Introduction:

PID Control stands for Proportional-Integral-Derivative control. It is also called a three –term control because of the involvement of the three blocks. PID control is a feedback control mechanism that calculates and controls three parameters – the proportional, integral and derivative of how much a process variable (PV) deviates from the user defined set point (SP) value. Complex electronic systems are equipped with PID control systems and industrial applications of this control system can be widely observed in PLC (Programmable Logic Controllers). The main objective of implementing a PID control system is to minimize the extent of steady state error and providing overall stability to the system that cannot be achieved as conveniently and efficiently through employing PI or PD systems alone.

Working Principle:

PID control system contains three calculation processes – proportionality, integration and differentiation. Hence the actuating signal for the control system is –

$$e_{out}(t) = K_p\, e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(\tau)\, d\tau$$

Here,

$e(t)$ = Error function

K_p = Multiplication factor pertinent to proportionality

K_d = Multiplication factor pertinent to differentiation

K_i = Multiplication factor pertinent to integration

The error signal is the difference between the reference input signal or the set point, (SP) and the feedback signal.

Proportional Control: The actuating signal for this block is –

$$P(t) = K_p\, e(t)$$

The proportional block creates an output signal proportional to the magnitude of the error signal. Initially, the value of the error is maximum and as it gets multiplied with K_p, the margin of error starts to decrease and the rise time decreases too. However, as the output approaches the set point closely, there is a risk of overshoot involved with using the proportional block alone and the error cannot be completely eliminated. Increased overshoot also leads to an increased settling time for the output.

Integral Control: The actuating signal for this block is –

$$I(t) = K_i \int_0^t e(\tau)\, d\tau$$

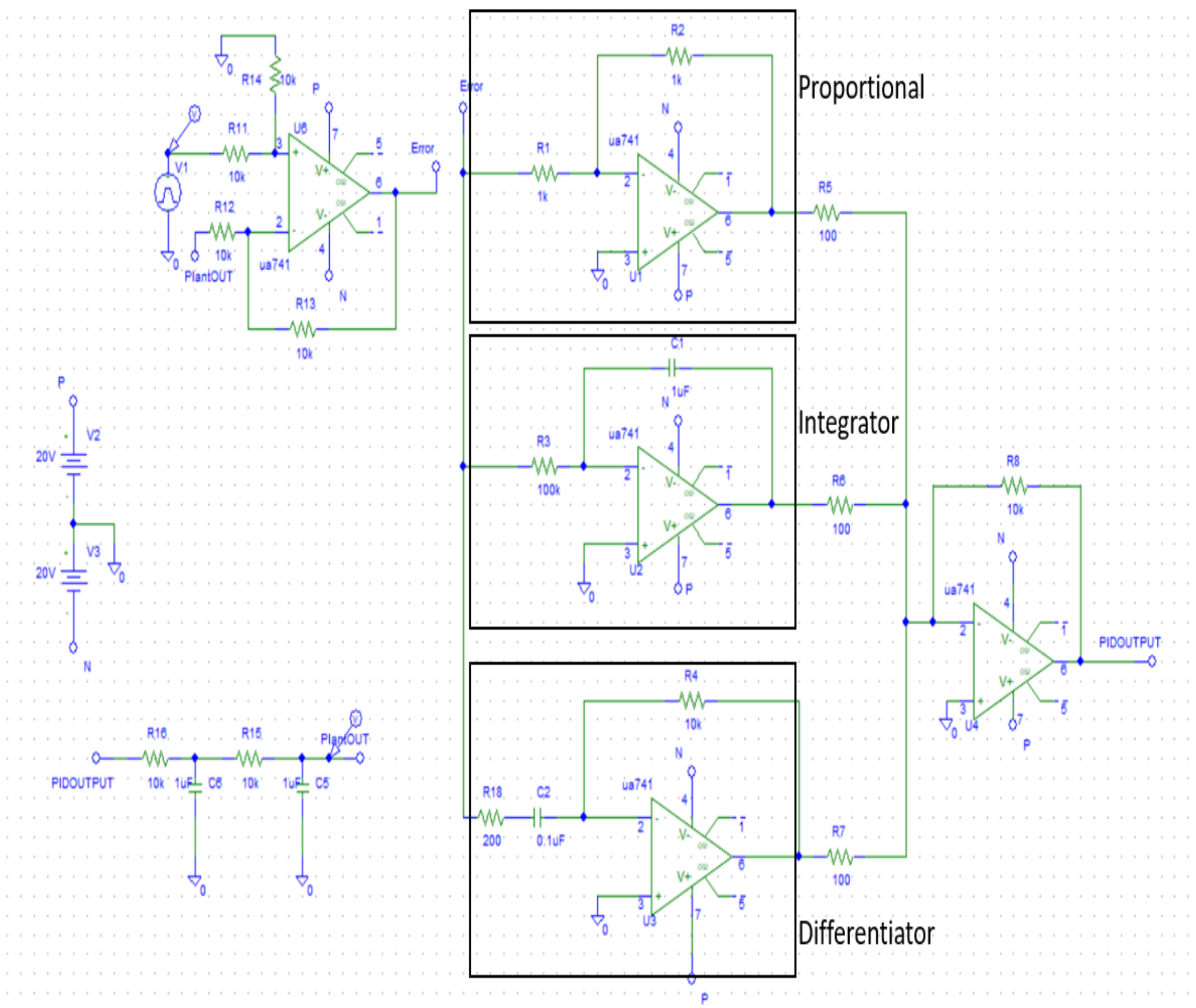
The integral block acts as a memory block storing past outputs. It integrates the error with previously stored value and hence eliminates the steady state error, while decreasing the rise time. However due to continuous accumulation of previous error values, this block too increases overshoot and settling time.

