

# Simulating 'Electronic Pressure Regulator' using MATLAB Simulink

Jayant Unde<sup>1</sup>

<sup>1</sup>Mechanical department, College of Engineering Pune, India.

\*\*\*

**Abstract** - We need to invest lot of time for manual tuning of PID controller with lot of iterations. There is still no guarantee that we will get the best possible values of gains of PID parameters i.e. values of  $K_p$ ,  $K_i$  and  $K_d$ . In this paper we have discussed the example of PID controller loop system for EPR which acronym for Electronic Pressure Regulator. EPR is used to obtain the desired set point pressure at the output with minimum response time. Help of MATLAB Simulink is taken for obtaining minimum response time and obtaining the optimum dimensions of the mechanical system as the PID parameters vary with slight changes in design and dimension. Complete EPR system is modelled in Simulink block diagram and simulated for getting the optimum gain values of two PID controller blocks. The EPR consist of four subsystems which are DC motor, PID controller for DC motor, Pneumatic circuit and feedback from the pressure sensor. Whole system of EPR is iterated in Simulink for getting the best possible values of gains of PID controller. Simulink model of EPR, first PID controller is used to control the DC motor rotation while second PID controller is used to control a mechanical pressure regulator knob to get the desired pressure at the outlet. Finally, with help Simulink auto tuner we can get the desired values of gains required for our system. It not just the values of gain, using MATLAB Simulink we can decide the required actuators without building the prototype. Hence saving lot of cost of building iterations of prototypes

**Key Words:** MATLAB, Simulink, PID controller, gain, Electronic Pressure Regulator, Robotics and Automation.

## 1. INTRODUCTION

Today's rapid growth of technology in robotics calls for many types of actuators and control system for controlling and operating the robots. In robotics mainly two sources are important for working of robot which are electric and pneumatics. There are multiple advantages of pneumatic sources over hydraulic sources as it has some unique features which suits robotics operation. In pneumatic system compressed air is used for actuation of pneumatic actuators.

To obtain the regulated air pressure from the pneumatic air reservoirs, a pneumatic pressure regulator is used.

This pressure regulator is a mechanical component in which we can set a desired pressure by manually adjusting the control knob. It works fine for single value of pressure but in case of variable operation we need something that can change the pressure output dynamically, in this case PPR is used which is a proportional pressure regulator. Many companies manufacture this product, but they are quite expensive, so there is need for developing a device which is

cheap and can perform pressure regulating operation with desired level of precision. In this paper we discuss the EPR electronic pressure regulator that we developed. EPR is a closed loop control system which utilizes PID algorithm to control the pressure control knob. For a PID controller it is very important to finely tune the system to obtain optimum stable system. The manual trial and error-based system for tuning requires many iterations with the system being not optimum.

So, for that we found solution of utilizing simulating whole circuit in MATLAB Simscape [1]. By simulating it we get the values for PID controller which are used with some modifications on actual hardware implementation. Using MATLAB, we are able save lot of time and got desired results.

## 2. EPR (electronic pressure regulator)

EPR stands for electronic pressure regulator it is cheaper version of PPR proportional pressure regulator available in the market. PPR is used to control the output pressure of given storage. Most commonly used PPR in market are Festo [2] and SMC [3]. EPR consist of IG 32 DC Motor with Encoder [4], Digital pressure sensor, and a mechanical pressure regulator (Spec.) EPR utilize ARM Cortex based STM 32 microcontroller [5] for its processing. In EPR there are two closed control loops; one for giving feedback of dc motor and other giving feedback of pressure. Both closed loops are used with PID controller for control action. EPR is designed for controlling pneumatic pressure in between 0 to 8 bars required for robotic application. In case of robotics application fast response of system is very important. In that case it's very necessary to set PID controller such that output pressure as quickly as possible. For that tuning of PID loop is utmost important. The values of PID controller such that it gives a minimum response time.

