IJAMML 15:1 (2021) 31-51 September 2021
ISSN: 2394-2258
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/ijamml_710012219

OPTIMAL TUNING OF FRACTIONAL ORDER $PI^{\lambda}D^{\mu}$ CONTROLLER USING WOUND HEALING ALGORITHM BASED ON CLONAL SELECTION PRINCIPLE

Mehmet Çinar

Tatvan Vocational School, Bitlis Eren University, Bitlis, Turkey

Abstract

Fractional-order PID (FOPID) controller is a generalization of standard PID controller using fractional calculus. Compared to PID controller, the tuning of FOPID is more complex and remains a challenge problem. This paper focuses on the design of FOPID controller using wound healing algorithm (WHA) based on clonal selection principle. The tuning of FOPID controller is formulated as a nonlinear optimization problem, in which the objective function is composed of overshoot, steady-state error, raising time and settling time. WHA algorithm, a newly developed evolutionary algorithm inspired by human immune system, is used as the optimizer to search the best parameters of FOPID controller. The designed WHA-FOPID controller is applied to various systems. Numerous numerical simulations and comparisons with other FOPID/PID controllers show that the WHA-FOPID controller can not only ensure good control performance with respect to reference input but also improve the system robustness with respect to model uncertainties.

*Corresponding author.

E-mail address: mcinar@beu.edu.tr (Mehmet Çinar).

2020 Mathematics Subject Classification: 68-W50.
Submitted by Jianqiang Gao.
Received August 6, 2021

Keywords: fractional-order systems, wound healing algorithm, $PI^{\lambda}D^{\mu}$ controller.

---

1. Introduction

The most important expectation from control systems; As a result of the change of input conditions, the output responds to this input in the shortest time and in a stable manner. The selection of a controller that meets the design criteria of the system is very important. For this reason, while selecting the controller, the most economical controller should be selected according to the system grade and control criteria. To achieve the performance characteristics expected from the control system, the process called tuning of the controller parameters is the acquisition of controller parameters. One of the most important problems in controller design is finding the best values for the controller parameters. Therefore, much attention has been paid to this issue. The best known methods for finding controller parameters for classical PID controllers are given by Ziegler-Nichols [1] and Åström-Hägglund [2]. Particularly if the mathematical model of the system to be controlled is unknown, the controller parameters can be obtained by using the Ziegler-Nichols method. With the Åström-Hägglund method, appropriate parameters that will provide the desired phase and gain margin are obtained. Applications of fractional order computation in control engineering started with Tustin's position control for large objects in 1958 [3-5]. Later in the 1960s, Manabe worked on noninteger integrals and their applications to control systems [A6], and system design using noninteger integrals [7].

A generalized and accepted method such as Routh Hurwitz, Root Locus, which is used in the stability analysis of integer order control systems, is not yet seen in the literature for the stability analysis of fractional order control system (FOCS). However, there are new studies examining the stability of FOCS in the literature. New studies on this subject can be in the form of adapting methods such as Routh-Hurwitz, which are used for integer order systems, to FOCS, as well as newly developed stability analysis methods. Therefore, serious studies are still needed in order to obtain techniques that can be used in this regard. Especially in recent years, new studies on finding the parameters of fractional-order PID controllers have been accelerating. The fractional order $PI^{\lambda}D^{\mu}$ controller was first presented in the work of Podlubny [8]. In this controller, λ represents the degree of the integral and μ the degree of the derivative. Since these controller parameters are real numbers, they also contain the integer values of the PID controller. Therefore, the fractional order $PI^{\lambda}D^{\mu}$ controller can be applied to both fractional order systems and integer order systems and provides better results.

The presence of five $(K_p, K_I, K_D, \lambda, \mu)$ parameters of the fractional order $PI^{\lambda}D^{\mu}$ controller makes the fractional order $PI^{\lambda}D^{\mu}$ controller more flexible and wider control area compared to the classical PID controller [9]. Since the integral and derivative degrees of the fractional order $PI^{\lambda}D^{\mu}$ controller can be real numbers, it also includes integers. This situation also paves the way for the fractional $PI^{\lambda}D^{\mu}$ controller to be a conventional PID controller if desired. Figure 1 is important in terms of comparing the fractional order $PI^{\lambda}D^{\mu}$ controller and the classical PID and components it contains, PI controller and PD controller control regions, and their flexibility against each other [10].

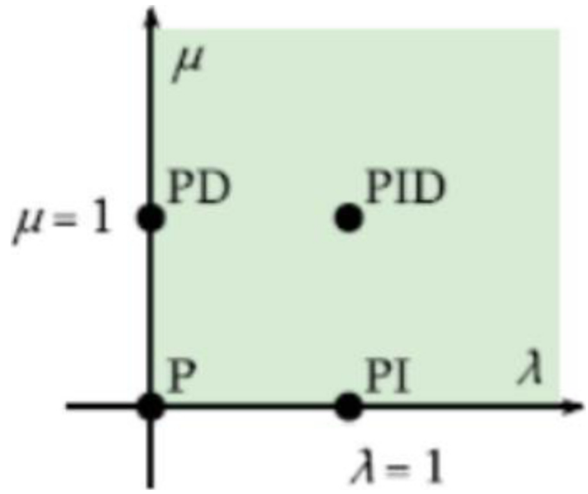

Figure 1. $PI^{\lambda}D^{\mu}$ controller control region.

