: Time characteristics with the controllers PI$^\lambda$, PID and PID + Delay

Controller	K_p	K_i λ	K_d μ	T_r (s)	T_s (s)	Overshoot (%)
PI$^\lambda$	0.0016323	0.001506, 1.004	0, 0	7.25	10	0
PID	0.0016323	0.001506, 1	0.007, 1	8.58	10	0
(PID) + PI 0.2 s Delay	0.08	0.02, 1	0, 0	0.27	2.5	1

6. Conclusions

After a short introduction of the FOS, PID and FOPID controllers and giving the fundamental definitions and properties of FO calculus, important design methods of FOPID controllers are reviewed. And then, the main topics of the paper are considered in Sections 3, 4 and exemplified in Section 5. The frequency-domain design specifications are given in Section 2 and frequency domain $PI^{\lambda}D^{\mu}$ control design formulas convenient for the 3D design method to realize these specifications are derived in Section 3. In particular, frequency-domain characteristics such as stability boundaries to yield the required gain and phase margins, phase flatness surfaces and curves to achieve the required iso-damping property, boundary surfaces LFODR and HFNR limits are considered. Section 4 illustrates how the derived formulas can be implemented for computer programming yielding 3D plots usable in choosing design parameters.

The 3D plots and their use in designing ordinary PID or FOPID controllers have not been considered before in the literature on this extent. The method itself and different 3D plots are illustrated by examples in Section 5. The formulas derived in Section 3 and the 3D plots described in Section 4 are verified by these examples.

The presented method applies to control IO as well as FO plants with time delay by using $PI^{\lambda}D^{\mu}$ controllers, in particular by using any of PI, PD, PI^{λ}, PD^{μ}, PID controller.

FOTF toolbox [73] is used for preparing the subprograms to obtain the 3D plots of different surfaces and curves from the obtained formulas. The programs are written using MATLAB notation. Oustaloup continuous approximation [53] is used with order $N = 5$, $\omega_b = 10^{-4}$, $\omega_h = 10^4$ rad/s for evaluation of step response by the FO Simulink blocks which appear in the FOTF. Although FOTF's subprograms have accelerated the work dealt with in the paper, there have been prepared many basic subroutines based on the formulations and computer implementations covered in Sections 3 and 4, respectively. Some of them are deserved to be mentioned without giving details as "Bass.m", "derivative.m", "distbound.m", "flatnessC.m", "FlatnsCurve.m", "flatsur.m", "noisebound.m", "performance.m" (as already mentione before), "RRBminMax.m", "RRBplot.m", "solution.m", "StabBound.m", "stepResp.m", and finally "testpoint.m" inclusion of which details in this article are not found to be appropriate from the viewpoints of neither contribution nor space.

The main advantage of the presented 3D design method of $PI^{\lambda}D^{\mu}$ controllers is that one view in 3D (K_p, K_i, K_d) space what ranges of the parameters can be used to achieve the required design specifications, which helps to find the optimum control set (K_p, K_i, K_d). Different 3D plots (stability boundaries, specified phase

and gain margin boundaries, disturbance and noise rejection boundaries) can be combined among themselves or altogether to improve the design parameters K_p, K_i, K_d to refine all the performance characteristics globally and simultaneously. This may cause confusion and crowded scenery on the 2D paperwork; however, it supplies great easiness on the MATLAB screen to decide on the choice of optimal controller parameters.

On account of its main advantage mentioned, there are a few restrictions that can be listed as follows:

i. The method assumes the FOs λ and μ to be known. Never mind, these can be determined by other conventional design techniques reviewed and referred in the literature surveys of Section 1 and particularly Section 2.3.

ii. Although the method is applicable for a large range of variety of plant transfer functions (FO or IO rational transfer functions with or without delay), it cannot be used for linear or nonlinear time-varying plants, for stochastic plants and plants having hysteresis behavior.

iii. Since the formulas derived in this work are valid for any of $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ controllers and its above-mentioned special forms, they cannot be used for other controller designs such as $(\mathrm{PID})^{\lambda}$ [38], DOF FOPID [48], TID (Tilt integral derivative, $\mathrm{P}^{1/n}\mathrm{ID}$) [43], FO lead-lag [69] controls.

