ISSN 2229-5518

Tuning of FOPID Controller Using Taylor Series Expansion

Ali Akbar Jalali, Shabnam Khosravi

Abstract— In this paper, a direct synthesis approach to fractional order controller design Is investigated. The proposed algorithm makes use of Taylor series of both desired closed-loop and actual closed-loop transfer function which is truncated to the first five terms. FOPID Controller parameters are synthesized in order to match the closed-loop response of the plant to the desired closed-loop response. The standard and stable second-order model is considered for both plant and the desired closed-loop transfer functions. Therefore for a given plant with damping ratio ξ_1 and natural frequency ω_{n_1}. The tuned FOPID controller results in the desired closed-loop response with damping ratio ξ_2 and natural frequency ω_{n_2}. An example is presented that indicates the designed FOPID results in actual closed-loop response very close to desired response rather than PID controller. It is shown that the proposed method performs better than Genetic Algorithm in obtaining the desired response.

Index Terms— FOPID controller, Taylor series expansion, second order model.

1 Introduction

For many decades, proportional-integral-derivative (PID) controllers have been very popular in industries for process control applications. The popularity and widespread use of PID controllers are attributed primarily to their simplicity and performance characteristics. Owing to the paramount importance of PID controllers, continuous efforts are being made to improve their quality and robustness [1], [2].

An elegant way of enhancing the performance of PID controllers is to use fractional order controllers where the integral and derivative operators have non-integer orders. Podlubny proposed the concept of fractional order control in 1999 [3]. In FOPID controller, despite of the proportional, integral and derivative constants, there are two more adjustable parameters: the power of s in integral and derivative operators, λ, μ respectively. Therefore this type of controllers is generalizations of PIDs and consequently has a wider scope of design, while retaining the advantages of classical ones.

Several methods have been reported for FOPID design. Vinagre, Podlubny, Dorcak, Feliu [4] proposed a frequency domain approach based on expected crossover frequency and phase margin. Petras came up with a method based on the pole distribution of the characteristic equation in the complex plane [5]. In recent years evolutionary algorithms are used for FOPID tuning. YICAO, LIANG, CAO [6], presented optimization of FOPID controller parameters based on Genetic Algorithm. A method based on Particle Swarm Optimization was proposed [7]. In this paper a tuning method for FOPID controller is proposed. Suppose a standard and stable second order plant such that desired response is not available. Tuning FOPID controller by the proposed method results in desired closed-loop response. The standard second order is considered for desired response. It is shown that the proposed method performs better than Genetic Algorithm in obtaining the desired response. The rest of the paper is organized as follows: In section 2 the tuning method for FOPID controller is described. An example is investigated in section 3 and finally Section 4 draws some conclusions.

2 Obtaining The Tuning Method For FOPID Controller

Consider the block diagram of feedback control system in fig. 1. The objective is design a FOPID controller, $G_c(s)$, such that for a given plant, $G_p(s)$, with standard second order model, the actual closed-loop response results in desired closed-loop response. Desired closed-loop response denoted by $G_d(s)$ and described by standard second order model as follows

$$G_d(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{1}$$

Where ξ, ω_n are damping ratio and natural frequency of desired response.

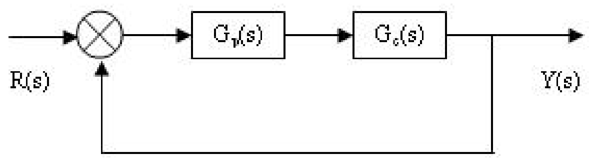

Fig. 1. Block diagram of feedback control system.

According to fig. 1. the actual closed-loop transfer function is given by

$$G_{Acl}(s)=\frac{G_p(s)G_c(s)}{1+G_p(s)G_c(s)} \tag{2}$$

In (2), transfer function of FOPID controller is given by

$$G_c(s)=k(1+\frac{k_i}{s^{\lambda}}+k_d s^{\mu}) \tag{3}$$

Where proportional, integral, derivative constants are denoted by k_c, k_i, k_d respectively. The orders of integral and derivative actions, λ, μ, include non-integer values as

$$0<\lambda,\mu<2 \tag{4}$$

M. Ramasy and Sundaramoorthy in [8] has used different structure for FOPID controller as

$$G_c(s)=\frac{k_c}{s}\tilde{G}_c(s) \tag{5}$$

Where

$$\tilde{G}_c(s)=(s+\frac{k_i}{s^{\lambda-1}}+k_d s^{\mu+1}) \tag{6}$$

Then for actual closed-loop transfer function in (2), we have

$$G_{Acl}(s)=\frac{k_c G_p(s)\tilde{G}_c(s)}{s+k_c G_p(s)\tilde{G}_c(s)} \tag{7}$$

