ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 5, May 2017

Study and Analysis of Fractional Order PID Controller for Pressure Process Station

[1] Chetan K. Hinganikar, [2] Kavitha. P
[1] [2] School of Electrical Engineering VIT University Vellore, India
[1] chetanh1808@gmail.com, [2] kavitha.p@vit.ac.in

Abstract: ***This project mainly focused on designing a next generation controller for Pressure Process Station. The plant model was obtained using system identification toolbox. Using Ziegler Nichols tuning technique the controlling parameter for conventional control algorithm was initially implemented in MATLAB. In order to overcome the drawbacks of conventional controller, Fractional Order Proportional Integral Derivative controller (FOPID) which was implemented using Fractional- Order Modeling and control (FOMCON) toolbox in MATLAB. The hardware implementation was done by interfacing MATLAB with Arduino Mega 2560 as a low cost Data acquisition unit. The performance of the Fractional-Order Proportional Integral Derivative controller parameter was compared with that of the conventional controller.***

Keywords: ***MATLAB-Arduino; fractional-order; data acquisition unit; control valve; E/P transducer; signal conversion; transmitter; real time; simulink; simulation .***

I. INTRODUCTION

In modern day technology, the performance of the systems are supported by the use of control system. By the survey and study on controller in the most of the industries, 30% of the controller works on manual mode and over 60% are not tuned properly [12]. Most of the industries uses properly tuned controllers for their real time hardware systems, while some industry uses controllers without tuning which badly affects the performance of the systems. This paper focuses on tuning of controllers for a pressure process system by using FOPID and Conventional PID controllers and then their performances be compared with each other.

Before going into the designing and real time hardware implementation section, survey was conducted and analyzed on various MATLAB-Arduino based projects which incorporates different controllers [6]. It has given a clear idea about interfacing and implementation which will help in preparing, analyzing the requirements and specifications new system so a precise design could be done [13]. MATLAB – Arduino interfacing is the best way of communication. Interfacing is possible using simulink support package. Arduino is one of the major impactful part of project as a low cost controller [4]. Low cost systems allow unexperienced users to do real time hardware implementation. So, most of people gives preference to Arduino mega 2560 for data acquisition. An Arduino IO toolbox for MATLAB simulink is available for code generation [5].

A proportional integral derivative controller, known as PID controller, is a negative feedback mechanism used in industrial systems based on three terms constants K_p, K_i, K_d. The negative feedback signal is known as the process variable. The error signal is the difference between the set point signal and process variable signal. The set point is the desired value of the process variable and the process is the recent status of any process.

Meanwhile the amount of error signal is needed to increase or decreases the process variable signal before its equal to the set point [14]. There are few types of tuning method such as Ziegler-Nichols, Cohen-Coon and the Kappa-Tau for controllers. Since, widely used and easiest tuning method is the Ziegler-Nichols which is easy to understand [15]. Fractional PID controllers are generalization of PID controller. FOPID controller are also known as $PI^{\lambda}D^{\mu}$ controller where λ and μ are the integration and differentiation orders, if both values are 1 the result called as integer PID that is usual PID controller [3]. A new toolbox is introduced for MATLAB that is FOMCON ("Fractional-order Modeling and Control") is developed by [1]. There are many advantages over manual control which is satisfied by automated process control. The application include mining facilities, waste water treatment plants, and the automotive industries etc. Maintaining process like pressure, flow, level, temperature, and pH within a desired operating range is most important in Process industries. Learning experience

IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 5, May 2017

with process control and instrumentation system will be the first step toward a successful career in the process control industry.

II. EASE OF USE

The operator or user should be able easily to tune the system using the MATLAB software. No more modifications is required for tuning purpose. There is flexible options to pause the control action and change the set point from the simulink model as per requirement and better trends. At a same time operator or user can modify it by simply doing code or simulink in MATLAB software. Within running mode also the user can change set point (psi) of control algorithm.

III. METHODOLOGY

A. Mathematical Model of FOPID and PID Controller

The most common form of fractional order PID controller is the P��� ��� controller, including � as order of integrator and � as order of differentiator whereas � and � can be any real number. The general transfer function of FOPID controller form

$G(S)= U(S)E(S)= Kp+KiS\lambda+ KdS\mu$, $(\lambda,\mu>0)$ (1)
Where, $G(S)$ is the transfer function of the FOPID controller
U(S) is controller's output
E(S) is an error Fig 1 shows block diagram of FOPID controller [7].