There are a few points that should be cared on when using the presented 3D method of design of $PI^{\lambda}D^{\mu}$ controllers:

1. Due to the exponential (delay) term in the plant transfer function and due to the FOs in the transfer functions of the controller and/or plant as well as the multivalued solutions of some algebraic equations in the formulations, the boundaries computed for the stability, flatness, noise and disturbance rejections may be multiple. One has to decide which of them are true ones and which sides of these boundaries are permissible regions for the searched characteristics satisfying the given requirements. To decide this, some test points should be tried by the program "performance" and the convenient regions should be chosen accordingly [30].

2. When the Nyquist plots of the open-loop gain cross negative real axis (multiple phase crossovers) and unity gain circle (multiple gain crossovers), improper GM and PM values may result. This surely occurs when the delay term ($e^{-\theta s}$) is present in the plant transfer function and may occur when the plot travels in the right half-plane due to non-minimum phase transfer functions or the multiple values of inverse trigonometric functions (especially inverse tangent). One should investigate the actual Bode plots given by "performance" instead of looking at the numerical results only;

that is, both the numerical results obtained by "performance" and the 3D plots used for different characteristics should always be judged.

We terminate the conclusions by noting that similar approaches valid for the 3D design of $PI^{\lambda}D^{\mu}$ controllers presented in this paper can be worked out for other types of FO controllers and the relevant formulations and computer implementations can be programmed as a future work.

References

[1] A.A. Aldair and W.J. Wang: Design of fractional order controller based on evolutionary algorithm for a full vehicle nonlinear active suspension systems. *International Journal of Control and Automation*, **3**(4), (2010), 33–46.

[2] A. Alfi and H. Modares: System identification and control using adaptive particle swarm optimization. *Applied Mathematical Modelling*, **35**(3), (2011), 1210–1221. DOI: 10.1016/j.apm.2010.08.008

[3] A.T. Azar and F.E. Serrano: Fractional order sliding mode PID controller/observer for continuous nonlinear switched systems with PSO parameter tuning. *Proceedings of International Conference on Advanced Machine Learning Technologies and Applications*, Springer, Cham, (2008), 13–22. DOI: 10.1007/978-3-319-74690-6_2

[4] K.J. Åström and T. Hägglund: *PID Controllers: Theory, Design, and Tuning*. Vol. 2. Research Triangle Park, NC: Instrument Society of America, 1995.

[5] S.S. Bhase and B.M. Patre: Robust FOPI controller design for power control of PHWR under step-back condition. *Nuclear Engineering and Design*, **274** (2014), 20–29. DOI: 10.1016/j.nucengdes.2014.03.041

[6] A. Biswas, S. Das, A. Abraham and S. Dasgupta: Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution. *Engineering Applications of Artificial Intelligence*, **22**(2), (2009), 343–350. DOI: 10.1016/j.engappai.2008.06.003

[7] B. Boudjehem and D. Boudjehem: Fractional order controller design for desired response. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, **227**(2), (2013), 243–251. DOI: 10.1177/0959651812456647

[8] M. Cai, X. Pan and Y. Du: New elite multi-parent crossover evolutionary optimization algorithm of parameters tuning of fractional-order PID controller and its application. *Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control*, (2009), 64–67. Kaohsiung, Taiwan. DOI: 10.1109/ICICIC.2009.277

[9] A.J. Calderón, B.M. Vinagre and V. Feliu: Fractional order control strategies for power electronic buck converters. *Signal Processing*, **86**(10), (2006), 2803–2819. DOI: 10.1016/j.sigpro.2006.02.022

[10] J.Y. Cao, J. Liang and B.G. Cao: Optimization of fractional order PID controllers based on genetic algorithms. *Proceedings of the 2005 International Conference on Machine Learning and Cybernetics*, **9** (2005), 5686–5689. Guangzhou, China. DOI: 10.1109/ICMLC.2005.1527950

[11] J.Y. Cao and B.G. Cao: Design of fractional order controllers based on particle swarm optimization. *Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications*, (2006), 1–6. Singapore, DOI: 10.1109/ICIEA.2006.257091

[12] R. Caponetto: *Fractional Order Systems: Modeling and Control Applications*. 72 World Scientific, 2010.

