Contents

PID SIMULATION

A Thesis submitted in partial fulfilment for the 7th semester project for the Degree

INTEGRATED MSC
in
PHYSICS
by

RAVI PRAKASH SINGH
Roll Number- 1811122

Supervisor: **Prof. Pratap Kumar Sahoo**
(Associate Professor,(SPS) NISER)

Cosupervisor: **Dr. Gunda Santosh Babu**
(Scientific Officer-E,(SPS) NISER)

SCHOOL OF PHYSICAL SCIENCES
NATIONAL INSTITUTION OF SCIENCE EDUCATION AND RESEARCH
BHUBANESWAR

Abstract

In this project, we focus on the PID simulation using Python code and also by doing circuit simulation using Multi-sim. Among controllers, PID controllers are being widely used in industry due to their well-grounded established theory, simplicity, maintenance requirements, and ease of re-tuning online.I have done the relative study of PID by changing the values of P, D and I term.Here we have figured out which of the parameters affect what thing in the control so that we can get to our desired tuning in small time. We have made a circuit which simulates as if a heater and have used to see the PID function in Multi-sim.The circuit is the most basis where we may feel the need of an PID.

ACKNOWLEDGEMENT

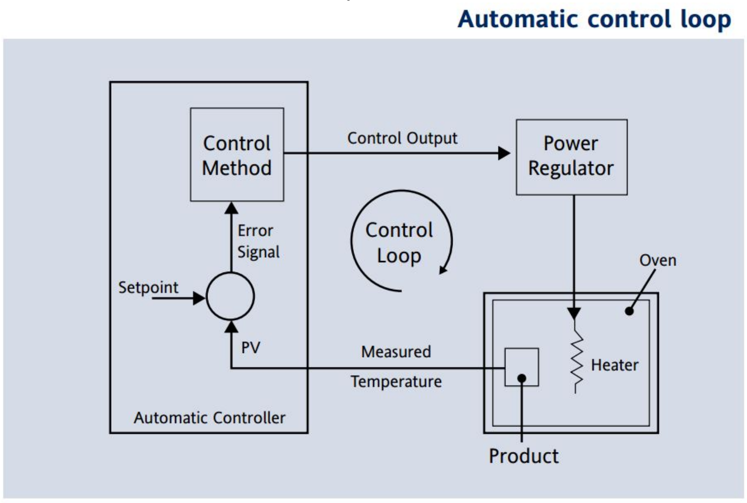
I sincerely thank **Prof. Pratap Kumar Sahoo** for granting me this opportunity to study and simulate PID. From the core of my heart I am extremely thankful to **Dr. Gunda Santosh Babu** for providing the help and guidance while learning and simulating PID, without whom this developing the PID simulation might not be possible in time. Once again I thank him giving me invaluable knowledge and insight into the field and guiding me towards the proper direction each time I went astray.
Last but not the least I would like to thank my family, friends and PhDs who supported me way over emotionally . This noble work won't be fruitful without you people, I thank you all from my heart and tranquility.

1 Objective

- Study PID controller and see the functions of P,I and D term separately in the PID.
- To simulate PID algorithm in Python program and analyse it's functions.
- To simulate PID circuit in multisim and see it's application on a heater like device.

2 Why do we need Control theory?

Control theory deals with the control of dynamic systems in manufactured processes and machines. The goal is to create a control model or algorithm that uses system inputs to optimize the system while minimizing latency, spikes, or idle errors and ensuring a stable level of control. Often with the aim of achieving a certain level of perfection.
A controller is a device that generates an output signal based on the input signal it receives. The input signal is actually an error signal, which is the difference between the measured variable and the desired value as can be shown in feedback control system.



This requires a control with the necessary corrective behavior. This controller controls the regulated process variable (PV) and compares it with a reference or setpoint (SP). The difference between the actual and target value of the process variable, called the error signal or SP-PV error, is used as feedback to generate a control action in order to bring the process variable under control to the same target value. Other aspects to consider are controllability and observability. This is the foundation for the advanced type of automation that has revolutionized the manufacturing, aerospace, communications, and other industries. It is a feedback control in which a sensor is measured and the measured variable is kept in a certain range by a "final control", for example a control valve.

2.1 Requirements of a good Control System

The essential requirements of a good Control System can be listed as follows:

1. Accuracy: Accuracy must be very high as error arising should be corrected. Accuracy can be improved by the use of feedback element.
2. Sensitivity: A good control system senses quick changes in the output due to an environment, parametric changes, internal and external disturbances.
3. Noise: Noise is a unwanted signal and a good control system should be sensitive to these type of disturbances.
4. Stability: The stable systems has bounded input and bounded output. A good control system should response to the undesirable changes in the stability.
5. Bandwidth: To obtain a good frequency response, bandwidth of a system should be large.
6. Speed: A good control system should have high speed that is the output of the system should be fast as possible.
7. Oscillation: For a good control system oscillation in the output should be constant or at least has small oscillation.

2.2 Controller Modes

There are many control modes as follows:

1. ON-OFF controller/two position controller as temperature controller used for domestic heating system.
2. Three-position controller
3. Proportional Action Control
4. Integral/Reset Action Control
5. Derivative/Rate Action Control
6. P+I Control
7. P+D Control
8. P+I+D Control

3 On/Off Control

The heating power is either fully On when the temperature is below setpoint or fully Off when it is above. As a result the temperature oscillates about the setpoint.
The amplitude and time period of the oscillation is a function of the thermal lag between the heating source and the temperature sensor. To prevent the output 'chattering' as the measured temperature crosses the setpoint, the controller does not turn On and Off at precisely the same point. Instead a small differential known as the 'hysteresis' is applied. A typical value is 1°C On/Off control is satisfactory for non-critical heating applications where some oscillation in the temperature is permissible.

4 PID Controller

A proportional–integral–derivative controller (PID controller or three-term controller) is a feedback-based control loop commonly used in industrial control systems and other applications that require continuously modulated control. A PID controller calculates an error value $e(t)$ as the difference between a desired setpoint (SP) and a measured process variable (PV) on a continuous basis and applies a correction using proportional, integral, and derivative terms (denoted P, I, and D respectively). So, before diving into PID, it's important to understand what proportional, integral, and derivative terms mean and how they work.

4.1 P(Proportional Term)

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain constant.

The proportional term is given by

$$P_{\text{out}} = K_{\text{p}} e(t).$$

A high proportional gain results in a large change in the output for a given change in the error. If **the proportional gain is too high, the system can become unstable**. In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the **proportional gain is too low, the control action may be too small** when responding to system disturbances. Tuning theory and industrial practice indicate that the proportional term should contribute the bulk of the output change

4.2 I(Integral Term)

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output.

The integral term is given by

$$I_{\mathrm{out}} = K_{\mathrm{i}} \int_0^t e(\tau)\, d\tau.$$

The integral term accelerates the movement of the process towards setpoint and eliminates the residual steady-state error that occurs with a pure proportional controller. However, since the **integral term responds to accumulated errors from the past, it can cause the present value to overshoot the setpoint value**.

4.3 D(Derivative term)

The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, Kd.

The derivative term is given by

$$D_{\mathrm{out}} = K_{\mathrm{d}} \frac{de(t)}{dt}.$$

Derivative action predicts system behavior and thus improves settling time and stability of the system.An ideal derivative is not causal, so that implementations of PID controllers include an additional low-pass filtering for the derivative term to limit the high-frequency gain and noise.

