App Based – Comparative Study Of PID And FOPID Controllers

Rahul Chowdary Paladugula, Ramesh Sai Prasanth, Dr. Ananya Parameswaran

- *Rahul Chowdary Paladugula is currently pursuing undergrad degree program in electronics and communication engineering in PES Institute of Technology, Bangalore, India, PH-919491453513. E-mail: paladugula98@gmail.com*
- *Ramesh Sai Prasanth is currently pursuing undergrad degree program in electronics and communication engineering in PES Institute of Technology, Bangalore, India, PH-918919413489. E-mail: saiprasanthramesh@gmail.com*
- *Dr. Ananya Parameswaran is currently working as an assistant professor in dept. electronics and communication engineering in PES Institute of Technology, Bangalore, India, PH-918015401530. E-mail: anukv@pes.edu*

Abstract— PID controllers are being widely used in the industry due to their well grounded established theory, ease of retuning and simplicity. A controller is a device that is used to regulate the behavior or response of a process so as to get desired response. Designing a PID controller to meet gain and phase margin specification is a well known design technique but introduction of fractional order calculus in PID controllers gave a start to the new era with addition of two more parameters to tune for efficient system response. This has been compared with the help of an app based on Matlab Script for the designed models of PID and FOPID controllers with DC motor as a plant of the system.

Index Terms— DC Motor, Fractional Order, Gain, Integral Order, Matlab, PID controller, Simulink

1 Introduction

THE aim of this work is to design a matlab based app that does the comparative study of FOPID and PID controllers with DC motor as an application.

In this work, the advantage of FOPID controller over a conventional PID controller is discussed using MATLAB/Simulink based app. PID (proportional-integral-derivative) controller is one of the earliest control strategies. Its early implementation was in pneumatic devices, followed by vaccum and solid state analog electronics., before arriving at today's digital implementation of microprocessors. But recently, fractional-order dynamic systems and controllers has been studying widely in many areas of engineering and science.[1].

Introduction of FOPID will fine tune the response as it adds two more tuning parameters for the conventional integral order PID controller. The parameters are the orders of differentiation and integration denoted by λ and μ. The use of these along the parameters K_p, K_d, and K_i will improve the efficiency of the response of the system as it will be fine tuned.

The controllers being refered with respect to this paper as designed using Simulink with a Simulink based DC motor designed with optimized control parameters. It will be an armature DC motor as plant in both the controllers.

The comparision between the two controllers is proposed to be represented graphically with the help of a MATLAB based application which displays the output response of PID and FOPID controllers for the corresponding instantiated parameters for the selected model.

2 Background and Development

2.1 Existing PID controller

The PID controller is the most common general purpose controller in today's industries. It can be used as a single unit or it can be a part of distributed computer control system. After implementing the PID controller, now we have to tune the controller; and there are different approaches to tune the PID parameters like P,I and D. The proportional (P) part is responsible for following the desired set-point while integral (I) and Derivative (D) part account for the accumulation of past errors and the rate of change of error in the process or plant, respectively. PID controller consists of three types of control i.e. Proportional, Integral and Derivative control.

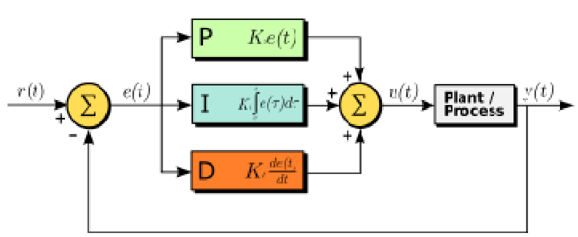


Fig.2.1: Schematic of PID controller

ISSN 2229-5518

The system transfer function is continuous s-domain are given as

$$G_c(s) = P + I + D = K_p + \frac{K_i}{S} + K_d S$$

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right)$$

Where, K_p is proportional gain, K_i is integration coefficient, K_d is the derivative coefficient, T_i is known as the integral action time or reset time and T_d is the derivative action time or rate time.

