Robust PID and Fractional PI Controllers Tuning for General Plant Model

Dr. Basil H. Jasim.
Department of electrical Engineering
University of Basrah
College of Engineering

***Abstract*-In this paper, a design procedure which assumes general integer or noninteger order plant models 'also can be unknown' has been adopted to tune PID and fractional order PI (FOPI) controller. The design procedure depends on some specifications of frequency response of open loop system to ensure performance and robustness of step response of closed loop system. Firstly, the procedure is applied to integer order conventional PID (IOPID) controller, and then it has been extended to FOPI. Extensive simulation study has been made to investigate the performance of the obtained controllers, and also to compare between the two controllers. The simulation study has showed the validity and that the proposed controllers have good features in all of control demands, where it shows that these controllers have fast rise time with no overshoot and negligible steady state error. Also, it has showed that FOPI controller performs better than IOPID one.**

I. Introduction

By fractional calculus, we mean the mathematics which deals with differential equations and integration with non-integer orders, instead of integer orders as it is the case in the known or traditional calculus. Obviously, Fractional calculus gives more flexibility to describe dynamic systems. Also, it can give more flexibility in control design in many cases [1]. Fractional order PID (FOPID) controllers are the controllers in which the derivative and/or integration parts have non integer orders. While by IOPID we mean the well-known classical PID.

The fractional order calculus (FOC) is the extension for integer (traditional) calculus (IOC). In fact, it appeared nearly at the same time with the integer order or classical calculus. In other words, it has about 300 years history [1]. However, because of its complexity, FOC has not been used in science and engineering applications until recently.

Nowadays, and because of advances in computer programming, there are lots methods to approximate fractional derivative and integral. Consequently, FOC has been applied in many applications [2-6]. In control theory, it has been used as a modeler and also in controllers. It has been shown that fractional order system (FOS) could model many objects and phenomena more accurately than classical IOS [7,8]. Also, many design fractional order controllers to replace some known classical controllers have been proposed. It has been shown that enhancement can be obtained with these controllers.

Among the most important FO controllers introduced in literature, are the FO PI/PID controllers [2,3] (often, known as PI^{τ} and $PI^{\tau}D^{\mu}$), the FO lead-lag compensator [4] and optimal fractional controllers [6].

The FOPID controller is an expansion to traditional PID controller. It has two more adjustable parameters, namely the power of integral and derivative parts of the controller. It has been shown by many designs that FOPID controller can gives better performance and robustness than the conventional IOPID [10]. Many design procedures for specific types of controlled plant models have been proposed for FOPID controllers [3,4].

In the present work, a design procedure based on frequency response parameters of controlled plant has been adopted for FOPI controller. The proposed design procedure needs only the absolute and argument values of the controlled plant at a specific or selected frequency which represents the required cross over frequency for the overall system. These two values can be experimentally obtained by applying a sinusoidal signal with the required cross over frequency to the input of the pant, and getting the output, then calculating the gain and argument. So, in the proposed design method, no need for plant model to be known. The design procedure assumes general integer or noninteger order plant model. The procedure is firstly applied to IOPID controller, and then it has been extended for FOPI controller.

II. Problem formulation

Consider the unity feedback control system with C(s) is the transfer function (T.F.) of controller and P(s) the T.F. of the controlled plant. The controlled plant is a general single input single output (SISO) system which can be represented by the following general T.F.:-

$$P(s) = \frac{k_p N(s) e^{-ls}}{D(s)} \ldots\ldots\ldots\ldots\ldots (1)$$

Where, K_p is the gain of the plant, l is the time delay. N(s) and D(s) are integer or fractional order polynomials of s.

Our aim in this paper is to design or tune two types PI/PID controllers to control the previous plant for specified performance and robustness to gain loop variations. The first controller (C_1) is a traditional IOPID, while the second one is a FOPI controller.