4.4 PI (Proportional + integral controllers)

In control engineering, a PI Controller (proportional-integral controller) is a feedback controller which drives the plant to be controlled by a weighted sum of the error (difference between the output and desired set-point) and the integral of that value. It is a special case of the PID controller in which the derivative (D) part of the error is not used. The PI controller is mathematically denoted as:

$$\mathrm{G_c} = \mathrm{K_p} + \frac{\mathrm{Ki}}{\mathrm{s}}$$

or

$$\mathrm{G_c} = \mathrm{K_p}\left(1 + \frac{1}{\mathrm{sT_i}}\right)$$

Integral control action added to the proportional controller converts the original system into high order. Hence the control system may become unstable for a large value of $\mathrm{K_p}$ since roots of the characteristic eqn. may have positive real part. In this control, proportional control action tends to stabilize the system, while the integral control action tends to eliminate or reduce steady-state error in response to various inputs. As the value of T_i is increased,

- Overshoot tends to be smaller
- Speed of the response tends to be slower.

4.5 PD (Proportional + derivative controllers)

Proportional-Derivative or PD control combines proportional control and derivative control in parallel. Derivative action acts on the derivative or rate of change of the control error. This provides a fast response, as opposed to the integral action, but cannot accommodate constant errors (i.e. the derivative of a constant, nonzero error is 0). Derivatives have a phase of +90 degrees leading to an anticipatory or predictive response. However, derivative control will produce large control signals in response to high frequency control errors such as set point changes (step command) and measurement noise [5].

In order to use derivative control the transfer functions must be proper. This often requires a pole to be added to the controller.

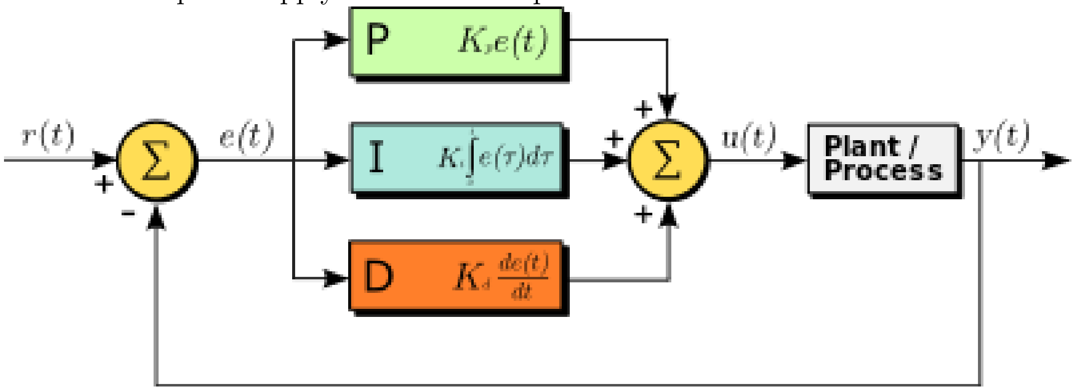
$$\begin{aligned}\mathrm{G_{pd}(s)} &= \mathrm{K_p} + \mathrm{K_d S} \text{ or}\\ &= \mathrm{K_p}\left(1 + \mathrm{T_d s}\right)\end{aligned}$$

With the increase of $\mathrm{T_d}$

- Overshoot tends to be smaller
- Slower rise time but similar settling time.

4.6 PID

The distinguishing feature of the PID controller is the ability to use the three control terms of proportional, integral and derivative influence on the controller output to apply accurate and optimal control.



In the block diagram above we can see the principles of how these terms are generated and applied. It shows a PID controller, which continuously calculates an error value $e(t)$ as the difference between a desired setpoint $\mathrm{SP} = r(t)$ and a measured process variable $\mathrm{PV} = y(t) : e(t) = r(t) - y(t)$ and applies a correction based on proportional, integral, and derivative terms. The controller attempts to minimize the error over time by adjustment of a control variable$u(t)$, such as the opening of a control valve, to a new value determined by a weighted sum of the control terms.

4.6.1 Mathematical Form of PID

The overall control function

$$u(t) = K_{\mathbf{p}} e(t) + K_{\mathbf{i}} \int_0^t e(\tau)\, \mathrm{d}\tau + K_{\mathbf{d}} \frac{\mathrm{d}e(t)}{\mathrm{d}t},$$

where K_p, K_i, and K_d, all non-negative, denote the coefficients for the proportional, integral, and derivative terms respectively (sometimes denoted P, I, and D).

In the standard form of the equation, K_i and K_d are respectively replaced by $K_\mathrm{p}/T_\mathrm{i}$ and $K_\mathrm{p}T_\mathrm{d}$; the advantage of this being that T_i and T_d have some understandable physical meaning, as they represent the integration time and the derivative time respectively.

$$u(t) = K_\mathrm{p} \left(e(t) + \frac{1}{T_\mathrm{i}} \int_0^t e(\tau)\, \mathrm{d}\tau + T_\mathrm{d} \frac{\mathrm{d}e(t)}{\mathrm{d}t} \right),$$

5 Observation

5.1 PID in Python

5.1.1 Seeing PID Algorithm in Python program.

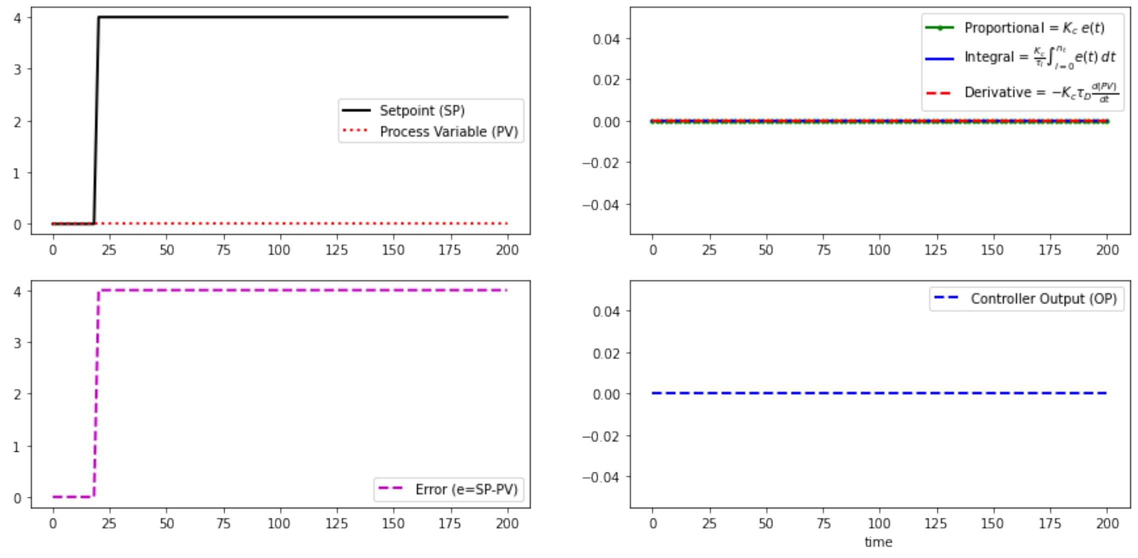
This algorithm I wrote to simulate PID in python and I have posted the results of simulation in coming sections.

```python
# Algorithm of PID
def pidPlot(Kc,tauI,tauD):
    t = np.linspace(0,tf,n) # create time vector
    P= np.zeros(n)          # initialize proportional term
    I = np.zeros(n)         # initialize integral term
    D = np.zeros(n)         # initialize derivative term
    e = np.zeros(n)         # initialize error
    OP = np.zeros(n)        # initialize controller output
    PV = np.zeros(n)        # initialize process variable




#   Setting a setfunction to follow
    SP = np.zeros(n)        # initialize setpoint
    SP_step = int(SP_start/(tf/(n-1))+1) # setpoint start
    SP[0:SP_step] = 0.0     # define setpoint
    SP[SP_step:n] = 4    # step up
    y0 = 0.0                # initial condition



#   loop through all time steps
    for i in range(1,n):
        # simulate process for one time step
        ts = [t[i-1],t[i]]         # time interval

#       Calling the process to calculate
        y = odeint(process,y0,ts,args=(OP[i-1],))  # compute next step
        y0 = y[1]                  # record new initial condition
#       calculate new OP with PID
        PV[i] = y[1]               # record PV
        e[i] = SP[i] - PV[i]       # calculate error = SP - PV
        dt = t[i] - t[i-1]         # calculate time step
        P[i] = Kc * e[i]           # calculate proportional term
        I[i] = I[i-1] + (Kc/tauI) * e[i] * dt  # calculate integral term
        D[i] = -Kc * tauD * (PV[i]-PV[i-1])/dt # calculate derivative term
        OP[i] = P[i] + I[i] + D[i] # calculate new controller output

#   Plot PID response
    plt.figure()
```

