A Software Realization of Disturbance Rejection Optimal FOPID Controller Design Methodology by Using Soft Computing Techniques

Sevilay Tufenkci, Baris Baykant Alagoz, Celaleddin Yeroglu, Bilal Senol

Abstract— **This study presents a soft computing tool for the computer-aided design of disturbance rejection FOPID controllers based on the maximization of Reference to Disturbance Ratio (RDR) index. The study illustrates the utilization of software routines to implement a soft computing scheme in order to solve a closed loop disturbance rejection FOPID control system design problem for a target gain margin specification. Authors demonstrate that the complex design efforts, which involve a high level of mathematical knowledge, can be easily performed by using basic software routines when soft computing techniques are employed effectively in the computation processes. Illustrative design examples are shown to show effectiveness of the proposed design method.**

Index Terms— **Disturbance rejection control, FOPID controller, genetic algorithm, phase margin, RDR**

I. INTRODUCTION

FRACTIONAL CALCULUS has come out at the end of the 17th century, and today it becomes a popular mathematical tool to solve modeling and design problems in engineering and applied science [1-3]. However, its utilization in control engineering practice has been delayed due to the computational load of fractional order operators. Approximate realization of fractional order operators and development of soft computation tools contributes to the solution of these computationally complex design problems.

Time-domain design methodologies can yield more relevant results for the control system design practice because they perform a design effort based on the time response of the control systems. This design strategy is practically reasonable because the control system performances of the real systems are generally evaluated according to their time responses by using test signals that are applied to the system. However, derivation of the analytical controller design rules regarding the time responses of higher order control systems or fractional order systems is rather difficult to obtain in the time domain design, and this turns into an important design limitation for development of analytical tuning rules in the time domain for complex system models. However, frequency domain analysis has an important role in control theory, since complex control systems can be easily studied in the frequency domain [4]. In the frequency domain, there are several graphical methods that can work not only with low-order systems but also with high-order systems and/or fractional order systems. The main reasons for preferring the frequency domain in the analysis and design of fractional order control systems are that stability analysis and analytical solutions in order to obtain controller design rules are more straightforward in the frequency domain due to allowing rather simple expressions of the linear time invariant high-order system models in the frequency domain compared to the time domain models. For these reasons, intensive studies have been carried out especially on the controller design based on the frequency domain modeling [5]. To evaluate behavior of the control system in the frequency domain, several graphical tools such as the Bode diagram, the Nichols and Nyquist diagrams and the root locus plots have been used.

In the frequency domain controller design, the phase margin is an important measure in order to express the robustness of the system. The phase margin is also related to controller performance; for instance, it affects the damping rate of the

SEVILAY TUFENKCI, is with Department of Computer Technology Department, Malatya Turgut Ozal University, Malatya, Turkey(e-mail: sevilay.tufenkci@gmail.com)

https://orcid.org/0000-0001-9815-7724

BARIS BAYKANT ALAGOZ, is with Department of Computer Engineering, Inonu University, Malatya, Turkey (e-mail: baykant.alagoz@inonu.edu.tr).

https://orcid.org/ 0000-0001-5238-6433

CELALEDDIN YEROGLU, is with Department of Computer Engineering, Inonu University, Malatya, Turkey (e-mail: c.yeroglu@inonu.edu.tr).

https://orcid.org/0000-0002-6106-2374

BILAL SENOL, is with Department of Computer Engineering, Inonu University, Malatya, Turkey (e-mail: bilal.senol@inonu.edu.tr).

https://orcid.org/0000-0002-3734-8807

Manuscript received Mar 27, 2022; accepted Nov 19, 2022.
DOI: 10.17694/bajece.1092971

system [6]. Several controller design topics cover the phase margin together with the gain margin [6, 7]. The results associated with phase and gain margin can be graphically displayed by using the Bode diagrams.

In real-world control applications, the control systems are exposed to a number of disturbance effects due to the system noise or external factors acting on the control system. Those disturbances may be unpredictable and they may influence the control systems in real control actions and prevent them from showing the expected control performance in the real-world systems. Therefore, it is very substantial to minimize the impacts of disturbances on the system response while designing a controller for the real-world control applications. Various studies have been presented in order to reduce the undesirable effects of disturbances on the control performance and improve the robust stability of the control systems [8-11, 12, 13].

In this study, the reference to disturbance ratio (RDR) in cooperation with the gain margin specification is employed in the design task to increase the disturbance rejection capacity of closed loop control systems. The RDR measure is used to calculate the quantitative evaluation of the dominance of reference input signal on the input disturbance signal at the system output [14, 15, 27]. While a control system shows satisfactory disturbance rejection performance in the case of $RDR >> 1$, the control system does not show any disturbance rejection skills and performance of the control system is vulnerable to the impacts of disturbances in the case of $RDR \leq 1$.

Today, closed loop control systems are widely used in many industrial control applications. The classical controller design process involves determining optimal controller coefficients in a way that responses of the designed control system meet several design specifications such as robust stability, disturbance rejection, fault tolerance etc. Therefore, multi-objective optimization methods have been widely used in the controller design problems. The optimization is the process of choosing the best possible tuning options under certain criteria. Many different optimization algorithms have been implemented to perform this task. The selection of a suitable optimization method is important in the design stage and assets of optimization methods have a role in optimal controller tuning problems. While one algorithm can find a satisfactory solution for the problem, the other optimization algorithm may not reach the desired result. Metaheuristic algorithms have been commonly utilized for obtaining the near-optimum solutions. They are preferred to reach the optimal solution in an acceptable time, particularly in the case of large-scale, mathematically not-well structured, complicated optimization problems. Today, metaheuristic optimization algorithms can be classified into several categories, for example the biology-based, the physics-based, swarm- based, the social-based etc. This algorithm has been frequently used in optimal tuning of fractional-order controls [16-18]. Recent works have been implemented the chaotic yellow saddle goatfish algorithm[19], hybrid Lévy flight distribution and simulated annealing algorithm[20], a multi-objective genetic algorithm (MOGA) and particle swarm optimization[21]. These works reveal that metaheuristic methods can deal with the complicated optimization problems for controller design, However they haven't clearly stated and discussed the soft computation details in their works. In the current study, we present soft computation details that were used for the proposed design scheme.

Mathematical complication in the solution of constrained optimization problems can be alleviated by using soft constraints [22]. The soft constraints refer to a constraint that can be violated in the optimization problem at a penalty cost and the optimization algorithm can progressively continue for the reduction or removal of the penalty costs [22]. Hard constraints cannot be violated at any cost. For further theoretical consideration, a deepened theoretical discussion on the soft and hard constraints was presented in [23] and their roles in the optimization process [24]. However, the practice of the soft constraints has appeared and developed through the soft computing applications [25, 26].

