International Journal of Control, Automation, and Systems (2015) 13(1):81-90
DOI 10.1007/s12555-013-0335-y
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Discrete-Time Fractional-Order PID Controller: Definition, Tuning, Digital Realization and Some Applications

Farshad Merrikh-Bayat*, Nafiseh Mirebrahimi, and Mohammad Reza Khalili

Abstract: In some of the complicated control problems we have to use the controllers that apply nonlocal operators to the error signal to generate the control. Currently, the most famous controller with nonlocal operators is the fractional-order PID (FOPID). Commonly, after tuning the parameters of FOPID controller, its transfer function is discretized (for realization purposes) using the so-called generating function. This discretization is the origin of some errors and unexpected results in feedback systems. It may even happen that the controller obtained by discretizing a FOPID controller works worse than a directly-tuned discrete-time classical PID controller. Moreover, FOPID controllers cannot directly be applied to the processes modeled by, e.g., the ARMA or ARMAX model. The aim of this paper is to propose a discrete-time version of the FOPID controller and discuss on its properties and applications. Similar to the FOPID controller, the proposed structure applies nonlocal operators (with adjustable memory length) to the error signal. Two methods for tuning the parameters of the proposed controller are developed and it is shown that the proposed controller has the capacity of solving complicated control problems.

Keywords: ARMA, ARMAX, discrete-time controller, fractional-order PID, long memory, tuning.

1. INTRODUCTION

During the past seven decades PID controllers have been successfully used in a wide variety of industrial applications [1,2]. Currently, various continuous and discrete-time versions of this type of controller are available, which can be applied to the processes modeled by linear differential or linear difference equations, respectively [1-3]. Successful applications of PID controllers to control nonlinear processes can also be found in the literature [4].

According to the high achievement and the simplicity of design and implementation of PID controllers many researchers tried to enhance the performance of these controllers by innovating new structures and tuning methods [5-10]. One of these attempts led to a new generation of PID controllers, known as the fractional-order PID (FOPID) or $PI^{\lambda}D^{\mu}$ controller, which was first proposed by Podlubny in 1999 [10]. In FOPID controllers the error and control, respectively denoted as $e(t)$ and $u(t)$, are related from the following equation

$$u(t) = k_p e(t) + k_i D_t^{-\lambda} e(t) + k_d D_t^{\mu} e(t), \tag{1}$$

where $k_p, k_i, k_d \in \mathbb{R}$ and $\lambda, \mu \in \mathbb{R}^+$ are the parameters of controller to be tuned, and $D_t^{-\lambda}$ and D_t^{μ} are the fractional integral and differential operator respectively, often defined by the Riemann-Liouville definition as the following [11]:

$$D_t^{-\lambda} f(t) = \frac{1}{\Gamma(\lambda)} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-\lambda}} d\tau, \tag{2}$$

$$D_t^{\mu} f(t) = \frac{1}{\Gamma(m-\mu)} \left(\frac{d}{dt}\right)^m \int_0^t \frac{f(\tau)}{(t-\tau)^{1+\mu-m}} d\tau, \tag{3}$$

where m is a positive integer such that $m-1 < \mu \le m$ and $\Gamma(.)$ is the well-known gamma function. Other definitions for fractional differential operators can also be found in the literature [11]. Note that according to (2) and (3), fractional integral and derivative are nonlocal operators which apply the past values of f to determine $D_t^{-\lambda} f(t)$ and $D_t^{\mu} f(t)$. Hence, unlike classical PID controllers, the derivative term of the FOPID controller is actually a nonlocal operator acts on the error signal. Taking the Laplace transform from both sides of (1) leads to the following transfer function for FOPID controller [10]:

$$C(s) = \frac{U(s)}{E(s)} = k_p + k_i s^{-\lambda} + k_d s^{\mu}. \tag{4}$$

Here it is worth to mention that, unlike the classical PID controllers, currently there is no direct definition available for discrete-time FOPID controllers. However, for realization purposes, it is common practice to first design the FOPID controller (4) and then approximate it with a discrete-time system. This approximation is often performed by using the so-called *generating function*

Manuscript received July 13, 2013; revised October 18, 2013, January 11, 2014, and April 30, 2014; accepted June 15, 2014. Recommended by Associate Editor Do Wan Kim under the direction of Editor PooGyeon Park.

Farshad Merrikh-Bayat, Nafiseh Mirebrahimi, and Mohammad Reza Khalili are with the Faculty of Electrical and Computer Engineering, University of Zanjan, Zanjan, Iran (e-mails: {f.bayat, n.mirebrahimi}@znu.ac.ir, adasbios2@gmail.com).
* Corresponding author.

[12,13]. In this technique the Laplace variable s in (4) is substituted with a certain function of z (using, e.g., the Tustin transform) and then the power series expansion (PSE) of the resulted expression is obtained in terms of z. Finally, since any practical discrete-time system must necessarily use a limited memory, the resulted PSE is truncated. This approach suffers from many drawbacks, the source of all is the unavoidable mismatch between the frequency responses of continuous and discrete transfer functions. The aim of this paper is to propose a long-memory discrete-time PID (LDPID) controller which removes the limitations of the above-mentioned discretization scheme and still has the high performance of FOPID controllers.

There are many good reasons for defining and using LDPID controllers. First of all, it is very common practice to model a real-world continuous-time process by a (discrete) transfer function in the z variable. For example, such a model is obtained when an unknown process is identified using ARMA or ARMAX model. Obviously, in this case it is more reasonable to directly design a discrete-time controller as well (instead of designing a continuous-time controller and then approximating it with a discrete-time one). Another reason for developing LDPID controllers is that even when both the controller and process are continuous-time, the controller is more likely to be realized using digital microprocessors. As a classical fact, some unwanted effects (such as decreasing the phase margin) may occur in the feedback system when the continuous-time controller is replaced with an approximate discrete-time controller. Moreover, FOPID controllers have some features which are not preserved after approximating them with discrete transfer functions. For example, one important feature of every FOPID controller is the so-called *long memory principle*, which is lost after approximating it with an integer-order transfer function. Another property that is lost after approximation is the optimality of controller. In fact, the optimal controller designed for a certain continuous-time process will no longer be optimal after approximating it with a discrete-time system. In addition to the above-mentioned points, it should also be noted that the derivative term of (4) cannot exactly be realized in practice since it is a non-causal operator and a low-pass filter in series with it is required in practice. The proposed LDPID controller will remove all of these difficulties.