The T.Fs. for the two controllers are listed below [2,11,12]:-

$$C_1(s) = K_c\left(1 + \frac{1}{T_i s} + T_d s\right) \ldots\ldots\ldots\ldots\ldots (2)$$

$$C_2(s) = K_c\left(1 + \frac{1}{T_i s^{\tau}}\right) \ldots\ldots\ldots\ldots\ldots (3)$$

Where, K_c is the proportional gain, T_i and T_d are integral time constant and derivative time constant, respectively, τ is a real number representing the order of the S-domain variable (s).

III. Design procedure

The controller design is based on some constraints of the open loop system (C(s).P(s)) in frequency response. These constraints can ensure some specified specifications of the closed loop system.

The first condition is the phase margin constraint which can be demonstrated mathematically as follow [11]:-

$$\arg[C(j\omega_c)P(j\omega_c)] = -\pi + \varphi_m \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Where, ω_c is the gain crossover frequency and φ_m is the desired phase margin.

The relationship between phase margin and the transient response is well known. So, by this condition the step response performance can be determined.

The second condition is the gain crossover frequency constraint [11,12]:-

$$|C(j\omega_c)P(j\omega_c)| = 1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

The third condition is given by the following [12]:-

$$\left.\frac{d\,(\arg[C(j\omega)P(j\omega)])}{d\omega}\right|_{\omega=\omega_c} = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

This constraint ensures that the phase plot is flat around the gain crossover frequency, which in turn ensures the robustness of the closed loop system to plant gain variations.

IV. Controllers design

4.1 IOPID controller

Equation 2 describes the T.F. of IOPID controller. Replacing S by $j\omega$ in Eq.2, then with some manipulation, the resultant equation can be rewritten as real and imaginary parts. Then, using the resultant equation, the absolute and the argument of the controller equation can be easily obtained as demonstrated by the following equations:-

$$|C(j\omega)| = \frac{kc\sqrt{(T_i\omega)^2 + (1 - T_dT_i\omega^2)^2}}{T_i\omega} \ldots\ldots\ldots\ldots\ldots\ldots (7)$$

$$\arg[(j\omega)] = tan^{-1}\left(\frac{Ti\omega}{(1 - T_dT_i\omega^2)}\right) - \frac{\pi}{2} \ldots\ldots\ldots\ldots\ldots\ldots (8)$$

Now, by constracting the three design conditions on this controller, one can obtain the following:-

- Phase margin constraint:-

$$\underline{\lfloor c(i\omega_c)P(j\omega_c)} = \underline{\lfloor C(j\omega_c)} + \underline{\lfloor P(\omega_c)} = -\pi + \varphi_m \ldots\ldots (9)$$

Substituting Eq.8 in Eq.9 and rearranging the resulting equation, the following will result:-

$$\frac{T_i\omega_c}{1 - T_dT_i{\omega_c}^2} + c_1 = 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

Where,

$$c_1 = -\tan\{\varphi_m - \frac{\pi}{2} - \arg[P(j\omega_c)]\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

-Gain crossover frequency constraint:-

$$|C(j\omega_c)P(j\omega_c)| = \frac{\sqrt{(T_i\omega_c)^2 + (1 - T_dT_i{\omega_c}^2)^2}}{T_i\omega_c}.|P(j\omega_c)| = 1 \ldots\ldots (12)$$

Eq.12 can be rearranged as follow:-

$$\frac{{T_i}^2{\omega_c}^2}{(T_i\omega_c)^2 + (1 - T_dT_i{\omega_c}^2)^2} = {k_c}^2c_2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots (13)$$

Where,

$$c_2 = (|P(j\omega_c)|)^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (14)$$

-Robustness constraint:-

$$\left.\frac{d\underline{\lfloor c(\omega)P(j\omega)}}{d\omega}\right|_{\omega=\omega_c} = \frac{T_i(T_iTd{\omega_c}^2+1)}{(1-T_iT_d{\omega_c}^2)^2+{T_i}^2{\omega_c}^2} + c_3 = 0 \ldots\ldots\ldots\ldots (15)$$

Where,

$$c_3 = \left.\frac{d\{\arg[P(j\omega)]\}}{d\omega}\right|_{\omega=\omega_c} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (16)$$

Now, we have three nonlinear equations, namely Eq.10, 13 and 15 and three design parameters (k_c, T_i and T_d) which we try to find. These equations can be solved numerically or by using optimization techniques. Matlab "which is the software used in this work" provides some functions for that purpose, such as fsolve which provide numerical solver for nonlinear equations and fminunc function which provides optimization tool for unconstrained nonlinear equations. To use these functions in Matlab, the system representing the problem to be solved should be formed as a nonlinear equations. Then, these functions can be used to solve the formulated system of equations by calling one of these functions by its name in Matlab workspace and passing the system of equations and an initial condition as arguments for the function.