Derivative Control: The actuating signal for this block is –

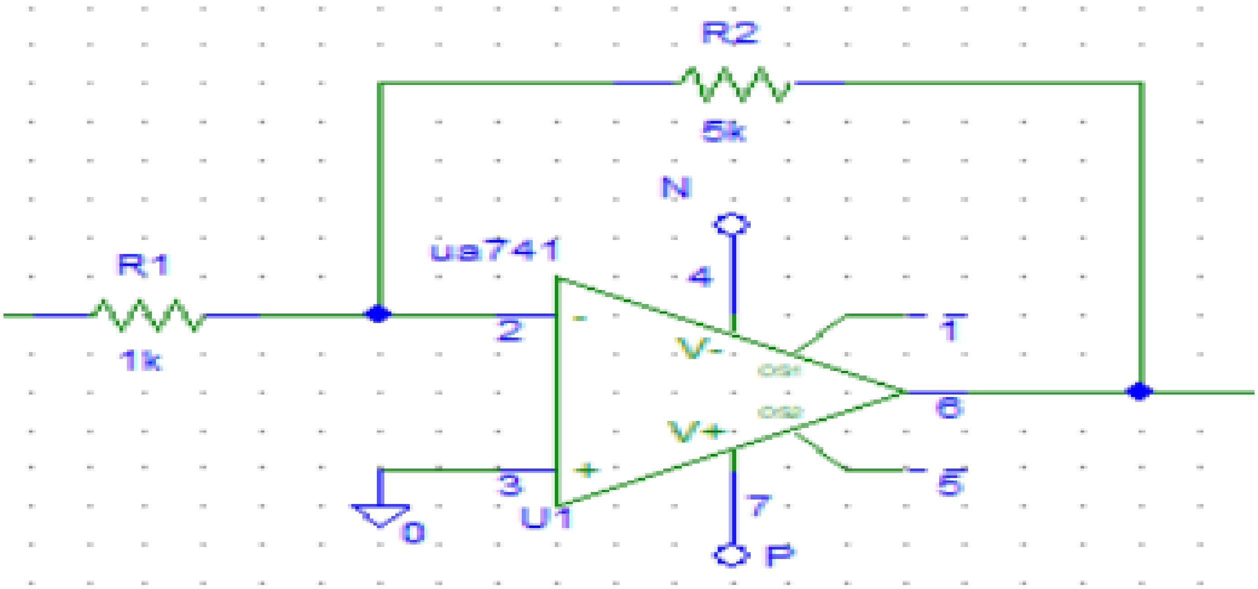
$$D(t) = K_d \frac{de(t)}{dt}$$

The derivative block plays the role of eliminating overshoot error. The derivative block creates an output signal proportional to the rate of the change of the error signal as graphically, slope of the error signal, i.e., the time rate of change can be measured by differentiating it with respect to time. The faster the error changes, the larger the derivative output is. Derivative control looks ahead to see what the error will be in the future and hence decreases the overshoot and settling time.

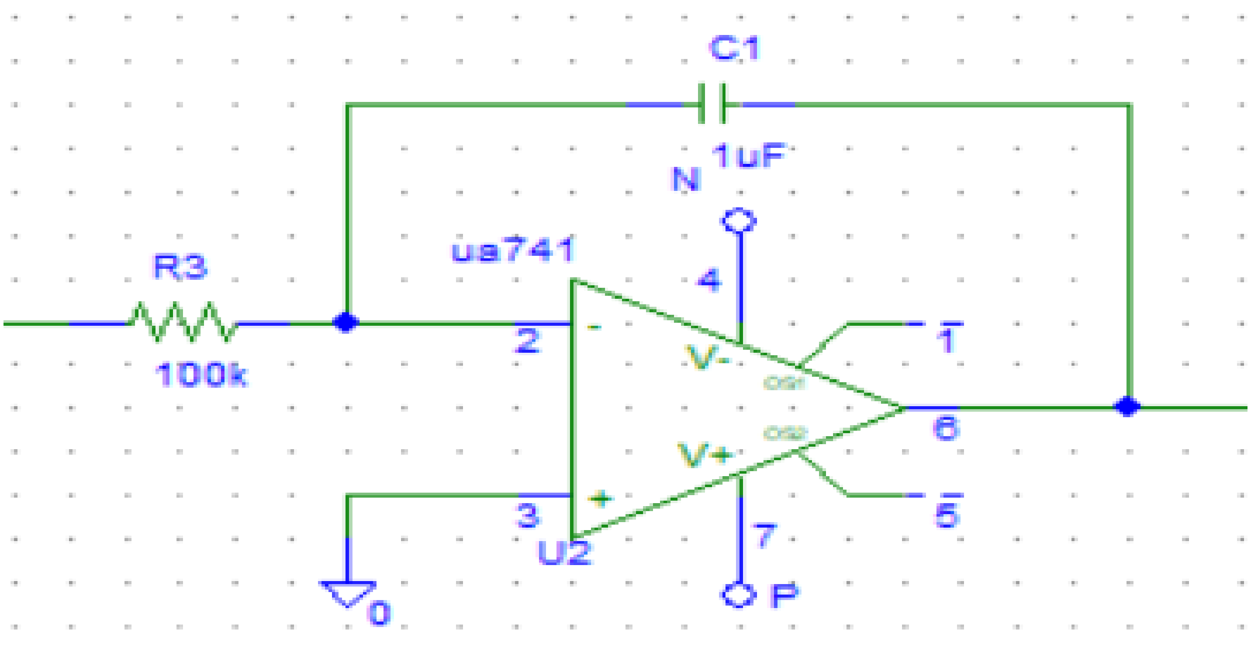
Schematic Diagram of the PID Controller with Plant:



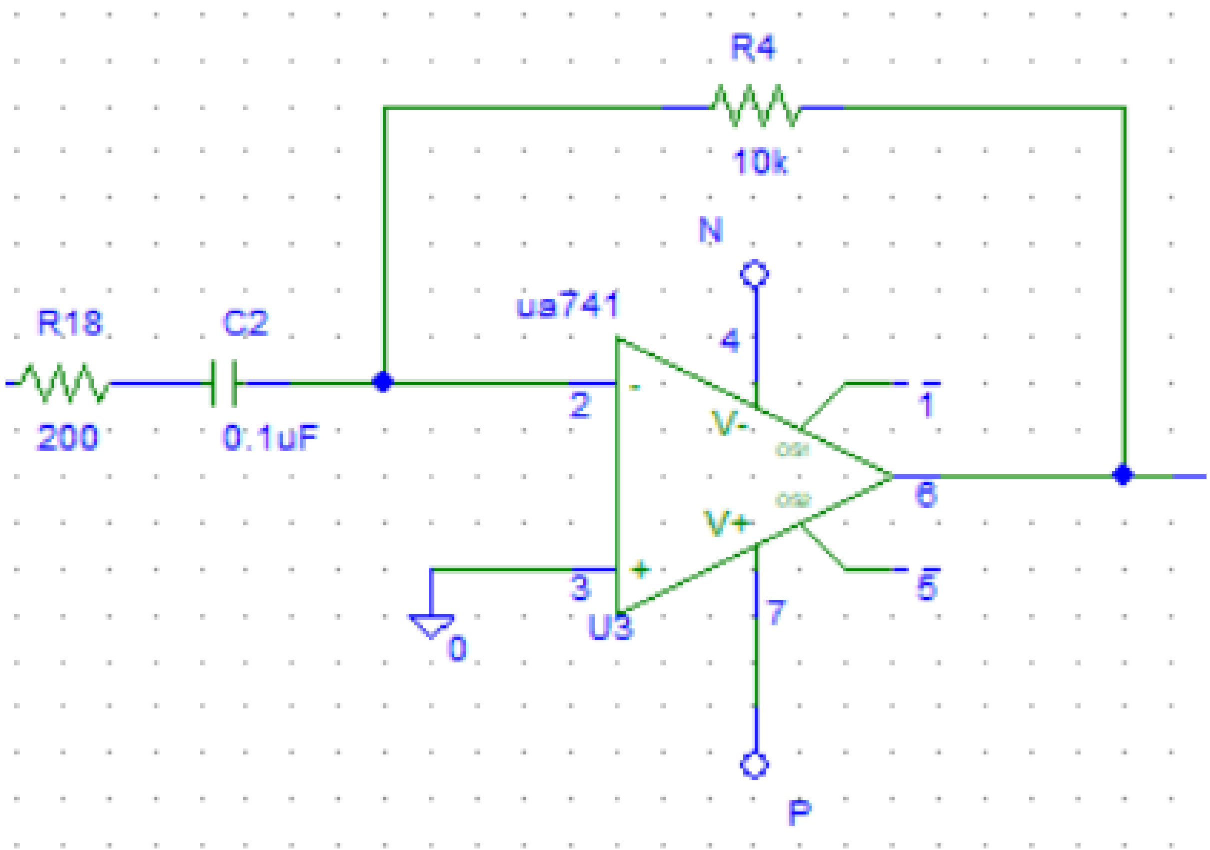
Proportional Block:



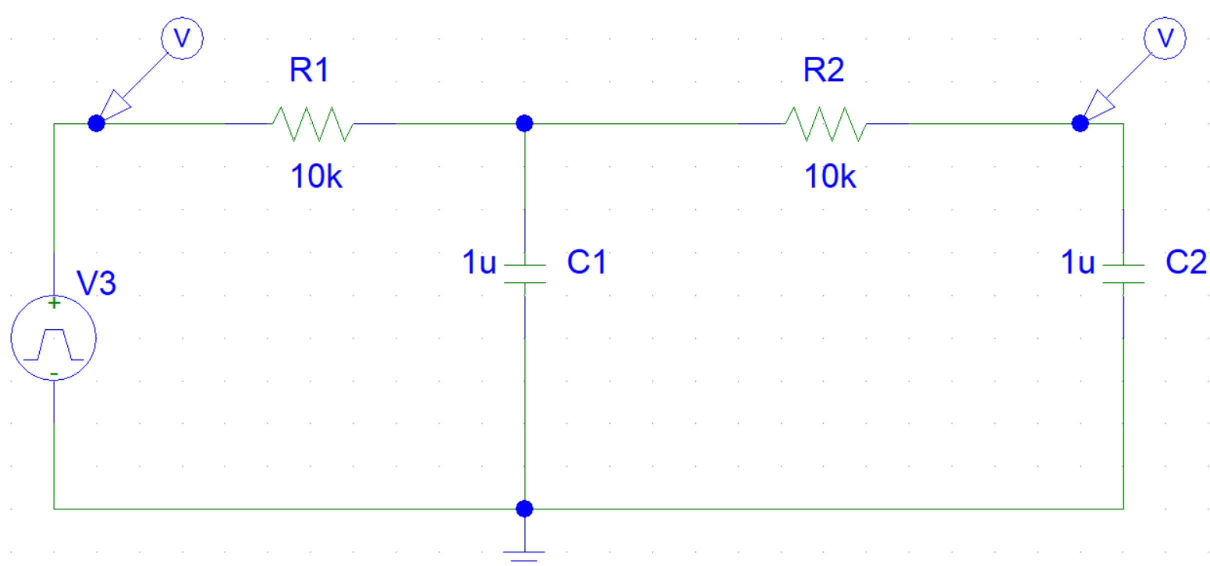
Integrator Block:



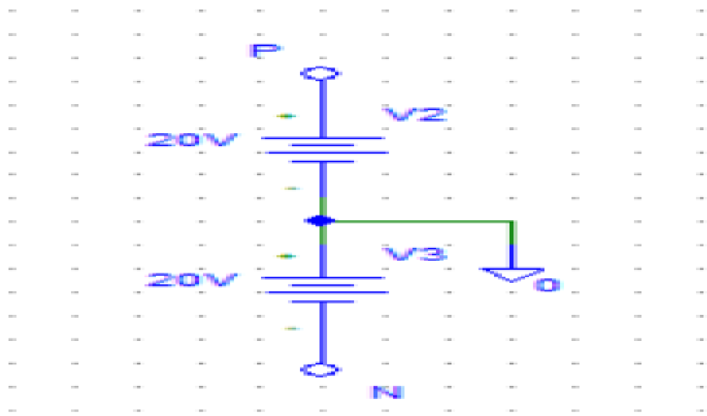
Differential Block:



Plant (Mechanical Analogous Circuit of Motor carrying a load):



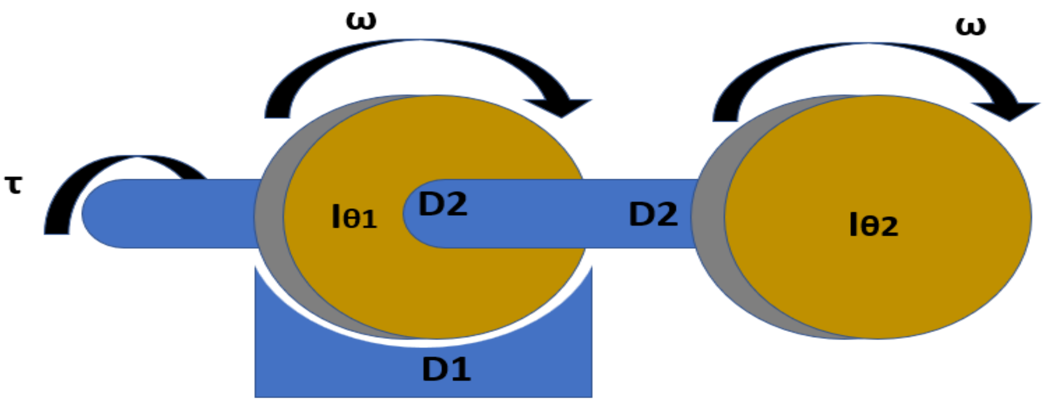
Biasing Configuration:



Experimental Analysis:

For our experiment we have used the mechanical output of a motor as a plant. We know that as we power a motor with power supply its shaft starts to move but it takes some time to reach the expected speed because of the inertia of the motor. Besides there are damping forces also. As a result, due to a change in the voltage, we can see that there is sufficient delay to attain the expected output speed. Here we use an analogous circuit of mechanical output of a motor as our plant to run our experiment to show how a PID controller helps to minimize the instability and overshoot and run the motor at our reference speed.

The Mechanical Model of Plant:

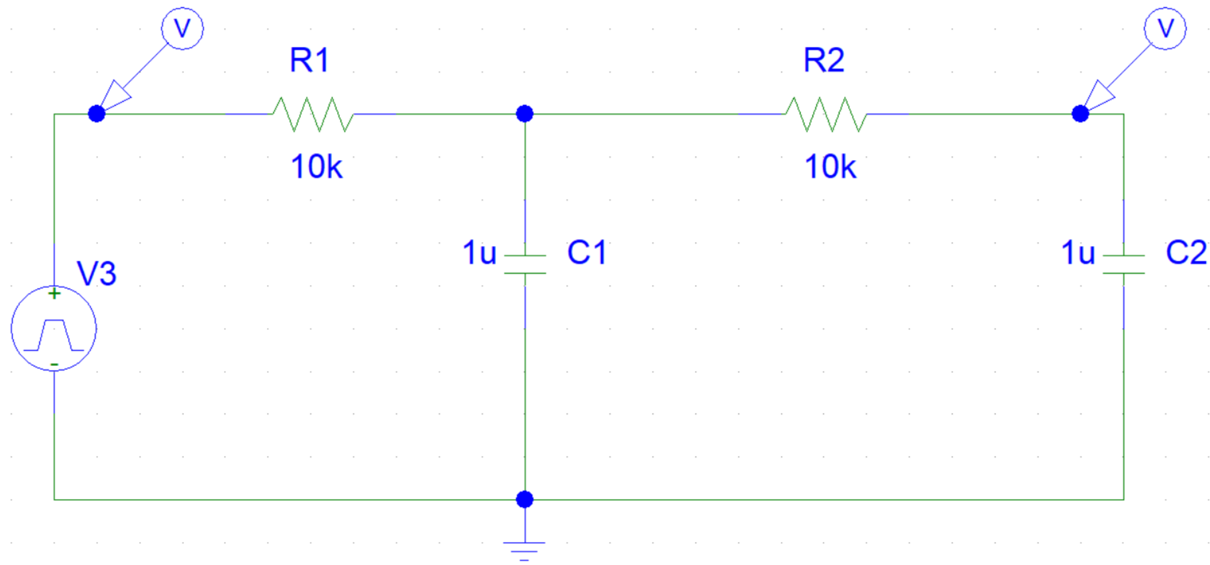


This figure represents the mechanical model that we used as the plant. It is the mechanical analogous system that can be considered as any motor which is operating with a load. Here, $I_{\theta 1}$ represents the inertia of rotational body inside the motor such as shaft, rotor circuit etc. and $I_{\theta 2}$ represents the load. D_1 is the damping force acting inside the motor and D_2 is the damping force between shaft and flywheel connection. The shaft is producing a torque to rotate the motor. With PID we will control the speed of the shaft as the load changes or torque changes. As desired speed, response time and stability are necessary for stable operation.

Mechanical to Electrical Analog Quantity:

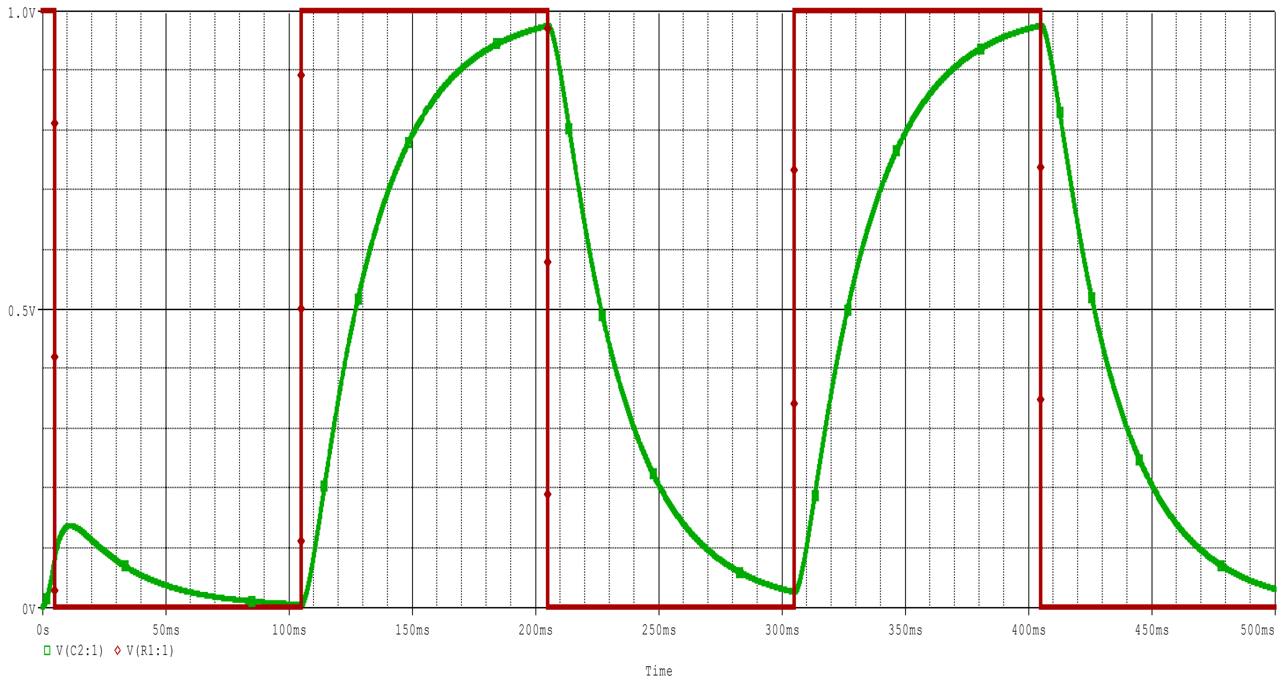
Mechanical	**Electrical**
Torque, τ	Current, I
Angular frequency, ω	Voltage, V
Inertia, I	Capacitance, C
Dampening Force, D	Resistance, R

Electrically Equivalent Circuit of The Plant:



Now at first, we run the motor without any controller plant and observe the angular velocity.

Output without PID Controller:

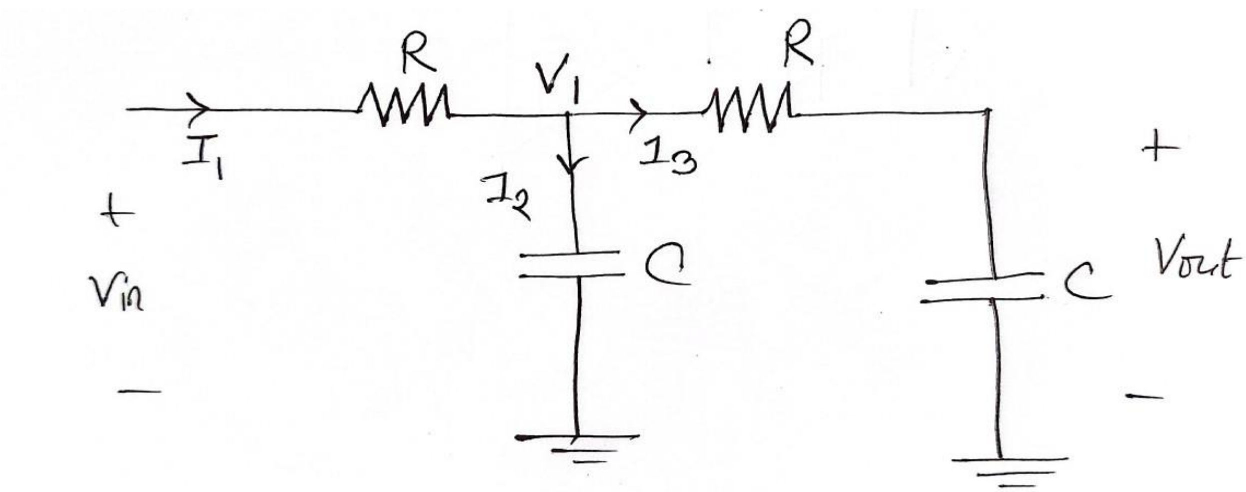


We can observe that because of the inertia, every time when power supply is given to the motor, which produces torque, the shaft has a quite large rise time - almost equals to the full duty cycle of the PWM. Besides when the motor is set to turn off, it again takes a good amount of time to come to rest. This plot indicates the larger react time of the motor and its less consistent behavior on the verge of a sudden change. Both are quite unexpected for our use.

Theoretical Analysis of the plant:

The plant is a plant of second order which contains two poles in its transfer function. The reason behind choosing a simple second order plant is that in real life we mostly deal with different 2nd order system as with the increase of order the system starts to become more unstable. Another reason to choose a second order plant is that in first order plant as there is no chance of getting complex conjugate poles so we cannot observe overshoot. So, we reasonably take the 2nd order plant to show the effect of PID on overshoot and unstable condition.

Transfer Function of the plant:



$$V_1 = I_3 R + V_{out}$$

$$I_3 = \frac{V_1}{R + \frac{1}{sC}}$$

$$I_3 = \frac{V_1(sC)}{1 + RsC}$$

$$\therefore V_1 = V_1 \frac{sCR}{1+RsC} + V_{out}$$

$$\therefore V_1 = \frac{V_{out}}{1 - \frac{sCR}{1+RsC}}$$

$$V_1 = \frac{V_{out}}{\frac{1+RsC-RsC}{1+RsC}}$$

$$\therefore V_1 = V_{out}(1+RsC)$$

$$I_1 = I_2 + I_3$$

$$= \frac{V_1}{\frac{1}{sC}} + \frac{V_1(sC)}{1+RsC}$$

$$= V_1 \left(sC + \frac{sC}{1+RsC}\right)$$

$$= V_1 \left(\frac{sC + Rs^2C^2 + sC}{1+RsC}\right)$$

$$= V_1 \left(\frac{2sC + Rs^2C^2}{1+RsC}\right)$$

$$V_{in} = V_1 + I_1 R$$

$$= V_1 + V_1 \frac{(2sC + Rs^2C^2)}{1+RsC} \times R$$

$$= V_1 \left(\frac{1+RsC + 2RsC + R^2s^2C^2}{1+RsC}\right)$$

$$= V_1 \left(\frac{1 + 3RsC + R^2s^2C^2}{1+RsC}\right)$$

$$= V_{out} \frac{(1+RsC)(1+3RsC+R^2s^2C^2)}{(1+RsC)}$$

$$V_{in} = V_{out}\,(1+3RsC + R^2s^2C^2)$$

$$\therefore \frac{V_{out}}{V_{in}} = \frac{1}{1+3RsC + R^2s^2C^2}$$

$$H(s) = \frac{1}{R^2C^2\left(s^2 + \frac{3s}{RC} + \frac{1}{R^2C^2}\right)}$$

New for $R = 10k$

$C = 1uF$

$$H(s) = \frac{10\times10^3}{(s^2 + 300s + 10000)}$$

for step response $u(t)$

$$Y(s) = \frac{1}{s}\frac{10\times10^3}{(s^2 + 300s + 10000)}$$

$$\therefore Y(s) = \frac{1}{s}\frac{10\times10^3}{(s^2 + \alpha_1 s + \alpha_2)}$$

$$= \frac{1}{s} - \frac{10\times10^3}{(s+\beta_1)(s+\beta_2)}$$

Observed Parameters:

Rise Time: The time needed for the output to rise from 10% to 90% of the desired output is called rise time.

Overshoots: The sudden increase in the amplitude of a signal due the oscillatory nature of a signal is called overshoot.

Steady State Error: An error which becomes constant after the transient phase of a signal is called Steady state error. It's the difference between the output and the desired output at steady state

Settling time: The time needed for an oscillation to die out completely (up to 90-95%)

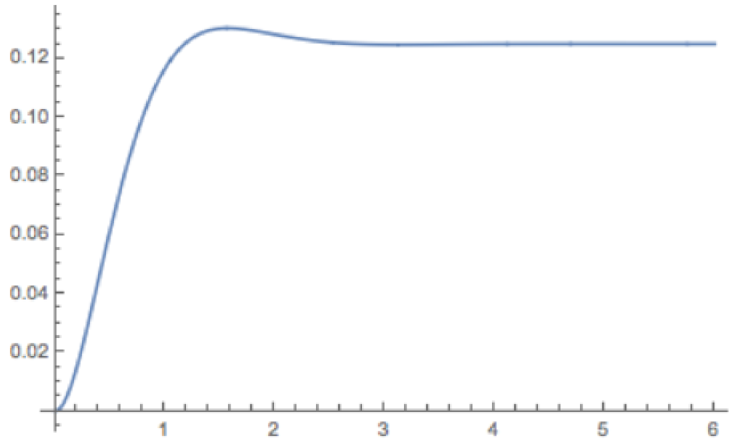
Dependence of output on the pole:

If both pole β_1 and β_2 are complex:

As complex roots come in pair and they are conjugate to each other. The response in the time domain will be-

y(t)= $e^{-\alpha t}[\sin(\gamma t) + \cos(\gamma t)]$, where $\beta = \alpha + j\gamma$

So, the nature of the output will be oscillatory, like in the figure -

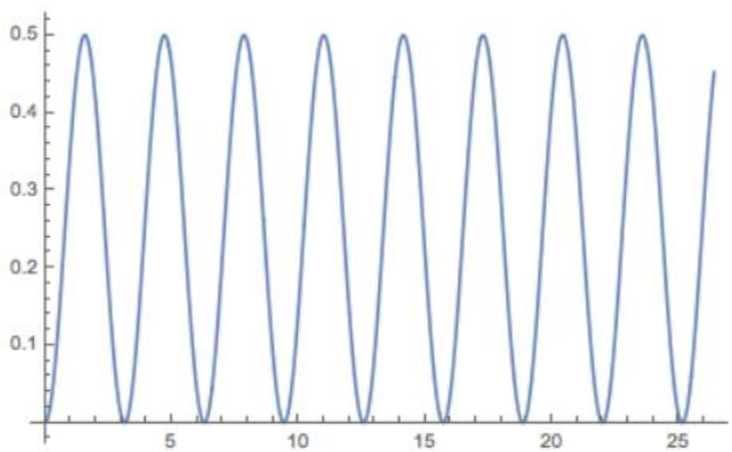


If both poles are imaginary:

If both of the poles are on the imaginary axis, then $\beta = j\gamma$

These poles correspond in the time domain to an output y(t)= $\sin(\beta_1 t) + \cos(\beta_2 t)$, which introduce oscillation.

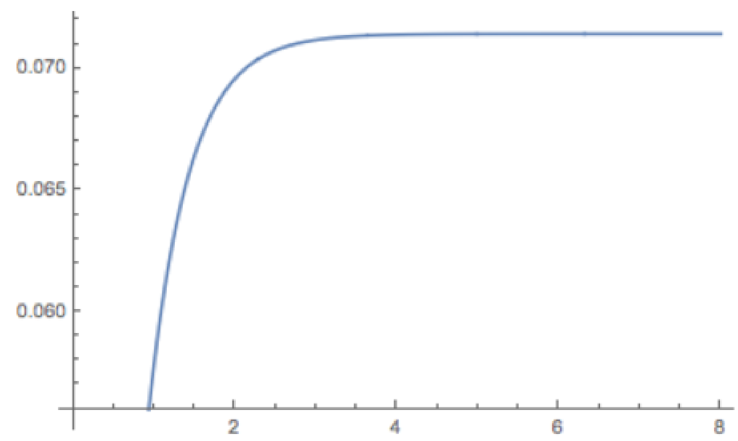
The plot of the output will look like-



If both the poles are real:

When both of the pole are real poles then the nature of the output will be decaying and stable in time domain. So real pole results in signal like in the figure-