### 2.1 Design of EPR

CAD modelling of EPR is done in Solid works. EPR consist of a 12 v DC motor with encoder, a mechanical pressure regulator, Janatics Digital pressure sensor [6], flexible coupling and custom designed mounting plate for assembly. DC motor has shaft diameter 6 mm and the mechanical pressure regulator knob is square having each side 6mm. To connect them a custom flexible coupling is used which has 6mm diameter hole on one side and square hole with side of 6mm on the other side. Flexible coupling can compensate

center misalignment of shaft hence it used. Mounting plate is made of aluminum and manufactured by laser cutting and CNC bending which offers accuracy, because of that assembly of component become quite simple.



Fig.1 Rendered image of EPR

### 3. Closed loop control PID

There are several ways to tune the PID loop. The most effective method generally involves developing a form of the process model and includes selecting P, I, and D values based on dynamic model parameters. However, this method sometimes requires complicated and time-consuming mathematical use. The manual tuning method can be relatively inefficient, especially if the response time of the loop is on the order of only a few minutes or more. Likewise, the manual tuning method also requires many trials, and the error takes time to tune the PID.

#### 3.1 Manual tuning

With this adjustment method it is necessary to set the values of  $K_i$  and  $K_d$  to zero and increase  $K_p$  until the output of the loop oscillates and set  $K_p$  to about half of the response for the "quarter amplitude decay" type. The value of  $K_i$  then increases until the offset is corrected within an enough time for a given process. However, when  $K_i$  is high, it becomes unstable. Therefore, caution is required while selecting the value of  $K_i$ . Finally, if necessary, raise the value of  $K_d$  until the criterion is sufficiently fast until the criterion is reached after the disturbance. However, too much  $K_d$  causes excessive response and overshoot [1]. Normally, fast PID loop tuning slightly overshoots until the set point is reached. However, since some systems cannot accept overshoot, an over damped closed-loop system that requires a  $K_p$  setting of less than half of the  $K_p$  setting is required.

$K_p$ : Proportional gain, a tuning parameter

$K_i$ : Integral gain, a tuning parameter

$K_d$ : Derivative gain, a tuning parameter

$$\text{Error} = \text{Set Point} - \text{PV}$$

Table -1: Ziegler Nicolas method tuning based on time response

| Type of controller | $K_p$   | $T_i$    | $T_d$ |
|--------------------|---------|----------|-------|
| P                  | T/L     | $\infty$ | 0     |
| PI                 | 0.9 T/L | L/0.3    | 0     |
| PID                | 1.2 T/L | 2L       | 0.5L  |

### 4. Matlab Simulink

MATLAB Simulink is very powerful tool to simulate various physical system and solved to get the required result. In Simulink blocks are and connection are used to model a problem it relinquishes user from knowing a programming language that without using a programming language you can still model your physical system into MATLAB and simulate it to get the result. Simulink helps user to simplify a model and build a better understanding of a problem and it's very easy to use powerful as well.

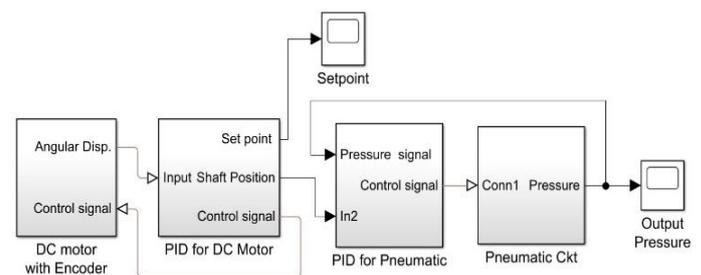
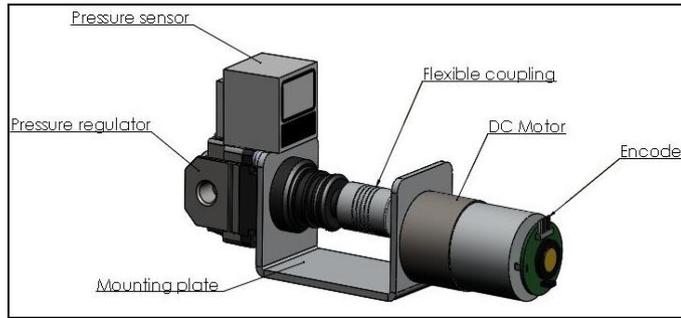