When using these functions or any other optimization or numerical technique, some difficulty may be faced to select initial values, namely the starting point. To overcome this problem, we suggest here alternative graphical procedure. The detail of this procedure is as follow:-

Given the plant model, c_1, c_2 and c_3 can be found. If the plant model is not available, these parameters can be found experimentally as explained previously. Now, from Eq.10 plot Td w.r.t T_i and from 15 also plot the same graph on the same plot. Then, T_i and T_d can be found from the intersection point of the two curves. From Eq.13 k_c can be calculated.

To explain the procedure, let us take the following T.F. as a plant model to be controlled:-

$$p(s) = \frac{10e^{-ls}}{3s^2 + 0.5s + 1} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (17)$$

Which is one of the models used in simulation study.

Now, replacing s by $j\omega$, the absolute and argument values can be obtained. Also, the derivative of the argument is required. Now, selecting a suitable values for ω_c and φ_m "say 0.9rad/s and π/2.8 rad respectively", then by replacing ω in equetions 11, 14 and 16 with selected ω_c value, c_1, c_2 and c_3 can be obtained.

Now, rearrange Eq.10 for T_d as illustrated bellow:-

$$T_d = \frac{c_1 + \omega_cT_i}{c_1T_i{\omega_c}^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, l.(18)$$

This equation have been used to obtain a plot for T_d w.r.t T_i by changing T_i from 0 to 50 with 0.01 step and finding T_d from the equation. Matlab programing has been used for that purpose using m-file utility.

Equation 15 can be rewritten as a second order homogeneous polynomial in T_d as illustrated bellow:-

$$(c_2{T_i}^2{\omega_c}^4){T_d}^2 + \left({T_i}^2{\omega_c}^2 - 2c_2\,T_i{\omega_c}^2\right)T_d + (c_2{T_i}^2{\omega_c}^2 + T_i + c_2) = 0 \ldots\ldots\ldots (19)$$

Numerical solution have been used to find the roots of this equation where T_i have been changed from 0 to 50 with 0.01 step, for each step the roots (T_d) of this equation is obtained. If the two roots obtained are positive integer, then we should apply the procedure for both, if one of the two roots is negative or complex, then it neglected. After choosing one of the two roots, the sets of this root (T_d values) are ploted w.r.t the corresponding values of T_i on the same plot obtained from Eq.18. Fig.1 shows this plot for our example.

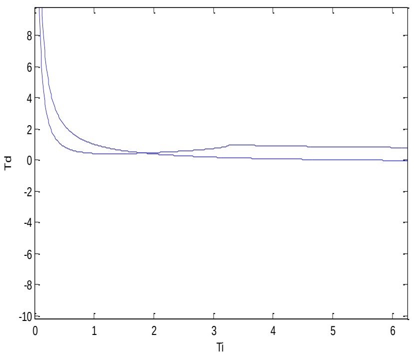

Fig. 1 Ti versus Td plot.

The intersection point of the two curves will represent the solution. This point has been found to be T_i=1.85 and T_d=0.423. Now, from Eq.13 K_c can be found, and it is equal to 0.719.