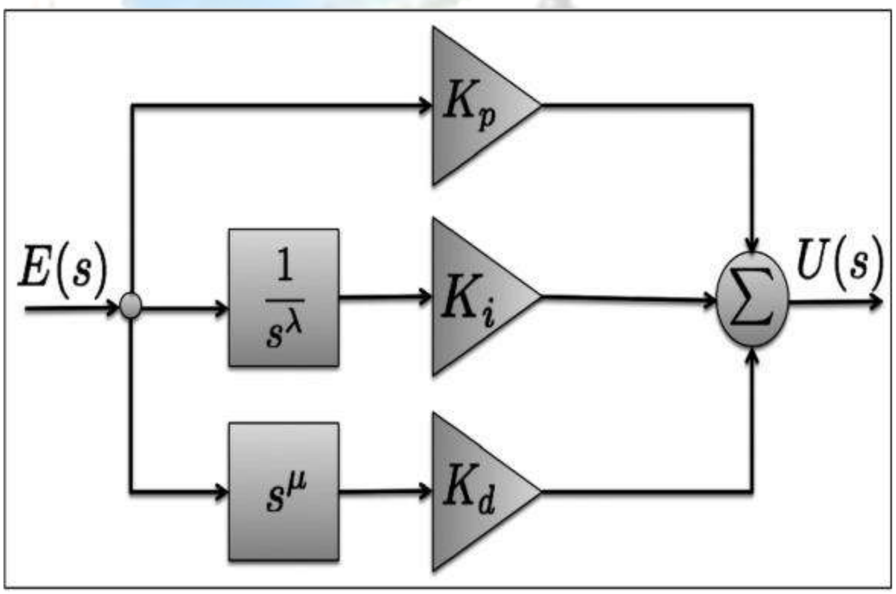

Fig. 1 Block Diagram of Fractional Order PID Controller

The selections of λ=0, μ1 and λ=1, μ=0 corresponds to conventional PD and PI controllers respectively. The classical types of controllers are exceptional cases of the fractional order controller [8]. Fractional order i.e. λ and μ might enhance the performance of control system. The major advantage of the $PI\lambda D\mu$ controller is better control of dynamical system, which is described by FOPID mathematical models [9]. FOPID controllers are less sensitive parameter changes. The performance evaluation of the model mainly depends on system error. The system error decides the difference between real response and the desired response of the system. The genetic algorithm based optimization technique is used for tuning of minimizing the Integral Square Error (ISE). The general form of PID controller is given by [7].

$u(t)= Kpe(t)+Ki∫e(\tau)d\tau+t0Kdde(t)dt$ (2)
Where, KP= proportional gain KI=integral gain
KD=derivative gain

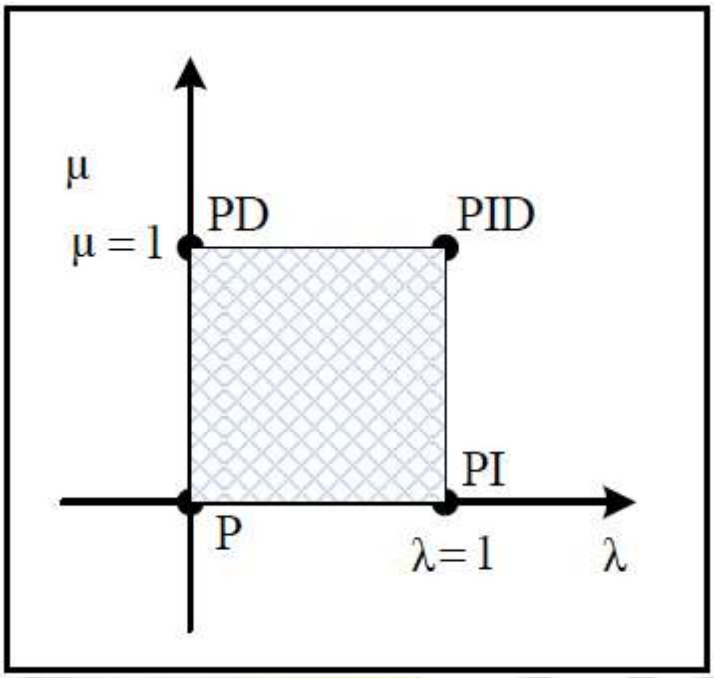

Fig. 2 Degree of freedom of FOPID

Conventional PID controller has 3 degree of freedom whereas FOPID has 5 degree of freedom.