5.1.2 PID Python Simulation Plots

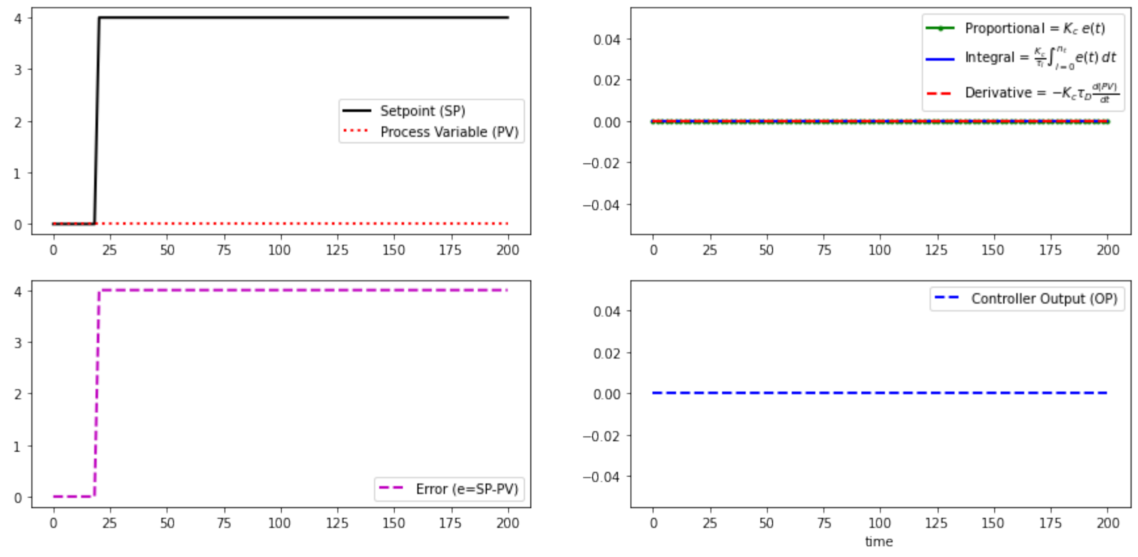
Note
Ki=Kp/I
Kd=Kp*D

- Kp=0, Ki=0, Kd=0

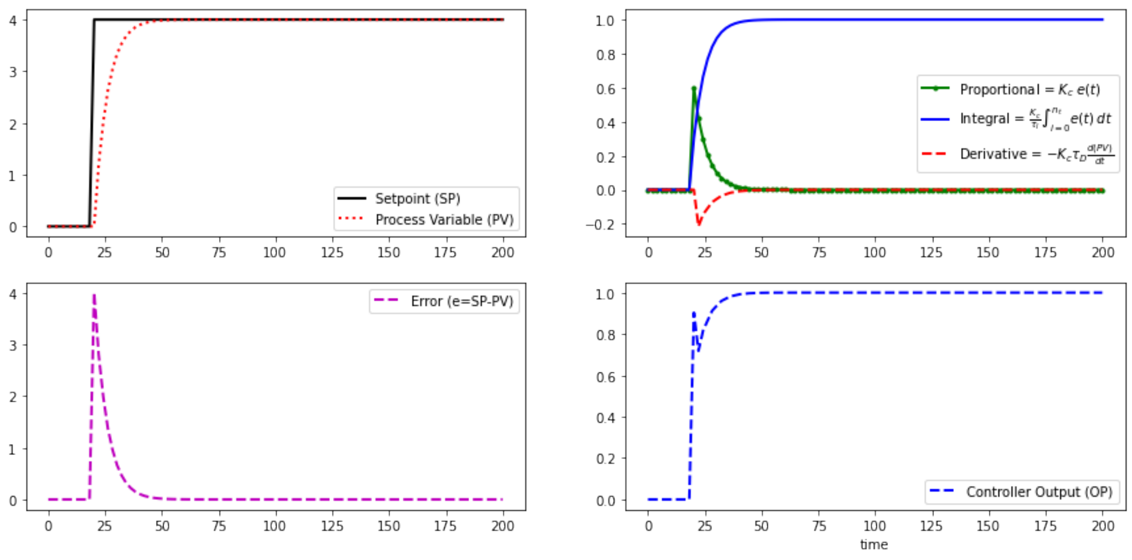


- I=4, D=2

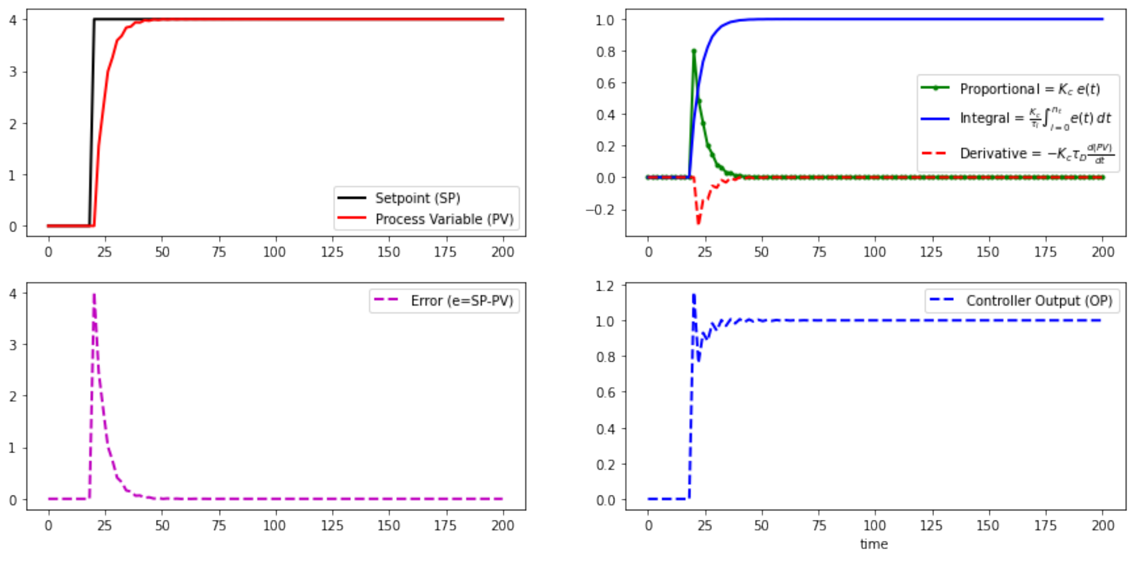
1. Kp=0



2. Kp=0.15

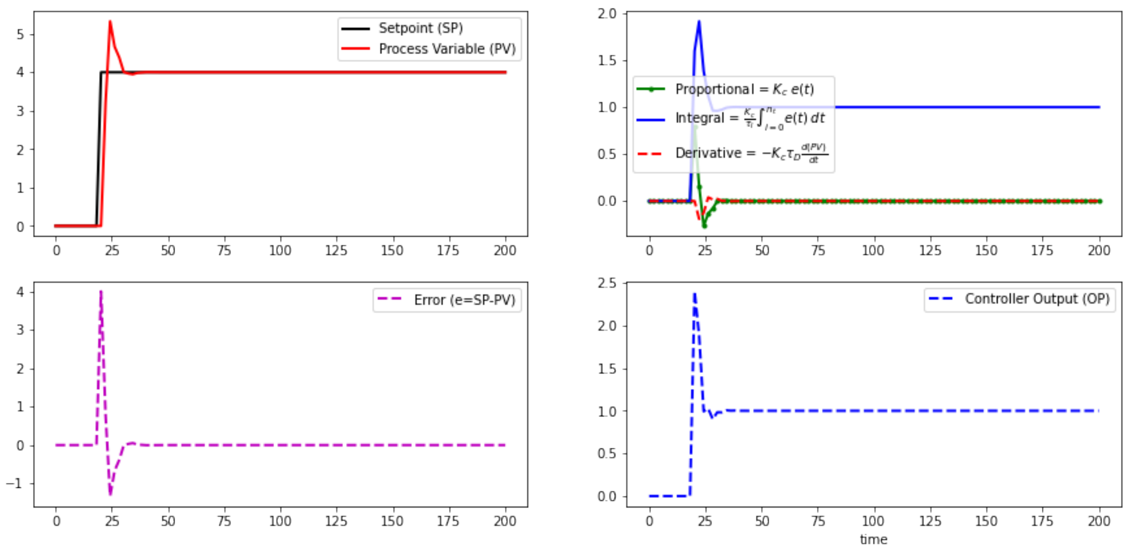


3. Kp=0.2

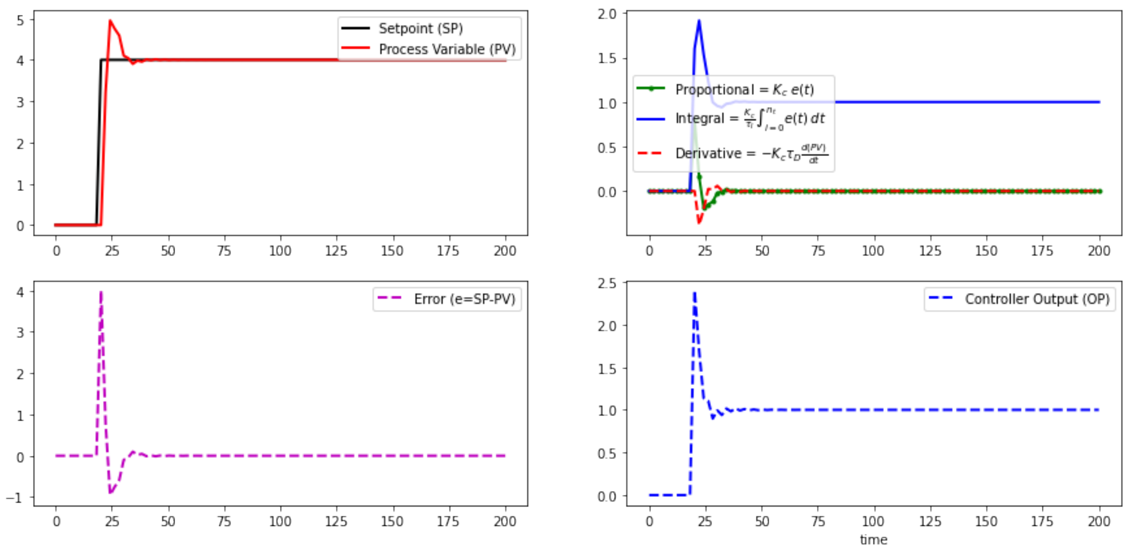


- Kp=0.2, I=1.01

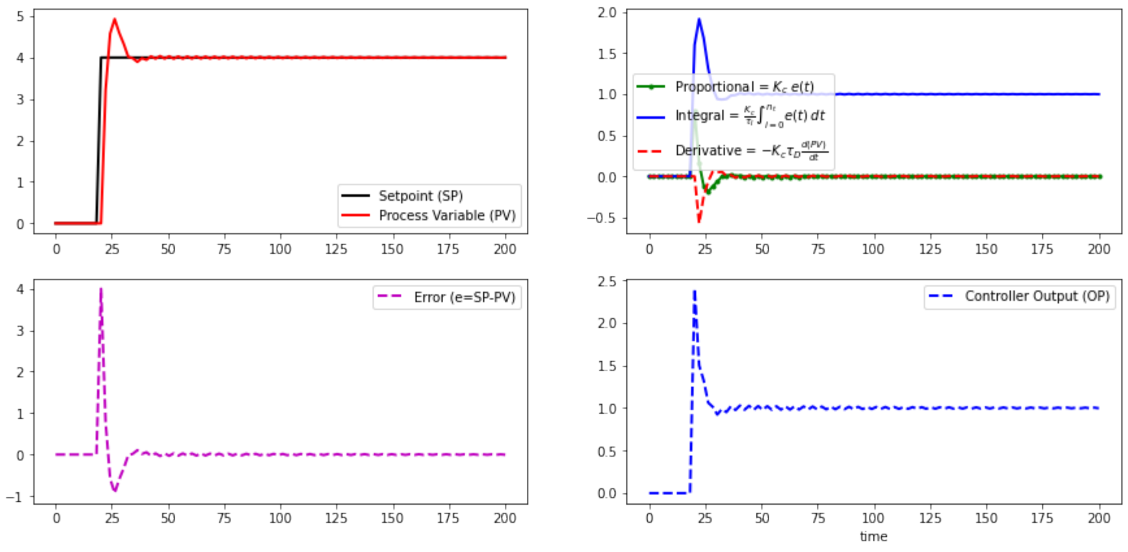