Fig. 2 Main block in Simulink

In EPR as there are two PID controller for getting the desired output pressure it was very painful to tune the PID. Here, MATLAB Simulink comes handy. EPR is completely modeled in MATLAB Simulink to simulate it before manufacturing. It gave us idea about whole system and most importantly setting the control parameter of each PID loop. As EPR is required to work on variable condition such as every time it's not used to control the output pressure for same Pneumatic cylinder. It is used with different cylinders with different signal dimensions. So, in this case also Simulink helped us for simulating the entire pneumatic circuit. One can easily

get the displacement of cylinder with respect to time as well as the output pressure. Figure [] shows the model of EPR in Simulink which consists of four subsystem which are dc motor with encoder for rotating the knob of mechanical pressure control valve, a PID for controlling DC motor, another PID for controlling output pressure and pneumatic circuit. Along with subsystem two scope are used for showing set point and viewing resultant output pressure.



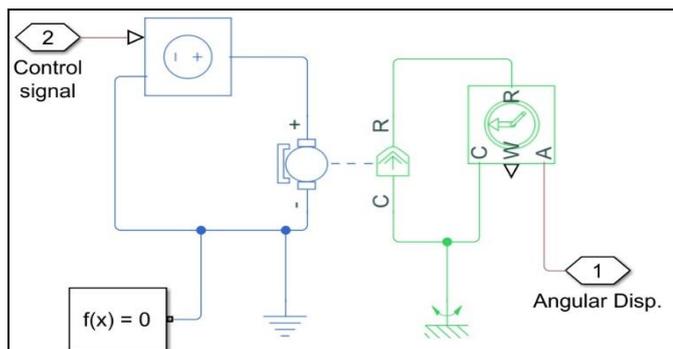
**Fig.3.**CAD model of EPR with component architecture

**4.1 Subsystem of EPR in Simulink block**

Simulink model of EPR consist of total four subsystem which are Dc motor, PID for DC motor, Pneumatic system and PID for pneumatic system. The interconnection between all the blocks are as shown in fig .2. In this figure the scope of setpoint and output pressure scope is shown along with other subsystems.

**4.1.1 DC Motor**

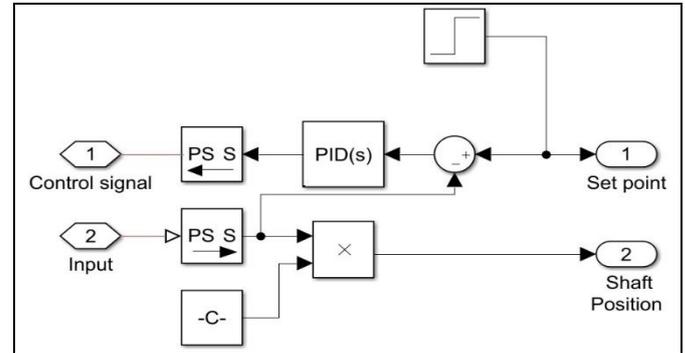
This subsystem consists of DC motor block along with voltage source and an ideal rotation sensor block. Ideal rotation sensor block measures the output angular velocity of the DC motor which can be used as feedback signal for closed loop control system. And the output of PID controller is given to the voltage regulator for controlling the output of the DC motor. In actual prototype DC motor with encoder is used as shown in fig.2. The motor used is IG32 12 V motor having 100 rpm and 1.2 N/m torque.