B. Identification and Tuning Method

In this paper the FOPID tuning technique was used which will be useful for users using FOMCON toolbox which is associated with the design of controller for a system with input – output delay. To solve this, a fractional order first order with dead time system was taken, which was treated it in continuous time case. Taking a fractional order transfer function and it is expected that the actuator saturates at u= ±1. By chance, the transfer function is not available then we have to find out its test information data from the setup. The primary task was to find out or identify the system from datasheet of experimental station using System Identification toolbox. For this a simple command (fotfid) is used that will help to user. This will open FOTF GUI from the observation of transient response where we have decided a first order model with static gain, in addition to some input output delay and entered into GUI then normalized it. Finally the FOPID controller system was obtained for pressure control process station [11].

Conventional PID controller tuning and FOPID optimization steps:
Step 1:- Use iopid_tune GUI to approximate FOPID model by conventional model.
Step 2:- Apply classical tuning formulae to obtain PID controller parameters.
Step 3:- Use Ziegler-Nichols tuning formula to get Kp, Ki, Kd.
Step 4:-To tune FOPID (fpid_optim) GUI helps to get better response.
Step 5:- Using fpid_optimize_Gp.mdl FOPID system will get with tuned system.

IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 5, May 2017

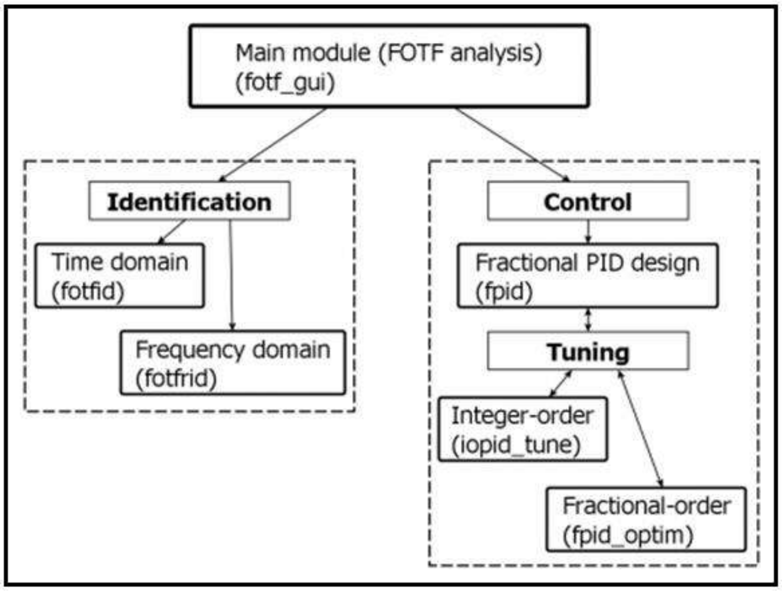

Fig. 3 FOPID Analysis Module

C. Mathematical Terms and Tuning Algorithm of FOPID (PIλDμ) Controller

1) Outstaloup's Approximation Algorithm:

Oustaloup algorithm is widely used algorithm in fractional order [17]. The algorithm which is used for a frequency domain realization. The approximation algorithm should be fix by a rank of integer-order filters to fractional order derivative. This method is based on the recursive-distribution of zero's and pole's. The range of desired frequency band to be fit which is given by ωA and ωB. Oustaloup algorithm approximates the general fractional order operator Sα (0< α <1) to an integer-order form within selected frequency band. Let (ωA , ωB) be the frequency band. Even within the selected frequency band both phase and gain have ripples, which decreases as rank of order of approximation algorithm increases. The approximation method is given as follows:

$S\alpha=K\ \Pi S+\omega'kS+\omega KNk=-N$ (3)

Where, K = Gain = $\omega B\alpha$

N = Order of approximation

$\omega'k=\omega A(\omega B\omega A)k+N+12(1-\alpha)2N+1=Zeros\ \omega k=\omega A(\omega B\omega A)k+N+12(1+\alpha)2N+1=Poles$

$\omega'k$, ωk are the rank of order of K for zeros and poles respectively and 2N+1 is the total number of poles or zeros. The condition of Oustaloup's approximation method may not be sufficient in frequency bands, near the selected fitting frequency bands.