Soft computation methods can facilitate very complicated design tasks in engineering problems. In this perspective, the main motivation of this study is the proposal and implementation of some soft computation techniques in an optimal disturbance rejection FOPID controller design problem. Analytical solution of this optimization problem is quite complicated because of difficulties in fractional calculus and the mathematical modeling of environmental disturbances. Authors aim to demonstrate that such complicated design tasks can be easily performed by using basic software routines. Different from the similar works, the study introduces these soft computation routines (soft constraints, crossover frequency calculation via zero crossing detection etc.) in detail, and illustrates an application of them in a disturbance reject FOPID design problem.

In the current study, fractional order control coefficients were adjusted by combining the phase margin constraint and RDR index. The RDR index is maximized for a gain margin specification in order to improve the disturbance rejection performance of the FOPID control systems. The phase margin specification is defined as the soft constraints of the optimization problem. Thus enables software realization of the complex phase margin analyses and eliminates a need for complicated mathematical derivations. Such a softening of the hard constraints also modifies the search spaces and contributes to search performance of the metaheuristic optimization methods by allowing progressive approximation to optimal solutions. For the solution of this optimal tuning problem, authors implemented the most fundamental and popular metaheuristic search method that is a genetic algorithm. Two illustrative FOPID controller design examples were shown to evaluate the efficiency of the proposed optimal disturbance reject FOPID controller design methodology.

II. PRELIMINARIES AND PROBLEM STATEMENTS

A. RDR Analysis for Disturbance Rejection

RDR analyses were suggested for quantitative evaluation of disturbance rejection capacity of the closed-loop control systems [14, 15, 27]. Consideration of the RDR index in fractional-order FOPID control designs allows improving disturbance rejection control performance [15, 27-31]. Effective disturbance rejection controller design approaches based on the RDR index have also been presented and their improvements for disturbance rejection control were shown [27-31].

The communication channel based analysis on the negative feedback loop yields the following RDR formulation for the control systems [14, 15, 29].

$$RDR(\omega)=\left|C(j\omega)\right|^2 \quad (1)$$

Since the RDR index can take very high values, their values are expressed in decibel(dB) as follows:

$$RDR_{dB}(\omega)=20\log\left|C(j\omega)\right| \quad (2)$$

To improve disturbance rejection performance, the RDR constraint that expresses a lower boundary in the RDR spectrum was suggested for the operating frequency range $[\omega_{\min},\omega_{\max}]$ of control systems [27] and a minimum RDR constraint was expressed as

$$\min\{RDR_{dB}(\omega)\}\geq M \text{ for } \omega\in[\omega_{\min},\omega_{\max}]. \quad (3)$$

Figure 1 shows a closed loop FOPID control system with the additive input disturbance model $D(s)$ for the RDR index. The function $C(s)$ is the transfer function of FOPID controller and it is widely expressed as

$$C(s)=k_p+k_d s^{\mu}+\frac{k_i}{s^{\lambda}} \quad . \quad (4)$$

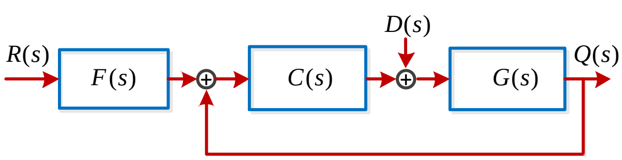

Fig.1. FOPID control system with ste-point filter

In contrast to the classical PID controller, a FOPID controller have two additional fractional order coefficients, which are fractional integrator orders λ and fractional derivative orders μ in addition to the k_p, k_d and k_i gain coefficients. These additional fractional order integral and derivative orders provide an opportunity to improve control system performance [3, 5]. By considering Equation (1), the RDR formulas of the FOPID control systems were derived as follows [15]

$$RDR(\omega)=(k_p+k_i\cos(\frac{\pi}{2}\lambda)\omega^{-\lambda}+k_d\cos(\frac{\pi}{2}\mu)\omega^{\mu})^2$$
$$+(k_d\sin(\frac{\pi}{2}\mu)\omega^{\mu}-k_i\sin(\frac{\pi}{2}\lambda)\omega^{-\lambda})^2 \quad (5)$$

The phase margin is a well-known property in classical control systems and it is expressed as an important criterion for ensuring the robustness of the system stability [6, 7]. The phase margin also affects the closed-loop damping ratio of the second order systems [6].

The phase margin is widely used for the stabilization of control systems, and the general form of the phase margin (φ_m) for the open loop transfer function $L(s)=C(s)G(s)$ is written by

$$\varphi_m=Arg\left[G(j\omega_c)C(j\omega_c)\right]+\pi\,. \quad (6)$$

Here, ω_c refers to the crossover frequency that is defined as the angular frequency that satisfy $\left|G(j\omega_c)C(j\omega_c)\right|=1$. Therefore, the calculation of the phase margin requires a solution of nonlinear, complex valued equations and it causes difficulties in the mathematical solution of this problem. The phase margin basically represents the difference between the phase at the crossover frequency of the open loop control system and the $-\pi$ angle and rather easy to solve graphically.

B. A Brief Review of Genetic Algorithm

Genetic algorithm (GA) is one of the most popular metaheuristic search techniques, which can provide easy solutions for today's complex design problems. The GA is based on the natural selection principles and the algorithm was suggested by John Holland in 1975. The basic idea in this method is based on the survival of the good individuals (solutions) in the genetic pool (solution population). It tries to find the best result or the closest one to be the best in its search space [32-34]. The search mechanism of the genetic algorithm resembles the transmission of the physical and biological characteristics of living things to the next generation through the genes. As each generation is formed by the combination of better features from the previous generations, the best individual survives with higher chance in selection mechanisms and each generation gets better as generations progress. Fitness values of each individual are used in a selection mechanism that tends to select the better fitting individuals. After selection of more fitting individuals to the solution, the next generation of individuals is reproduced through a series of genetic processes such as mutation and crossover. The basic algorithmic steps are shown in Figure 2 [35]. The GA algorithm has become a very effective evolutionary computational method.

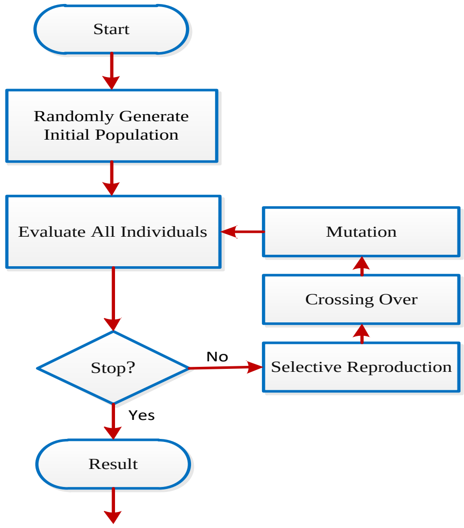

Fig.2. A fundamental flow chart for the genetic algorithm

III. SOFTWARE REALIZATION OF OPTIMAL FRACTIONAL ORDER PID CONTROL SYSTEM DESIGN

This section summarizes the mathematical background and the software realization of the computer-aided design method that are implemented for the optimal tuning problem of FOPID controls coefficients based on the maximization of RDR index and complying with the phase margin specification.