1. D =0.60



2. D =1.20

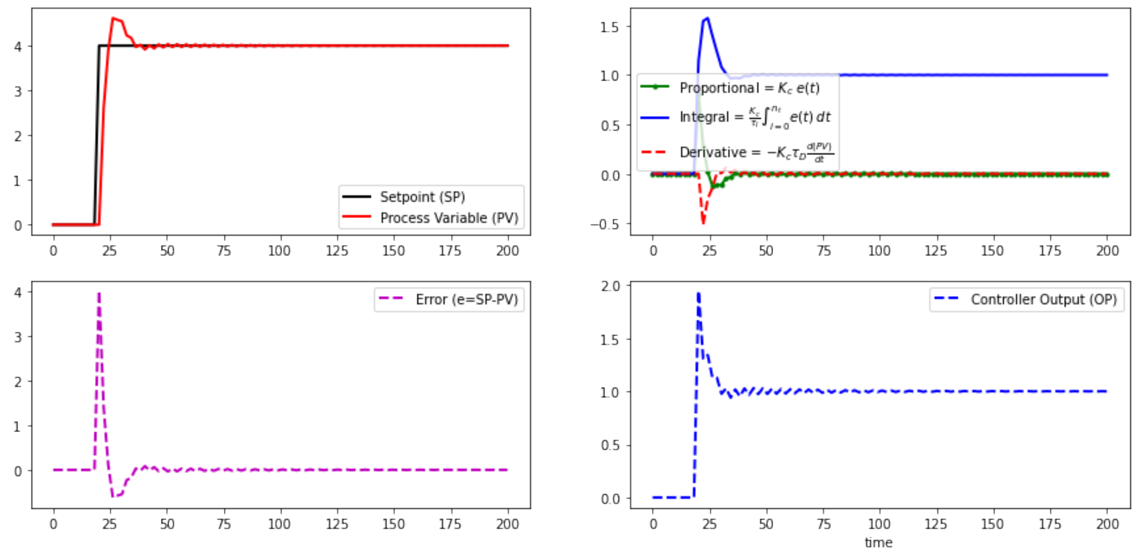


3. D =1.80

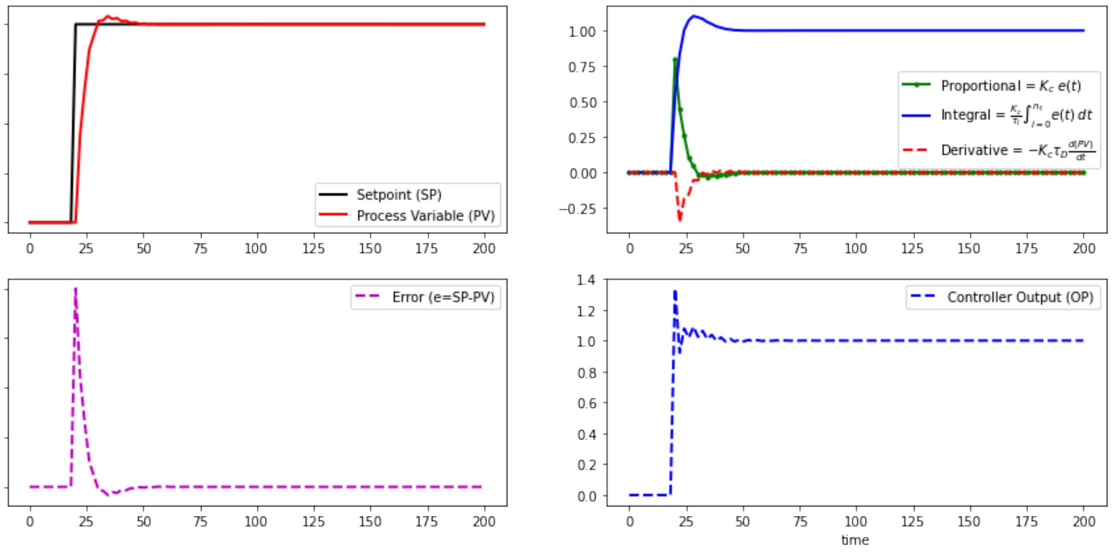


- Kp=0.2, D=2

1. I =1.41

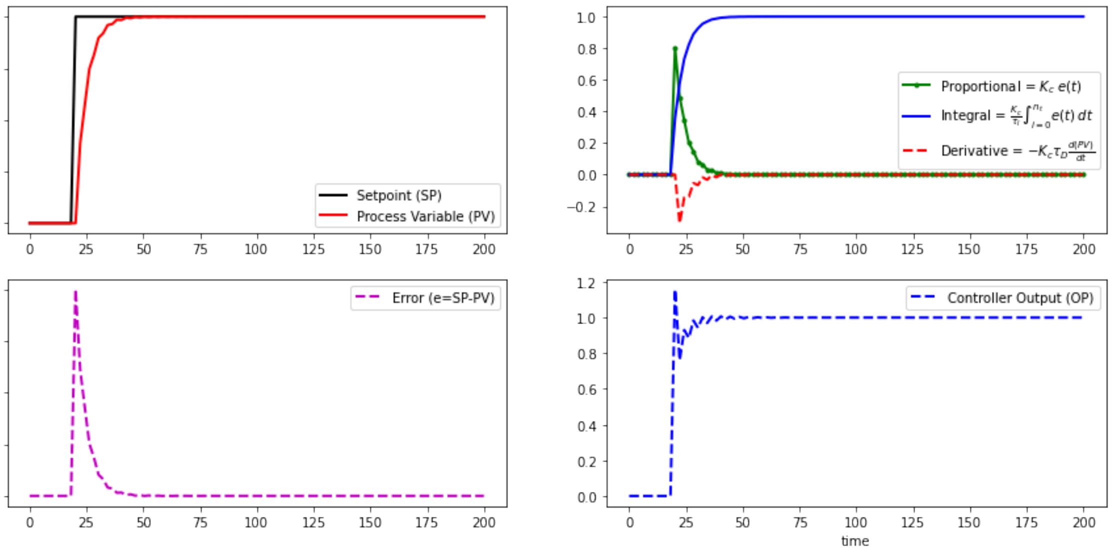


2. I =3.10



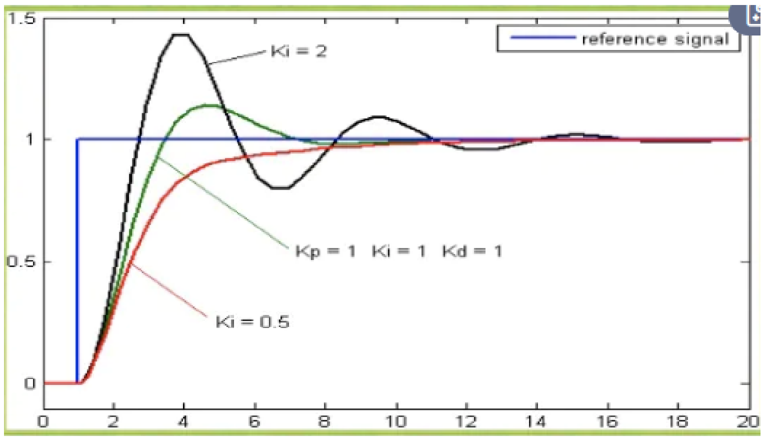
Setpoint (SP)
Process Variable (PV)
Proportional = $K_c\, e(t)$
Integral = $\frac{K_c}{\tau_I}\int_{i=0}^{n_t} e(t)\, dt$
Derivative = $-K_c \tau_D \frac{d(PV)}{dt}$