[13] M. Caputo: Linear model of dissipiation whose Q is almost frequencyy is independent-II. *Geophysical Journal International*, **13**(5), (1967), 529–539. DOI: 10.1111/j.1365-246X.1967.tb02303.x

[14] L.Y. Chang and H.C. Chen: Tuning of fractional PID controllers using adaptive genetic algorithm for active magnetic bearing system. *WSEAS Transactions on Systems*, **8**(1), (2009), 158–167.

[15] A. Charef, H.H. Sun, Y.Y. Tsao and B. Onaral: Fractal system as represented by singularity function. *IEEE Transactions on Automatic Control*, **37**(9), (1992), 1465–1470. DOI: 10.1109/9.159595

[16] Y. Chen, C. Hu and K.L. Moore: Relay feedback tuning of robust PID controllers with iso-damping property. *Proceedings of the 42nd IEEE International Conference on Decision and Control*, **3**, (2003), 2180–2185. Maui, HI, USA. DOI: 10.1109/CDC.2003.1272941

[17] E. Cokmez, S. Atiç, F. Peker and I. Kaya: Fractional-order PI controller design for integrating processes based on gain and phase margin specifications. *IFAC-PapersOnLine*, **51**(4), (2018), 751–756. DOI: 10.1016/j.ifacol.2018.06.206

[18] A. Dabiri, B.P. Moghaddam and J.T. Machado: Optimal variable-order fractional PID controllers for dynamical systems. *Journal of Computational and Applied Mathematics*, **339** (2018), 40–48. DOI: 10.1016/j.cam.2018.02.029

[19] S. Das, S. Saha, S. Das and A. Gupta: On the selection of tuning methodology of FOPID controllers for the control of higher order processes. *ISA Transactions*, **50**(3), (2011), 376–388. DOI: 10.1016/j.isatra.2011.02.003

[20] S. Das, S. Das and A. Gupta: Fractional order modeling of a PHWR under step-back condition and control of its global power with a robust and $PI^{\lambda}D^{\mu}$ controller. *IEEE Transactions on Nuclear Science*, **58**(5), (2011), 2431–2441. DOI: 10.48550/arXiv.1202.5684

[21] S. Das, I. Pan, S. Das and A. Gupta: A novel fractional order fuzzy PID controller and its optimal time domain tuning based on integral performance indices. *Engineering Applications of Artificial Intelligence*, **25**(2), (2012), 430–442. DOI: 10.1016/j.engappai.2011.10.004

[22] I. Dimeas, I. Petras and C. Psychalinos: New analog implementation technique for fractional-order controller: a DC motor control. *AEU-International Journal of Electronics and Communications*, **78** (2017), 192–200. DOI: 10.1016/j.aeue.2017.03.010

[23] R. El-Khazali: Fractional-order $PI^{\lambda}D^{\mu}$ controller design. *Computers & Mathematics with Applications*, **66**(5), (2013), 639–646. DOI: 10.1016/j.camwa.2013.02.015

[24] A.M.A El-Sayed, A.E.M. El-Mesiry and H.A.A. El-Saka: On the fractional-order logistic equation. *Applied Mathematics Letters*, **20**(7), (2007), 817–823. DOI: 10.1016/j.aml.2006.08.013

[25] V. Feliu-Batlle, R.R. Perez and L.S. Rodriguez: Fractional robust control of main irrigation canals with variable dynamic parameters. *Control Engineering Practice*, **15**(6), (2007), 673–686. DOI: 10.1016/j.conengprac.2006.11.018

[26] N.F. Ferreira and J.T. Machado: Fractional-order hybrid control of robotic manipulators. *Proceedings of the 11th International Conference on Advanced Robotics*, **1** (2003), 393–398. San Diego, CA, USA. DOI: 10.1109/ICSMC.1998.725510

[27] G.F. Franklin, J.D. Powell and M.L. Workman: *Digital Control of Dynamic Systems*. 3 Menlo Park, CA: Addison-wesley, 1998.