2.2 Fractional Order PID controller

The Fractional Order Control systems are described by fractional order differential equations. Fractional calculus allows the derivatives and integrals to be any real number. The FOPID is an extension of conventional PID controller where a new integral factor and a new derivative factor have fractional values add more flexibility and make the system less sensitive to parameters changes.[2]. The differential equation of the P controller can be described as follows:

$$u(t) = K_p e(t) + K_i D^{-\lambda} e(t) + K_d D^{\mu} e(t)$$

Where e(t) is the error between the measured process output variable and a desired set point and is the control output. Thus the transfer laplace function of the controller becomes:

$$G(s) = K_p + K_i S^{-\lambda} + K_d S^{\mu}$$

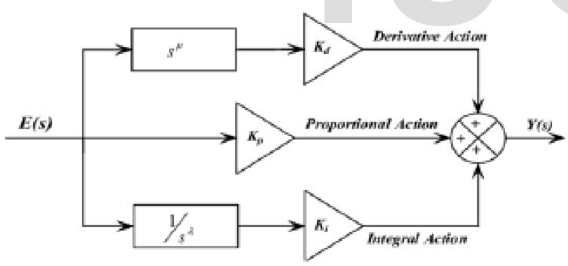


Fig.2.2: Schematic of FOPID controller

2.3 Chosen Design for Plant

An Armature DC motor is chosen as the application for both the controllers. The current objective of our project is to stabilize the operating speed of DC motorto a provided threshold value to meet the requirements of application.[3]

The transfer function for the DC motor speed control is derived as follows:

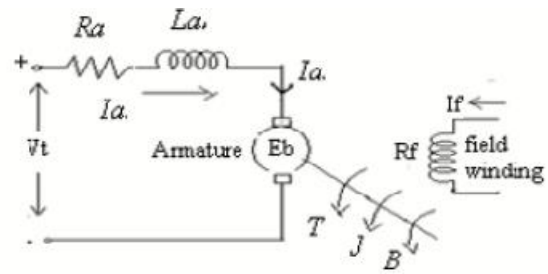


Fig.2.3: Schematic of Armature DC Motor

We know that the Air Gap flux is proportional to the field current i.e.,

$$\phi = K_f I_f \tag{1}$$

The torque is proportional to the armature current and air gap flux i.e.,

$$T = K I_a \tag{2}$$

The motor back emf is proportional to speed that is given by

$$E_b = K_b \frac{d\theta}{dt} \tag{3}$$

Now, applying KVL to the armature circuit we obtain the following equation i.e.,

$$V = R_a I_a + L_a \frac{d\theta}{dt} + E_b \tag{4}$$

The dynamic equation with moment of inertia and co-efficient of friction with load torque is

$$T = J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} + T_L \tag{5}$$

Now taking the laplace transform of eqn (2),(3),(4),(5);

$$T(s) = K I_a(s)$$
$$E_b(s) = K_b S\theta(s)$$
$$V(s) = I_a(s)(R_a + sL_a) + E_b$$
$$T(s) = (Js^2 + sB)\theta(s)$$
$$T(s) = (Js + B)s\theta(s)$$

The transfer function of DC motor speed with respect to input voltage is:

$$G(s) = \frac{W(s)}{V(s)}$$

$$\Rightarrow G(s) = \frac{K_t}{(R_a + sL_a)(Js + B) + K_b K_T} \tag{6}$$

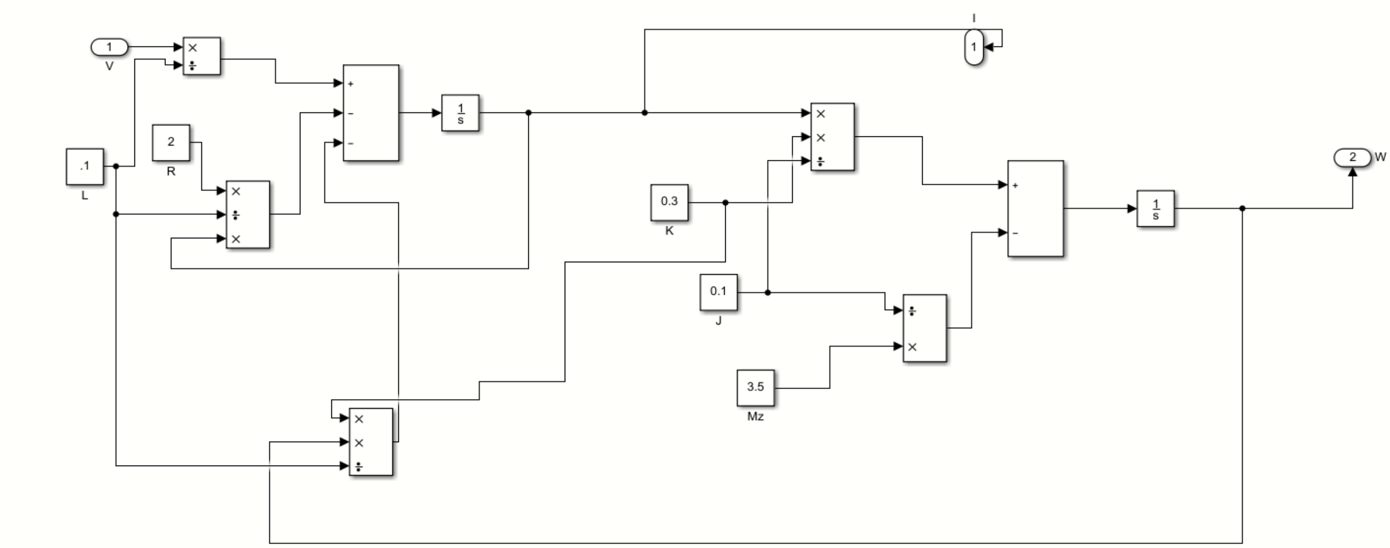


Fig.2.4: DC motor block built from considering various parameters

Furthur simplifying after approximations, the transfer function becomes;

(7) $$G(s)=\frac{K_m}{s\tau+1}$$

Where,

$$K_m=\frac{K_T}{R_a B+K_b K_T},\qquad \tau=\frac{R_a J}{R_a B+K_b K_T}$$

Finally used DC motor block can be seen in fig.2.4.

3 PID AND FOPID IMPLEMENTATION

3.1 PID Implementation

A discrete PID controller is implemented using Simulink to measure Speed, Torque and Voltage of DC motor [1] and then send it back as feedback. The PID controller will analyse the error signal between measured speed and desired speed and this error signal is used to calculate the voltage required to command the motor. The desired speed and actual speed are constantly compared.The error is calculated based on which the PID is tuned to attain the desired speed. The parameters K_p,K_d,K_i can be tuned with the response tuner to further tune the response according to the user requirement. Automatic tuning can also be done using matlab code of corresponding tuning algorithm if needed.

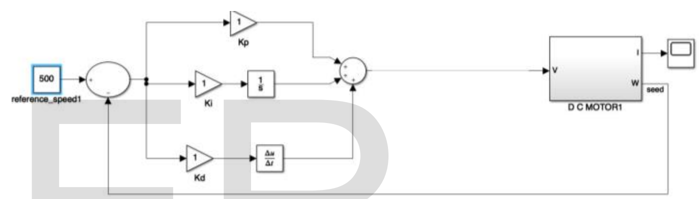


Fig.3.1: Implemented PID controller

3.2 FOPID Implementation

A parallel form of fractional order PID Controller has been implemented using Simulink to have more tuning freedom and thus also a wider region of parameters that stabilize the plant under control, offering improvements in robustness.

In general, to make a valid comparison between the performance of PID and FOPID controllers one can turn to global optimization based methods for tuning both controllers because that way the best possible controller gains and orders are assumed to be obtained. [4]

The Desired speed and actual speed are constantly compared. The error is calculated based on which the FOPID is tuned to attain the desired speed. The parametersof Kp,Kd, Ki are added with two more fine tuning parameters l and μ as the fractional order paramaters which can be tuned with the to get much finer and efficient response. FOPID controller usually has to use approximations which are often more complicated and require considerable computational resources. Still, modern embedded software solutions have been found to easily handle the additional implementation complexity (Tepl- jakov (2017)). Thus, as far as this comparative project is concerned, we focus on the benefits of FOPID controllers with respect to achievable performance improvement. [5],[6],[7].

ISSN 2229-5518

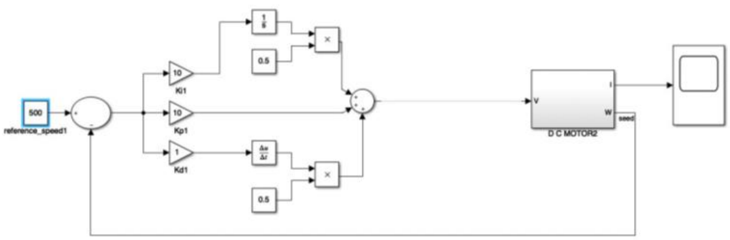


Fig.3.2: Implemented FOPID controller

4 Proposed Comparision App

4.1 App Overview

The user end interface (refer fig.4.1) have been developed using matlab script to present the comparative results of PID and FOPID for various parameters.