**Fig.4.**DC Motor Subsystem

**4.1.2 PID controller for DC motor**

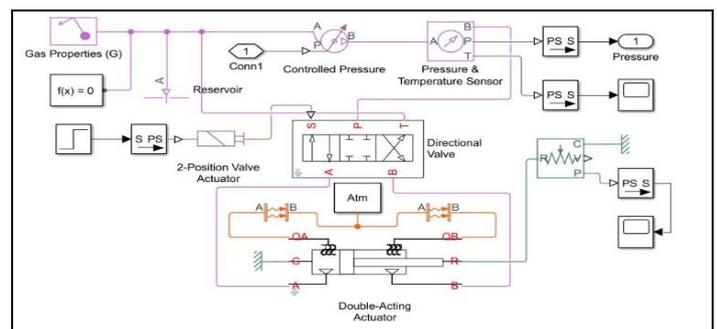
This subsystem has the PID controller for closed loop control of DC motor. It consists of a PS to S converter which convert physical output signal of ideal rotational signal into electronic signal. Also signal from set point is taken for displaying it in scope.



**Fig.5.** PID for DC Motor Subsystem

**4.1.3 Pneumatic subsystem**

In this subsystem multiple blocks are used to model physical pneumatic circuit from Simulink's pneumatic library. Here, a Pneumatic reservoir is used to store air at the pressure of 6 bar. Controlled pressure block function as mechanical pressure regulator and controlled by PID. Pressure and temperature sensor are used for sensing output pressure and temperature of fluid which required for closed loop feedback circuit. A 4-2 directional control valve is used to actuate the pneumatic cylinder which is in turn is operated by position valve actuator. By actuating direction control valve, one can get forward and reverse stroke. A double acting piston cylinder actuated by directional control valve and its actuation sensed by translation sensor. From translation sensor we can get position and actuation velocity which can be viewed with help of scope. In given circuit PS to S and S to PS converter block is majorly used. They are used to convert physical signal to electrical signal and an electric signal to a physical signal respectively.



**Fig.6.** Pneumatic Subsystem

#### 4.1.4 PID controller for pressure regulator

This subsystem consists of 2nd PID controller for controlling the output of pressure regulator. Input signal to this block comes from DC motor subsystem after adding into gain which for converting it into Pascal value of pressure and takes as set point. PID controller signal are then fed to S to PS block for converting them from electronic signal into a mechanical signal.

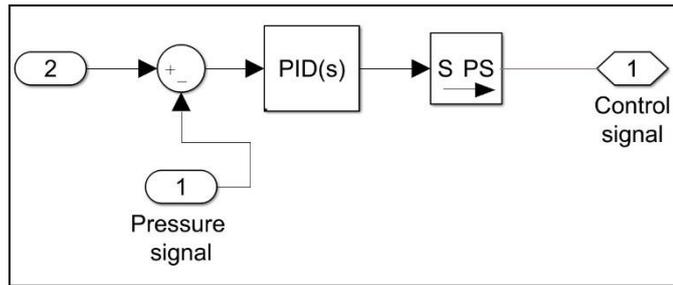


Fig. 7. PID for Pneumatic Subsystem

### 5. GRAPHS AND RESULT

Following graphs shows the response of PID controller before tuning it for optimum response. We can see that the graph shows fluctuation before attaining the desired value of pressure i.e. set point pressure. Such fluctuation will lead to strain on the hardware of the system and in extreme cases failure of system. Thus, it is necessary to get the optimum values of gain such that it will give minimum response time and with no fluctuation which is a smooth response.