As seen in Figure 1, the fractional $PI^{\lambda}D^{\mu}$ controller contains the classical PID controller and its components, the PI controller and the PD controller. In addition, the fractional $PI^{\lambda}D^{\mu}$ controller provides control in a wider and more flexible region than these controllers. When the necessary conditions are met, the fractional $PI^{\lambda}D^{\mu}$ controller can be converted to the classical PID controller, PI controller and PD controller by selecting the appropriate λ and μ values.

The transfer function of the fractional order $PI^{\lambda}D^{\mu}$ controller is given in Equation (1).

$$C(s) = K_p + \frac{K_I}{s^{\lambda}} + K_D s^{\mu}. \tag{1}$$

If $\lambda = 1$ and $\mu = 1$ in Equation (1), classical PID controller is obtained.

The block diagram of a feedback control system is shown in Figure 2.

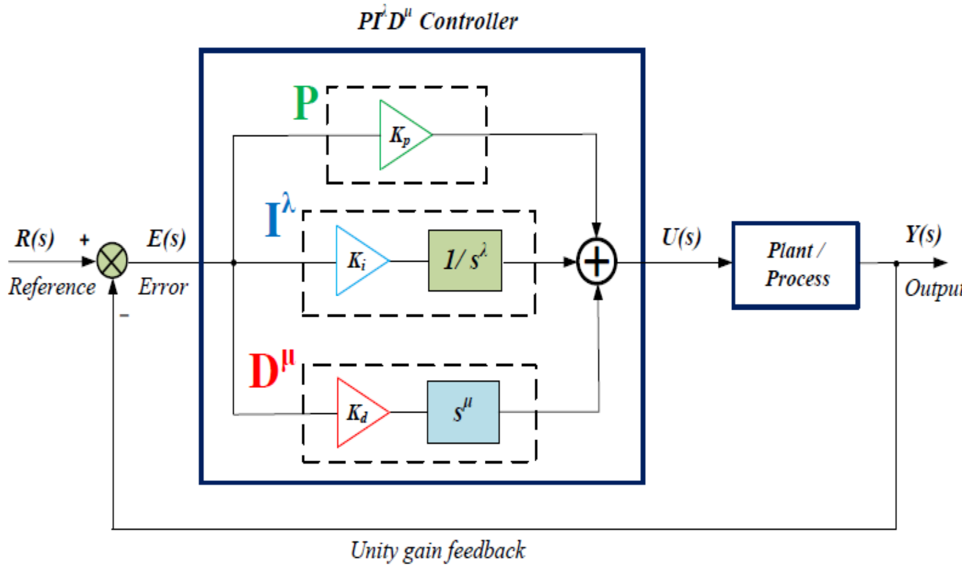

Figure 2. Block diagram of a feedback system [11].

2. Fractional Order Systems

Fractional order systems are a generalization of integer order systems. The non-integer derivative and integral operator ${}_aD_t^{\alpha}$ are defined as in Equation (2). Here a and t represent the upper and lower limits of the integral operation, α is the fractional degree.

$$
{}_aD_t^{\alpha} = \begin{cases} \dfrac{d^{\alpha}}{dt^{\alpha}}; & \alpha > 0, \\ 1; & \alpha = 1, \\ \displaystyle\int_a^t (d\tau)^{-\alpha}; & \alpha < 1. \end{cases} \tag{2}
$$

There are many definitions for fractional derivatives and integrals, the most popular being Grünwald-Letnikov and Riemann-Liouville. Grünwald-Letnikov and the integral definition are given by Equation (3) and Equation (4). Here h denotes the step interval.

$$ {}_aD_t^{\alpha} f(t) = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^j \frac{\Gamma(\alpha+1)}{j!\,\Gamma(\alpha-j+1)} f(t-jh), \tag{3} $$

$$ {}_aD_t^{-\alpha} f(t) = \lim_{h \to 0} h^{\alpha} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} \frac{\Gamma(\alpha+1)}{j!\,\Gamma(\alpha)} f(t-jh). \tag{4} $$

The Grünwald-Letnikov definition can be explained by taking a function $f(t) = \sin(t)$ as an example. It is known that the integer derivative of the sine function is obtained with a phase shift of 900 of the sine function. The transfer function of a fractional dynamical system is expressed as:

$$ G(s) = \frac{Y(s)}{R(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \ldots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \ldots + a_0 s^{\alpha_0}}. \tag{5} $$

Various methods have been developed to express fractional order systems with integer order approximation models. Among these, Matsuda and Fujii [12], Oustaloup et al. [13], Carlson and Halijak [14], Chareff et al. [15], and CFE (continuous fraction expansion) [16] can be counted as the most well-known methods.