[28] S. Ghasemi, A. Tabesh and J. Askari-Marnani: Application of fractional calculus theory to robust controller design for wind turbine generators. *IEEE Transactions on Energy Conversion*, **29**(3), (2014), 780–787. DOI: 10.1109/TEC.2014.2321792

[29] S.E. Hamamci: An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers. *IEEE Transactions on Automatic Control*, **52**(10), (2007), 1964–1969. DOI: 10.1109/TAC.2007.906243

[30] S.E. Hamamci: Stabilization using fractional-order PI and PID controllers. *Nonlinear Dynamics*, **51**(1-2), (2008), 329–343. DOI: 10.1007/s11071-007-9214-5

[31] S.E. Hamamci and M. Koksal: Calculation of all stabilizing fractional-order PD controllers for integrating time delay systems. *Computers & Mathematics with Applications*, **59**(5), (2010), 1621–1629. DOI: 10.1016/j.camwa.2009.08.049

[32] C. Hwang and Y.C. Cheng: A numerical algorithm for stability testing of fractional delay systems. *Automatica*, **42**(5), (2006), 825–831. DOI: 10.1016/j.automatica.2006.01.008

[33] I.S. Jesus and J.T. Machado: Fractional control of heat diffusion systems. *Nonlinear Dynamics*, **54**(3), (2008), 263–282. DOI: 10.1007/s11071-007-9322-2

[34] T. Kaczorek: Fractional positive continuous-time linear systems and their reachability. *International Journal of Applied Mathematics and Computer Science*, **18**(2), (2008), 223–228. DOI: 10.2478/v10006-008-0020-0

[35] V.K. Kadiyala, R.K. Jatoth and S. Pothalaiah: Design and implementation of Fractional Order PID controller for aerofin control system. *Proceedings of 2009 World Congress on Nature & Biologically Inspired Computing*, (2009), 696–701. Coimbatore, India. DOI: 10.1109/NABIC.2009.5393470

[36] I. Kheirizad, A.A. Jalali and K. Khandani: Stabilization of fractional-order unstable delay systems by fractional-order controllers. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, **226**(9), (2012), 1166–1173. DOI: 10.1177/0959651812453668

[37] I. Kheirizad, K. Khandani and A.A. Jalali: Stabilisability analysis of high-order unstable processes by fractional-order controllers. *International Journal of Control*, **86**(2), (2013), 244–252. DOI: 10.1080/00207179.2012.723138

[38] N. Lachhab, F. Svaricek, F. Wobbe and H. Rabba: Fractional order PID controller (FOPID)-toolbox. *Proceedings of the 2013 European control conference (ECC)*, (2013), 3694–3699. Zurich, Switzerland. DOI: 10.23919/ECC.2013.6669365

[39] C.H. Lee and F.K. Chang: Fractional-order PID controller optimization via improved electromagnetism-like algorithm. *Expert Systems with Applications*, **37**(12), (2010), 8871–8878. DOI: 10.1016/j.eswa.2010.06.009

[40] H. Li and Y. Chen: A fractional order proportional and derivative (FOPD) controller tuning algorithm. *Proceedings of 2008 Chinese Control and Decision Conference*, (2008), 4059–4063. Yantai, China. DOI: 10.1109/CCDC.2008.4598094

[41] J. Liouville: Mémoire sur le Théoreme des fonctions complémentaires. *Journal für die Reine und Angewandte Matematik*, **1834**(11), (1834), 1–19.

[42] Y. Luo, Y.Q. Chen, C.Y. Wang and Y.G. Pi: Tuning fractional order proportional integral controllers for fractional order systems. *Journal of Process Control*, **20**(7), (2010), 823–831. DOI: 10.1016/j.jprocont.2010.04.011

[43] B.J. LURIE: *Three-Parameter Tunable Tilt-Integral-Derivative (TID) Controller*. US Patent no 5,371,670. 1994.

[44] C. Ma and Y. Hori: Fractional order control and its application of pi/sup/spl alpha//d controller for robust two-inertia speed control. *Proceedings of the 4th International Power Electronics and Motion Control Conference*, **3** (2004), 1477–1482. Xi'an, China.