2) Optimization Algorithms:

a) Nelder Mead Method:

The Nelder mead introduced a simple algorithm, first issued in 1965 which is a most popular direct search method for optimization. Nelder Mead method was used to find local minimum of function of interacted variables. In this method, triangle is used as simplex for two variables and its design search that function value at 3 vertices of triangle. When we get value which is out of region then we have to discard that value and take new vertex. Then forming of new triangle search is continued. Eventually it will acquire smaller and smaller vertices. Area of the triangle is minimized and minimum point is found. This is simple method for finding approximate roots of convex equation. Detailed explanation of Nelder Mead is shown in the following Fig 4.

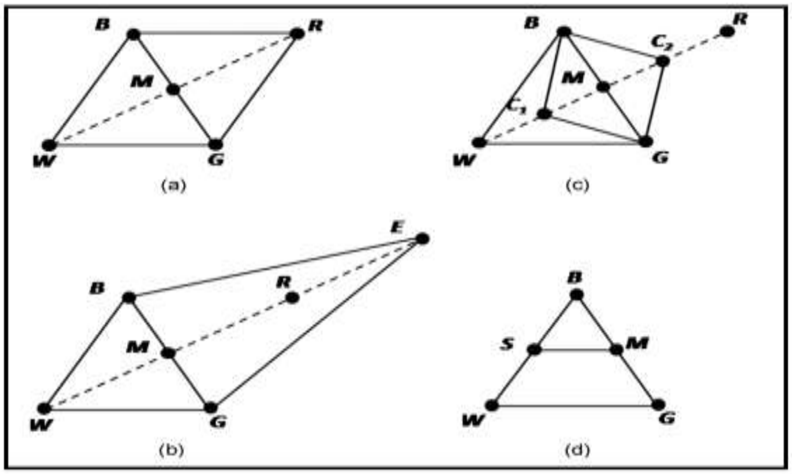

Fig. 4 Nelder-Mead for Value Optimization

IV. EXPERIMENTAL SETUP

A. Process Hardware Setup

The experimental setup consists of pressure process station, Arduino Mega 2560 and PC. The process station in Fig. 5 is available in laboratory in VIT University. The specifications of pressure process station are given in Table 1. The main component of system is cylindrical metallic tank with valve. The pressure inside the tank is the controlled variable and outlet flow rate is the manipulated variable [6]. The pneumatic control valve is paced at outlet side of tank. The transmitter needed a 12-24 V fixed DC supply to work it properly. The pressure transmitter measures the pressure inside the tank, transmitter converts pressure in 0-75 psi to 4-20 mA current signals. The 4-20mA current signal act as an input for the controller. The output of controller would be 4-20 mA current signal. The E/P converter gives pneumatic signal in 3-15 psi range according to controller generated controller signal.

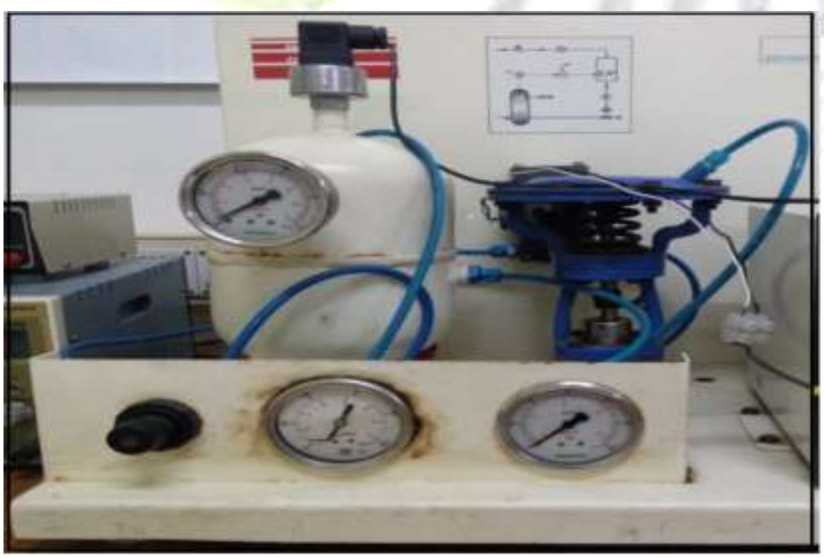

Fig. 5 Pressure Process System

IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 5, May 2017

The pneumatic signal operates the control valve depending on the control signal.