We consider a fractional plant function that is expressed in a general form as follows:

$$G(s)=\frac{a_0}{b_2 s^k + b_1 s^{\alpha} + b_0} \quad (7)$$

The closed loop transfer function, which is composed of the controlled system model (Equation (7)) and the fractional order PID controller (Equation (4)), can be written by

$$T(s)=\frac{Q(s)}{R(s)}=\frac{a_0 k_d s^{(\lambda+\mu)}+a_0 k_p s^{\lambda}+a_0 k_i}{b_2 s^{(\lambda+k)}+a_0 k_d s^{(\lambda+\mu)}+b_1 s^{(\lambda+\alpha)} +(b_0+a_0 k_p)s^{\lambda}+a_0 k_i} \quad .(8)$$

This transfer function is implemented by using the *fotf()* function of *fotf toolbox* [36]. Figure 3 shows the pseudocode that describes a software implementation of the closed loop FOPID control system model.

A pseudocode for RDR index calculation according to the equation (5) is shown in Figure 4. The code returns the minimum RDR value in a frequency range ($[\omega_{\min},\omega_{\max}]$) to implement the $RDR_c(x)=\min_{RDR}\{RDR_{dB}(\omega,x), \quad \omega\in[\omega_{\min},\omega_{\max}]\}$.

The crossover frequency (ω_c) is commonly found by solving $|G(j\omega_c)C(j\omega_c)|=1$. One can write this equation by forming open loop transfer function $L(s)=C(s)G(s)|_{s\to j\omega}$ in the frequency region and writing the magnitude of the resulting complex rational function by

$$\left|\frac{a_0 k_d (j\omega_c)^{\mu+\lambda}+a_0 k_p (j\omega_c)^{\lambda}+a_0 k_i}{b_2 (j\omega_c)^{k+\lambda}+b_1 (j\omega_c)^{\alpha+\lambda}+b_0 (j\omega_c)^{\lambda}}\right|=1 \,. \quad (9)$$

```
Algorithm fractional order transfer function is
Inputs: Coefficients of G(s) a0,b2,b1,b0,
        Order of G(s) k, alpha
        Coefficients of C(s) kp,kd,ki,
        Order of C(s) u, lamda
        Coefficients of T(s) c0,c1,c2,c3,c4,d2,d1,d0
        Order of T(s) beta0, beta1, beta2, beta3,
beta4, gamma0, gamma1, gamma2,
Outputs: Symbolic model of T(s) Ts,

% Coefficients of closed loop transfer function
c0=a0*ki;
c1=a0*kp+b0;
c2=a0*kd;
c3=b1;
c4=b2;
beta0=0;
beta1=lamda;
beta2=lamda+u;
beta3=lamda+alfa;
beta4=k+lamda;
d2=a0*kd;
d1=kp*a0;
d0=ki*a0;
gamma2=(lamda+u);
gamma1=lamda;
gamma0=0;
% Software realization of the closed loop transfer
function by using fotf toolbox
Ts=fotf([c4 c3 c2 c1 c0],[beta4 beta3 beta2 beta1
beta0],[d2 d1 d0],[gamma2 gamma1 gamma0]);
Return Ts
```

Fig.3. A pseudocode for implementation of fractional order transfer functions by using fotf toolbox

```
Algorithm minimum RDR is
Inputs: Coefficients of C(s) kp,kd,ki,
        Order of C(s) u, lamda,
        Angular frequency vector w,
         RDR value vector RDRdB
Outputs: Minimum RDR value minRDRdB

% Calculation of logarithmic RDR values according
equations(5)
RDRdB=20*log10((kp+ki*cos((pi/2)*lamda)*w.^(-
lamda)+kd*cos((pi/2)*u)*w.^(u)).^2+(kd*sin((pi/2)*
u)*w.^(u)-ki*sin((pi/2)*lamda)*w.^(-lamda)).^2);

% Calculation of minimum RDR index
minRDRdB =min(RDRdB);
Return minRDRdB
```

Fig.4. A pseudocode for RDR the index and the minimum RDR calculation for the realization of the equation (9)

The equation 9 requires a nonlinear equation solving method to find crossover frequency (ω_c). To solve this complex equation by using the soft computing technique, one can form a zero crossing by taking the logarithm of both side of the equation and obtain $f(\omega_c) = \log_{10}|G(j\omega_c)C(j\omega_c)| = 0$. To solve this equation and roughly find the sampled crossover frequency ω_c and the phase at the crossover frequency ($Arg[G(j\omega_i)C(j\omega_i)]$), a zero-crossing detection mechanism as depicted in Figure 5 is used. Thus, this nonlinear equation can be easily solved by using zero crossing detection with an error (tolerance) less than the unit sampling distance ($|\omega_i - \omega_{i-1}|$).

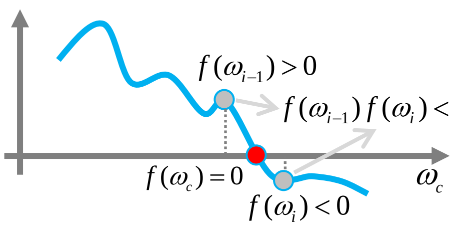

Fig.5. A soft computing technique for searching the crossover frequency ω_c via the zero crossing detection

Figure (6) shows a pseudocode for the zero-crossing detection algorithm that can be used for approximate calculation of the crossover frequency as in Figure 5.

```
Algorithm zero-crossing detection is
Inputs: Value vector of the function f,
        Angular frequency vector w,
Outputs: Crossover frequency wc
-----------------------------------------
For i is from 1 to length of f
if f(i-1)*f(i)<0
wc=w(i);
end
Return wc
```

Fig.6. A pseudocode for the zero-crossing detection algorithm that can be used for approximate calculation of the crossover frequency