3. Wound Healing Algorithm

The basic structure of the wound healing algorithm is based on clonal selection [17]. The number of clones generated in the clonal selection algorithm appears in Equation (6) [18].

$$ Nc = \sum_{i=1}^{n} \mathrm{round}\left(\frac{N}{i}\right), \tag{6} $$

where

Nc: Number of clones produced from each Ag;

N: Total Ab number;

n: Number of antibodies selected;

Ab: Solution population;

Ag: Antigen population.

In the developed wound healing algorithm, the parameters α (cloning factor) and f (acceleration factor of the cloning process) have been added to Equation (6). This can be seen from Equation (9). These additional parameters gave better results. The flow chart of the developed algorithm is given in Figure 3. The starting population P is produced first. The selection process then selects n antibodies with the best affinity to create a new population of P_n. Affinity means the attraction that binds the antibody to the antigen, namely the immune response. The basic principle within the selection process is the affinity value of the antibodies. Individuals in this population are cloned using a cloning process in order to create a new population. The number of clones is related to the affinity value of the antibodies. After that, with the help of the hypermutation process, the clones are transferred to create a new population. The fundamental rule of the mutation process: high-affinity antibodies have a lower mutation rate or vice versa. Clones with low-affinity values have a higher mutation rate. This is because antibodies closer to the local optimum value are closer to the solution value; but antibodies that are far from optimal solution. They undergo excessive mutations to move towards the optimum or the best solution. The re-selection process checks whether the best clones are better than their family. Finally, low-affinity antibodies are replaced by new antibodies. The process of selection, cloning, and mutation move the population towards the best solution.

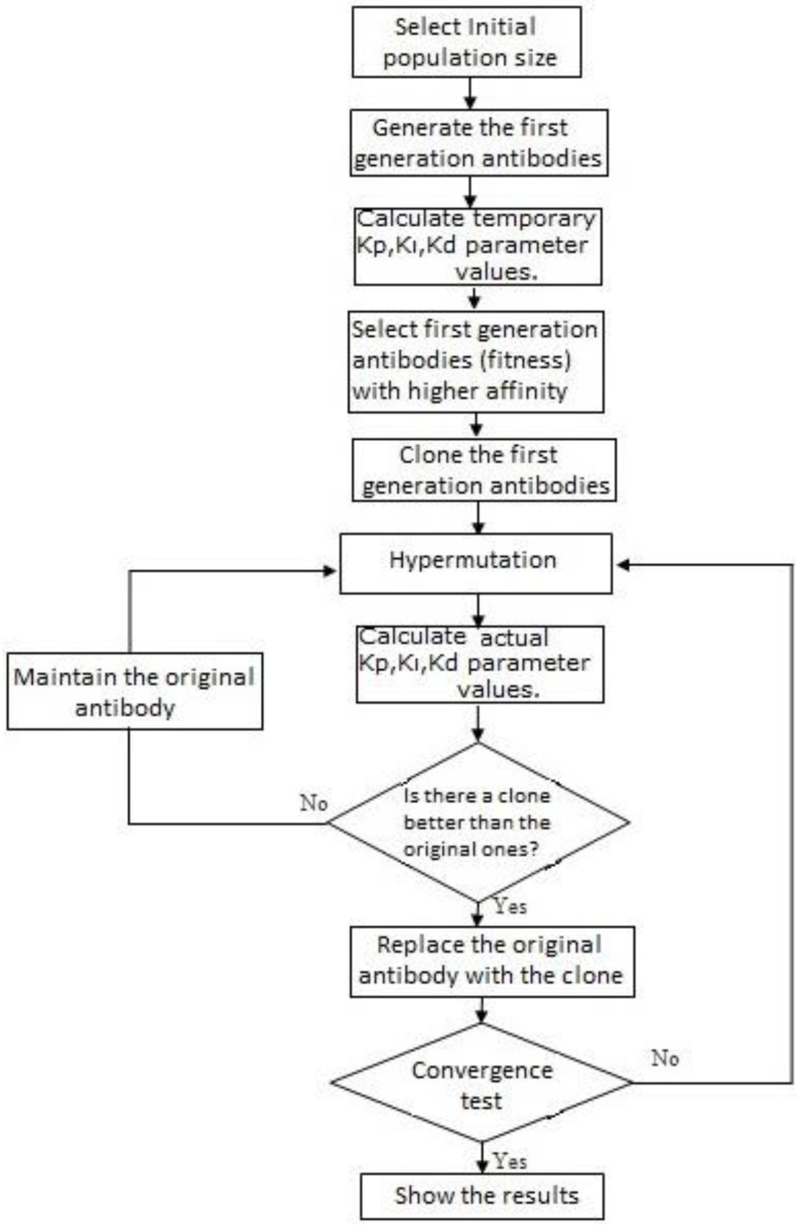

Figure 3. The flowchart of wound healing algorithm.