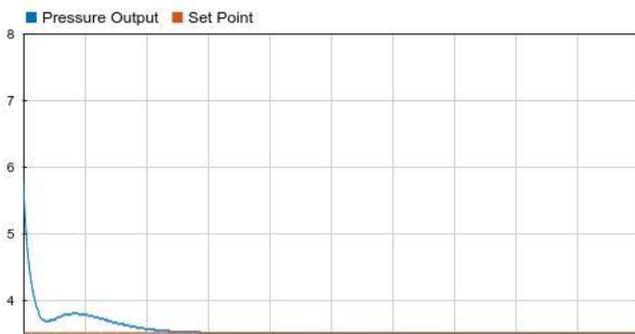


Fig.8. Before tuning

MATLAB Simulink has a very unique feature for tuning the PID controller which is auto tuner. In auto tuner one can get the desired trajectory of PID controller response. One can iterate the whole process of setting the gain value of PID without straining the actual hardware which gives designer a motivation to use MATLAB Simulink as it gives chance to optimize the system without building. As there is no hardware part required, so without straining the actual hardware which can cause damage of expensive hardware in

other words Simulink let you optimize your system without braking.

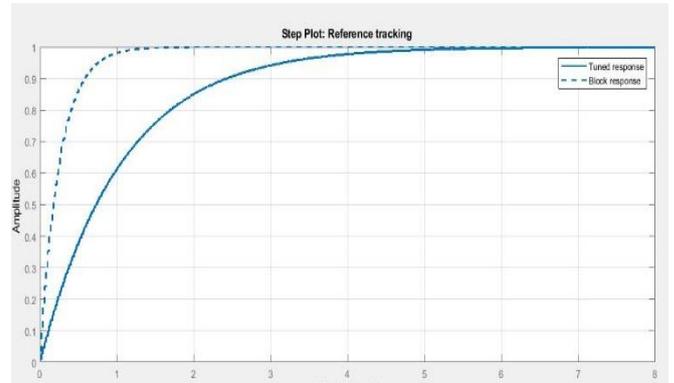


Fig. 9. Simulink auto tuner

In auto tuner one can tune the system for minimum response time or without excessive fluctuation as per the need. We can set gain of PID to get the optimize gain. In case of PID controller one can get minimum possible response time at the expense of smooth trajectory response so one has to compromise between smooth response and minimum response time. MATLAB Simulink gives user flexibility to set both the parameters as per the need of system. In automatic PID tuning auto tuner automatically adjusts the gain of the PID controller and fine-tunes it interactively with the user. It can also Adjust many PID controllers in batch mode.

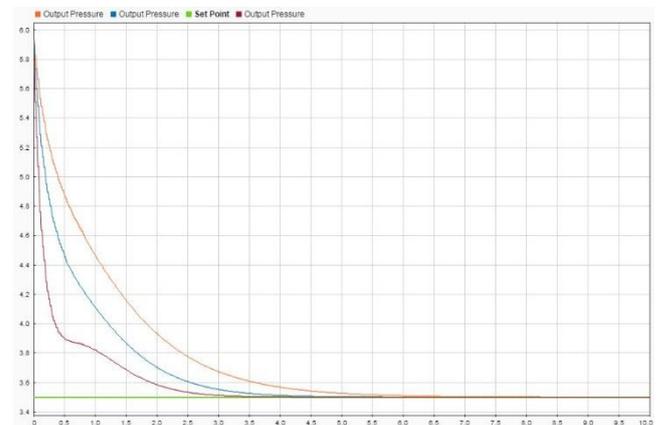
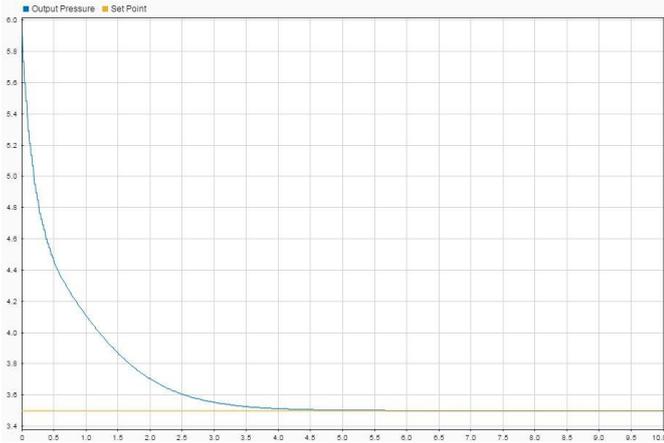