4.2 Working

Step1: Enter values for K_p,K_i,K_d & derivative order and store it. The stored values will appear in the command window.
Step2: Load the parameters using the *load_param* command.
Step3: Open simulink for model selection.
Step4: Simulate and obtain PID (fig.4.2(a)), FOPID (fig.4.2(b)) and comparative (fig.4.2(c)) responses.

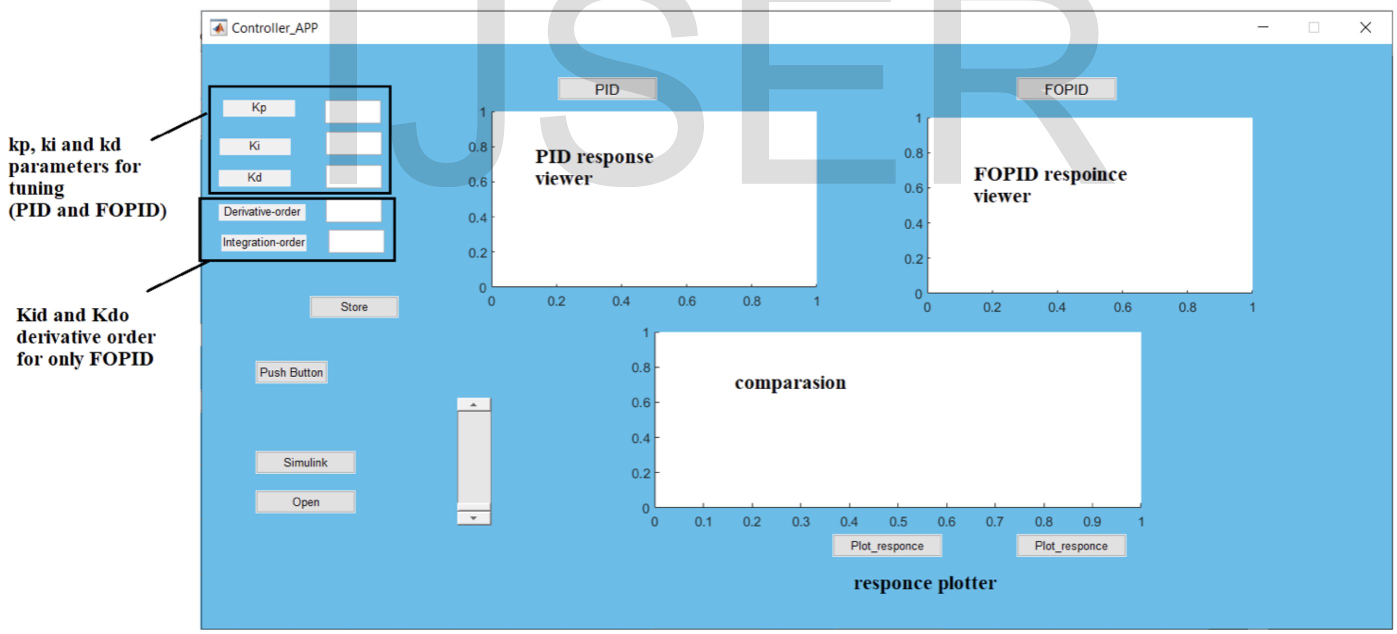


Fig4.1: screenshot of the total app interface.