TABLE I. PRESSURE PROCESS STATION SPECIFICATION

Operational Specifications	**Functional Details**
Operating Voltage of Transmitter	24 V
Transmitter Type	2 – Wire
Input/ Output Signal range	4 – 20 mA
Pressure range (Tank)	0 – 100 psi
Control valve Type	Air to close
Pneumatic pressure	3 – 15 psig / 20 – 100 kPa

An Arduino Mega 2560 board is used for implementation of data acquisition unit. It is a microcontroller board based on the ATmega 2560. Arduino Mega 2560 has 54 digital input / output pins (of which 15 can be used for PWM outputs), 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 16 analog inputs, a USB connection, a power jack, an ICSP header, and a reset button. Arduino Mega 2560 can be connected to a computer with a USB cable or a 5 V battery or can be powered with an AC – to – DC (Arduino, 2017). Design will be in MATLAB (Simulink) and it can be used as a platform to code in Arduino Mega 2560 for acquiring and transmitting data. Support package i.e. Arduino IO library for Arduino Mega 2560 is available as freeware on MathWork website, thus it makes Arduino a low cost DAQ unit.

In two wire system pressure transmitter works on 4 – 20 mA current loop. To work properly pressure transmitter works on 12 – 24 V fixed DC supply which is taking from external source. To acquire the signal, using a current to voltage converter circuit which consist of 250 Ω resistance should beresistance because if we want 1 – 5 V by using Ohm's law we can get it exactly. Arduino Mega 2560 then reads the signal it will be in the form of 10 bit samples, so the voltage will be in range of 1 – 5 V then Arduino mapped it into 0 – 1023 value. These mapped values should be calibrated to the pressure range values using MATLAB (simulink). This shows lot of noise in response so, we have used low pass filter which is added in simulink to get smooth response.

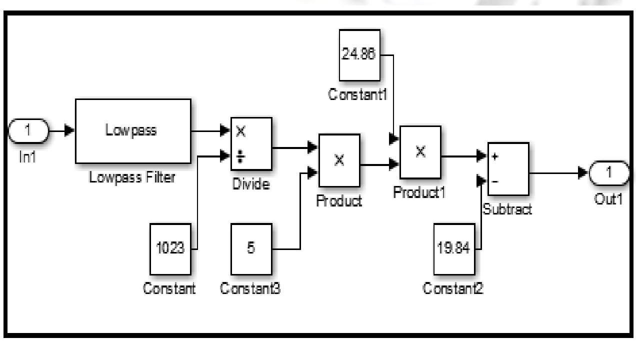

Fig. 6 Calibration and Data acquisition (V/P)

The pressure transmitter converts pressure into current in the range of 4 – 20 mA. Arduino Mega 2560 boards can transmit output signal only in the PWM (Pulse Width Modulation) or digital form. Hence, we use PWM it will depend on output. The average of duty cycle of PWM will depend on the output ranges from 0 –255 (8 bit samples) for output pin. Sensitivity plays significant role in real time hardware setup so, to get better sensitivity a bias with multiplication factor be added before it. The voltage from the PWM output pin will be average of it. The output voltage of PWM pin is then converted to 4 – 20 mA by using ideal voltage to current converter and then transmitted back to the E/P or I/P converter for controlling action. Hence the pressure will controlled.

IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 5, May 2017

V. REAL TIME IMPLEMENTATION

A. Simulink Model of PID Controller

First, PI controller was designed in MATLAB. The Ziegler – Nichols tuning method was used to calculate the tuning parameter K_p (Proportional gain), K_i (Integral gain) for PI controller. To convert the input Arduino voltage to pressure, the voltage to pressure (V/P) block is used.

P, PI controller implementation: For testing purpose of the DAQ a simple P controller has been implemented to run the system. The PI-controller is designed in MATLAB (simulink) using Arduino IO library as shown in Fig 7. Arduino reads the signal (value) from the pin 3 and provides it to next block, which is then converted from voltage to pressure. Then the control action will takes place, that value is give back to controller generated controller signal.