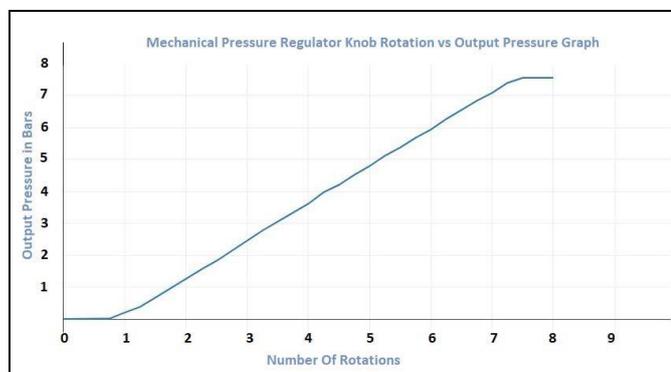
Fig.10. comparing response

Above graph shows the comparison between response time and output trajectory of PID controller. It is easily noticed that by decreasing the response time below certain threshold value the response of PID controller start fluctuating due to this user must set the optimum value of gain for the system. One can easily perform this task using MATLAB Simulink without expending much time. User just have to build his entire system in Simulink and after that he can iterate his system for multiple time.



**Fig.** Smooth response

With help of this MATLAB Simulink, we are able to tune the Electronic pressure regulator with ease and without number of iterations. The results of Simulink model were realistic. We were able to use those values with little changes as it is in our prototype. This prototype of electronic pressure regulator was used in robotics project of disc shooting robot. The Simulink model is also used for selecting the right actuator for the robot, as the model can also measure output forces of Pneumatic cylinder. Because of that selection of pneumatic actuator is done without building actual prototype beforehand.



**Fig. 11.** Plot of rotations of pressure control knob vs. change in actual output pressure.

Above plot shows the output pressure of actual build model of EPR vs. number of rotations of the knob of mechanical pressure regulator. In this graph input and output are proportional to each other.

## 6. CONCLUSIONS

MATLAB Simulink is a versatile tool which can be used in variety of problem solving situations. This tool can greatly reduce effort required in manual tuning of PID controller and can give realistic values of gain which can be directly used to tune the controller. Also building a model in Simulink

can help in selection of actuator for task in hand without building a prototype. This helps in reducing the cost of the project as multiple iteration are done virtually without building prototype and stressing the physical hardware components; which could sometimes result in breaking of component. So, Simulink helps in designing system without building and breaking. One can easily check feasibility of design (weight/speed/power etc.) and verify preliminary algorithms using MATLAB Simulink, saving lot time and funds.

## REFERENCES

- [1] Gaurav and Amrit Kaur, "Comparison between Conventional PID and Fuzzy Logic Controller for Liquid Flow Control: Performance Evaluation of Fuzzy Logic and PID Controller by Using MATLAB/Simulink," International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-1, Issue-1, June 2012
- [2] [https://www.festo.com/net/SupportPortal/Files/26930/info\\_241\\_en.pdf](https://www.festo.com/net/SupportPortal/Files/26930/info_241_en.pdf)
- [3] [http://ca01.smcworld.com/catalog/en/frl/ITV-D-E/6-6-p0893-0946-itv\\_en/pageview.html](http://ca01.smcworld.com/catalog/en/frl/ITV-D-E/6-6-p0893-0946-itv_en/pageview.html)
- [4] <https://www.superdroidrobots.com/images/ig-32-datasheet-1.jpg>
- [5] Steven Barrett, "Embedded System Design with the Atmel AVR Microcontroller: Part I," in Embedded System Design with the Atmel AVR Microcontroller: Part I,1, Morgan & Claypool, 2009, pp.176- doi: 10.2200/S00225ED1V01Y200910DCS02
- [6] JANATICS Pneumatic, "Digital Pressure Sensor (Two output) Series PS," PS - 01 - 01 - B datasheet