The rest of this paper is organized as the following. In Section 2 we introduce the proposed LDPID controller and develop two methods for its tuning. Three simulated examples and an experimental study are presented in Section 3. Finally, Section 4 concludes the paper.

2. DISCRETE-TIME FRACTIONAL-ORDER PID CONTROLLER

2.1. Formulation of the proposed controller

Consider the FOPID controller given in (4). In the following first we develop a method for discretizing the derivative term of this controller based on the prewarped Tustin method and then we extend the results to the integrative term. Next, based on these results we will propose a LDPID controller which can be thought of as the discrete-time dual of FOPID controllers.

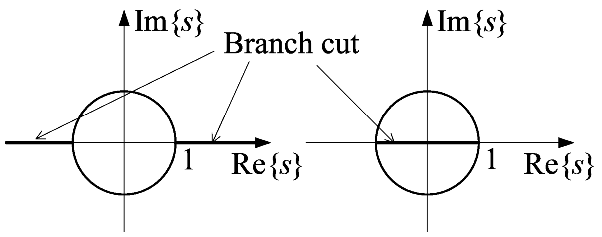


Fig. 1. Two possible branch cuts for (5).

Assuming that the sampling period of system is equal to T, applying prewarped Tustin method leads to the following approximation for the derivative term of (4):

$$s^{\mu} = \left(\frac{\omega_c}{\tan(\omega_c T/2)} \times \frac{1-z^{-1}}{1+z^{-1}} \right)^{\mu} = \alpha^{\mu} \left(\frac{1-z^{-1}}{1+z^{-1}} \right)^{\mu}, \quad (5)$$

where $\alpha \triangleq \omega_c / \tan(\omega_c T/2)$ and ω_c is the gain crossover frequency of the open-loop transfer function. (Here we have applied Tustin method since it is more accurate compared to other transforms such as backward difference. Section 2.3. makes a connection between different possible transforms.) To proceed, we need to calculate the PSE of the expression in the right hand-side of (5), which cannot be performed without determining its region of convergence (ROC) in the complex z-plane. Note that this expression is actually a multi-valued function of z which has two branch points at $z=1$ and $z=-1$ located on the unit circle. Mathematically, the corresponding branch cut (BC) can be considered either inside or outside the unit circle as shown in Fig. 1. Clearly, the choice of BC affects the ROC and consequently, causality of the resulted system. Similar to the classical results [14], here considering the BC inside the unit circle and the ROC as $|z|>1$ yields a causal system as it is desired. Hence, assuming $w=z^{-1}$ and $|z|>1$ the PSE of the expression in the right hand-side of (5) is obtained as the following:

$$\alpha^{\mu} \left(\frac{1-w}{1+w} \right)^{\mu} = \alpha^{\mu} \sum_{k=0}^{\infty} f_k(\mu) w^k, \qquad |w|<1, \quad (6)$$

where

$$f_k(\mu) = \frac{1}{k!} \times \frac{d^k}{dw^k} \left(\frac{1-w}{1+w} \right) \Bigg|_{w=0}. \quad (7)$$

Substitution of $w=z^{-1}$ in (6), and then the resulted equation in (5) yields the following PSE for the fractional-order differentiator:

$$s^{\mu} = \alpha^{\mu} \sum_{k=0}^{\infty} f_k(\mu) z^{-k}, \qquad |z|<1, \quad (8)$$

where again the coefficients $f_k(\mu)$ are calculated from

(7). It can be shown (using Maple) that the first few coefficients in (8) are $f_0(\mu)=1$, $f_1(\mu)=-2\mu$, $f_2(\mu)=2\mu^2$, $f_3(\mu)=-\frac{4}{3}\mu^3-\frac{2}{3}\mu$, $f_4(\mu)=\frac{2}{3}\mu^4+\frac{4}{3}\mu^2$, $f_5(\mu)=-\frac{4}{15}\mu^5-\frac{4}{3}\mu^3-\frac{2}{5}\mu$, ... Note that (8) holds for both the positive and negative values of μ. Hence, one may try to expand the integral term of (4) in a similar manner and arrive at an equation like (8) in λ, but the problem with such an expansion is that the resulted series does not have infinite DC gain (considering the fact that any infinite series must be truncated in practice), which is essential for tracking the step command without steady-state error. In order to find a series approximation for $s^{-\lambda}$ in terms of z^{-1} which has infinite DC gain, first we write it as $s^{-\lambda}=(1/s)\times s^{1-\lambda}$ and then apply the prewarped Tustin method to it. Applying this technique yields

$$s^{-\lambda}=\alpha^{-\lambda}\frac{1+z^{-1}}{1-z^{-1}}\sum_{k=0}^{\infty}f_k(1-\lambda)z^{-k}, \qquad |z|>1, \tag{9}$$

where $f_k(1-\lambda)$ are again calculated from (7).

Substitution of (8) and (9) in (4) results in the following formulation for the LDPID controller

$$\begin{aligned}C_d(z)&=K_p+K_d\sum_{k=0}^{\infty}f_k(\mu)z^{-k}\\&\quad+K_i\frac{1+z^{-1}}{1-z^{-1}}\sum_{k=0}^{\infty}f_k(1-\lambda)z^{-k},\end{aligned} \tag{10}$$

where

$$K_p=k_p,\quad K_d=k_d\alpha^{\mu},\quad K_i=k_i\alpha^{-\lambda}. \tag{11}$$

Clearly, in practice the upper bound of sigmas in (10) cannot be considered equal to infinity. Restricting the number of memory units to M, the following formula is proposed for the Mth-order LDPID controller:

$$\begin{aligned}C_d(z)&=K_p+K_d\sum_{k=0}^{M}f_k(\mu)z^{-k}\\&\quad+K_i\frac{1+z^{-1}}{1-z^{-1}}\sum_{k=0}^{M}f_k(1-\lambda)z^{-k}.\end{aligned} \tag{12}$$

In the rest of this paper whenever we refer to the LDPID controller, a system with transfer function (12) is under consideration. Fig. 2 shows the block diagram of the proposed LDPID controller where C/D and D/C stand for the ideal continuous-to-discrete-time and discrete-to-continuous-time converters, respectively [15]. Obviously, in practice the C/D is realized using a sample-and-hold (or an A/D converter) and the output of adder in Fig. 2 can directly be applied to the process. For simulation in Matlab, the C/D is modeled with a sample-and-hold, and the D/C is simply omitted.