Then, the objective function to minimize is written as [29]

$$E(x) = \left(\frac{1}{RDR_c(x) + \varepsilon_0} \right)^2, \tag{10}$$

where $\varepsilon_0 > 0$ is a very small real number in order to avoid zero divisions in the case of a zero value of RDR_c. Here, the vector $x = [k_p, k_i, k_d, \lambda, \mu]$ is the controller coefficients. The standard definition of the optimization problem is written as

$$\min_x E(x), \tag{11}$$

$$\text{S.t.:} \quad |\varphi_m - \varphi_T| < \varepsilon. \tag{12}$$

This objective function maximizes the RDR_c index that allows an improved input disturbance rejection performance for the additive input disturbance signal $D(s)$ (See Figure 1). The phase margin specification of the system is introduced by the inequality constraint. This phase margin constraint is necessary for the stabilization of the control system. The target phase margin value is set $\varphi_T = \frac{2\pi}{3}$ for robust stabilization of the system. According to Equation (12), the parameter φ_m stands for the phase margin of the current system, the φ_T denotes the target phase value. The ε is a small positive number to define an approximation of system phase margin (φ_m) to the target phase margin (φ_T) within the ε tolerance range. This range allows the enhancement of RDR performance during the optimization process of genetic algorithms in the expanse of a small allowable deviation from target phase margin. Thus it softens the phase constraint and facilitates the constrained optimization problem ((Equations (11) and (12)). Figure 7 shows a pseudocode to implement this optimization problem according to a soft constraint $|\varphi_m - \varphi_T| < \varepsilon$. In this code, when the phase constraint $|\varphi_m - \varphi_T| < \varepsilon$ is not satisfied, the absolute phase margin error $|\varphi_m - \varphi_T|$ is amplified by a factor of 10^{10} in order to reduce search possibility of such candidate solutions in the search space. Assignment of very high error values for these candidate solutions prevents the survival of them in next generations and eliminates these solutions that violate the soft constraint.

```
Algorithm objective function is
Inputs: Target phase margin phi_m,
          Calculated phase margin phi_t,
           Minimum RDR value minRDRdB
Outputs: Objective function to minimize E
-----------------------------------------
if | phi_m- phi_t| <0.01
E=(1/(( minRDRdB)^2)+0.001)
else
E=| phi_m- phi_t| *10^10
end
Return E
```

Fig.7. A pseudocode for solving optimization problem according to a soft constraint ($\varepsilon = 0.01$, $\varepsilon_0 = 0.001$)

A software realization of this design problem in Matlab is carried out by employing soft computing techniques. This avoids the requirement of solving very difficult analytical equations to develop practical computer-aided design tools. The Matlab codes that were written for this design tool are presented in the Appendix section.

IV. ILLUSTRATIVE EXAMPLES

In this section, two illustrative design examples are presented.

Example 1: Let's design a disturbance rejection FOPID control for the fractional order plant model given below [37]:

$$G(s)=\frac{1}{0.8s^{2.2}+0.5s^{0.9}+1} \tag{13}$$

The closed loop transfer function of the system is written as

$$T(s)=\frac{k_d s^{(\lambda+\mu)}+k_p s^{\lambda}+k_i}{0.8s^{(\lambda+2.2)}+k_d s^{(\lambda+\mu)}+0.5s^{(\lambda+0.9)}+(1+k_p)s^{\lambda}+k_i}\,. \tag{14}$$

Target phase margin for the control system is set as $\varphi_T=\frac{2\pi}{3}$. Optimal FOPID controller coefficients were searched by using GA optimization for the fractional order controller parameter ranges LB = [1 1 1 0.3 0.3] and UB = [500 500 500 2 2] according to $x=[k_p,k_i,k_d,\lambda,\mu]$. The optimal disturbance reject FOPID controller was obtained as

$$C(s)=307.4346+145.9923s^{0.8984}+\frac{452.6156}{s^{1.2533}}\,. \tag{15}$$

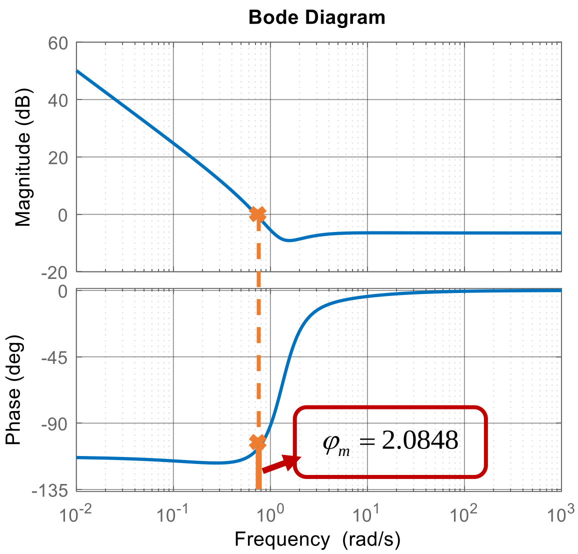

Fig.8. Bode diagram of the designed control system

The stability of systems can be evaluated by examining the Bode diagram. When the amplitude graph is 0 dB, the distance of the system phase to -180 degrees ($-\pi$ radian) represents the phase margin. In Figure 8, the phase margin is shown with a thick orange solid line. The phase margin value for the designed system is obtained as $\varphi_m = 2.0848$. Figure 9 shows the step responses of the proposed FOPID controller and Zhao et al.'s optimal FOPID controller. Figure 10 shows a close view of Figure 9 in order to compare step performances in terms of the rise time, the settling time and the maximum overshoot. The figure indicates improvement of the step response performance by means of the proposed FOPID controller in this example.

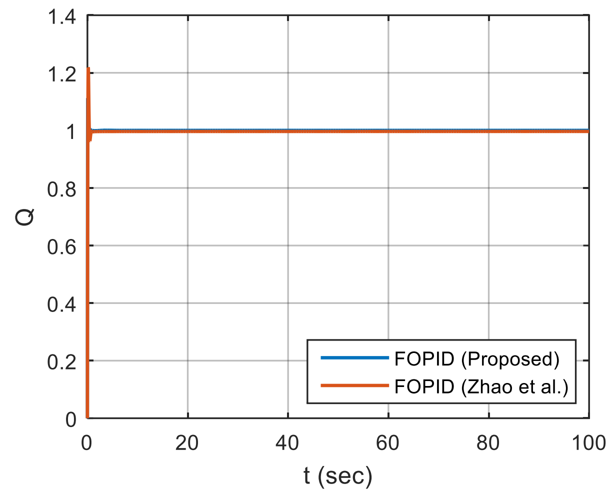

Fig.9. Step responses of the control systems

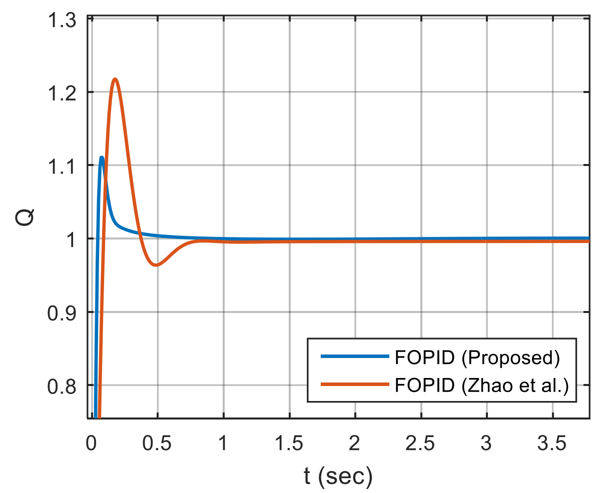

Fig.10. A close view of the step responses in Figure 9

Figure 11 shows the RDR spectrum of the FOPID control systems. Higher RDR values indicate the improved disturbance rejection control performance in the corresponding frequency. The proposed FOPID controller can provide higher RDR values in the majority of the RDR spectrum and this indicates the proposed FOPID controller can present a better disturbance rejection performance, particularly at the higher frequency region compared to Zhao et al.'s FOPID controller. To further improve setpoint control performance, authors used the 2DOF setpoint filter FOPID control system (See Figure 1) in disturbance rejection control simulations. The setpoint filter is configured as $F(s)=\frac{1}{3s+1}$ that avoids high overshoot step response when settling to the setpoint 1 as in Figure 12. (Zhao et al.'s FOPID control system does not include a set-point filter) These simulations were performed in the Simulink environment by using the *fotf toolbox* [36]. Figure 12 shows step disturbance responses of the controllers. An additive step disturbance was applied to the control system at 40 sec simulation time. Figure 13 shows a close view of disturbance rejection control for both controllers.