For designing FOPID controller, it is not possible to set both $G_{Acl}(s)$ and $G_d(s)$ equal directly. For each closed-loop transfer function, there exist an equal expression but in different structure. This equal expression is Taylor series expansion that can be represented for both $G_{Acl}(s)$ and $G_d(s)$ in $s=\alpha$ as follows

$$G_{Acl}(s)=G_{Acl}(\alpha)+(s-\alpha)G'_{Acl}(\alpha)+\frac{(s-\alpha)^2 G''_{Acl}(\alpha)}{2}+...... \tag{8}$$

$$G_d(s)=G_d(\alpha)+(s-\alpha)G'_d(\alpha)+\frac{(s-\alpha)^2 G''_d(\alpha)}{2}+....... \tag{9}$$

In (8), (9), expressions for derivatives of actual closed-loop transfer function involve derivatives of FOPID controller.

$$G'_c(s)=k_c(-k_i\lambda s^{-\lambda-1}+k_d\mu s^{\mu-1}) \tag{10a}$$

$$G''_c(s)=k_c(-k_i\lambda(-\lambda-1)s^{-\lambda-2}+k_d\mu(\mu-1)s^{\mu-2}) \tag{10b}$$

$$G'''_c(s)=k_c(-k_i\lambda(-\lambda-1)(-\lambda-2)s^{-\lambda-3} +k_d\mu(\mu-1)(\mu-2)s^{\mu-3}) \tag{10c}$$

$$G^{(4)}_c(s)=k_c(-k_i\lambda(-\lambda-1)(-\lambda-2)(-\lambda-3)s^{-\lambda-4} +k_d\mu(\mu-1)(\mu-2)(\mu-3)s^{\mu-4}) \tag{10d}$$

$$G^{(5)}_c(s)=k_c(-k_i\lambda(-\lambda-1)(-\lambda-2)(-\lambda-3)(-\lambda-4)s^{-\lambda-5} +k_d\mu(\mu-1)(\mu-2)(\mu-3)(\mu-4)s^{\mu-5}) \tag{10e}$$

However derivatives of FOPID controller are not defined at $s=0$. For convenience and avoiding complexity that causes by non-integer orders of the Laplace variable s, it is proposed to evaluate (8) and (9) at $\alpha=1$.

FOPID controller has five tuning parameters. Therefore five independent equations are needed for tuning FOPID controller.

$$G_{Acl}(1)=G_d(1) \tag{11a}$$

$$G'_{Acl}(1)=G'_d(1) \tag{11b}$$

$$G''_{Acl}(1)=G''_d(1) \tag{11c}$$

$$G'''_{Acl}(1)=G'''_d(1) \tag{11d}$$

$$G^{(4)}_{Acl}(1)=G^{(4)}_d(1) \tag{11e}$$

$$G^{(5)}_{Acl}(1)=G^{(5)}_d(1) \tag{11f}$$

In obtaining design parameters of FOPID controller, the first terms in (8) and (9) are not considered. According to (1) and (7), $G_{Acl}(s)$, $G_d(s)$ are equal in $s=0$. Probably these values are nearly equal in $s=1$. Using (2), the equations in (11) can be written as

$$\frac{G'_c(1)G_p(1)+G_c(1)G'_p(1)}{(1+G_c(1)G_p(1))^2}-G'_d(1)=0 \tag{12a}$$

$$\frac{(G''_c(1)G_p(1)+2G'_c(1)G'_p(1)+G_c(1)G''_p(1))(1+G_c(1)G_p(1))-2(G'_c(1)G_p(1)+G_c(1)G'_p(1))^2}{(1+G_c(1)G_p(1))^3}-G''_d(1)=0 \tag{12b}$$