Considering the fact that the FOPID and LDPID controllers (as defined in (4) and (12), respectively) have different number of parameters to tune, and taking into account the effect of sample-and-hold, it is evident that

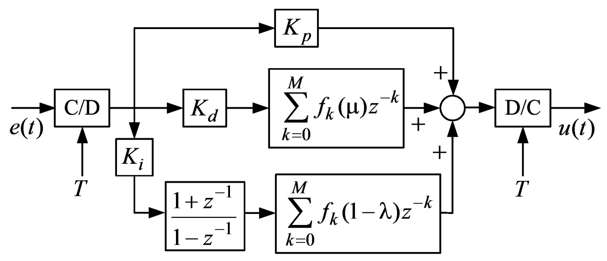

K_p
C/D
$e(t)$
T
K_d
$\sum_{k=0}^{M} f_k(\mu)z^{-k}$
K_i
$\frac{1+z^{-1}}{1-z^{-1}}$
$\sum_{k=0}^{M} f_k(1-\lambda)z^{-k}$
+
+
+
D/C
$u(t)$
T

Fig. 2. Block diagram of the Mth-order LDPID.

in practice the parameters of these two controllers cannot simply be related according to (11). In other words, the transfer function of a certain LDPID controller is not, in general, obtained by applying the Tustin transform to any FOPID controller. It concludes that the parameters of the proposed LDPID controller should be tuned directly. For this purpose, a certain value can be assigned to M and then the value of other parameters be calculated such that a predetermined set of objectives is met. The other possible approach is to consider M as a tuning parameter and then find the values of K_p, K_d, K_i, μ, λ, and M such that the objectives under consideration are met. The first approach is used in the rest of this paper.

2.2. Two methods for tuning the proposed LDPID controller

Two methods for tuning the parameters of the LDPID controller defined in (12) are developed in this section. The first method is the discrete-time equivalent of the method proposed in [16] for tuning the parameters of FOPID controller. The second method is based on minimization of a certain integral performance index by suitable choice of the unknown parameters of controller. This approach is similar to the method used in [9] for optimal tuning the FOPID controllers. In both of the methods first we assign a suitable value to M. This value should be chosen considering the limitations of the hardware used to realize the controller. Evidently, increasing the value of M increases the computational cost and the memory usage, and indeed puts a limitation on the minimum possible value for sampling period of system. (it was observed that the final results are not so sensitive to the special value assigned to M provided that other parameters are selected properly.) Note that according to (12) the difference equation relating $e[n]$ to $u[n]$ in Fig. 2 is as the following:

$$\begin{aligned}u[n]&=u[n-1]+K_p\left(e[n]-e[n-1]\right)\\&\quad+\sum_{k=0}^{M}K_df_k(\mu)\left(e[n-k]-e[n-k-1]\right)\\&\quad+\sum_{k=0}^{M}K_if_k(1-\lambda)\left(e[n-k]-e[n-k-1]\right).\end{aligned} \tag{13}$$

In (13) calculation of each sigma needs M+1 (floating point) multiplications and M+1 summations (assuming that $K_df_k(\mu)$ and $K_if_k(1-\lambda)$ are calculated beforehand and are known parameters). Hence, it can be easily verified that calculation of $u[n]$ from (13) totally needs

2(M+1)+1=2M+3 multiplications and 2(M+1)+4=2M+6 summations at each sampling period (note that (13) is not a minimal description for (12) and more effective formulations can also be obtained [15]). Considering the fact that any floating point multiplication is much more time consuming than any summation, the computational cost of the proposed controller can be approximated by the number of multiplications, which is equal to $2M+6$ at each sampling period. So, in order to determine the suitable value of M first we should determine the suitable sampling period of system and then choose the value of M such that the digital processor can perform at least $2M+6$ floating point multiplications at each sampling period.

After determining the value of M, the values of the remaining five parameters of controller are determined such that the following five conditions are satisfied simultaneously ($P(s)$ is the process transfer function, which is located in a standard unity feedback system in series with the LDPID controller):

- The gain crossover frequency of the open-loop system, ω_c, be equal to the desired value, that is the following equality holds for the desired ω_c:

$$\left|C_d(e^{j\omega_c})P(j\omega_c)\right| = 0\,\text{dB}. \tag{14}$$

- The phase margin of the feedback system, φ_m, be equal to the desired value, that is the equality

$$\arg\left\{C_d(e^{j\omega_c})P(j\omega_c)\right\} = \pi + \varphi_m, \tag{15}$$

holds for the desired φ_m.

- The feedback system exhibits a good robustness to variations in the gain of process, which can be achieved by satisfying the following equality

$$\left.\frac{d\left(C_d(e^{j\omega})P(j\omega)\right)}{d\omega}\right|_{\omega=\omega_c} = 0. \tag{16}$$

- The feedback system attenuates the high frequency noise, which is achieved by satisfying the inequality:

$$\left|\frac{C_d(e^{j\omega})P(j\omega)}{1+C_d(e^{j\omega})P(j\omega)}\right| \le A\,\text{dB} \qquad \omega \ge \omega_t \text{ rad/s}, \tag{17}$$

where A and ω_t are desired constants.

- The feedback system rejects the disturbance, which is achieved by satisfying the inequality:

$$\left|\frac{1}{1+C_d(e^{j\omega})P(j\omega)}\right| \le B\,\text{dB} \qquad \omega \le \omega_s \text{ rad/s}, \tag{18}$$

where A and ω_t are desired constants.