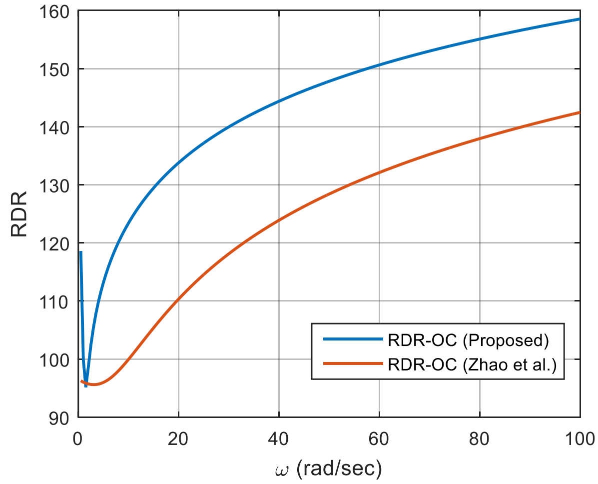

Fig.11. The RDR spectrums of the control systems

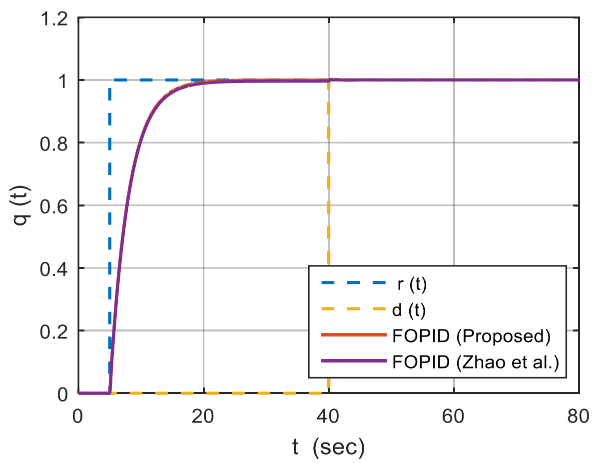

Fig.12. Step responses and disturbance responses of the FOPID controllers in case of an additive step disturbance insertion into the plant input at 40 sec

Figure 14 shows system responses when a sinusoidal disturbance signal with the amplitude of 1 and a frequency of 3.14 rad/sec was inserted into the control system. The figure shows that the disturbance rejection control performance is improved by the proposed FOPID control system in this frequency.

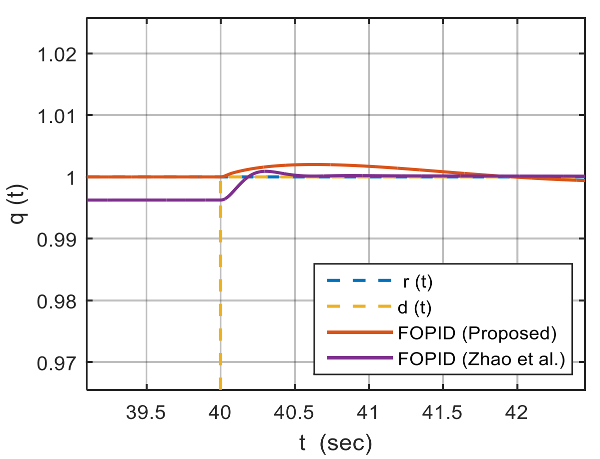

Fig.13. A close view of the disturbance responses in Figure 12

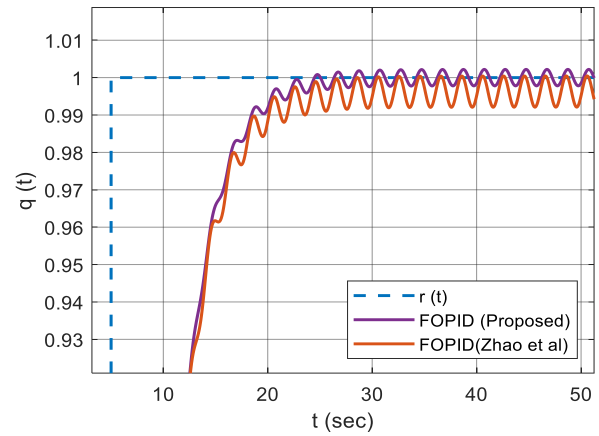

Fig.14. Sinusoidal disturbance responses of the control systems

Example 2: Let's design a disturbance rejection FOPID control for the fractional order plant model given below [38]:

$$G(s) = \frac{0.5}{1.5s^{1.3} + 1} \tag{16}$$

The closed loop transfer function of the system is written as

$$T(s) = \frac{0.5k_d s^{(\lambda+\mu)} + 0.5k_p s^{\lambda} + 0.5k_i}{0.5k_d s^{(\lambda+\mu)} + 1.5s^{(\lambda+1.3)} + (1+0.5k_p)s^{\lambda} + 0.5k_i} \tag{17}$$

A target phase margin to be achieved by using the control system is set as $\varphi_T = \frac{2\pi}{3}$. The optimal FOPID controller coefficients obtained by using GA optimization for a FOPID parameter search ranges LB = [1 1 1 0.5 0.3] and UB = [50 50 50 2 2] according to $x = [k_p, k_i, k_d, \lambda, \mu]$. Then, optimal disturbance reject FOPID controller design is obtained as

$$C(s) = 49.9128 + 18.3780s^{0.7535} + \frac{32.1662}{s^{0.5358}}. \tag{18}$$

Figure 15 shows a Bode diagram of the designed control system. The phase margin is indicated with a thick orange solid line in this figure. The phase margin value for the designed system is obtained as $\varphi_m = 2.1027$. Figure 16 shows step responses of the proposed FOPID controller and Tabatabaei et al.'s optimal FOPID controller[38]. Figure 17 shows a close view of Figure 16 in order to compare step performances. The figure indicates the improvement of the step response performance by means of the proposed FOPID controller in this example.