[45] D. Maiti, S. Biswas and A. Konar: Design of a fractional order PID controller using particle swarm optimization technique. *arXiv preprint arXiv:0810.3776*, 2008. DOI: 10.48550/arXiv.0810.3776

[46] D. Maiti, A. Acharya, M. Chakraborty, A. Konar and R. Janarthanan: Tuning PID and $PI^{\lambda}D^{\sigma}$ controllers using the integral time absolute error criterion. *Proceedings of the 4th International Conference on Information and Automation for Sustainability*, (2008), 457–462. Colombo, Sri Lanka. DOI: 10.1109/ICIAFS.2008.4783932

[47] D. Matignon: Generalized fractional differential and difference equations: Stability. *ESAIM Proceedings*, **5** (1998), 145–158.

[48] H. Meneses, E. Guevara, O. Arrieta, F. Padula, R. Vilanova and A. Visioli: Improvement of the control system performance based on fractional-order PID controllers and models with robustness considerations. IFAC-PapersOnLine, **51**(4), (2018), 551–556. DOI: 10.1016/j.ifacol.2018.06.153

[49] S. Mete, S. Ozer and H. Zorlu: System identification using Hammerstein model optimized with differential evolution algorithm. AEU-International Journal of Electronics and Communications, **70**(12), (2016), 1667–1675. DOI: 10.1016/j.aeue.2016.10.005

[50] K.S. Miller and B. Ross: *An Introduction to the Fractional Calculus and Fractional Differential Equations*. John Wiley & Sons, Inc., New York, 1993.

[51] C.A. Monje, B.M. Vinagre, V. Feliu and Y. Chen: Tuning and auto-tuning of fractional order controllers for industry applications. *Control Engineering Practice*, **16**(7), (2008), 798–812. DOI: 10.1016/j.conengprac.2007.08.006

[52] A. Oustaloup, J. Sabatier, P. Lanusse, R. Malti, P. Melchior, X. Moreau and M. Moze: An overview of the CRONE approach in system analysis, modeling and identification, observation and control. *IFAC Proceedings*, **41**(2), (2008), 14254–14265. DOI: 10.3182/20080706-5-KR-1001.02416

[53] A. Oustaloup, F. Levron, B. Mathieu and F.M. Nanot: Frequency-band complex noninteger differentiator: characterization and synthesis. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **47**(1), (2000), 25–39. DOI: 10.1109/81.817385

[54] H. Özbay, C. Bonnet and A.R. Fioravanti: PID controller design for fractional-order systems with time delays. Systems & Control Letters, **61**(1), (2012), 18–23. DOI: 10.1016/j.sysconle.2011.09.011

[55] I. Pan and S. Das: Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}|$ controller in AVR system. *International Journal of Electrical Power & Energy Systems*, **43**(1), (2012), 393–407. DOI: 10.1016/j.ijepes.2012.06.034

[56] F. Padula and A. Visioli: Tuning rules for optimal PID and fractional-order PID controllers. *Journal of Process Control*, **21**(1), (2011), 69–81. DOI: 10.1016/j.jprocont.2010.10.006

[57] I. Petras, L. Dorcak, P. O'Leary, B.M. Vinagre and I. Podlubny: The modelling and analysis of fractional-order control systems in frequency domain. *arXiv preprint math/0008186*, (2000). DOI: 10.48550/arXiv.math/0008186

[58] I. Podlubny: Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers. *IEEE Transactions on Automatic Control*, **44**(1), (1999), 208–214. DOI: 10.1109/9.739144

[59] A. Rajasekhar, A. Abraham and M. Pant: A hybrid differential artificial bee colony algorithm based tuning of fractional order controller for permanent magnet synchronous motor drive. *International Journal of Machine Learning and Cybernetics*, **5**(3), (2014), 327–337. DOI: 10.1007/s13042-012-0136-2

[60] A. Rajasekhar, S. Das and P.N. Suganthan: Design of fractional order controller for a servohydraulic positioning system with micro artificial bee colony algorithm. *Proceedings of the Congress on Evolutionary Computation*, (2012), 1–8. Brisbane, QLD, Australia. DOI: 10.1109/CEC.2012.6256644