Similar to [16], in this paper (14) is considered as the main object of optimization and (15)-(18) are considered as the corresponding constrains. More precisely, we have applied the genetic algorithm to find the values of K_p, K_d, K_i, μ, and λ in (12) such that $||C_d(e^{j\omega_c})P(j\omega_c)|-1|$ is minimized and simultaneously the equality constraints (15) and (16), and inequality constraints (17) and (18) are fulfilled. Clearly, it may happen that the above optimization problem does not have any solution, but even approximate solutions (which violate the constrains to some extent) are useful in practice. Numerical simulations performed by authors show that the genetic algorithm toolbox of Matlab can effectively solve such a complicated constrained optimization problem in a relatively short time.

The second method that can be used for tuning the parameters of LDPID controller is to assign a certain value to M and then calculate the values of K_p, K_d, K_i, μ, and λ in (12) such that an integral performance index (e.g., the IAE or ISE performance index corresponding to the tracking error of step command) is minimized. In this method, the genetic algorithm (or any other meta-heuristic optimization algorithm such as PSO) can be used to search the five-dimensional space to find the optimal solution. For this purpose, the corresponding integral performance index should be considered as the object of minimization and the equation obtained by equating the number of unstable poles of the closed-loop system to zero can be considered as the constraint of optimization.

3.1. Generalization

The coefficients $f_k(\mu)$ and $f_k(1-\lambda)$ of the proposed LDPID controller (12) are calculated from (7). The question that may arise at this point is: Can one propose another reasonable method instead of (7) for calculation of these coefficients? To provide an answer for this question first recall that any PID controller combines a proportion, derivative and integral of the error signal to generate the control. It motivates us first to study the general behavior of discrete-time derivative operators. Based on these results we can conclude that any alternating-sign function of k provides us with a natural definition for $f_k(\mu)$ to be used in (12). Finally, we briefly extend the results to discrete-time integrators.

Suppose that we want to approximate the time derivative of the function $e(t)$ only by using its samples $e_i = e(t_i)$ $(i=1,\ldots,n)$ where $T = t_i - t_{i-1}$. Mathematicall y, this task can be performed by using one of the following formulas [17]:

$$\dot{e}(t_i) = \frac{e_{i+1}-e_i}{T} + O(T), \tag{19}$$

$$\dot{e}(t_i) = \frac{e_i-e_{i-1}}{T} + O(T), \tag{20}$$

$$\dot{e}(t_i) = \frac{e_{i+1}-e_{i-1}}{2T} + O(T^2), \tag{21}$$

which are called forward, backward, and centered difference approximations, respectively. In each case one can increase the accuracy of the resulted time derivative by using larger number of sample points. For example, backward difference approximations with second and third order errors are obtained as the following [17]:

$$\dot{e}(t_i) = \frac{3e_i - 4e_{i-1} + e_{i-2}}{2T} + O(T^2), \tag{22}$$

$$\dot{e}(t_i) = \frac{11e_i - 18e_{i-1} + 9e_{i-2} - 2e_{i-3}}{6T} + O(T^3). \tag{23}$$

The general formula for approximating the time derivative of a function with n th-order error using the backward difference method is as the following [18]:

$$\dot{e}(t_i) = \frac{1}{T}\sum_{k=-n}^{0} g_{k,n}^{B,1} + O(T^n), \tag{24}$$

where the coefficients $g_{k,n}^{B,1}$ are calculated from the following iterative procedure:

$$g_{0,n}^{B,1} = \sum_{j=1}^{n}(1/j), \qquad g_{-1,n}^{B,1} = -n, \tag{25}$$

$$g_{-k,n}^{B,1} = -g_{-k+1,n}^{B,1}(k-1)(n-k+1)/k^2, \quad k = 2,\ldots,n. \tag{26}$$

Very similar formulas can also be found in [18] for forward and centered difference approximations. The intersection point of all of these finite-difference formulas is that they approximate the time derivative of a function by the weighted sum of its samples, where these weights change sign decussately as the k is increased. For example, according to (26) it is obvious that the weights $g_{k,n}^{B,1}$ in (24) are positive for $k = 0,-2,-4,\ldots$ and negative for $k = -1,-3,-5,\ldots$. Equation (24) leads us to the fact that in digital control system design one can use the following n th-order difference equation to calculate the derivative of error signal from its samples with an arbitrary precision:

$$e_{dot}[i] \approx \frac{1}{T}\sum_{k=0}^{n} g_{-k,n}^{B,1} e[i-k], \tag{27}$$

where $e[i]$ and $e_{dot}[i]$ stand for the samples of error signal and its derivative, respectively, and the alternating-sign weights $g_{-k,n}^{B,1}$ are defined similar to (25) and (26). Recall that according to (12) the proposed LDPID controller relates the samples of error signal to its derivative through the following difference equation

$$e_{dot}[i] = K_d \sum_{k=0}^{M} f_k(\mu) e[i-k], \tag{28}$$

where the weights $f_k(\mu)$ are calculated from (7). It can be easily verified that the plot of $f_k(\mu)$ versus k (assuming a certain value for μ) changes sign decussately similar to the weights $g_{-k,n}^{B,1}$ in (27). It means that the derivative term of the proposed LDPID controller also has the property of subtracting many two successive (positive-weighted) samples of the error signal and then forming the cumulative sum of results. However, the difference equation (28) is, compared to (27), advantageous in the way that one can adjust the amplitude of weights simply by changing the value of μ (note that in case of using (27) we have no control on the amplitude of the weights $g_{-k,n}^{B,1}$ for the given n). For this reason, it is not surprising if the derivative action of the LDPID controller as given in (28) reduces to a classical discrete-time differentiator for some μ and M. For example, assuming $\mu = 0.5$ and $M = 1$ equation (28) yields $e_{dot}[i] = K_d(e[i] - e[i-1])$, which is the classical backward difference approximation for derivative operator.