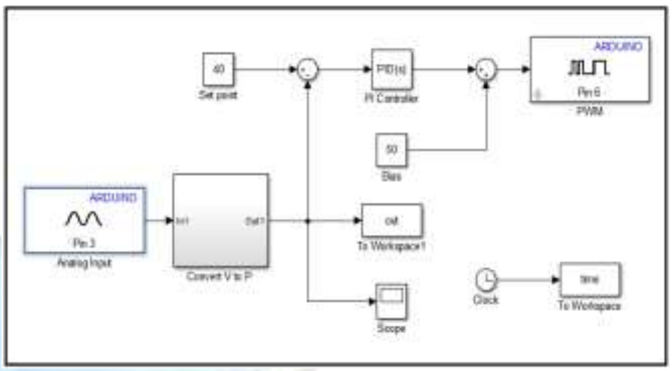

Fig. 7 Simulink Model of PID controller

PID Controller: It is the most widely used controller in industries. PID controller cannot provide enough performance in nonlinear and uncertain processes. In this project the transfer function is FOPDT, so for that it will not give proper response. For the proper output response of FOPDT we have to use Fuzzy Controller or Smith Predictor. But it will be more complex to implement it, hence PID controller are not used.

B. Simulink Model of Fractional order Controller

The same simulink model used for Fractional order controller (FOPID) instead of PI controller.

FOPID Controller: FOPID controller is an updated version of PID controller. In this the transfer function is FOPDT system in addition with FOPID controller also used in simulink. For dead time smith predictor can also use that make the response better. For dead time system the combination of FOPID and Smith predictor is always use in simulink. Smith predictor shows better response for FOPID than PID controller.

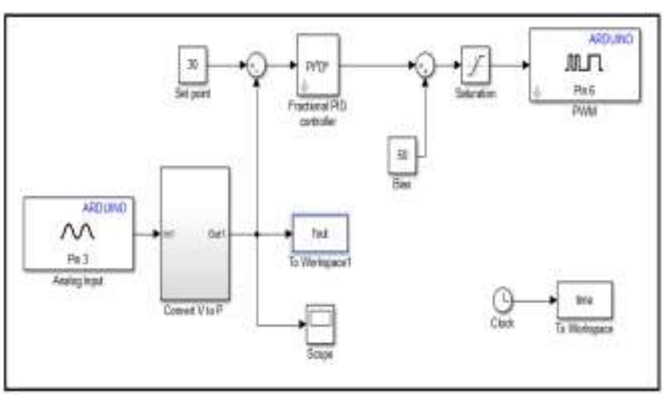

Fig.8 Simulink Model of FOPID controller

IJIRMET ISSN (Online): 2456-0448
International Journal Of Innovative Research In Management, Engineering And Technology
Vol. 2, Issue 5, May 2017

VI. RESULTS AND DISCUSSIONS

Initially, the real time hardware tested for the FOPDT model. The FOPDT system model used for this paper used is given by

$$y(s) = e^{-\alpha s}\frac{K}{\tau s+1}u(s) = e^{-0.5s}\frac{1.01}{18.42s+1}u(s) \quad (4)$$

The initial pressure of the process system was set as 25 psi inside the tank and 30, 40 psi set point was given to the simulink controller model for PI and FOPID respectively. The tracking of set point of pressure process system is shown in Fig 9, 10. The pressure was settled without any overshoot for FOPID as compare to PI controller at provided set point [6]. The Fig 10 shows that the pressure response with FOPID controller for a set point of 30 psi. The mostly used PID controller was considered for pressure process system [15]. Using transfer function model PI, FOPID controller parameters were tuned.

PI Controller: The PI-controller was designed in MATLAB (simulink) using Arduino IO library as shown in Fig 6. Arduino reads the signal (value) from the pin 3 and provides it to next block, which is then converted from voltage to pressure. Then the control action will takes place, that value is give back to Arduino PWM pin which will be written on. The simulink model was tested with a PI controller tuning parameter i.e. $Kp=20$, $Ki=0.0266$ with 40 psi. The output response of the system is shown in Fig 10, it clearly shows that even though there is offset but still the response is significantly better as compare to P controller. The PI controller response takes longer time to settle. The response of PI controller is good so it will be compared with FOPID controller.

FOPID Controller: The simulink model for FOPID controller tuning parameter i.e. $Kp=43.7703$, $Ki=43.7703$, $Kd=10.9426$, $\lambda=0.9$, $\mu=0.5$ was tested for 30 psi shown in Fig 8. In this the transfer function is FOPDT system in addition with FOPID controller also used in simulink. Because of addition parameter of FOPID over PID controller, FOPID provides better response for its tuning parameter.