4.2 FOPI controller

Equation 3 describes the T.F. of this controller. Using the following relation[12]:-

$$j^{\tau}=\cos\left(\frac{\pi\tau}{2}\right)+j\sin\left(\frac{\pi\tau}{2}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(20)$$

The angle and gain of Eq.3 can be easily obtained, and these are constructed in equations bellow:-

$$\arg[C(j\omega)]=-\left(\frac{\pi\tau}{2}\right)+\tan^{-1}\left[\frac{T_i\omega^{\tau}\,sin\left(\frac{\pi\tau}{2}\right)}{1+T_i\omega^{\tau}\,cos\left(\frac{\pi\tau}{2}\right)}\right]\ldots(21)$$

$$|C(j\omega)|=\frac{kc}{T_i w^{\tau}}\sqrt{1+2T_i\omega^{\tau}\cos\left(\frac{\pi\tau}{2}\right)+{T_i}^2\omega^{2\tau}}\ \ldots\ldots(22)$$

Then, the three design constraint can be easily obtained as follow:-

-Phase margin constraint

Using Eq.9, substituting Eq.21 and rearranging, the following can be obtained:-

$$\frac{T_i{\omega_c}^{\tau}\sin\left(\frac{\pi\tau}{2}\right)}{1+T_i{\omega_c}^{\tau}\cos\left(\frac{\pi\tau}{2}\right)}=C_1(\tau)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(23)$$

Where,

$$C_1(\tau)=\tan\{\frac{\pi\tau}{2}-\pi+\varphi_m-\arg[P(j\omega_c)]\}$$

-gain crossover frequency

Using Eq.5, substituting Eq.22 and rearranging, the following can be obtained:-

$$\frac{{T_i}^2{\omega_c}^{2\tau}}{1+2T_i{\omega_c}^{\tau}\cos\left(\frac{\pi\tau}{2}\right)+{T_i}^2{\omega_c}^{2\tau}}=kc^2C_2\ldots\ldots\ldots\ldots\ldots(24)$$

Where, C_2 is as described by Eq.14.

-Robustness constraint

Using Eq.6, getting the derivative of Eq.21, the following can be obtained:-

$$\frac{\tau T_i{\omega_c}^{\tau}\sin\left(\frac{\pi\tau}{2}\right)}{\omega_c[{\omega_c}^{2\tau}\cos\left(\frac{\pi\tau}{2}\right)+1]}+C_3=0\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(25)$$

Where, C_3 is as described by Eq.16.

Now, as for IOPID controller we have three equations and three design parameters. The same procedure suggested for IOPID to solve these equations can be applied here. The procedure can be applied as follow:- From Eq.23 plot Ti w.r.t τ and from 25 also plot the same graph on the same plot. Then, T_i and τ can be found from the intersection point of the two curves. From Eq.24, k_c can be calculated.

V. Simulation study

In this section, the validity and performance of the obtained tuning method have been investigated by an extensive simulation study made by Matlab, where m-file tool has been used to solve the nonlinear equations using the described method while Simulink utility has been used to simulate the systems. Four different process models have been chosen to be used as the controlled processes.

1- First order plus delay time (FOPDT)

The FOPDT is a model used widely to approximate high order process by a first order model with delay time. The general T.F. for FOPDT is:-

$$p(s)=\frac{K_pe^{-ls}}{Ts+1}$$

Where K_p is the process gain, T is the time constant and l the dead time.

This model has been used with K_p=10, T=5 and l=0.5. These values are arbitrary selected.

For integer order PID controller, we have selected ω_c=0.9 rad/sec. and φ_m=π/2.3rad, and for FOPI ω_c=0.9 rad/sec. and φ_m=π/4rad. ω_c and φ_m has been suitably selected to ensure good performance for the overall system.

Now, by applying the procedure described in 4.1 and 4.2, the parameters for the two controllers can be obtained. For IOPID controllers, the controller parameters are k_c=0.213, T_i=3.54 and T_d=0.377, while for FOPI k_c=0.138, T_i=1.256 and τ=0.77.

Fig.2 shows the step response for the two systems.

2- Second order plus delay time (SOPDT)

This model is also used to approximate high order processes. The general T.F. for this model is shown below:-

$$p(s)=\frac{K_pe^{-ls}}{t_2s^2+t_1s+1}$$

The used model have been chosen with the following parameters:- K_p=10, t_1=0.5 sec., t_2=3 sec. and l=0.5.