3.1. Wound healing algorithm solution stages

1. Generate the initial population (P) with N antibodies.

2. Determine affinity for each antibody in the P population. Select n antibodies with the best affinity (N_s) and generate the P_n population. The affinity value between antibody and antigen is calculated by Equation (7):

$$d = \sum_{i=1}^{N} (Ag_i - Ab_i)^2. \tag{7}$$

The calculated threshold value d is compared with λ and the E marking error is calculated as follows:

$$E = d - \lambda. \tag{8}$$

If $E > 0$, the antibody does not recognize the antigen and there is no affinity between them. If the E value is between 0 and 1, there is an affinity between them.

3. Clone the n antibodies selected in Step 2 and create temporary clone population N_c. Equation (9) is used to form N_c.

$$Nc_i = \text{round}\left(\frac{\alpha^* N_s{}^* f}{i}\right), \tag{9}$$

where

α: Cloning coefficient (value ranges from 0 to 1);

f: Cloning acceleration factor (value range from 0.9 to 0.99);

N_s: Best number of antibodies selected in Step 2;

4. Hypermutate the N_c clone population. Build the subpopulation N_c^*. Hypermutation is proportional to the affinity value of antibodies [19].

5. Calculate the affinity value of each antibody of the N_c^* subpopulation and select the antibodies with the best value from which to generate $N_{c(n)}^*$ and add to the initial population.

6. Replace antibodies with low affinity values by new antibodies.

7. If the value of the P population is less than $N,$ produce antibodies to complete the population. Converge your test. If the test is successful, stop the program. Otherwise, continue the process.

4. Implementation of the Optimization Algorithm

The performance of control systems is often evaluates from the point of transient behaviour. It is expected from control systems that the rise time, settling time and percent overshootvalue should not be high. These values should be as small as possible. In a control system, the controller design is of great importance to obtain the transient behaviour parameters of the system. One of the methods used to design the system behaviour with the controller is optimization. Optimization is to use the available data in the most efficient way, to minimize or maximize a function mathematically. While optimization methods are used in control systems, objective functions are used to minimize the error in the system. These objective functions used are called integral performance criteria. The error occurring in the control system is generally the difference between the input and the output and is shown in Equation (10).

$$e(t) = r(t) - y(t). \tag{10}$$

Table 1. Provides integral performance criteria that are frequently used in control systems

Performance indices		
1	$J_{\mathrm{ISE}} = \int_0^{\infty} e^2(t)\,dt$	Integral squared error (ISE)
2	$J_{\mathrm{IAE}} = \int_0^{\infty} \lvert e(t)\rvert\,dt$	Integral absolute error (IAE)
3	$J_{\mathrm{ITSE}} = \int_0^{\infty} t\,e^2(t)\,dt$	Integral time multiplied squared error (ITSE)
4	$J_{\mathrm{ITAE}} = \int_0^{\infty} t\lvert e(t)\rvert\,dt$	Integral time multiplied absolute error (ITAE)

Let's consider the following fractional order system to perform PID controller design according to the proposed method:

4.1. Example transfer function

$$G(s) = \frac{1}{2.2s^{2.2} + 1.2s^{1.2} + 0.2s^{0.2}}. \tag{11}$$

The fractional order system is modelled by using Oustaloup's 5th order integer approach given in the Appendix. In this example, the error in the control system is minimized by using the ITAE (Integral time multiplied absolute error) criterion. Taking $C(s) = 1$, the controllerless closed-loop transfer function is obtained as follows:

$$T(s) = \frac{\begin{array}{c} s^5 + 56.87s^4 + 442.3s^3 + 531.7s^2 \\ + 98.83s + 2.512 \end{array}}{\begin{array}{c} 5.526s^7 + 222.4s^6 + 1373s^5 + 2113s^4 + 1558s^3 \\ + 736.1s^2 + 112.2s + 2.712 \end{array}}. \tag{12}$$

Figure 4 shows the step response of the closed loop transfer function (Equation (12)) calculated with the help of Matlab program by taking $C(s) = 1$.