Error (e=SP-PV)
Controller Output (OP)
time

3. I =4.15



Setpoint (SP)
Process Variable (PV)
Proportional = $K_c\, e(t)$
Integral = $\frac{K_c}{\tau_I}\int_{i=0}^{n_t} e(t)\, dt$
Derivative = $-K_c \tau_D \frac{d(PV)}{dt}$
Error (e=SP-PV)
Controller Output (OP)
time

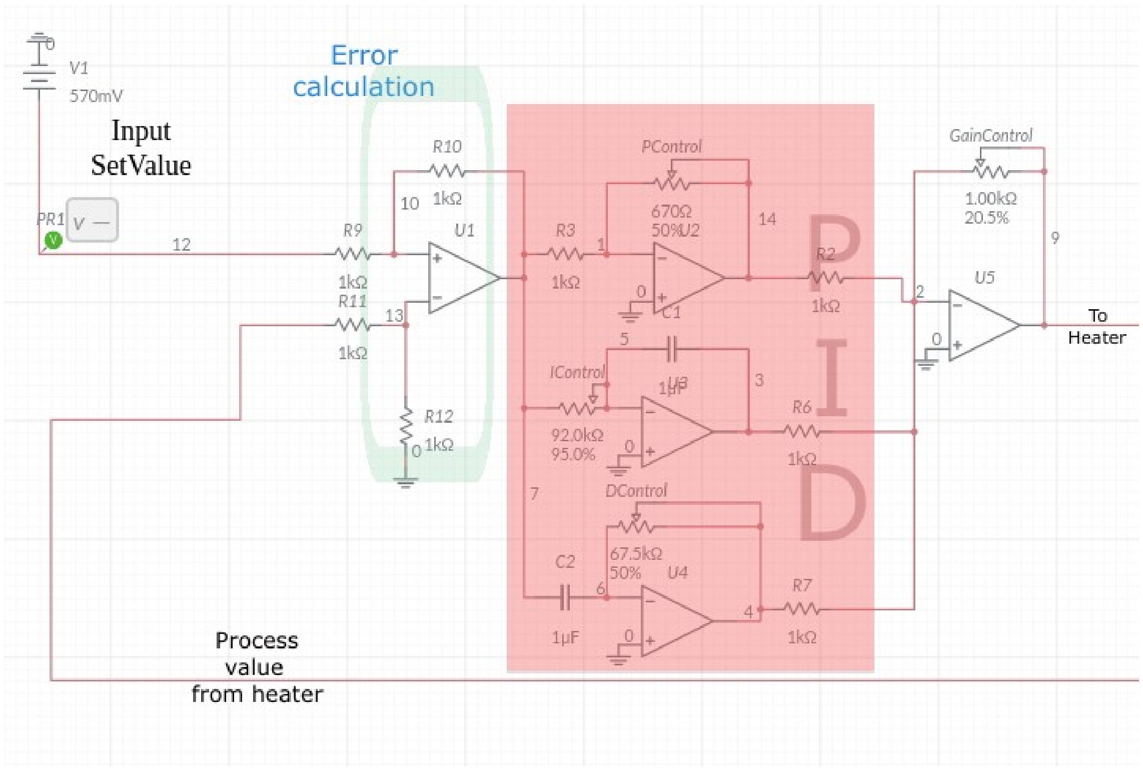
****
The Image Below is not mine plot it is from Internet.