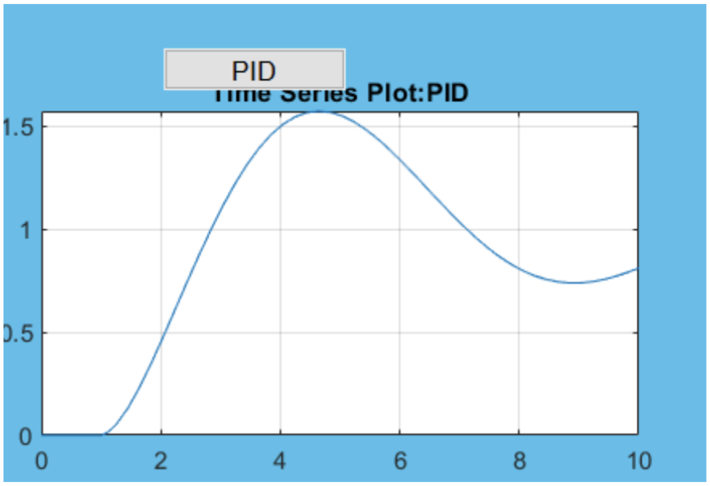


Fig.4.2(a): Implemented PID controller

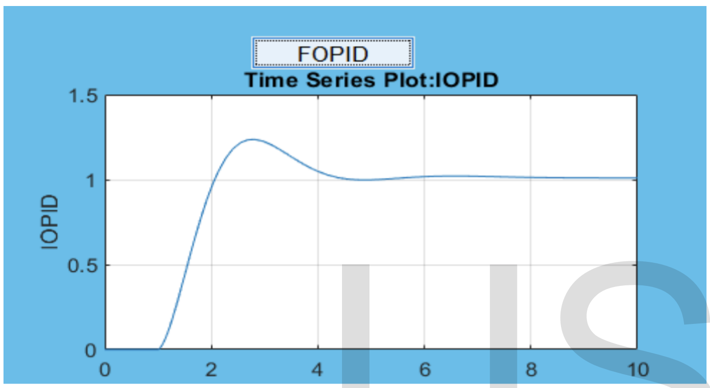


Fig.4.2(b): Implemented FOPID controller

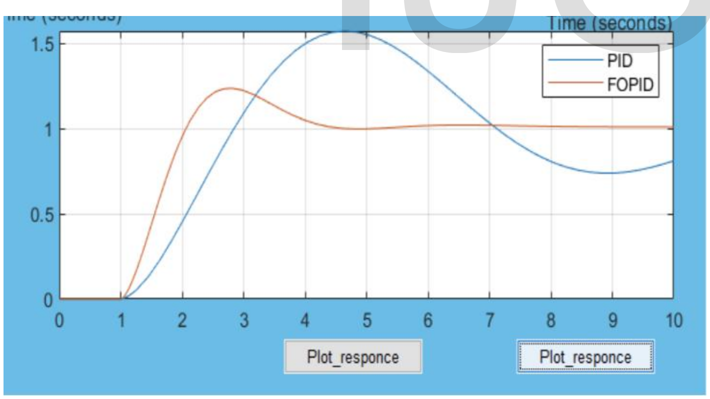


Fig.4.2(c): Comparative graph of PID and FOPID controllers

5 Result Analysis

The comparision of PID and FOPID response is done for a given set of parameters. The performance output parameters for the given inputs are overshoot, Rise time, Peak time and steady state error (settling time).

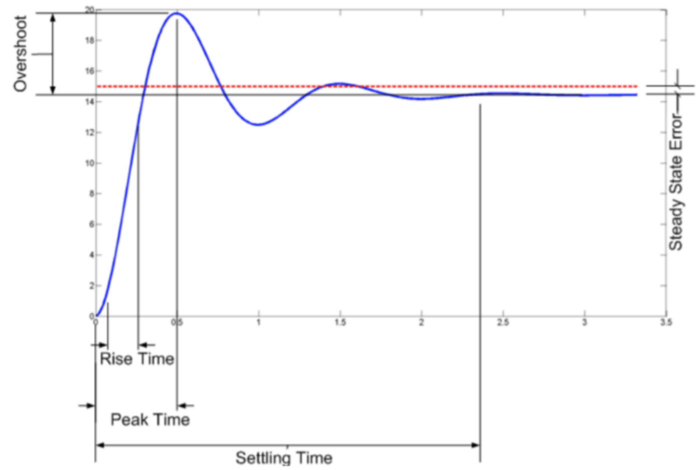
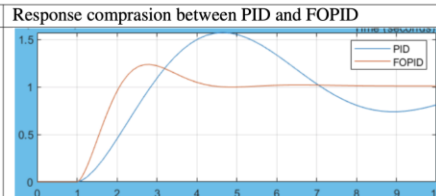
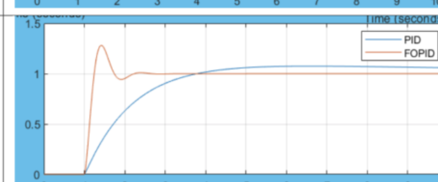
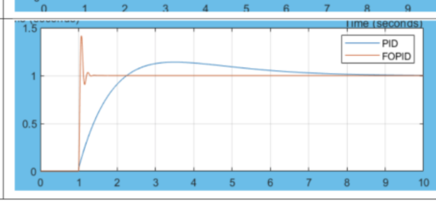