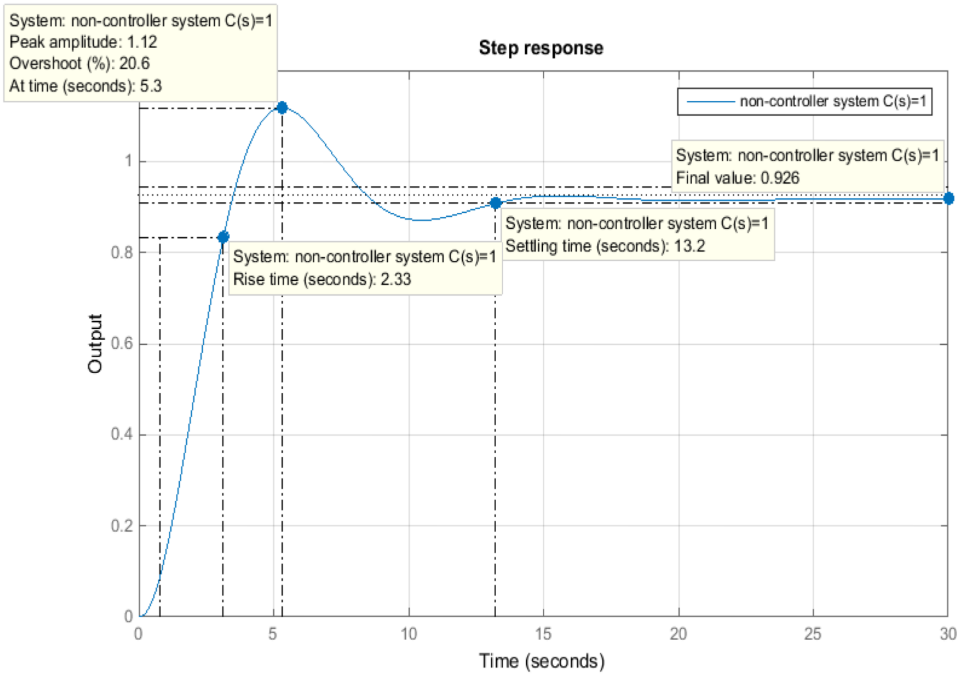

Figure 4. Step response of the closed loop transfer function of the system in Equation (11) $(C(s) = 1)$.

As seen in Figure 4, when the system controller is not used, the overshoot is 20.6%, the settling time is 13.2 seconds, the rise time is 2.3 seconds, and the steady state value is 0.926. The system does not provide the steady state error value and is unstable. The system needs to be controlled by the controller. PID controller is used for this. The simulation model created to determine the PID controller parameters is given in Figure 5.

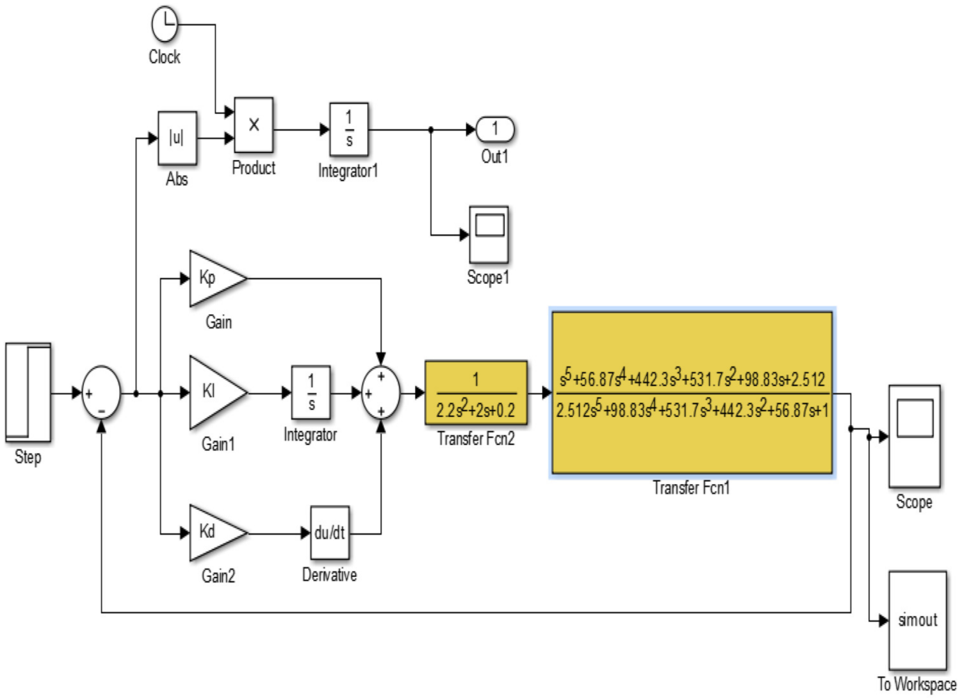

Figure 5. Closed loop control system's Simulink model with ITAE.

The proposed wound healing algorithm was used to determine the PID controller parameters (K_p, K_I and K_D). The parameters and their values used in the algorithm are given in Table 2.

Table 2. Wound healing algorithm parameters and initial values

Parameter	Value	Parameter	Value
Population number	150	Affinity coefficient	1.3
Iteration number	150	Clone coefficient (α)	0.4
Run number	30	Clone acceleration (f)	0.95
Healing coefficient	1.3		

Wound healing algorithm is a stochastic algorithm and the program was run 30 times to obtain optimum K_p, K_I, and K_D values. The screenshot of the program developed to calculate the optimum PID parameters of the Equation (11) with the aid of the wound healing algorithm is Figure 6.