$$\frac{\begin{array}{l}(G'''_c(1)G_p(1)+3G''_c(1)G'_p(1)+3G'_c(1)G''_p(1)+G_c(1)G'''_p(1))(1+G_c(1)G_p(1))^2-6(G'_c(1)G_p(1)+G_c(1)G'_p(1))\\(G''_c(1)G_p(1)+2G'_c(1)G'_p+G_c(1)G''_p)(1+G_c(1)G_p)+6(G'_c(1)G_p(1)+G_c(1)G'_p(1))^3)\end{array}}{(1+G_c(1)G_p(1))^4}-G'''_d(1)=0 \tag{12c}$$

$$\frac{\begin{aligned}&(G_c^{(4)}(1)G_p(1)+4G_c'''(1)G_p'(1)+6G_c''(1)G_p''(1)\\&+4G_c'(1)G_p'''(1)+G_c(1)G_p^{(4)}(1))(1+G_c(1)G_p(1))^3\\&-6(G_c''(1)G_p(1)+2G_c'(1)G_p'(1)+G_c(1)G_p''(1))^2\\&(1+G_c(1)G_p(1))^2-8(G_c'(1)G_p(1)+G_c(1)G_p'(1))\\&(G_c'''(1)G_p(1)+3G_c''(1)G_p'(1)+3G_c'(1)G_p''(1)+\\&G_c(1)G_p'''(1))(1+G_c(1)G_p(1))^2+36(G_c'(1)G_p(1)\\&+G_c(1)G_p'(1))^2(G_c''(1)G_p(1)+2G_c'(1)G_p'(1)+G_c(1)\\&G_p''(1))(1+G_c(1)G_p(1))-24(G_c'(1)G_p(1)+G_c(1)\\&G_p'(1))^4)\end{aligned}}{(1+G_c(1)G_p(1))^5}-G_d^{(4)}(1)=0 \tag{12d}$$

$$\frac{\begin{aligned}&(G_c^{(5)}(1)G_p(1)+5G_c^{(4)}(1)G_p'(1)+10G_c'''(1)G_p''(1)+\\&10G_c''(1)G_p'''(1)+5kG_c'(1)G_p^{(4)}(1)+G_c(1)G_p^{(5)}(1))\\&(1+G_c(1)G_p(1))^4-20(G_c''(1)G_p(1)+2G_c'(1)G_p'(1)\\&+G_c(1)G_p''(1))(G_c'''(1)G_p(1)+3G_c''(1)G_p'(1)+3G_c'(1)\\&G_p''(1)+G_c(1)G_p'''(1))(1+G_c(1)G_p(1))^3+90(G_c''(1)\\&G_p(1)+2G_c'(1)G_p'(1)+G_c(1)G_p''(1))^2(G_c'(1)G_p(1)\\&+G_c(1)G_p'(1))(1+G_c(1)G_p(1))^2-10(G_c'(1)G_p(1)\\&+G_c(1)G_p'(1))(G_c^{(4)}(1)G_p(1)+4G_c'''(1))G_p'(1)+6\\&G_c''(1)G_p''(1)+4G_c'(1)G_p'''(1)+G_c(1)G_p^{(4)}(1))(1+G_c(1)\\&G_p(1))^3+60(G_c'(1)G_p(1)+G_c(1)G_p'(1))^2(G_c'''(1)G_p(1)\\&+3k_cG_c''(1)G_p'(1)+3G_c'(1)\ G_p''(1)+G_c(1)G_p'''(1))(1+\\&G_c(1)G_p(1))^2-240(G_c'(1)G_p(1)+G_c(1)G_p'(1))^3(G_c''(1)\\&G_p(1)+2G_c'(1)G_p'(1)+G_c(1)G_p''(1))(1+G_c(1)G_p(1))+\\&120(G_c'(1)G_p(1)+G_c(1)G_p'(1))^5\end{aligned}}{(1+G_c(1)G_p(1))^6}-G_d^{(5)}(1)=0 \tag{12e}$$