Fig.5.1: Different Parameters of Response Graph

K_p,K_i, and K_d should be varied as per algorithm to obtain the optimal response for PID and K_p,K_i, and K_d along with integrator and derivative order for FOPID as it can be seen for the below set of parameters it shows that FOPID response time is better than PID.[8] The criteria for better performance are

- The rise time should be fast.
- The overshoot and peak time should be less.
- The steady state Error or the settling time should be minimal.

Table 5.3
Graphical comparision with λ and μ set to 0.5 instantially.

S.no.	kp	kd	ki	Response comprasion between PID and FOPID
1	1	1	1	
2	10	1	10	
3	232	100	150	

Furthur FOPID an additional 2 extra tuning parameters exist i.e derivative and integration orders:

Table 5.1
Comparision of paramters for different combinations of λ>1 and μ>1

Integral Order	Derivative Order	Peak Overshoot	Peak Time	Settling time
0.5	0.5	26.5058	2.1536	3.5961
0.5	0.7	26.7243	2.1446	3.6187
0.5	0.9	26.9232	2.1085	3.4517
0.7	0.5	19.0836	2.1777	2.9069
0.7	0.7	19.2968	2.3212	3.0267
0.7	0.9	19.6501	2.2879	2.8258
0.9	0.5	15.0713	2.3615	3.0890
0.9	0.7	15.1803	2.3340	3.0644
0.9	0.9	15.1968	2.4555	3.0036

Table 5.3
Comparision of paramters for different combinations of λ>1 and μ>1.

Integral Order	Derivative Order	Peak Overshoot	Peak Time	Settling time
1.05	1.05	13.2209	2.4806	3.05914
1.05	1.1	11.8799	2.4159	2.9683
1.05	1.15	7.1071	2.7573	2.9573
1.1	1.05	12.7180	2.4831	3.0413
1.1	1.1	11.5889	2.5615	2.9388
1.1	1.15	6.9922	2.7544	2.9544
1.15	1.05	12.2015	2.4725	3.0215
1.15	1.1	11.2334	2.5383	3.0984
1.15	1.15	4.8675	2.7519	2.9519

6 Conclusion and Future Scope

6.1 Conclusion

From the above Analysis and Results, it is quite evident that FOPID controller performs better than PID controller for the selected application, DC motor. The user interface (app) simplifies the comparision by providing an easy way to instantialise different parameters for the controllers to work. Hence, the app based graphical comparision of responses of Fractional and Integral order PID controllers is concluded.

6.2 Future Scope

Fractional order PID controller being more flexible than its integral counter part can be applied to other types of plants such as cruise control, inverted pendulum, ball suspension, etc to enhance its control performance. The app made can be furthur optimized to check different tuning algorithms for different applications by appending the source code with a code for automatic tuning of the controllers.

6 References

[1] O'Dwyer, Aidan. *Handbook of PI and PID controller tuning rules*. Imperial college press, 2009.

[2] Shah, Pritesh, and Sudhir Agashe. "Review of fractional PID controller." *Mechatronics* 38 (2016): 29-41.

[3] Aditya Pratap singh, "*Speed Control of DC motor using PID controller based on Matlab* " Innovative Systems Design and Engineering, ISSN 2222-1727, Vol.4, No.6, 2013

[4] "*Fractional order Controllers versus Integral Order Controllers* ",10th International Conference Interdisciplinarity in Engineering, INTER-ENG – 2016

[5] *On Fractional order PID controllers*, Department of Elcetronics and Electrical Engineering, University of Strathclyde Glasgow, United Kingdom., IFAC Papersonline 51-4(2018) 739-744

[6] Igor Podlubny, "*Fractional Order Systems and Fractional Order Controllers* ", Department of Control Engineering, University of Technology, Kosice, Slovakia.

[7] C.H.Lubich, "Discretized Fractional Calculus", Academic Press, New York, (1974).

[8] Padula, F., Visioli, A., (2010a). Tuning rules for optimal PID and FOPID controllers, Journal of Process Control, doi:10.1016j.jprocont.2010.10.006