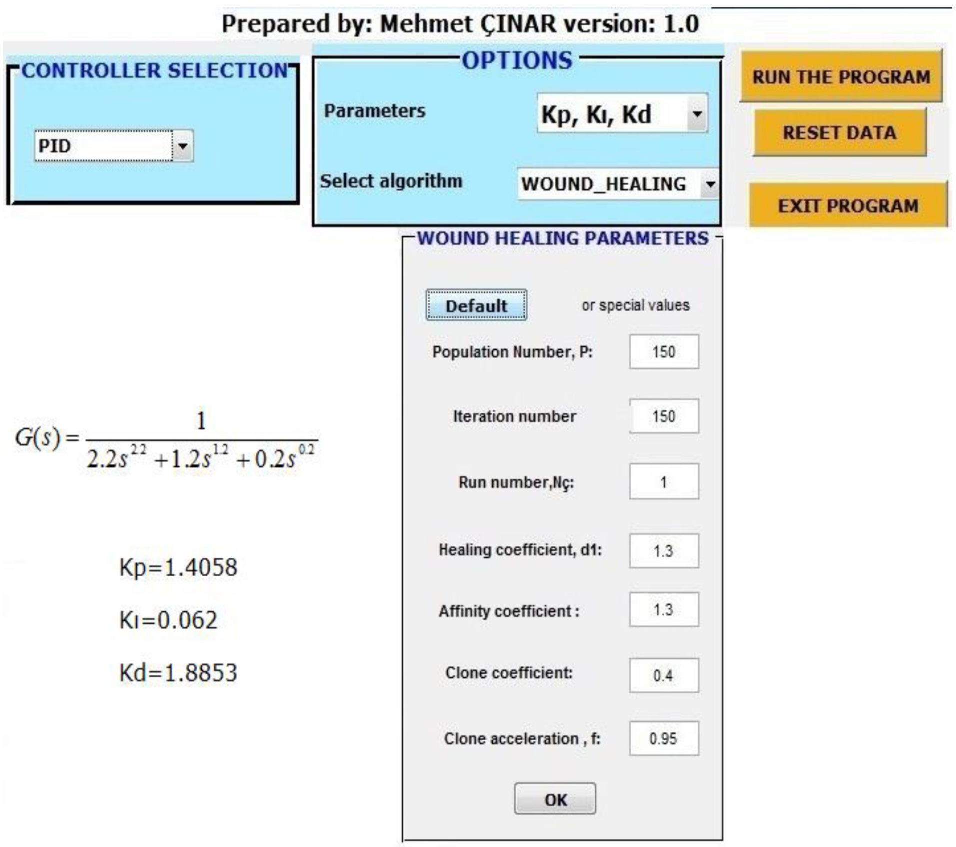

Figure 6. Developed program.

Optimization is initiated by entering the initial values into the controller parameters in the wound healing algorithm. When the stopping criterion is met, the controller parameters are determined. For the system given in Equation (11), with the ITAE performance criterion, K_p, K_I, and K_D are obtained as 1.4058, 0.062, and 1.8853, respectively.

Substituting these parameters in the PID controller equation, the following equation is obtained:

$$C(s) = K_p + \frac{K_I}{s} + K_D s = 1.4058 + \frac{0.062}{s} + 1.8853s. \tag{13}$$

If the system in Equation (11) is controlled with PID controller parameters as in Equation (13), the following closed-loop transfer function is obtained with the help of Matlab program.

$$T(s) = \frac{\begin{array}{r} 1.88s^7 + 108.6s^6 + 913.9s^5 + 1628s^4 + 961.2s^3 \\ + 176.6s^2 + 9.659s + 0.1557 \end{array}}{\begin{array}{r} 5.526s^8 + 224.3s^7 + 1477s^6 + 2970s^5 + 2744s^4 \\ + 1166s^3 + 190s^2 + 9.859s + 0.1557 \end{array}}. \tag{14}$$

Figure 7 shows the closed-loop unit step response curve of the system in Equation (14) calculated with the PID controller parameters obtained as a result of the wound healing algorithm. As seen in Figure 4, while the percent exceedance value was 20.6% in the system before the controller was applied, it decreased to 0% after the controller was applied. While the settling time was 13.2 seconds before the controller and 3.48 seconds after the controller, while the steady state error was 0.926 before the controller was applied, the system became stable by taking the value of 1 after the controller. Thus, with the PID controller parameters calculated as a result of the wound healing algorithm, the percent overshoot value and steady state error were eliminated in the system and the settling time was considerably shortened.