Selecting ω_c=0.9 rad/sec. and φ_m=π/2.8 rad for IOPID, ω_c=0.9 rad/sec. and φ_m=π/2.8 for FOPI and applying the solution procedures, the controller parameters have been obtained:- k_c=0.719, T_i=1.85 and T_d=0.423 for IOPID, k_c=0.054, T_i=1.225 and τ=0.93 for FOPI

Fig.3 shows the step response for the two systems.

3- In this part of simulation study, two general T.Fs. have been selected. The two equations below describe these T.Fs.:-

$$P_1(s)=\frac{10}{(0.2s+1)(0.3s+1)(0.5s+1)(0.7s+1)}$$

$$P_2(s)=\frac{10(0.8s+1)(0.4s+1)}{(0.2s+1)(0.3s+1)(0.5s+1)(0.7s+1)}$$

For P1, we have selected ω_c=0.9 and φ_m=π/4.2 for IOPID and ω_c=1.1 and φ_m=π/12 for FOPI. For P2, ω_c=2 and φ_m=π/3 for IOPID, ω_c=2 and φ_m=π/4.2 for FOPI. Using similar procedure, the controller parameters have been obtained:-

P1: kc=0.0266, Ti=0.97, Td=0.787 for IOPID, kc=0.043, Ti=1.446 and τ=1.02 for FOPI.

P2: kc=0.06, Ti=0.7, Td=0.147 for IOPID, kc=0.049, Ti=0.575 and τ=0.95 for FOPI.

Figs.4 and 5 show the step response for the two T.Fs.
To study the robustness of the proposed controllers, FOPDT and SOPDT models have been simulated with different values of plant gain (K_p). The nominal value for K_p which was used to obtain the controllers parameters for both plants is 10. Then K_p has been changed to the values 4, 8, 10,12 14, 16, and for each value the system is simulated with the same controller.
Fig 6 and 7 show the step response for all of these values of K_p for FOPDT model and for IOPID and FOPI controllers respectively. Fig.8 and 9 show the step response for all of K_p values for SOPDT model and for IOPID and FOPI controllers respectively.

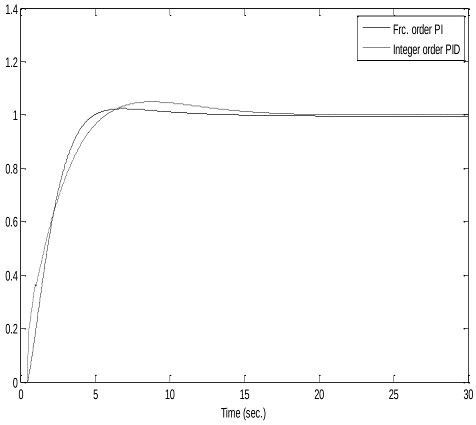

Fig. 2 step response for FOPDT.

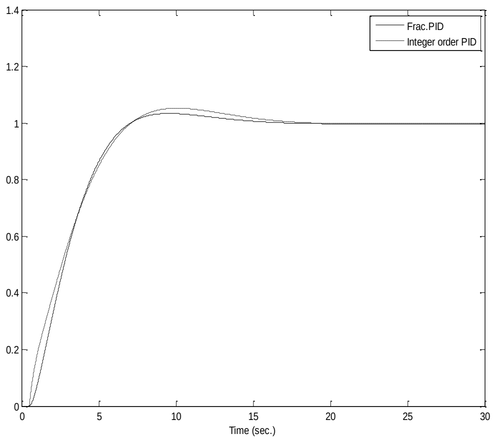

Fig. 3 step response for SOPDT.