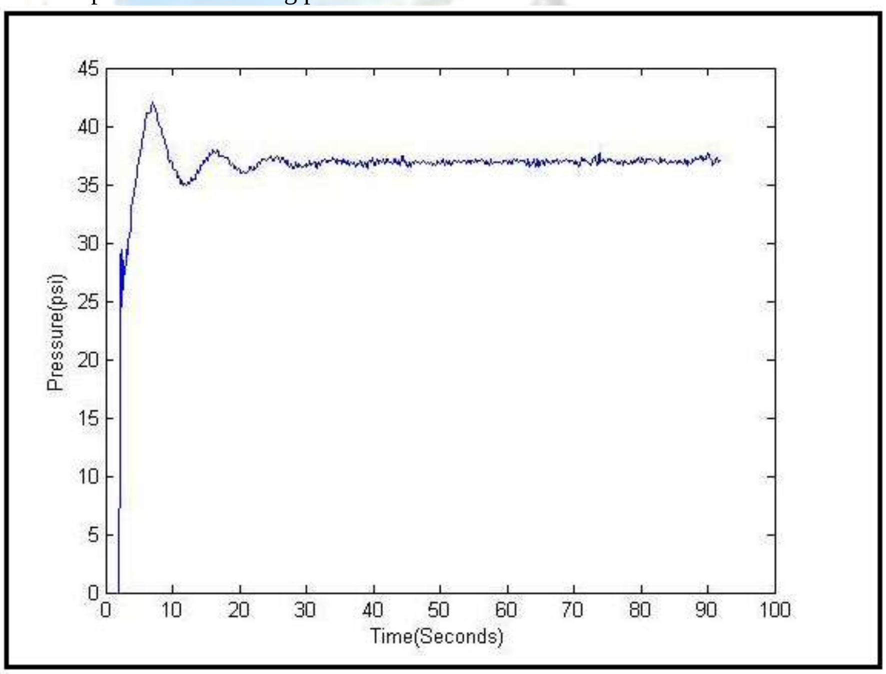

Fig.9 Output Response of PI Controller

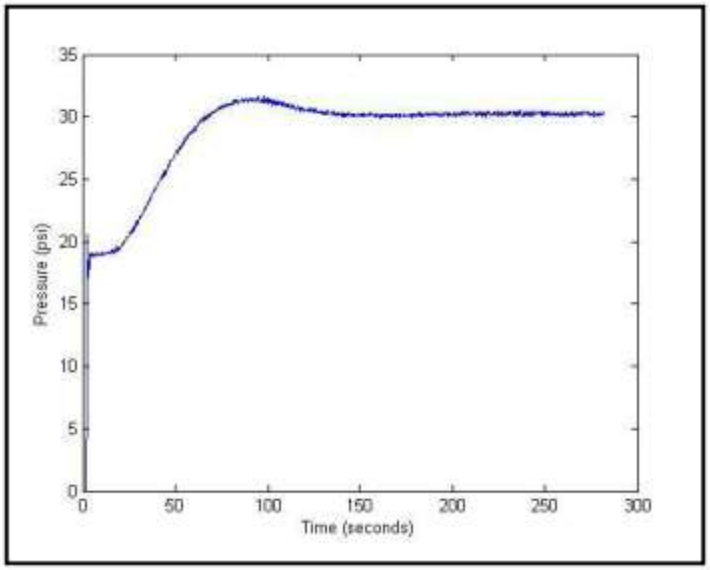

Fig.10 Output Response of FOPID Controller

It clearly shows that even though there is some offset then also it track the set point quickly and settle down with less rise time without oscillation as compare to PI controller.

VII. CONCLUSION

A next generation controller (FOPID) was used to control the Pressure process setup by use of simple interfacing unit. The performance of FOPID controller were tested for different value of parameter i.e. λ and μ by varying it. The FOMCOM toolbox along with MATLAB gives a simple interface with Arduino to implement P, PI, and FOPID controller in real time. The performance of the FOPID controller was desired as compare to other controllers.

TABLE II. Comparison between Conventional and Advanced PID Controller

PI Controller	**FOPID Controller**
K_p: 20	K_p: 43.7703, K_i: 43.7703, K_d: 10.9426
K_i: 0.02216	λ: 0.9, μ: 0.5
Overshoot: 13.6193%	Overshoot: 7.3041%
Settling time: 22.0683 Sec for 40 psi	Settling time: 103.8444 Sec for 30 psi
Peak: 42.0195	Peak: 31.1392

Hence, a next generation method of PID controller is used for controlling the pressure process system with interface unit. The data acquisition (DAQ) system and signal conditioning circuit were implemented on a low cost concept on the real time pressure process system and it proved to work efficiently. The main contribution in this work is the successful implementation of controllers on software with real time hardware system using a low cost data acquisition unit on very complicated pressure setup hardware for controlling the pressure of the process system. FOPID controller shows successful attempt to get better performance in terms of offset, rise time and settling time.