All the thing in this graph can be seen very clear in the graph I have plotted using Different P, I and D value.
****

5.2 PID Circuit

This circuit I draw to simulate PID in multi-sim and I have posted the results of simulation in coming sections.



But there was a problem while doing PID in Multisim, as there was no any feature of using Heater directly. So I had to make something by circuit which can pretend to be heater.

5.2.1 Making something similar to Heater

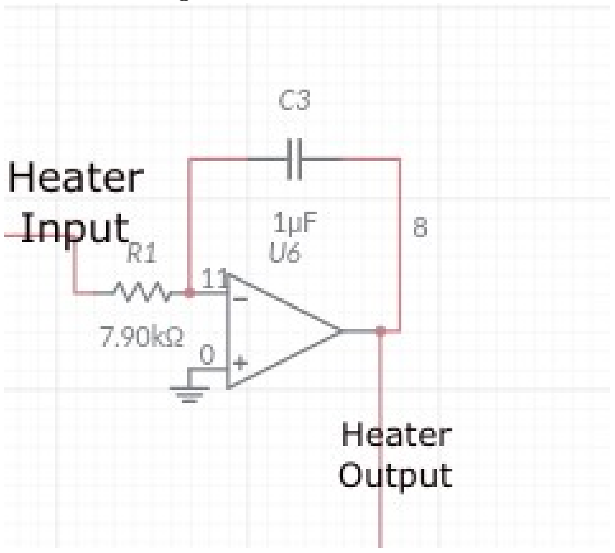
Basically the actual benefit of using a PID is that it gives control for Let say TEMP if we have direct control of POWER (i.e. heat). Means we have to control Integral Output if we have control over Integrant, which is similar to saying

$$\text{Integral Output } \propto \int \text{Integrant dt}$$

i.e.

$$\text{TEMP } \propto \int \text{POWER dt}$$

So I used an Integrator circuit to have this use.

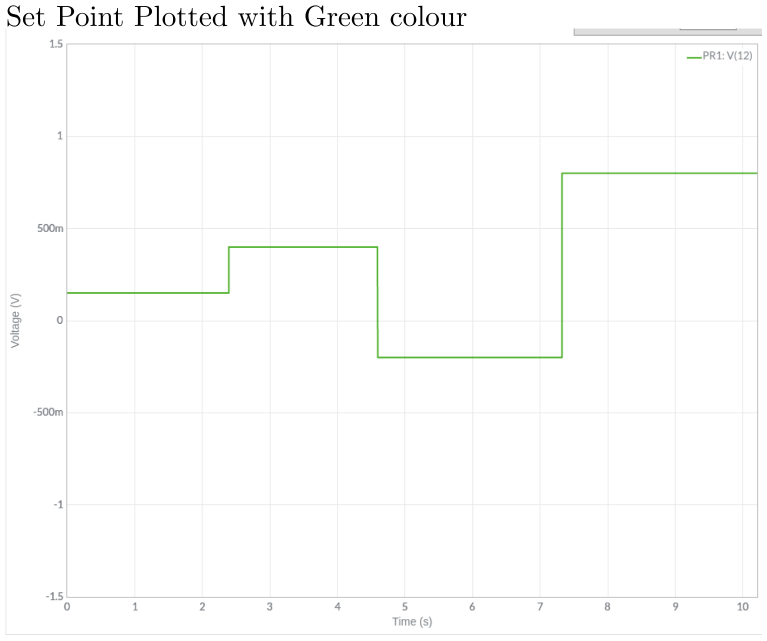


We can have some process having many more similar Integrator or combinations as well. Overall the device we formulated above is a basic or first step when we can feel the need of control system like PID.

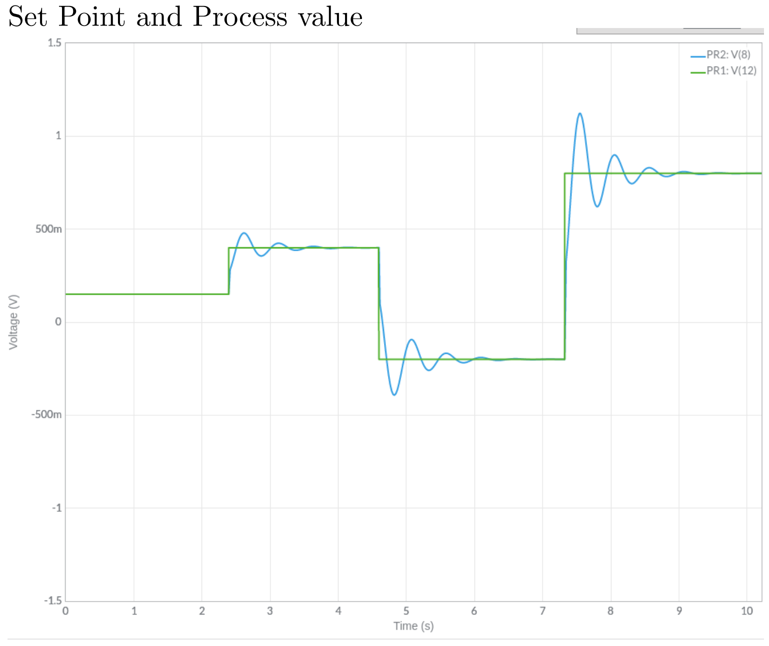
5.2.2 PID Multisim Circuit based Simulation Plots

1. **Fixed P,I and D Value Changing Setvalue**

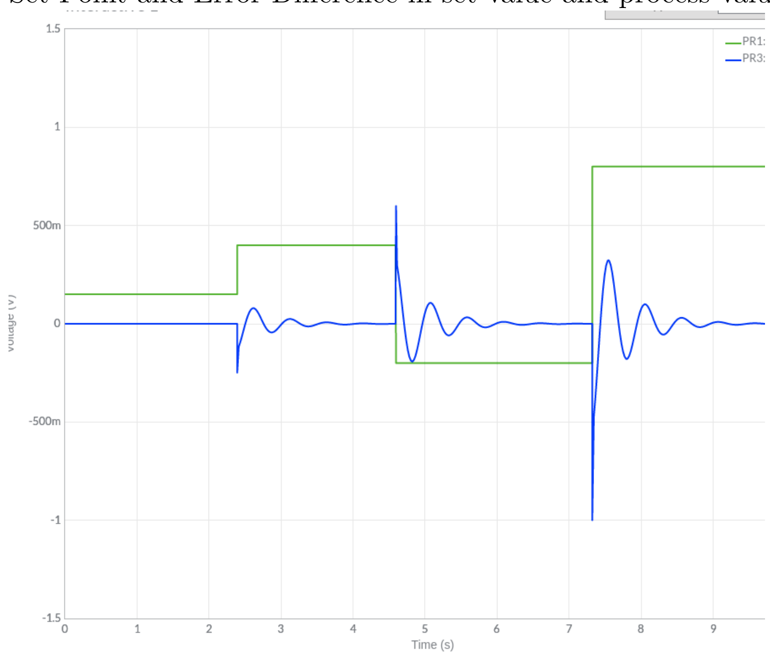
   - Set Point Plotted with Green colour