Where

$$G_c(1)=k_c(1+k_i+k_d) \tag{13a}$$

$$G_c'(1)=k_c(-k_i\lambda+k_d\mu) \tag{13b}$$

$$G_c''(1)=k_c(-k_i\lambda(-\lambda-1)+k_d\mu(\mu-1)) \tag{13c}$$

$$G_c'''(1)=k_c(-k_i\lambda(-\lambda-1)(-\lambda-2)+k_d\mu(\mu-1)(\mu-2)) \tag{13d}$$

$$G_c^{(4)}(1)=k_c(-k_i\lambda(-\lambda-1)(-\lambda-2)(-\lambda-3)+k_d\mu(\mu-1)(\mu-2)(\mu-3)) \tag{13e}$$

$$G_c^{(5)}(1)=k_c(-k_i\lambda(-\lambda-1)(-\lambda-2)(-\lambda-3)(-\lambda-4)+k_d\mu(\mu-1)(\mu-2)(\mu-3)(\mu-4)) \tag{13f}$$

The nonlinear equations in (12) are complicated to obtain design parameters, $k_c, k_i, k_d, \lambda, \mu$. thus a nonlinear optimization problem must be solved. The fsolve command in optimization toolbox of Matlab is a sufficient tool for solving the set of nonlinear equations. The input arguments of this command are described as fun, x_0 and options. Fsolve starts at an initial value for design parameters, x_0, in order to solve the set of nonlinear equations described in fun. The argument of option determines the type of optimization algorithm which is used in solving the nonlinear equations. The output arguments are the solution of nonlinear equations, x, and the value of objective function at x. Consider the nonlinear equations in (12) as the objective functions. In this case the solution, x, is the design parameters of FOPID controller, $k_c, k_i, k_d, \lambda, \mu$. The optimization algorithm of Levenberg-Marquert and Gauss-Newton are considered. In the next section, the proposed tuning method is illustrated during an example.

3 Example

In this section, a process from [9] is considered. The example involves the speed control of a DC motor. Since the most basic requirement of a motor is that it should rotate at the desired speed, the steady-state error of the motor speed should be less than 1%. We want to have settling time of 2s and an overshoot of 4%. A desired closed-loop transfer function that includes all of the design specifications, can be defined as follows

$$G_d(s)=\frac{8.9401}{s^2+4.2398s+8.9401} \tag{14}$$

The transfer function of the process is defined by

$$G_p(s)=0.0999\frac{20.02}{s^2+12s+20.02} \tag{15}$$

Applying the values of $G_p(1), G_p'(1), G_p''(1), G_p'''(1), G_p^{(4)}(1)$ $G_p^{(5)}(1), G_d(1), G_d'(1), G_d''(1), G_d'''(1), G_d^{(4)}(1), G_d^{(5)}(1)$ in (12), five nonlinear equations are obtained. Using the fsolve command in optimization toolbox of Matlab, the fractional order controller is designed as

$$G_c(s)=18.27\left(1+\frac{1.11}{s^{1.02}}-0.563s^{0.1}\right) \tag{16}$$

After designing the FOPID controller, the values of nonlinear equations in (12), are $-10^{-4}, -10^{-4}, -10^{-4}, -10^{-4}, 10^{-3}$. It is obvious that the obtained values for design parameters, $k_c, k_i, k_d, \lambda, \mu$., are very close to roots of nonlinear equations in (12). Furthermore the values of both actual and desired closed-loop transfer functions at $s=1$ are nearly equal with difference of 10^{-3}. Thus the first six terms of Taylor series of actual closed-loop transfer function are equal to same order terms for desired one. The accuracy of how much the actual closed-loop response is close to desired response can be concluded from fig. 2. For comparison, the tuned FOPID controller using Genetic Algorithm is considered. Fig. 3 shows that the closed-loop response under FOPID controller, which is tuned by both GA and proposed methods, in comparison to desired response. Data about specifications of desired closed-loop response are collected in Table1.