The above discussion motivates us to define, in general, the derivative action of discrete-time LDPID controllers through the difference equation (28) where $f_k(\mu)$ ($\mu > 0$) can be considered equal to any alternating-sign function of k (which is not necessarily calculated from (7)). This definition for discrete-time derivative operator is consistent with the one used in Section 2.1 and the classical higher-order one presented in (27). However, application of, e.g., the backward difference approximation in (5) leads to a discrete-time approximation for derivative operator that is not consistent with the above generalized definition and a more general definition is needed to cover this case. The integral term of the proposed LDPID controller can be generalized similar to the above approach. In fact the generalized integrator can be defined as an operator that generates the positive-weighted sum of the current and past samples of error signal.

3. ILLUSTRATIVE EXAMPLES

Example 1: The aim of this example is to show that designing a classical PID controller for a FOPTD process and then discretizing it may lead to an unstable feedback system, while direct tuning the proposed LDPID controller can remove this difficulty. For this purpose consider a FOPTD process with transfer function $P(s) = 2e^{-3s}/(1+10s)$, which can be effectively controlled by the PID controller $C(s) = 1.1 + 0.1/s + 0.4s$ obtained by trial and error. The dashed curve in Fig. 3 shows the unit step response of the feedback system in this case. As it can be observed, the command following is quite satisfactory.

Now let us examine the performance of this feedback system when the controller is discretized and then applied. In this example the gain crossover frequency of the open-loop system is $\omega_c \approx 0.21$ rad/s and the phase margin is about 60°. Applying prewarped Tustin method to controller assuming $T = 0.1$ s leads to the following discrete-time controller:

$$C_{d1}(z) = 1.1 + 0.005\frac{1+z^{-1}}{1-z^{-1}} + 8\frac{1-z^{-1}}{1+z^{-1}}. \tag{29}$$

Fig. 4 shows the Bode plots of $C(s)P(s)$ and $C_{d1}(e^{sT})P(s)$ (the latter corresponds to the open-loop transfer function when the discrete-time controller is applied in series with ideal C/D and D/C converters). This figure clearly shows that the closed-loop system with discrete-time controller is unstable, while the original continuous-time feedback system was stable with a satisfactory phase margin (recall that any spike in

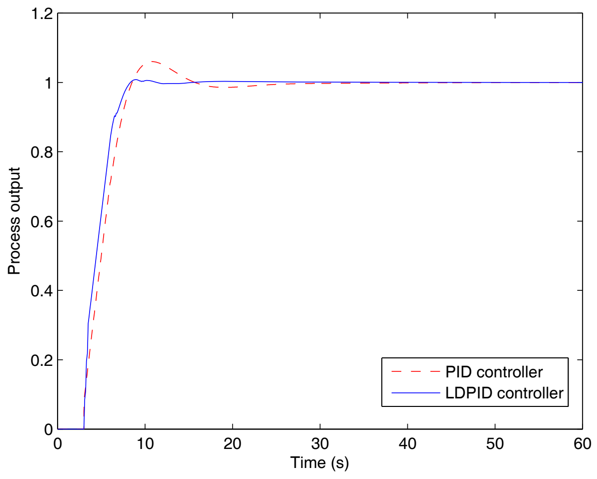


Fig. 3. Unit step response of the closed-loop system, corresponding to Example 1.

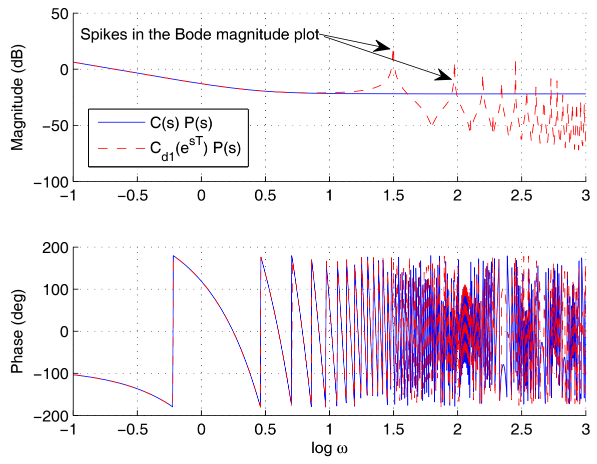


Fig. 4. Bode plots of $C(s)P(s)$ and $C_{d1}(e^{sT})P(s)$, corresponding to Example 1.

the Bode magnitude plot of $C_{d1}(e^{sT})P(s)$ in Fig. 4 corresponds to an encirclement around -1 in the corresponding Nyquist plot). One can also perform a time-domain simulation to justify the fact that the discretized controller (29) leads to an unstable feedback system. It is worth to mention that in this example one can hardly arrive at a stable feedback system only by changing to sampling period of system.

Assuming $M=5$ in (12), the (suboptimal) LDPID controller which minimizes the IAE performance index (corresponding to the tracking error of the unit step command) is obtained as the following:

$$C_{d2}(z)=2.8+1.5\sum_{k=0}^{5} f_k(1.03)z^{-k} \\ +0.004\frac{1+z^{-1}}{1-z^{-1}}\sum_{k=0}^{5} f_k(-0.1)z^{-k}. \tag{30}$$

The main advantage of this controller over $C(s)$ is that the unwanted effects caused by using ZOH are also taken into account during the controller design, and consequently, it is ready to be realized using microprocessors. Fig. 3 shows the unit step response of the closed-loop system when the LDPID controller (30) is applied. As it can be observed, the LDPID controller results in a satisfactory transient response.

Example 2: The following non-minimum phase transfer function appears in the one-link flexible arm robot [19]:

$$P(s)=\frac{-4.906s^2-0.5884s+335.17}{s^4+0.55437s^3+139.6s^2+27.91s}. \tag{31}$$

$P(s)$ has a non-minimum phase zero at 8.2057 and four poles at 0, 0.2, $0.1772\pm j11.8109$. Controlling a system with transfer function (31) is a relatively difficult task since trivial PID controllers often do not lead to satisfactory results when the process has both the non-minimum phase zero and complex conjugate poles with a very small damping ratio (here we have $\zeta=0.0150$) [1]. In the following we try the proposed LDPID controller.