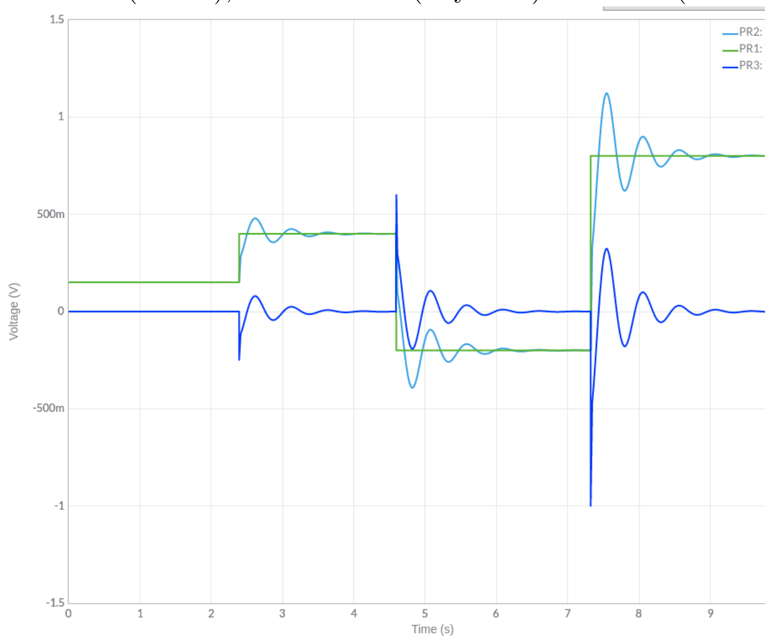
   - Set Point and Process value



- Set Point and Error Difference in set value and process value

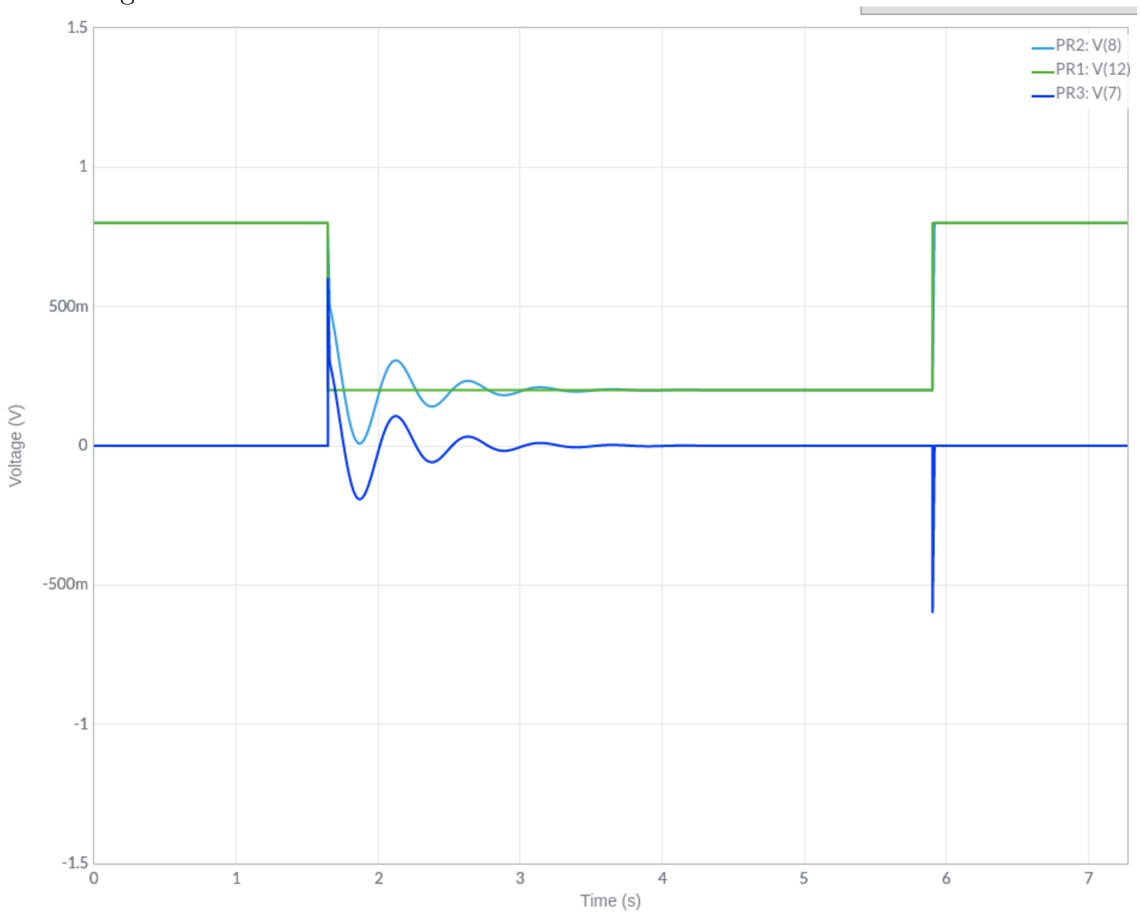


- Set Point(Green), Process value(Sky blue) and Error(Dark Blue)



2. **Fixed I and D Value Changing Setvalue and P**

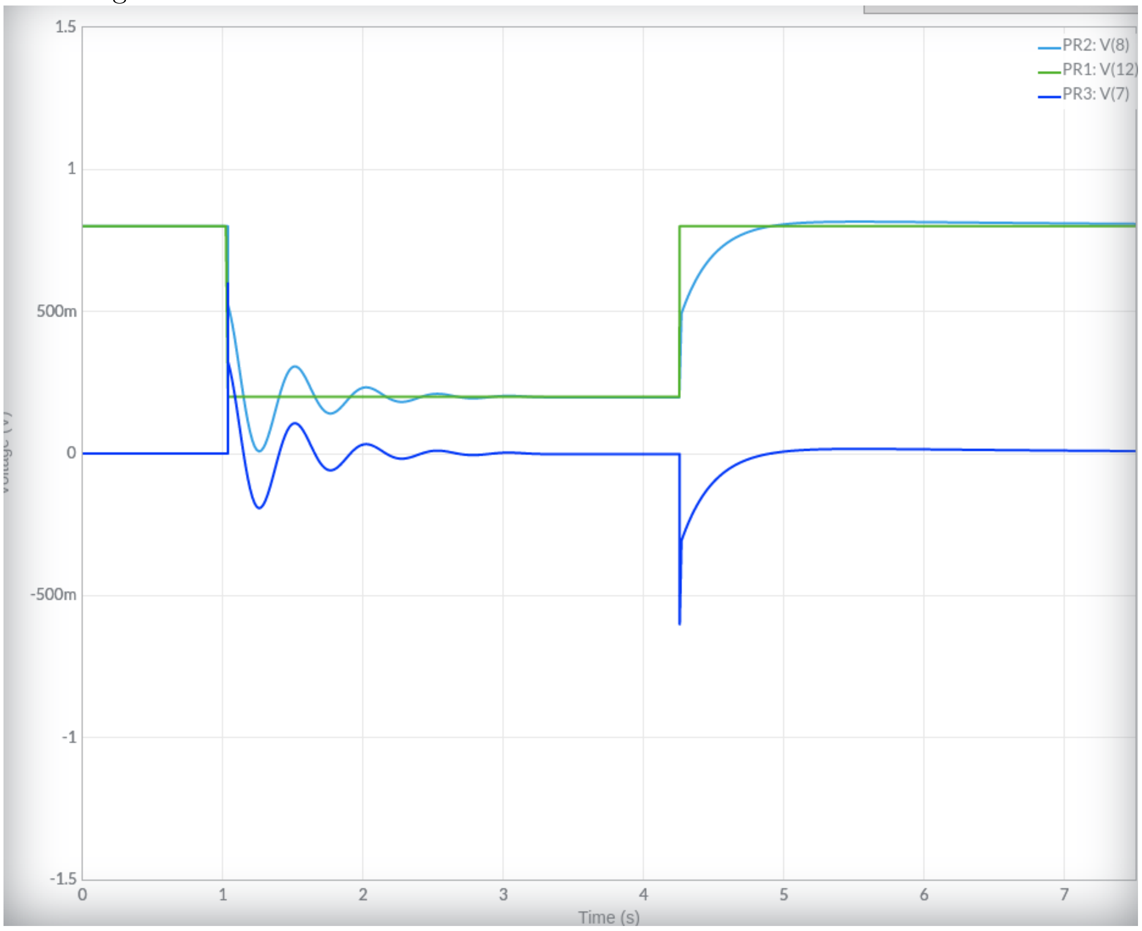
   Here P during second change is 100 times of initial P
   P is changed after 5 Seconds



   Initially it took large time to reach the set point but after change in P it reached Setpoint in no time.