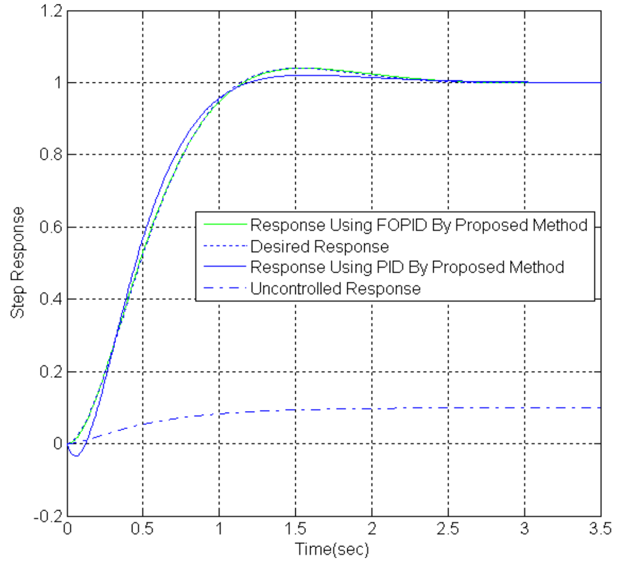

Fig. 2. Step response of closed-loop system using both FOPID and PID controllers, step response of the desired closed-loop system, step response of uncontrolled system.

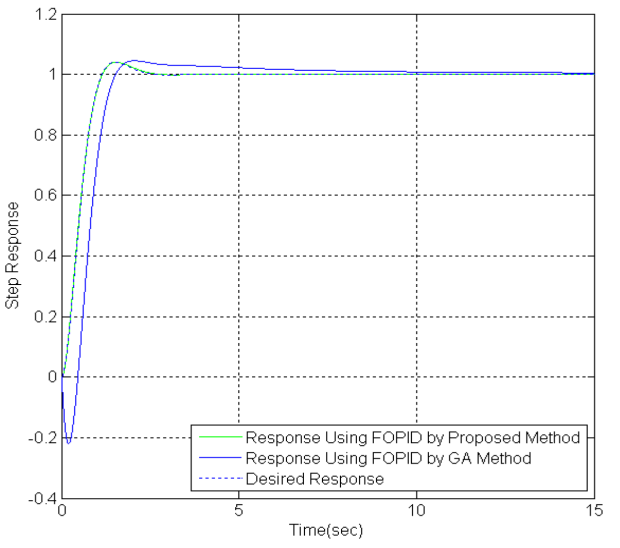

Fig. 3. Step response of closed-loop system using FOPID controller obtained by proposed method and Genetic Algorithm tuning method, step response of the desired closed-loop system.

TABLE 1
SPECIFICATIONS OF DESIRED CLOSED-LOOP RESPONSE

Maximum overshoot	Settling time	Rise time	Steady-state error
4	2	0.9	0

Table. 1. reports the maximum overshoot (in %), settling time and rise time (in second) and steady-state error (in %) for the closed-loop step response. Data about the performance of closed-loop system under FOPID and PID controllers against unit step, are collected in Table 2.

TABLE 2
SUMMERY OF THE PERFORMANCE OF CLOSED-LOOP SYSTEM UNDER FOPID AND PID CONTROLLER AGAINST UNIT STEP

Different controllers	Maximum overshoot	Settling time	Rise time	Steady-state error
FOPID using proposed method	4	2.08	0.9	0
FOPID using GA method	4.5	5.3	1.26	0
PID using proposed method	2	1.08	0.86	0

Fig. 2. compares actual closed-loop response under both PID and FOPID controllers and desired response. In fig. 2. PID and FOPID controllers are tuned by proposed method. Due to Table. 2. the actual closed-loop response under FOPID controller is very close to desired response rather than applying PID controller. For example the specifications of transient response, such as maximum overshoot, rise time, settling time and steady-state error when using FOPID controller, are very close to desired transient specifications. Using PID controller results in a behavior in t=0, which is not appeared when applying FOPID controller. This behavior is because of existing a zero near the origin. Furthermore due to fig.3. and Table. 2 the FOPID controller, which is tuned by proposed method, results in better performance rather than FOPID controller which is tuned by GA method.

Fig. 4. shows the bode diagram of open-loop systems, applying both FOPID and PID controllers.