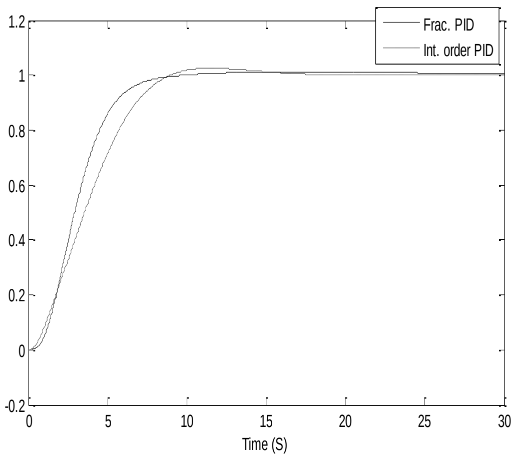

Fig. 4 step response for P1.

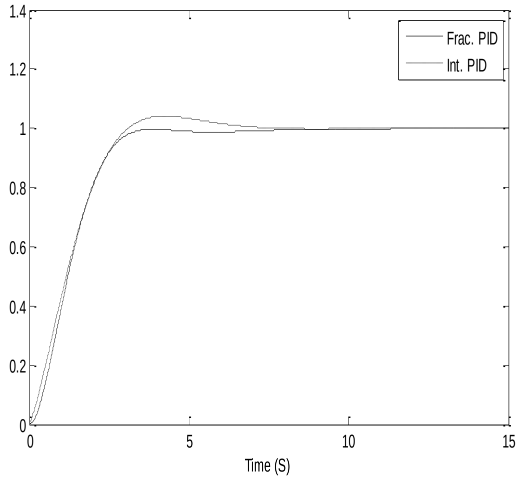

Fig. 5 step response for P2.

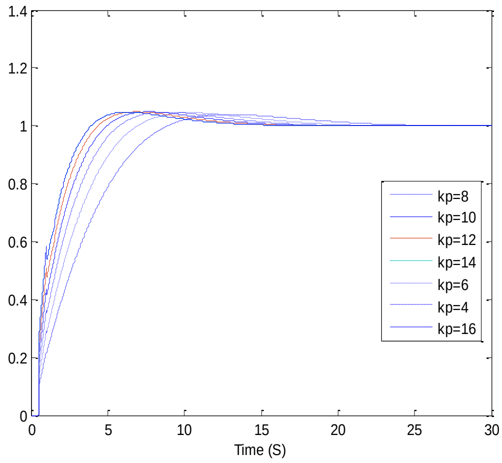

Fig. 6 step response for FOPDT for different values of Kp with IOPID controller.

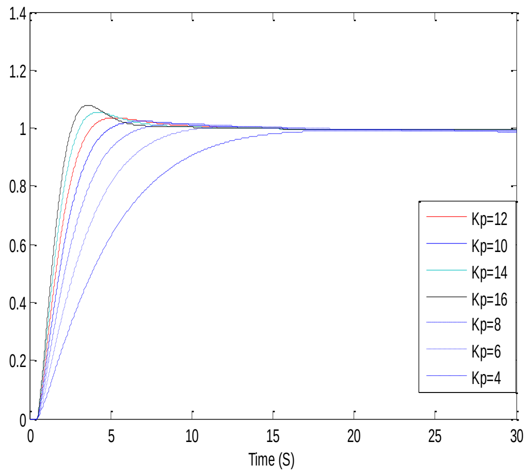

Fig. 7 step response for FOPDT for different values of Kp with FOPI controller.

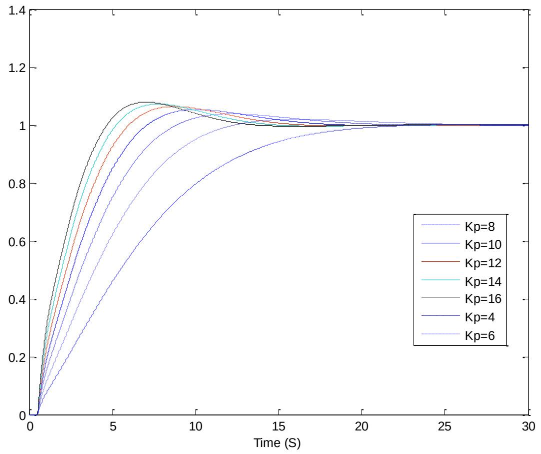

Fig. 8 step response for SOPDT for different values of Kp with IOPID controller.