3. **Fixed P and D Value Changing Setvalue and I**

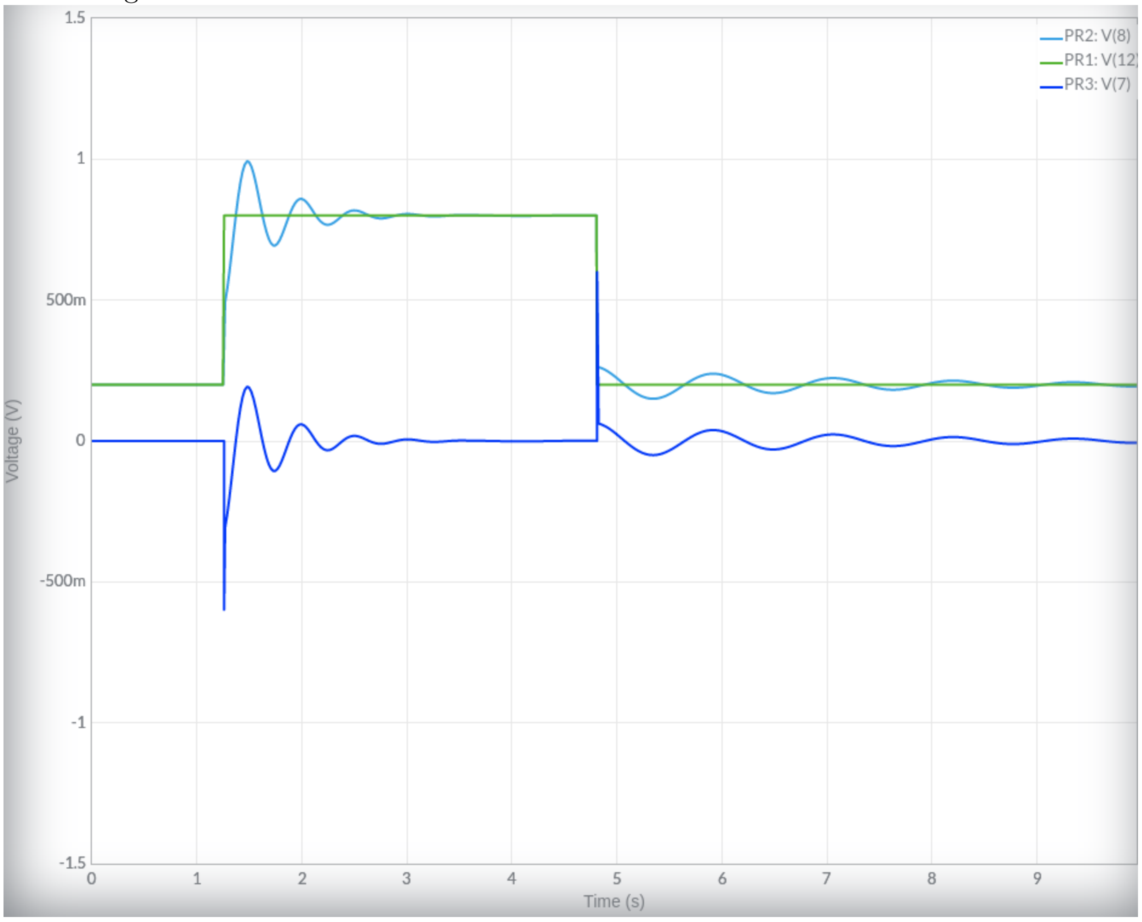
   Here I during second change is 100 times of initial I
   I is changed after 4 Seconds



   Initially it had a large overshoot but after change in I it just approached the set value.

4. **Fixed P and I Value Changing Setvalue and D**

   Here D during second change is 100 times of initial D
   D is changed after 4 Seconds



   Initially it had a large overshoot but after change in D the overshoot is decreased slightly but the major change is seen in time taken to reach the set point.

6 The tuning parameters

Correction to be made. The magnitude of the correction (change in controller output) is determined by the proportional mode of the controller.
The correction to be applied. The duration of the adjustment to the controller output is determined by the integral mode of the controller.
The correction to be applied. The speed at which a correction is made is determined by the derivative mode of the controller.

7 Analysis

- P depends on the present error.I on the accumulation of past errors.D is a prediction of future errors, based on current rate of change.

- A proportional controller (Kp) will have the effect of reducing the rise time and will reduce but never eliminate the steady state error.

- An integral control (Ki) will have the effect of eliminating the steadystate error for a constant or step input, but it may make the transient response slower.

- A derivative control (Kd) will have the effect of increasing the stability of the system, reducing the overshoot, and improving the transient response.

- In fact, changing one of these variables can change the effect of the other two.
- With the PID controller we can set the P + I + D values so that we will not have any Over or undershoot and reach set point directly.

- PID controller has all the necessary dynamics: fast reaction on change of the controller input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action inside control error area to eliminate oscillations (P mode).

- **This combination of{Present + Past + Future} makes it possible to control the application very well.**

8 Conclusions

- Proportional action gives an output signal proportional to the size of the error. Increasing the proportional feedback gain reduces steady-state errors, but high gains almost always destabilize the system.
- Integral action gives a signal which magnitude depends on the time the error has been there. Integral control provides robust reduction in steady-state errors, but often makes the system less stable.
- Derivative action gives a signal proportional to the change in the Error. It gives sort of ”anticipatory” control .Derivative control usually increases damping and improves stability, but has almost no effect on the steady state error.
- These three kinds of control combined from the classical PID controller.
- PID can be implemented in Hardware and software.
- The PI controller can be considered as Lag compensator, The PD controller can be considered as lead compensator and PID same as Lag-Dead compensator works to improve transient and steady state region.
- The tuning of the controller is one of the limitations of PID controller.
- Proportional and integral control modes are essential for most control loops, while derivative is useful only in some cases.
- Designing and tuning a PID controller appears to be conceptually intuitive, but can be hard in practice, if multiple (and often conflicting) objectives such as short transient and high stability are to be achieved.
- Control engineers usually prefer P-I controllers to control first order plants. On the other hand, P-I-D control is vastly used to control two or higher order plants.
- The major reasons behind the popularity of P-I-D controller are its simplicity in structure and the applicability to variety of processes. Moreover the controller can be tuned for a process, even without detailed mathematical model of the process.
- The choice of P-D, P-I or P-I-D structure de pends on the type of the process we intend to control.
- There are few more issues those need to be addressed while using P-I controller. The most important among them is the anti-windup control.

9 Codes uploaded on GitHub

Link of all my codes in (https://github.com/1raviprakash/OPENLAB$_{P}ROJECT$)

10 References

- Process Control and Optimization volume 2 by Taylor

- https://www.multisim.com/content/joAbzNDZrgyrkHnbeQEYtP/pid/open/

- https://en.wikipedia.org/wiki/PID$_{c}ontrollerhttps : //apmonitor.com/pdc/index.php/Main/F$

- https://github.com/PID