Figure 17 shows the RDR spectrum of the FOPID control systems. The proposed FOPID controller provides a superior RDR performance, and this indicates that the proposed FOPID controller presents a better disturbance rejection control performance compared to the Tabatabaei et al.'s FOPID controller. To improve setpoint control performance, authors used 2DOF setpoint filter FOPID control system (See Figure 1) in disturbance rejection control simulations for the proposed FOPID controller. The setpoint filter is configured as $F(s) = \frac{1}{3s+1}$ that avoids the high overshoot step response when settling to the setpoint 1. (Tabatabaei et al.'s FOPID control system does not include a set-point filter)

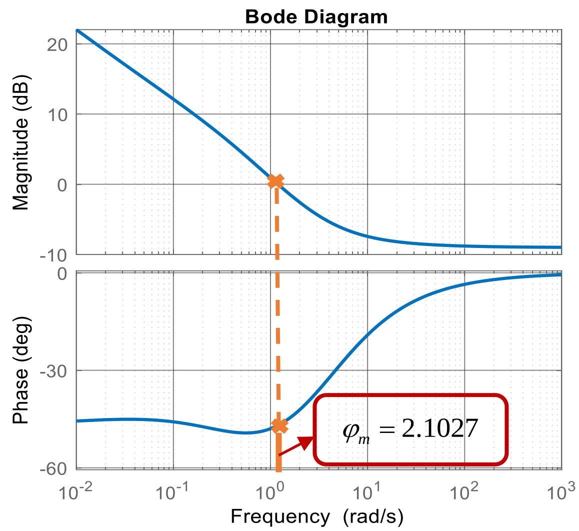

Fig.15. Bode diagram of the proposed control system

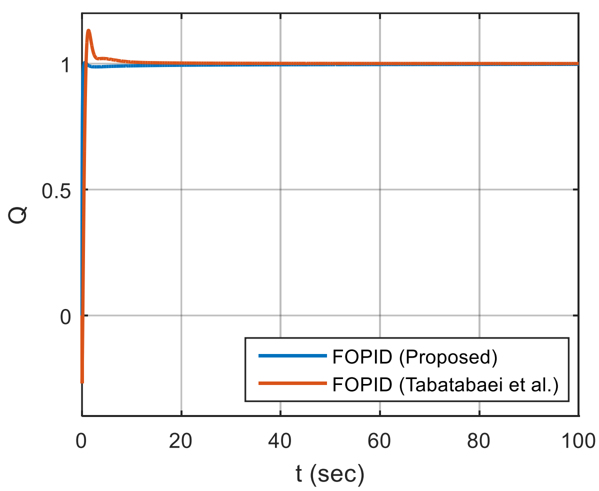

Fig.16. Step responses of the control systems

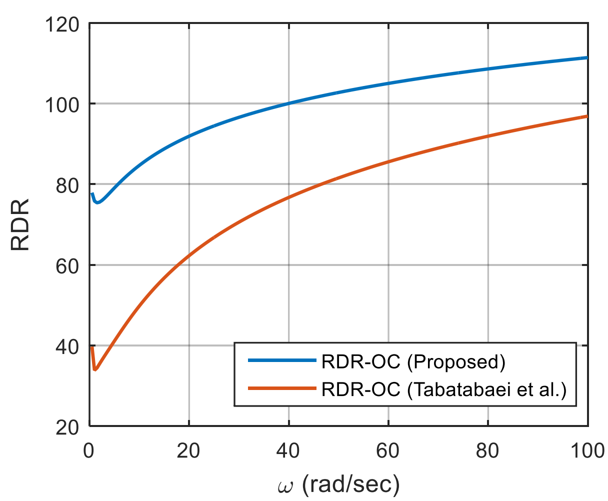

Fig.17. The RDR spectrums of the control systems

Figure 18 shows step disturbance response of the controllers. An additive step disturbance was applied to the control system at 40 sec simulation time. Figure 19 shows a close view of disturbance rejection control of both controllers.

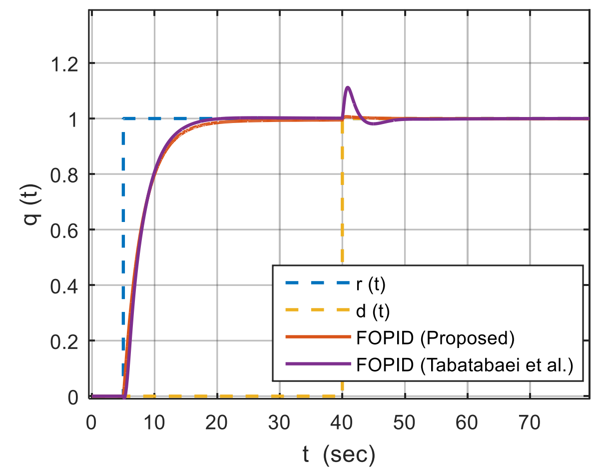

Fig.18. Step responses and disturbance responses of the FOPID controllers in case of an additive step disturbance insertion into the plant input at 40 sec

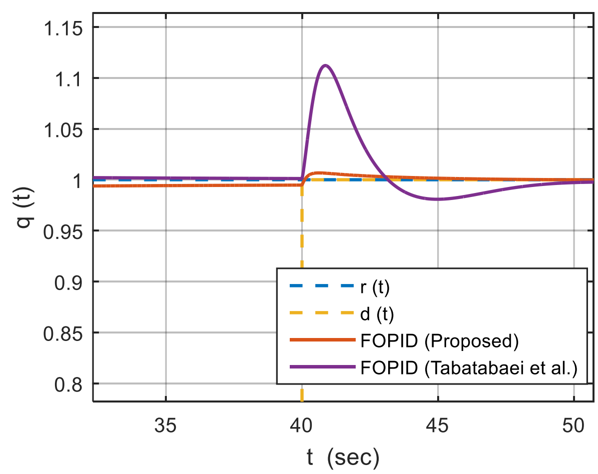

Fig.19. A close view of step responses in Figure 18

Figure 20 shows system responses when a sinusoidal input disturbance with the amplitude of 1 and a frequency of 3.14 rad/sec was inserted into the system. Figure shows that the disturbance rejection performance in this frequency is improved by the proposed FOPID control system.

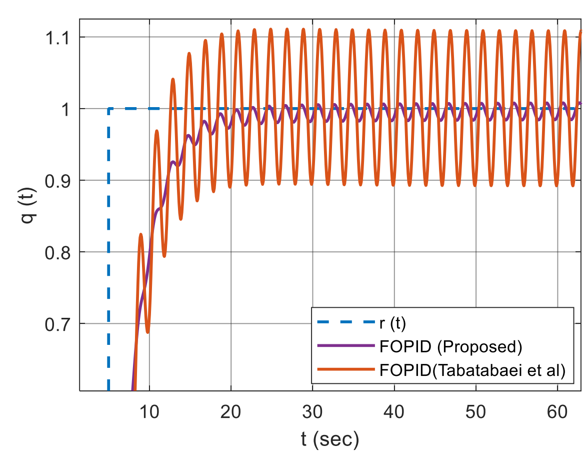

Fig.20. Sinusoidal disturbance responses of the control systems

V. CONCLUSIONS

In summary, this study demonstrated a software implementation of the disturbance reject FOPID controller design methodology that was based on the RDR index maximization subject to a soft constraining of the phase margin. Thus, a difficult mathematical optimization problem can be effectively solved by using soft computing techniques.