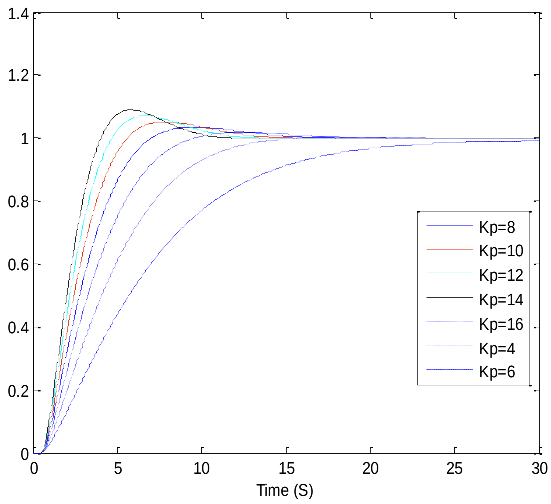

Fig. 9 step response for SOPDT for different values of Kp with FOPI controller.

VI. Conclusion

In this work, a design procedure based on some frequency response specifications for open loop system has been used to obtain the parameters of IOPID and FOPI controllers. The design procedure assumes general plant model which can be unknown, because the procedure gives the set of solution equations in terms of some frequency response parameters which can be obtained experimentally.
Simulation study using four different controlled plant model has showed that the method gives good results for both controllers, where as it is clear from the Figs, the response of all simulated cases have fast transient with nearly no overshoot and negligible steady state error.
If these results are used to compare the two controllers performance, then it is clear that the two controllers are nearly give identical performances with a small superiority for FOIP controller as it is clear from fig.4.
Simulation study included investigating the robustness of the proposed controllers to the gain variation. It has showed as it can be noted from Figs that the two controllers have good robustness and that IOPID controller performs better than FOPI.
Extending the method for other types of FOPI and FOPID controllers is our suggestion for future works.

VII. References

[1] Lin Kong Rong, “The analyses and comparing about several kinds of different definitions of the fractional step derivative [J]”, Minjiang river college journal in October, 2003, 24(5):3-4.
[2] I. Podlubny, “Fractional-order systems and PID controllers”, IEEE Transactions on Automatic Control 44 (1999) 208–214.
[3] S.E. Hamamci, “An algorithm for stabilization of fractional-order time delay systems using fractional-order PID controllers”, IEEE Transactions on Automatic Control 52 (2007) 1964–1969.
[4] H.F. Raynaud, A. Zergainoh, “State-space representation for fractional-order controllers”, Automatica 36 (2000) 1017–1021.
[5] A. Oustaloup, “The CRONE approach: theoretical developments and major applications”, in: Proc. Second IFAC Workshop on Fractional Differentiation and its Applications, Porto, Portugal, 2006, pp. 39–69.
[6] B.M. Vinagre, V. Feliu, “Optimal fractional controllers for rational order systems: a special case of the Wiener-Hopf spectral factorization method”, IEEE Transactions on Automatic Control 52 (2007) 2385–2389.
[7] H.S. Li, Y. Luo, Y.Q. Chen, “A fractional order proportional and derivative (FOPD) motion controller: tuning rule and experiments”, IEEE Transactions on Control Systems Technology 18 (2010) 516–520.
[8] I. Podlubny, “Fractional Differential Equations”, Academic Press, San Diego, 1999.
[9] R. Hilfer (Ed.), “Applications of Fractional Calculus in Physics”, World Scientific Pub Co., NJ, 2001.
[10] Mohammad Saleh Tavazoei, “Overshoot in the step response of fractional-order control systems”, Journal of Process Control 22 (2012) 90– 94.
[11] W. K. Ho, C. C. Hang and L. S. Cao, “Tunning of PI controllers based on gain and phase margin specifications”, ELSEVIER, Automatica volume31 issue 3, (1995) 496-502.
[12] D. Wang and X. Gao "Analytical Tuning of PI^{λ} Controllers with Phase Margin and Robustness”, IEEE the second international conference on intelligent control and information processing, 2011.