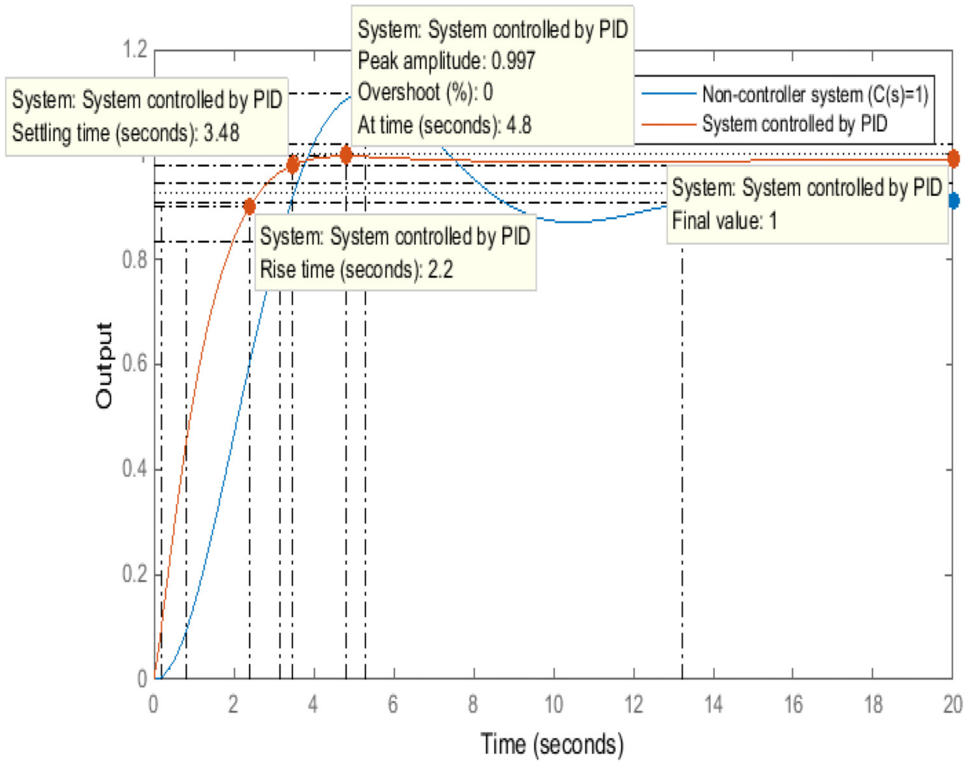

Figure 7. Step response for $G(s)$ with PID.

5. Conclusion

Optimization algorithms are one of the most common methods used to solve functions that are difficult to solve by analytical means and have a high process complexity. Although there are many analytical methods for controller design of control systems, these methods have many processing steps and sometimes require advanced mathematical operations. In this case, using an optimization algorithm for controller design is very useful in terms of shortening the process at the design stage and avoiding possible operational errors. However, parameters can be determined with the desired precision.

In this study, a program was developed in Matlab GUI environment for PID controller design with the aid of a wound healing algorithm. For this, an easy-to-use interface has been designed in a GUI environment by using the high mathematical computing capability of the Matlab program. Thus, a fast and easy design process has been passed, an application has been developed in which the PID controller coefficients can be optimized with the wound healing algorithm and the results can be presented graphically.

References

[1] K. Ogata, Modern Control Engineering, Prentice Hall, New Jersey, 2002.

[2] K. Aström and T. Hagglund, PID Controllers: Theory, Design and Tuning, Instrument Society of America, North Carolina, 1995.

[3] C. Ma and Y. Hori, Fractional-order control: Theory and applications in motion control, IEEE Industrial Electronics Magazine 1(4) (2007), 6-16.

DOI: https://doi.org/10.1109/MIE.2007.909703

[4] A. Tustin, J. T. Allanson, J. M. Layton and R. J. Jakeways, The design of systems for automatic control of the position of massive objects, Proceedings of the IEE - Part C: Monographs 105(1S) (1958), 1-57.

DOI: https://doi.org/10.1049/pi-c.1958.0001

[5] C. Ma, Experimental verification of fractional order controller's robustness using torsion test bench, Proceedings of the Chinese Control and Decision Conference (CCDC), 17-19 June 2009, Guilin, China, 335-340.

DOI: https://doi.org/10.1109/CCDC.2009.5195075

[6] S. Manabe, The Non-integer integral and its application to control systems, The Journal of the Institute of Electrical Engineers of Japan 80(860) (1960), 589-597.

DOI: https://doi.org/10.11526/ieejjournal1888.80.589

[7] S. Manabe, The system design by the use of non-integer integral and transport delay, The Journal of the Institute of Electrical Engineers of Japan 81(878) (1962), 1803-1812.

DOI: https://doi.org/10.11526/ieejjournal1888.81.1803

[8] I. Podlubny, Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers, IEEE Transactions on Automatic Control 44(1) (1999), 208-214.

DOI: https://doi.org/10.1109/9.739144

[9] V. Çelik, Effects of Fractional Order Structures on the Nonlinear Behaviors, PhD. Thesis, Firat Universitesi, Türkiye, 2010.

[10] A. Tepljakov, E. Petlenkov, J. Belikov, FOMCON: Fractional-order modelling and control toolbox for MATLAB, Proceedings of the 18th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES), Gliwice, Poland (2011), 684-689.

[11] I. Dimeas, Design of an Integrated Fractional-Order Controller, M.Sc. Thesis, University of Patras, 2017.

[12] K. Matsuda and H. Fujii, H_∞ optimized wave-absorbing control: Analytical and experimental results, Journal of Guidance, Control, and Dynamics 16(6) (1993), 1146-1153.

DOI: https://doi.org/10.2514/3.21139

[13] 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: https://doi.org/10.1109/81.817385

[14] G. Carlson and C. Halijak, Approximation of fractional capacitors $(1/s)^{(1/n)}$ by a regular Newton process, IEEE Transactions on Circuit Theory 11(2) (1964), 210-213.