The proposed control system performance was compared with other optimal FOPID system design methods. Simulation results reveal the improvement of the step response and the disturbance rejection performance compared to other optimal tuning methods. Future work can address the addition of more design constraints by using soft computing routines in order to solve more sophisticated design problems.

References

[1] K.B. Oldham, J. Spanier, “Fractional Calculus.” Academic Press Inc., 1974.

[2] R. Hilfer, “Application of Fractional Calculus in Physics.” World Scientific, 2000.

[3] Y.Q. Chen, T. Bhaskaran, and D. Xue, “Practical tuning rule development for fractional order proportional and integral controllers.” J. Comput. Nonlinear Dyn., vol. 3, 2, 2008, pp. 214031–214038.

[4] W. Jifeng, L. Yuankai, “Frequency domain Analysis and Applications for Fractional Order Control System.” IOP, Journal of Physics: Conf. 13, 2005, pp. 268- 273.

[5] D. Xue, Y.Q. Chen, “A Comparative Introduction of Four Fractional Order Controllers.” Proceedings of the 4th World Congress on Intelligent Control and Automation, Shanghai, P.R. China, 2002, pp. 3228-3235.

[6] G.F. Franklin, J.D. Powell, A.E. and Baeini, “Feedback Control of Dynamic Systems.” Addison-Wesley, Reading, MA, 1986.

[7] K. Ogata, “Modern Control Engineering.” 2nd edn. Prentice-Hall, Englewcod Cliffs, NJ, 1990.

[8] J.L. Chang, “Robust output feedback disturbance rejection control by simultaneously estimating state and disturbance.” J Control Sci Eng, 2011, pp. 1–13.

[9] Z. Vukic, O. Kuljaca, “Lect PID Controllers.”, Vol. 23, 2002.

[10] D. Vrancic, S. Strmčnik, J. Kocijan, P.B. de Moura Oliveirac, “Improving disturbance rejection of PID controllers by means of the magnitude optimum method.” ISA Trans, Vol. 49, 1, 2010, pp. 47–56.

[11] J.M.E. Vandeursen, J.A. Peperstraete, “Model-based and PID controllers for disturbance rejection in processes with time delay: a comparison.” ISA Trans, Vol. 35, 3, 1996, pp. 225–236.

[12] F. Fayaz and G.L. Pahuja, "Disturbance rejection based fractional order PID controller for system performance improvement of hybrid power system." Optimal Control Applications and Methods, 2022.

[13] P. Selvaraj, O.M. Kwon, & R. Sakthivel, “Disturbance and uncertainty rejection performance for fractional-order complex dynamical networks.” Neural Networks, Vol. 112, 2019, pp. 73-84.

[14] F.N., Deniz, C., Keles, B.B., Alagoz, and N. Tan, “Design of fractional-order PI controllers for disturbance rejection using RDR measure.” In International Conference on Fractional Differentiation and Its Applications 2014 (ICFDA 2014), 2014.

[15] B.B. Alagoz, F.N. Deniz, C. Keles, and N. Tan, “Disturbance rejection performance analyses of closed loop control systems by reference to disturbance ratio.” ISA Transactions, Vol. 55, 2015, pp. 63–71. doi:10.1016/j.isatra.2014.09.013.

[16] A. Biswas, S. Das, and A. Abraham, and S. Dasgupta, “Design of fractional-order PIλDμ controllers with an improved differential evolution.” Engineering Applications of Artificial Intelligence, Vol. 22, 2009,pp. 343-350.

[17] M. Zamani, M. Karimi-Ghartemani, N. Sadati, M. Parniani, “Design of a fractional order PID controller for an AVR using particle swarm optimization.” Control Engineering Practice, Vol. 17, 2009, pp.1380-1387.

[18] G. Zhe, X. Liao, “Rational approximation for fractionalorder system by particle swarm optimization,” Nonlinear Dynamics, Vol. 67, 2, 2012, pp. 1387-1395.

[19] M. Micev, M. Ćalasan, & D. Oliva, “Fractional order PID controller design for an AVR system using Chaotic Yellow Saddle Goatfish Algorithm.” Mathematics, Vol. 8, 7, 2020, 1182.

[20] D. Izci, S. Ekinci, & B. Hekimoğlu, “Fractional-order PID controller design for buck converter system via hybrid Lévy flight distribution and simulated annealing algorithm.” Arabian Journal for Science and Engineering, 2022, pp. 1-19.

[21] I. Paducel, C.O. Safirescu, & E.H. Dulf, “Fractional Order Controller Design for Wind Turbines.” Applied Sciences, Vol. 12, 17, 2022, 8400.

[22] A. Richards, “Fast Model Predictive Control with Soft Constraints.” European Control Conference (ECC) Julay, Zürich, Switzerland, pp. 17-19.

[23] J.C. Regin, “Using hard constratints for representing soft constraints, International Conference on AI and OR Techniques in Constriant Programming for Combinational Optimization Problems.” Springer, Berlin, Heideberg, 2011, pp. 176-189.

[24] P. Meseguer, F. Rossi, and T. Schiex, “Soft Constraints, Foundation of Artificial Intelligence.” Elsevier, 2006, pp. 281-328 (Soft global constraining page 312).

[25] M. Chandrasekaran, M. Muralidhar, C.M. Krishna, U.S. Dixit, “Application of soft computing techniques in machining performance prediction and optimization: a literature review.” The international journal of Advanced manufacturing Technology, Vol. 46, 5, 2010, pp. 445-464.

[26] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, “Oposition versus randomness in soft computing techniques,” Applied Soft Computing, Vol. 8, 2008, pp. 906-918.

[27] B.B. Alagoz, N. Tan, F.N. Deniz, C. Keles, “Implicit disturbance rejection performance analysis of closed loop control systems according to communication channel limitations.” IET Control Theory & Applications, Vol. 9, 17, 2015, pp. 2522-2531.

[28] A. Ates, B.B. Alagoz, C. Yeroglu, et al., “Disturbance rejection FOPID control of rotor by multi-objective bb-bc optimization algorithm.” In: ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 13th ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications Cleveland, Ohio, USA, Vol. 9, 6–9 (August 2017).

[29] S. Tufenkci, B. Senol, B.B. Alagoz, & R. Matušů, “Disturbance rejection FOPID controller design in v-domain.” Journal of Advanced Research, Vol. 25, 2020, pp. 171-180.