ACKNOWLEDGMENT

I would like to thanks my research guide who has provided me the valuable knowledge and opportunity to implement the project successfully. Also, support was from the School of Electrical Engineering, VIT University, India.

REFERENCES

[1] A. Tepljakov, E. Petlenkov, and J. Belikov, "FOMCON: Fractional order modeling and control toolbox for MATLAB," in Proc. 18th Int Mixed Design of Integrated Circuits and Systems (MIXDES) Conference, pp. 684-689, 2011.

[2] K. Astrom and T. Hagglund, "The future of PID control," Control Engineering Practice, vol. 9, no. 11, pp. 1163-1175, 2001.

[3] I.Podlubny, "Fractional order systems and PIλDμ controllers," IEEE Transactions on Automatic Control, vol. 44, no. 1, pp. 208-214, 1999.

[4] G. Berisch, H. Donath, F. Heinrich, I. Huebeneer, T. Lipke, T. Mende, M. Schriefer, H. Schulte, M. Zajac, "Design and Development of a low Cost Rapid Control Prototyping System Applied to an Air Suspension System" Proceeding of the 9th IFAC Symposium Advances in Control Education Nizhny Novgorod, Russia, June 19-21, 2012.

[5] http://www.arduino.cc/en/Main/arduinoboardMega2560

[6] V. Bagyaveereswaran, Tushar D. Mathur, Sukrit Gupta, P. Arulmozhivarman, "Performance comparison of next generation controller and MPC in real time for a SISO process with low cost DAQ unit" Alexandria Engineering Journal, pp. 2515-2524, 2016.

[7] J. Poovarasan, R. Kayalvizhi, R. K. Pongiannan, "Design of Fractional Order PID controllers for a CSTR process" International Refereed Journal of Engineering and Science (IRJES) ISSN (Online) 2319-183X, (Print) 2319-1821 Volume 3, Issue 1, pp. 08-14, 2014.

[8] B.M. Vinagre, C. A. Monje, A. J. Calder'on, J. I. Suarez, "Fractional PID controllers for industry application: a brief introduction", Journal of Vibration and Control, Volume 13, pp.1419-1429, 2007.

[9] C. A. Monje, Y. Q. Chen, B. M. Vinagre, D. Xue, V. Feliu, "Fractional order systems and controls", Springer London Dordrecht Heidelberg Newyork, ISSN 1430-9491, 2010.

[10] Sushant Narang, Sudha Ramasami, Prabhu Ramanathan, "Design of Self-Tuning Fuzzy Logic Controller", Sensors & Transducers, Vol. 190, Issue 7, pp. 63-71, pp. 2015.

[11] Aleksei Tepljakov, Eduard Petlenkov, Juri Belikov, Miroslav Halas, "Design and Implementation of Fractional-order PID Controller for a Fluid Tank System", Proceeding of The American Control Conference, 2013.

[12] D. B. Ender, Control Engineering, September 1993, pp. 180-190.

[13] Nawi Berahim, Sulaini Besar, Mohd Zain Abdul Rahim, Shamsul Aizam Zulkifli, Zairi Ismael Rizwan, "PID Voltage Control For DC Motor Using MATLAB Simulink and Arduino Microcontroller", Journal of Applied Environmental and Biological Science, 5(9), ISSN: 2090-4274,pp. 166-173, 2015.

[14] Abbas Emami, Naeini Gene, F. Franklin, J. David Powell, "Feedback Control of Dynamic System", Pearson, 6th edition, 2010.

[15] T. Hagglund, K. Astrom, "Automatic tuning of PID controllers", The Control Handbook, CRC Press, Boca Raton, FL, pp. 817-826,1996 .

[16] D. Chen, D. E. Seborg, "PI / PID controller design based on direct synthesis and disturbance rejection," Industrial and Engineering Chemistry Research, 41(19) pp. 4807-4822, 2002.

[17] A. Oustaloup, F. Levron, B. Mathieu, F. M. Nanot, "Frequency-band complex noninteger differentiator: characterization and synthesis," IEEE Trans. Circuits Syst., 44(1), pp. 25-39, 2000.