[61] H.P. Ren, J.T. Fan and O. Kaynak: Optimal design of a fractional order PID controller for a pneumatic position servo system. *IEEE Transactions on Industrial Electronics*, **66**(8), (2018), 6620–6629. DOI: 10.1109/TIE.2018.2870412

[62] H. Senberber and A. Bagis: Fractional PID controller design for fractional order systems using ABC algorithm. *Proceedings of IEEE 2017 Electronics*, (2017), 1–7. Palanga, Lithuania. DOI: 10.1109/ELECTRONICS.2017.7995218

[63] M. Schlegel and M. Čech: *Fractal System Identification For Robust Control–The Moment Approach*. Department of Cybernetics, University of West Bohemia in Pilsen Univerzitn, Pilsen, 2004.

[64] W.C. Schultz and V.C. Rideout: Control system performance measures: Past, present, and future. *IRE Transactions on Automatic Control*, **AC-6**(1), (1961), 22–35. DOI: 10.1109/TAC.1961.6429306

[65] P. Shah and S.D. Agashe: Design and optimization of fractional PID controller for higher order control system. *Proceedings of International Conference of ICART*, (2013), 588–592. Barcelona, Spain.

[66] A. Soltan, L. Xia, A. Jackson, G. Chester and P. Degenaar: Fractional order PID system for suppressing epileptic activities. *Proceedings of the 2018 IEEE International Conference on Applied System Invention* (ICASI), (2018), 338–341. Chiba, Japan. DOI: 10.1109/ICASI.2018.8394603

[67] N. Tan, I. Kaya, C. Yeroglu and D.P. Atherton: Computation of stabilizing PI and PID controllers using the stability boundary locus. *Energy Conversion and Management*, **47**(18-19), (2006), 3045–3058. DOI: 10.1016/j.enconman.2006.03.022

[68] M.S. Tavazoei: Notes on integral performance indices in fractional-order control systems. *Journal of Process Control*, **20**(3), (2010), 285–291. DOI: 10.1016/j.jprocont.2009.09.005

[69] A. Tepljakov, B.B. Alagoz, C. Yeroglu, E. Gonzalez, S.H. Hosseinnia and E. Petlenkov: FOPID controllers and their industrial applications: A survey of recent results. *IFAC-PapersOnLine*, **51**(4), (2018), 25–30. DOI: 10.1016/j.ifacol.2018.06.014

[70] D. Valerio and J.S. Da Costa: Ninteger: a non-integer control toolbox for MatLab. *Proceedings of the 1st IFAC Workshop on Fractional Differentiation and its Applications*, (2004), 1–6. Bordeaux, France.

[71] M. Veronesi and A. Visioli: Performance assessment and retuning of PID controllers for integral processes. *Journal of Process Control*, **20**(3), 2010, 261–269. DOI: 10.1016/j.jprocont.2009.12.007

[72] B.M. Vinagre, C.A. Monje, A.J. Calderón and J.I Suárez: Fractional PID controllers for industry application. A brief introduction. *Journal of Vibration and Control*, **13**(9-10), (2007), 1419–1429. DOI: 10.1177/1077546307077498

[73] D. Xue: *Fractional-Order Control Systems: Fundamentals and Numerical Implementations*. Walter de Gruyter GmbH and Co KG, 1 Berlin, 2017.

[74] M. Zamani, M. Karimi-Ghartemani, N. Sadati and M. Parniani: Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Engineering Practice*, **17**(12), (2009), 1380–1387. DOI: 10.1016/j.conengprac.2009.07.005

[75] C. Zhao, D. Xue and Y. Chen: A fractional order PID tuning algorithm for a class of fractional order plants. *Proceedings of IEEE International Conference Mechatronics and Automation*, **1** (2005), 216–221. Niagara Falls, ON, Canada. DOI: 10.1109/ICMA.2005.1626550

[76] W. Zheng, Y. Luo, X. Wang, Y. Pi and Y. Chen: Fractional order $PI^{\lambda}D^{\mu}$ controller design for satisfying time and frequency domain specifications simultaneously. *ISA Transactions*, **68** (2017), 212–222. DOI: 10.1016/j.isatra.2017.02.016