[30] N. Ozbey, C. Yeroglu, B.B. Alagoz, N. Herencsar, A. Kartci, & R. Sotner, “2DOF multi-objective optimal tuning of disturbance reject fractional order PIDA controllers according to improved consensus oriented random search method.” Journal of Advanced Research., Vol. 25, 2020, pp. 159-170.

[31] Q. Jin, Y. Shi, Q. Liu, M. Chu, & Y. Zhang, “Graphical robust PID tuning for disturbance rejection satisfying multiple objectives.” Chemical Engineering Communications, Vol. 205, 12, 2018, pp. 1701-1711.

[32] D. Goldberg, J. Holland, “Genetic Algorithms and Machine Learning.” Mach. Learn., Vol. 3, 2-3, 1988, pp. 95–99.

[33] K.F. Man, K.S. Tang, ve S. Kwong, “Genetic Algorithms.” Springer Publishing, 1999.

[34] D.E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning.” Addison-Wesley Publishing Company, 1989.

[35] A.R. Tavakolpour, I.Z. Mat Darus, O. Tokhi, M. Mailah, “Genetic Algorithm-Based Identification of Transfer Function Parameters for a Rectangular Flexible Plate System.” Eng. Appl. Artif. Intel., Vol. 23, 8, 2010, pp. 1388–1397.

[36] X. Dingyu, “Fractional-Order Control Systems: Fundamentals and Numerical Implementations,” Walter de Gruyter, Berlin, Germany, Boston, MA, USA, 2017.

[37] C. Zhao, D. Xue, & Y. Chen, “A fractional order PID tuning algorithm for a class of fractional order plants.” In IEEE International Conference Mechatronics and Automation, Vol. 1, 2005, pp. 216-221.

[38] M. Tabatabaei, R. Salehi, “Fractional order PID controller design based on Laguerre orthogonal functions.” International Journal of Dynamics and Control, Vol. 5, 3, 2017, pp. 542-550.

APPENDIX

The matlab code of this design tool is given below. The objective function is costFunc(x):

```
function E = costFunc(x)
% x chromosome represents candidate controller
coefficients
kp=x(1);
ki=x(2);
kd=x(3);
lamda=x(4);
u=x(5);
% frequency setting for RDR index
walt=0;
wust=100;
dw=0.5;
w=walt:dw:wust;
% Calculation of logarithmic RDR index to
implement equations(5)and(2)
RDRdB=20*log10((kp+ki*cos((pi/2)*lamda)*w.^(-
lamda)+kd*cos((pi/2)*u)*w.^(u)).^2+(kd*sin((pi
/2)*u)*w.^(u)-ki*sin((pi/2)*lamda)*w.^(-
lamda)).^2);
% Calculation of minimum RDR index that
implements equation(9)
rdr=min(RDRdB);
%Plant parameters
a0=1;
b0=1;
b1=0.5;
alfa=0.9;
b2=0.8;
k=2.2;
%Phase margin specification
TargetPhaseMar=2*pi/3;
Pc=-pi;
% Coefficients of closed loop transfer
function
c0=a0*ki;
c1=a0*kp+b0;
c2=a0*kd;
c3=b1;
c4=b2;
beta0=0;
beta1=lamda;
beta2=lamda+u;
beta3=lamda+alfa;
beta4=k+lamda;
d2=a0*kd;
d1=kp*a0;
d0=ki*a0;
gamma2=(lamda+u);
gamma1=lamda;
gamma0=0;
% The software realization of the closed loop
transfer function
sistem=fotf([c4 c3 c2 c1 c0],[beta4 beta3
beta2 beta1 beta0],[d2 d1 d0],[gamma2 gamma1
gamma0]);
% The software realization of phase margin
calculations
X=bode(sistem,w);
 Tw=squeeze(X.ResponseData);
 Tw1=squeeze(Tw);
 Mw=abs(Tw1);
 Pw=angle(Tw1);
% Determination of crossover frequency by
using soft computing
% The following code detects zero crossing
for i=2:length(Mw)
if log10(Mw(i))*log10(Mw(i-1))<0
         Wc=w(i);
         Pc=Pw(i);
break;
end
end
%Phase margin according to equation (6)
Qp=pi+Pc;
% Implementation of optimization problem
according to soft constraint
% for phase margin specification
if abs(TargetPhaseMar-Qp)<0.01
  E=(1/((rdr)^2)+0.01);
else
  E=abs(TargetPhaseMar-Qp)*1e+10;
end
end
```

The code runs Matlab genetic algorithm for the objective function *costFunc(x):*

```
clc;
clear all;
close all;
ObjectiveFunction = @costFunc;
nvars = 5;
LB = [1 1 1 0.5 0.3]; % Lower bound
UB = [50 50 50 2 2]; % Upper bound
opts =
gaoptimset('PlotFcn',{@gaplotbestf,@gaplotbest
indiv});
[x,fval,exitFlag,Output,population,scores] =
ga(ObjectiveFunction,nvars,[],[],[],[],LB,UB,[
],opts);
fprintf('****Optimization completed ****\n')
fprintf('kp : %f \n',x(1));
fprintf('ki : %f \n',x(2));
fprintf('kd : %f \n',x(3));
fprintf('lampda : %f \n',x(4));
fprintf('mu : %f \n',x(5));
fprintf('Minimum value of objective function :
%f \n',fval);
```

(To run this code for *fotf()* function, *fotf toolbox* should be placed in the same folder.)

BIOGRAPHIES

SEVILAY TUFENKCI graduated from Selçuk University Computer Engineering Department in 2017. She graduated her MSc program in Inonu University, department of Computer Engineering in 2019 and is continuing her PhD program in the Computer Engineering Department of Inonu University. Her research interests are control systems, metaheuristic optimization and reinforcement learning methods.

BARIS BAYKANT ALAGOZ, received bachelor's degree in Electronics and Communication Engineering from Istanbul Technical University. He received the Ph.D. degree in Electrical-electronics Engineering from Inonu University in 2015. His research interests are computational intelligence, intelligent systems, control systems, simulation and modeling.

CELALEDDİN YEROGLU, received the BSc degree in electronics engineering from Hacettepe University, in 1991, and the Master and PhD degree from the Department of Computer Engineering, Trakya University, in 2000. He received second PhD degree in 2011 from the Department of Electrical and Electronics Engineering, Inonu University. Since 2009, he has been working with the Computer Engineering Department, Inonu University. His research interests include intelligent systems, control theory and applications, simulation and modeling of networks.

BILAL SENOL received the B.S. and M.S. degrees in electrical and electronics engineering, and the Ph.D. degree in computer engineering from İnönü University, Malatya, Turkey, in 2009, 2011, and 2015, respectively. He is currently working as an Associate Professor with the Department of Computer Engineering, İnönü University. His research interests include fractional-order analysis and controller design. He also works related to computer-aided design and user-friendly interfaces for system analysis. Lately, he is having researches on the classification of medical images.