Since in this example the process itself has a pole at the origin, a LDPD controller is sufficient for the tracking of step command without steady-state error. In order to design the LDPD controller first we arbitrarily assume $M=5$ and then we obtain the parameters of the LDPD such that the IAE performance index (corresponding to the tracking error of step command) is minimized. Next, we slightly modify the parameters of the resulted controller by trial and error such that the closed-loop system exhibits a desired step response. This approach leads to the following LDPD controller:

$$C_d(z)=0.3+0.5\sum_{k=0}^{5} f_k(0.8)z^{-k}. \tag{32}$$

Unit step response of the corresponding closed-loop system is shown in Fig. 5. In this figure the rise time, the settling time and overshoot of the response are approximately equal to 5.4s, 15s, and 14%, respectively.

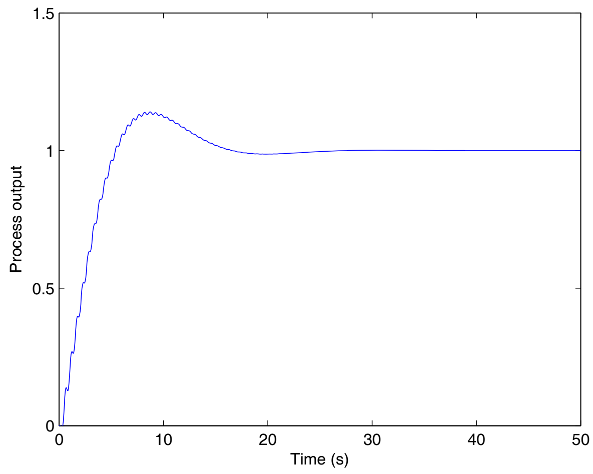


Fig. 5. Unit step response of the closed-loop system, corresponding to Example 2.

Numerous numerical simulations performed by authors show that a response with this characteristics cannot be achieved by applying any PD or PID type controller (i.e., either the rise time or the settling time or the overshoot of the response obtained by using a PID controller will be larger than the corresponding one obtained by using the proposed LDPD controller).

Example 3: Astrom and Hagglund [5] proposed the so-called AMIGO method for tuning the two degrees-of-freedom (2DOF) PID controllers, which have six parameters to tune. Application of this method to a process with transfer function $P(s)=e^{-s}/(1+0.05s)^2$ leads to a 2DOF PID controller with the following input-output relation:

$$u(t)=0.24(y_{sp}-y_f)+0.515\int_0^t(y_{sp}(\tau)-y_f(\tau))d\tau-0.032\frac{dy_f(t)}{dt}, \tag{33}$$

where y_{sp} is the set point, $Y_f(s)=G_f(s)Y(s)$ the filtered process variable, $u(t)$ the control, $G_f(s)=1/(1+0.1s)^2$, and $y(t)$ is the process output. On the other hand, minimization of the IAE performance index (i.e., the integral of the absolute error when the unit step command and the unit step disturbance are applied at $t=0$ and $t=10$, respectively in the standard unity feedback connection) leads to the following (suboptimal) LDPID controller:

$$C_d(z)=0.5+0.15\sum_{k=0}^{15}f_k(1.15)z^{-k}+7\times10^{-4}\frac{1+z^{-1}}{1-z^{-1}}\sum_{k=0}^{15}f_k(-0.2)z^{-k}. \tag{34}$$

Fig. 6 shows the process output and the corresponding control variable when the PID controllers (33) and (34) are applied. As it can be observed in this figure both controllers lead to almost the same step response, and both of them apply almost the same control effort. It concludes that the proposed controller can compete the 2DOF PID controller tuned by using the AMIGO method. However, the proposed controller is advantageous in the way that it is in the discrete-time form and ready for realization.

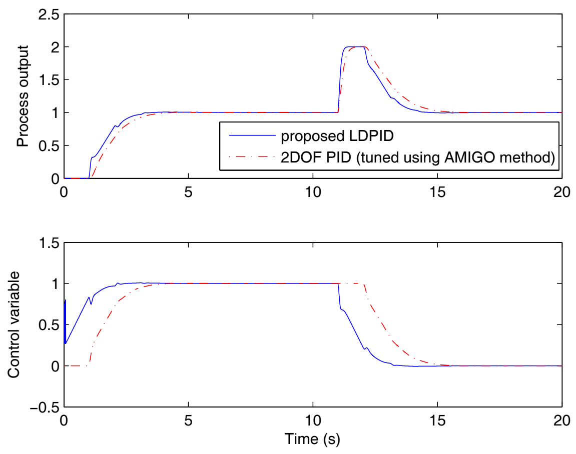


Fig. 6. Process output and control variable, corresponding to Example 3.

Example 4: In this example we study the application of the proposed LDPID controller for temperature control of an industrial heating box. In this system a DC voltage is applied to the heating element wire of the box and a PWM with adjustable duty cycle is used to control the on-off time of this wire. The DC voltage applied to wire does not have a certain value. In fact, an industrial rectifier provides a very high DC voltage which is applied to the series connection of the wires of many boxes of this type. Hence, the DC voltage across the wire of each box (when it is on) strictly depends on the number of wires connected in series (typically, in practice few hundred boxes are used in series).

For many reasons this process constitutes a difficult control problem. Firstly, the DC voltage applied to each wire (which, of course, affects the DC gain of process) has an uncertain value. In fact, in practice it is observed that this voltage may even be subjected to 200% changes from the nominal value. It is also observed that changing this voltage also changes the time constant of process. Secondly, the time constant of the process also depends on the type and weight of the material embedded in the box (to be warmed). Thirdly, taking into account the function of PWM it is obvious that the system is actually nonlinear. More precisely, the negative control signals generated by controller are simply neglected by PWM. Note also that applying a kind of anti-windup technique is also mandatory for digital realization of integrators.

At different working conditions each box can (very approximately) be modelled by the following first-order uncertain transfer function

$$P(s)=\frac{K}{1+sT}, \quad 14<K<34, \quad 380<T<570. \tag{35}$$

(In the above transfer function the input is duty cycle and the output is temperature in °C). The nominal process model is also considered as $P(s)=32/(1+425s)$ (note that the probability of occurring different uncertainties is not the same). Our aim here is to design a controller which leads to $\omega_c=1$ rad/s, $\varphi_m\geq 75^\circ$, $\omega_t=10$ rad/s, $\omega_s=0.1$ rad/s, and A=B=–20 dB (see Section 2.2). Moreover, according to the high uncertainty in the process model it is highly desired that the open-loop phase plot be as flat as possible at frequencies around ω_c.