DOI: https://doi.org/10.1109/TCT.1964.1082270

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

DOI: https://doi.org/10.1109/9.159595

[16] I. Podlubny, I. Petráš, B. M. Vinagre, P. O'Leary and L. Dorčák, Analogue realizations of fractional-order controllers, Nonlinear Dynamics 29(1) (2002), 281-296.

DOI: https://doi.org/10.1023/A:1016556604320

[17] M. Çınar and A. Kaygusuz, Artificial immunity based wound healing algorithm for power loss optimization in smart grids, Advances in Electrical and Computer Engineering 20(1) (2020), 11-18.

DOI: https://doi.org/10.4316/AECE.2020.01002

[18] E. D. Kaymaz, Predicting the Cellular Localization Sites of Protein Patterns with k-NN Classification Algorithm Based Artificial Immune System, Master of Thesis, Firat University, Turkey, 2007.

[19] S. S. F Souza, R. Romero and J. F. Franco, Artificial immune networks Copt-aiNet and Opt-aiNet applied to the reconfiguration problem of radial electrical distribution systems, Electric Power Systems Research 119 (2015), 304-312.

DOI: https://doi.org/10.1016/j.epsr.2014.10.012

[20] M. S. Tavazoei, Notes on integral performance indices in fractional-order control systems, Journal of Process Control 20(3) (2010), 285-291.

DOI: https://doi.org/10.1016/j.jprocont.2009.09.005

■

Appendix

Table 1. Oustaloup's fifth-order approximation table

$s^{0.1}$	$\dfrac{1.585s^5+68.37s^4+403.3s^3+367.9s^2+51.87s+1}{s^5+51.87s^4+367.9s^3+403.3s^2+68.37s+1.585}$
$s^{0.2}$	$\dfrac{2.512s^5+98.83s^4+531.7s^3+442.3s^2+56.87s+1}{s^5+56.87s^4+442.3s^3+531.7s^2+98.83s+2.512}$
$s^{0.3}$	$\dfrac{3.981s^5+142.9s^4+700.9s^3+531.7s^2+62.36s+1}{s^5+62.36s^4+531.7s^3+700.9s^2+142.9s+3.981}$
$s^{0.4}$	$\dfrac{6.31s^5+206.5s^4+924s^3+639.3s^2+68.37s+1}{s^5+68.37s^4+639.3s^3+924s^2+206.5s+6.31}$
$s^{0.5}$	$\dfrac{10s^5+298.5s^4+1218s^3+768.5s^2+74.97s+1}{s^5+74.97s^4+768.5s^3+1218s^2+298.5s+10}$
$s^{0.6}$	$\dfrac{15.85s^5+431.4s^4+1606s^3+924s^2+82.2s+1}{s^5+82.2s^4+924s^3+1606s^2+431.4s+15.85}$
$s^{0.7}$	$\dfrac{25.12s^5+623.6s^4+2117s^3+1111s^2+90.14s+1}{s^5+90.14s^4+1111s^3+2117s^2+623.6s-25.12}$
$s^{0.8}$	$\dfrac{39.81s^5+901.4s^4+2790s^3+1336s^2+98.83s+1}{s^5+98.83s^4+1336s^3+2790s^2+901.4s+39.81}$
$s^{0.9}$	$\dfrac{63.1s^5+1303s^4+3679s^3+1606s^2+108.4s+1}{s^5+108.4s^4+1606s^3+3679s^2+1303s+63.1}$

Table 2. Matsuda's fourth-order integer approximation table

$s^{0.1}$	$\frac{1.828s^4+102.7s^3+329.8s^2+78.91s+1}{s^4+78.91s^3+329.8s^2+102.7s+1.828}$
$s^{0.2}$	$\frac{3.357s^4+161s^3+453.9s^2+95s+1}{s^4+95s^3+453.9s^2+161s+3.357}$
$s^{0.3}$	$\frac{6.227s^4+256.4s^3+635s^2+116.1s+1}{s^4+116.1s^3+635s^2+256.4s+6.227}$
$s^{0.4}$	$\frac{11.74s^4+417.1s^3+907.9s^2+144.6s+1}{s^4+144.6s^3+907.9s^2+417.1s+11.74}$
$s^{0.5}$	$\frac{22.72s^4+698.8s^3+1337s^2+185s+1}{s^4+185s^3+1337s^2+698.8s+22.72}$
$s^{0.6}$	$\frac{45.73s^4+1222s^3+2056s^2+246.3s+1}{s^4+246.3s^3+2056s^2+1222s+45.73}$
$s^{0.7}$	$\frac{98.22s^4+2287s^3+3381s^2+349.4s+1}{s^4+349.4s^3+3381s^2+2287s+98.22}$
$s^{0.8}$	$\frac{237.8s^4+4833s^3+6277s^2+557s+1}{s^4+557s^3+6277s^2+4833s+237.8}$
$s^{0.9}$	$\frac{770s^4+13690s^3+15610s^2+1182s+1}{s^4+1182s^3+15610s^2+13690s+770}$