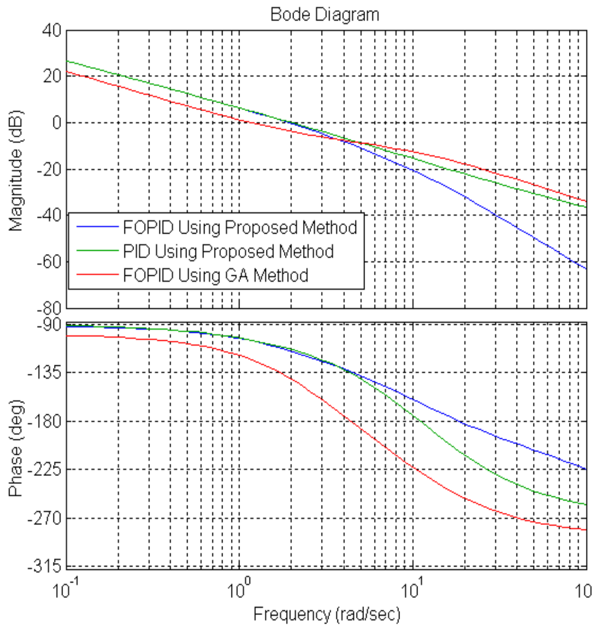

Fig. 4. Bode diagram of open-loop systems applying FOPID controller which is tuned by both proposed method and GA method, bode diagram of open-loop system applying PID controller which is tuned by proposed method.

According to fig.4. applying FOPID and PID controllers which is tuned by proposed method result in a same phase margin of 66 (deg). However the difference in values of gain margin is significant. The gain margin of 30.7 (dB) and 17(dB) are obtained for FOPID and PID controllers respectively. The FOPID controller which is tuned by Genetic Algorithm results in gain margin and phase margin of 8.2(dB) and 59 (deg) respectively. It can be concluded that better performances can be obtained by using the proposed method.

4 Conclusion

A design method for FOPID controller is proposed. This method is based on Taylor series of both actual and desired closed-loop transfer function. The design parameters of controller are used for matching the same order terms of both desired and actual closed-loop response. FOPID controller has two design parameters, λ, μ, more than PID controller. Thus two more terms in Taylor series are used to match closed-loop response to desired response. This causes increasing accuracy in tracking the desired response rather than using PID controller. Furthermore the FOPID controller which is tuned by proposed method performs better than FOPID controller which is tuned by Genetic Algorithm. It can be concluded that that better performances can be obtained by the proposed method.

References

[1] Y.X. Su, Dong Sun, B.Y. Duan, "*Design of an enhanced nonlinear PID controller*", Mechatronics, Vol 15, No. 8, pp. 1005–1024, October 2005.

[2] Deepyaman Maiti, Ayan Acharya, Mithun Chakraborty, Amit Konar, Ramadoss Janarthanan, *"Tuning PID and $PI^{\lambda}D^{\mu}$ Controllers using the Integral Time Absolute Error Criterion",* 4th IEEE International Conference on Information and Automation for Sustainability, Nov 2008.

[3] Igor Podlubny, *"Fractional-order systems and $PI^{\lambda}D^{\mu}$ controllers"*, IEEE Transactions on Automatic Control, Vol 44, No. 1, pp. 208–213, JANUARY 1999.

[4] Vinagre, B.M., Podlubny, I., Dorcak, L., Feliu, V., 2000. *"On fractional PID controllers: a frequency domain approach"*. In: Proceedings of IFAC Workshop on Digital Control-PID, 2000, Terrassa, Spain.

[5] Petras, I., 1999. "*The fractional order controllers: methods for their synthesis and application"*. Journal of Electrical Engineering 50 (9–10), 284–288, Apr 2000.

[6] JUN-YICAO, JIN LIANG, BING-GANG CAO, *"optimization of fractional order PID controllers based on genetic algorithms"*, proceeding of fourth International Conference on Machine Learning and Cybernetics, Guangzhou, Vol 9, pp. 5686-5689, 18-21 August 2005.

[7] Deepyaman Maiti, Sagnik Biswas, Amit Konar, *"Design of a Fractional Order PID Controller UsingParticle Swarm Optimization Technique"*, 2nd National Conference on Recent Trends in Information Systems Oct 2008.

[8] M. Ramasamya, S. Sundaramoorthy, *"PID controller tuning for desired closed-loop responses for SISO systems using impulse response"*, Computers and Chemical Engineering 32 (2008) 1773–1788.

[9] Arijit Biswas, Swagatam Das, Ajith Abraham, Sambarta Dasgupta, *"Design of fractional-order $PI^{\lambda}D^{\mu}$ controllers with an improved differential evolution"*, Enginnering Application of Artificial Intelligence, Vol 22, No. 2, pp. 343-350, March 2009.