Following the procedure presented in Section 2.2 the genetic algorithm leads to the following PID and LDPID controllers (T= 0.1 s):

$$C(s)=7.937+\frac{1.187}{s}-0.935s, \tag{36}$$

$$C_d(z)=7.109+0.711\sum_{k=0}^{5}f_k(0.077)z^{-k} \tag{37}$$

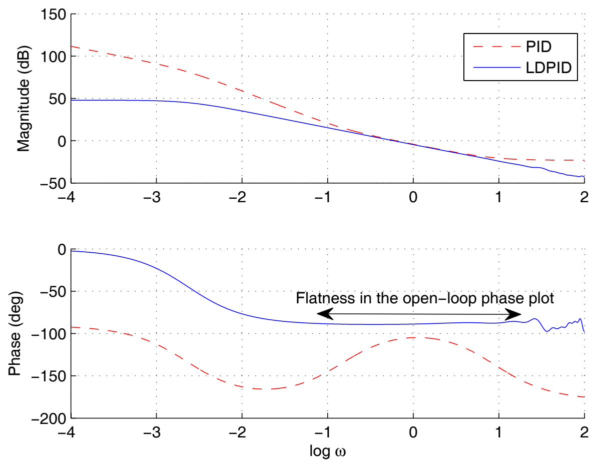


Fig. 7. Bode plots of $C(s)P(s)$ and $C_d(e^{sT})P(s)$, corresponding to Example 4.

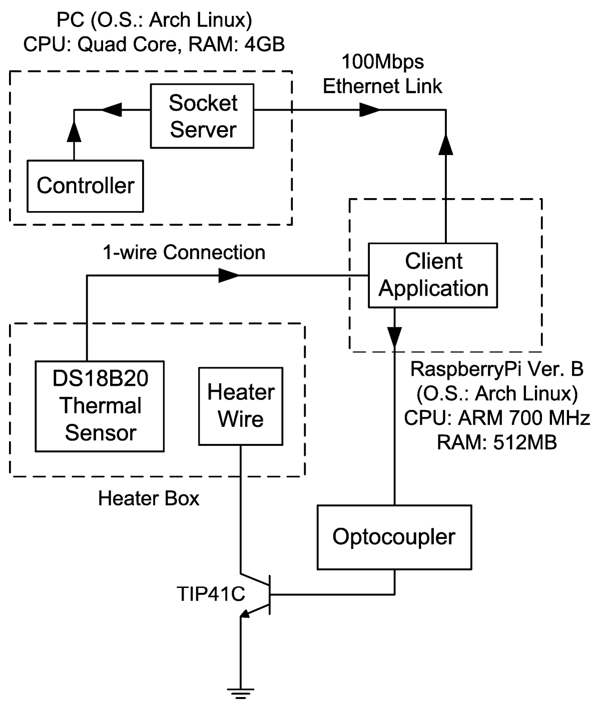


Fig. 8. The digital system used to realize the controllers of Example 4.

$$+0.750\frac{1+z^{-1}}{1-z^{-1}}\sum_{k=0}^{5} f_k(0.415)z^{-k}.$$

Fig. 7 shows the Bode phase and magnitude plots of $C(s)P(s)$ and $C_d(e^{sT})P(s)$ when the nominal process model is considered. In this example, PID and LDPID controllers lead to phase margins equal to 73.3° and 91°, respectively. As it can be observed in this figure the LDPID controller has perfectly satisfied the design requests. Especially, unlike the PID controller, the proposed LDPID has led to a very flat curve in the phase plot (for more than 2 decades), which is highly desired in dealing with the uncertain process under consideration. Using trivial simulations it can be easily verified that

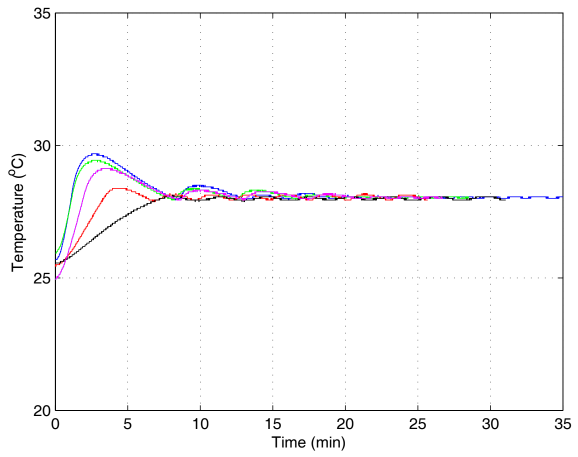


Fig. 9. Step responses of the practical closed-loop system when LDPID is applied, corresponding to Example 4.

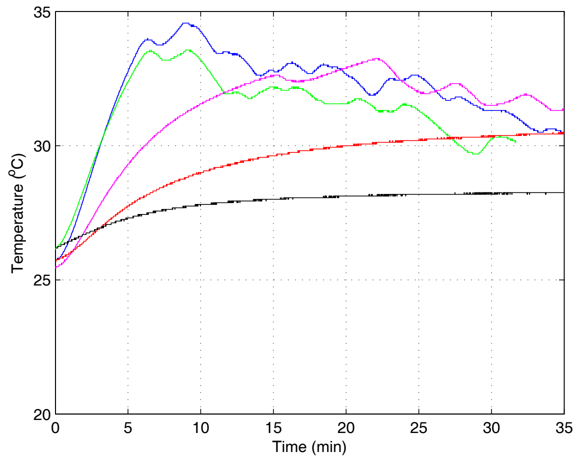


Fig. 10. Step responses of the practical closed-loop system when classical PID is applied, corresponding to Example 4.

both controllers lead to very similar and satisfactory time-domain responses when the nominal linear process model is considered.

Both of the above mentioned controllers are realized using the digital system shown in Fig. 8 (the classical PID is discretized using the Tustin method with pre-warping). Here it is worth to mention that since the 1-wire output of DS18B20 digital sensor cannot directly be connected to PC, a kind of transducer is needed. The RaspberryPi in Fig. 8 is used for this purpose. The input of optocoupler is connected to the software-generated PWM with frequency 50Hz. It was observed that in dealing with LDPID controller calculation of each control signal takes about 2ms in practice.

The time-domain responses of the practical closed-loop system when PID and LDPID controllers are applied (assuming that the reference temperature is equal to 28°C) are shown in Figs. 9 and 10, respectively in five different conditions. Note that the curves with similar colors in Figs. 9 and 10 are obtained under exactly the same conditions in practice (i.e., exactly the same DC voltages across the heating wire, the same materials in the box, etc.). Note also that the system itself has an initial condition and the vertical axis in Figs. 9 and 10 begins from 20°C. Fig. 9 clearly shows the superiority of the proposed LDPID controller. In fact, the maximum overshoot (in the worst case) caused by PID and LDPID controllers is equal to 23.43% and 6.1%, respectively. Moreover, fluctuations in the response are settled down much faster when LDPID is applied.

It should be emphasized that since the PWM cannot generate negative voltages (i.e., the process only has a heater but not a cooler) it is observed in Fig. 9 that the temperature is increased and decreased with two different time-constants. Note also that according to the thermal capacity of heater, the temperature in the box keeps increasing even after turning off the heater.

Consequently, since the derivative term of classical PID applies much larger controls compared to LDPID, it leads to larger overshoots and settling times in the response as it is observed in Fig. 10.

4. CONCLUSION

In this paper we proposed a new formulation for discrete-time fractional-order PID controllers. Experimental and numerical examples were also presented which showed that the proposed controller is capable of solving complicated control problems and has some advantages to the classical PID controllers. We also developed two methods for tuning the parameters of this controller. The main advantages of the proposed controller are: application of non-local derivative operator (as well as integrator) for calculation of error signal, direct realization of the derivative operator without the need to a series low-pass filter, taking into account the unwanted effects caused by using the sample-and-hold (such as decreasing PM) during the controller design, and applying adjustable number of memory units for realization of the controller.

REFERENCES

[1] K. J. Astrom and T. Hagglund, *Advanced PID Control*, ISA-The Instrumentation, Systems, and Automation Society, 2006.

[2] A. Visioli, *Practical PID Control*, Springer-Verlag, London, 2006.

[3] K. Ogata, *Discrete-Time Control Systems*, 2nd ed., Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[4] W.-D. Chang, R.-C. Hwang, and J.-G. Hsieh, “A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach,” *Journal of Process Control*, vol. 12, no. 2, pp. 233-242, February 2002.

[5] K. J. Astrom and T. Hagglund, “Revisiting the Ziegler-Nichols step response method for PID control,” *Journal of Process Control*, vol. 14, no. 6, pp. 635-650, September 2004.

[6] A. Madady, “An extended PID type iterative learning control,” *International Journal of Control, Automation and Systems*, vol. 11, no. 3, pp. 470-481, June 2013.

[7] M. Farahani and S. Ganjefar, “Intelligent control of static synchronous series compensator via an adaptive self-tuning PID controller for suppression of torsional oscillations,” *International Journal of Control, Automation and Systems*, vol. 10, no. 4, pp. 744-752, August 2012.

[8] V. Feliu-Batlle, R. Rivas-Perez, and F. J. Castillo-García, “Simple fractional order controller combined with a smith predictor for temperature control in a steel slab reheating furnace,” *International Journal of Control, Automation and Systems*, vol. 11, no. 3, pp. 533-544, June 2013.

[9] F. Merrikh-Bayat, “General rules for optimal tuning the $PI^{\lambda}D^{\mu}$ controllers with application to first-order plus time delay processes,” *Canadian Journal of Chemical Engineering*, vol. 90, no. 6, pp. 1400-1410, December 2012.

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

[11] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, 1999.

[12] J. A. T. Machado, “Discrete-time fractional-order controllers,” *Journal of Fractional Calculus and Applied Analysis*, vol. 4, no. 1, pp. 47-66, 2001.

[13] Y.-Q. Chen and K. L. Moore, “Discretization schemes for fractional-order differentiators and integrators,” *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 3, pp. 363-367, 2002.

[14] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals and Systems*, 2nd ed., Prentice Hall, 1996.

[15] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd ed., Prentice Hall, NJ, 2010.

[16] C. A. Monje, B. M. Vinagre, V. Feliu, and Y.-Q. Chen, “Tuning and auto-tuning of fractional order controllers for industry applications,” *Control Engineering Practice*, vol. 16, no. 7, pp. 798-812, July 2008.

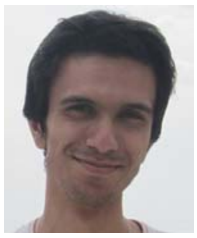
[17] R. W. Hornbeck, *Numerical Methods*, Quantum Publishers, New York, 1975.

[18] I. R. Khan and R. Ohba, “Closed-form expressions for the finite difference approximations of first and higher derivatives based on Taylor series,” *Journal of Computational and Applied Mathematics*, vol. 107, no. 2, pp. 179-193, July 1999.

[19] B.-S. Chen and T.-Y. Yang, “Robust optimal model matching control design for flexible manipulators,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, no. 1, pp. 173-178, March 1993.

Farshad Merrikh-Bayat received his B.Sc. in Electronics from K. N. Toosi University of technology in 2002, and M.Sc. and Ph.D. in Electrical Engineering (control) from Sharif University of Technology, in 2005 and 2009, respectively, all from Tehran, Iran. He is the author of 5 books, 16 journal and 22 conference papers. His research interests include meta-heuristic optimization algorithms, nonlinear dynamics and control, Fractional-order systems and memristive systems.

Nafiseh Mirebrahimi received her B.Sc. and M.Sc. degrees in Electrical Engineering (electronics) from the University of Zanjan, Iran, in 2009 and 2014, respectively. She is currently preparing for Ph.D. degree at the same university. Her research interests include digital signal processing and memristive systems.



Mohammad Reza Khalili received his B.Sc. degree in Computer Engineering from the University of Zanjan, Iran in 2014. His research interests include robotics and digital control systems.