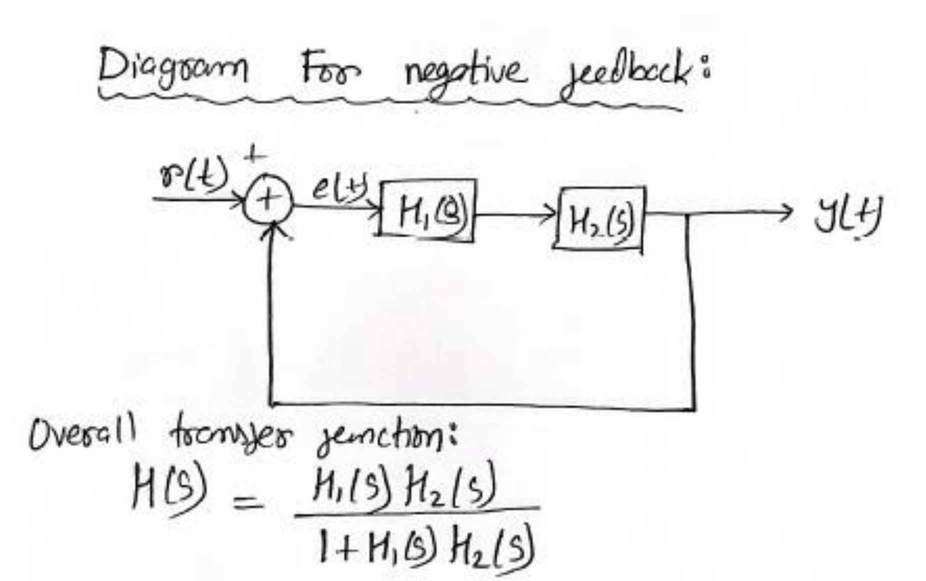
y(t) = $e^{-\alpha 1t}$+$e^{-\alpha 2t}$+ constant



So, depending on the position of the poles on complex plane the output can be either stable, decaying or oscillatory.

Mathematical Analysis of Effect of Different Block:

PID works in the negative feedback mechanism. Here, we will try to analyze the effect of each block in Laplace domain. The diagram of the overall transfer function will be-



Laplace Analysis of the effect of different block:

Proportional Block:

Here, $H_1(s) = K_p$ and $H_2(s) = \frac{1}{s^2+300s+10000}$

Therefore $H(S) = \frac{KpX\frac{1}{s^2+300s+10000}}{1+KpX\frac{1}{s^2+300s+10000}}$

$$= \frac{Kp}{s^2+300s+(10000+Kp)}$$

Now for complex pole roots of $s^2+300s+10000+Kp = 0$ will be complex.

$s^2+300s+10000+Kp = 0$

- ➔ $s^2+2\times150s+22500-12500+Kp = 0$
- ➔ $(s+150)^2 + (Kp-12500) = 0$
- ➔ $s = -150 \pm \sqrt{12500 - Kp}$

So, here with increasing K_p, we will get a complex conjugate pole and therefore the output of the plant will be oscillatory. Therefore, with increasing K_p, overshoot will increase.

Now, $H(s) = \frac{\boldsymbol{Kp}}{(\boldsymbol{s}+\mathbf{150})^2+(\boldsymbol{Kp}-\mathbf{12500})}$

So, the response in the time domain will be –

$$h(t) = \frac{\boldsymbol{Kp}}{\sqrt{\boldsymbol{Kp}-\mathbf{12500}}} \times \boldsymbol{e}^{-\mathbf{150}\boldsymbol{t}}\boldsymbol{sin}(\sqrt{\boldsymbol{Kp}-\mathbf{12500}}\boldsymbol{t})]u(t)$$

So, here we can see that with increasing K_p the frequency of the oscillatory sinusoidal term increases. As a result, increasing K_p will decrease the rise time.

Derivative Block:

For derivative block $H_1(s) = K_d \times s$

So, the overall transfer function will be,

$$H(s) = \frac{\boldsymbol{Kd\ s}}{\boldsymbol{s^2+(300+Kd)s+10000}}$$

Here we can see that introducing derivative block produces a zero in the overall transfer function. Besides, varying the value of K_d, we will be able to place the poles at the right side of the complex plane with real value. So, by the mitigating the effect of poles with introducing a zero and making real poles, derivative block ensures stability and reduces the overshoot.

Integral Block:

For integral block, $H_1(s)= \frac{Ki}{s}$

Then the overall transfer function will be,

$$H(s) = \frac{\boldsymbol{Ki}}{\boldsymbol{s^3+300s^2+10000s+Ki}}$$

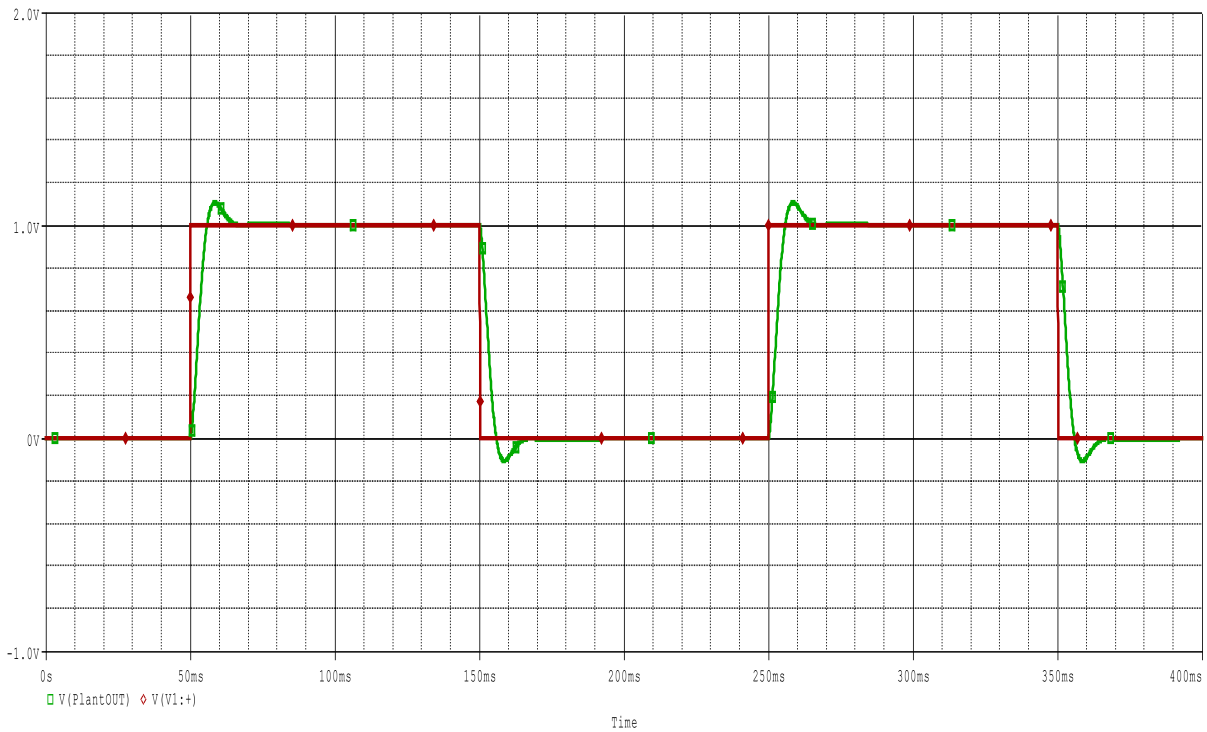
If we derive partial fraction then we will get an expression like this,

$$H(s) = \frac{\boldsymbol{A}}{\boldsymbol{s}} - \frac{\boldsymbol{B}}{\boldsymbol{s^2+300s+B}}$$ **;** where B is a constant

So, corresponding to A/s term we get a constant value in the time domain and based on the value of Ki, we can get either complex or real pole. But we can interpret here that with increasing value of Ki, the pole will be complex, and we will get more overshoot. But with time the oscillatory nature of the output with become less dominate and steady state value will set at the desired constant value with no steady state error. So, integrator eliminates steady state error but increasing the number of poles in the system make the system unstable with increasing overshoot.

Results and Output Analysis:

Output Characteristics After the Implementation of PID Controller:



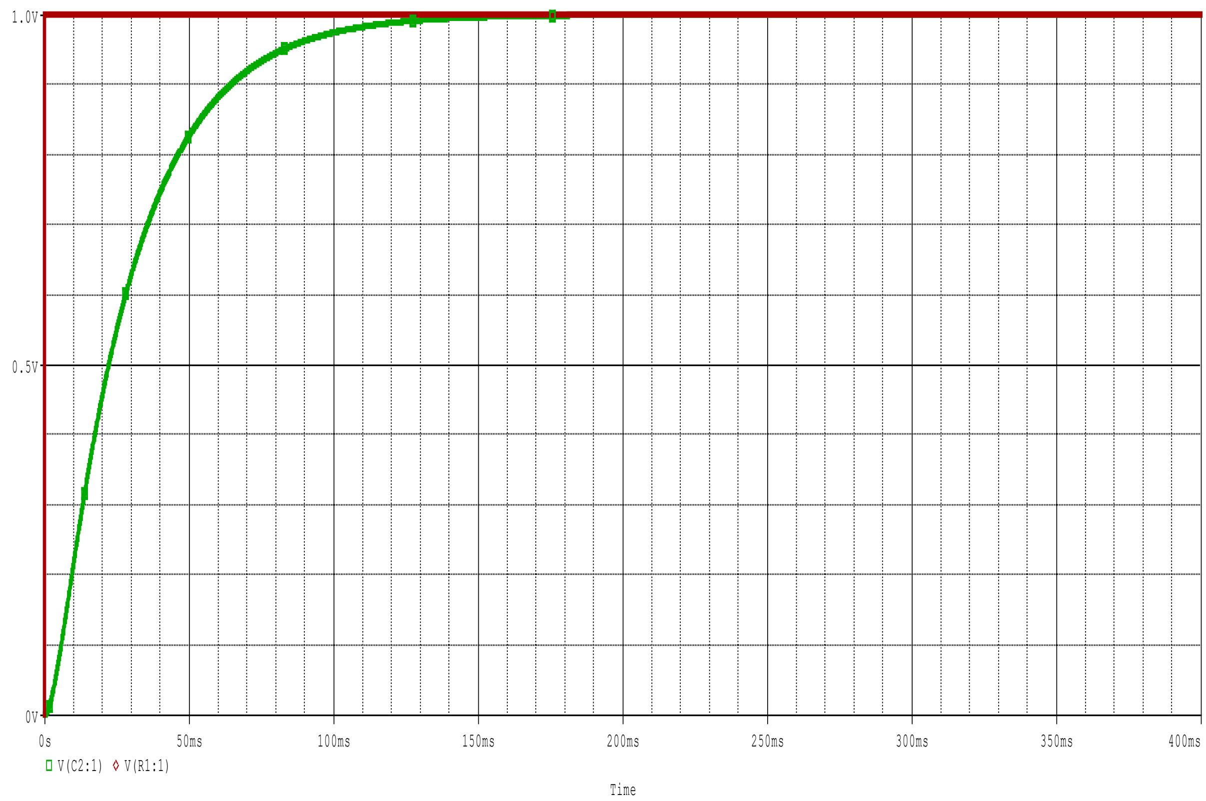
Output Analysis:

We can see that with PID controller the motor behaves in a more improved way. The motor speed is following the expected reference curve. The rising time becomes quite small and tolerable, and the setting time also get reduced. The motor is now adapting to sudden changes quickly. The overshoot decreases as well. So, with PID controller the motor is being controlled effectively.

Effects of Proportional, Integral and Differential Blocks:

Now, we will go through a thorough analysis. First, we will see the effect of different blocks and proportionality constant K_p, Integral constant K_i and Differential constant K_d on controlled output. For this we will analyze the step response of Proportional, Differential and Integral blocks accordingly.

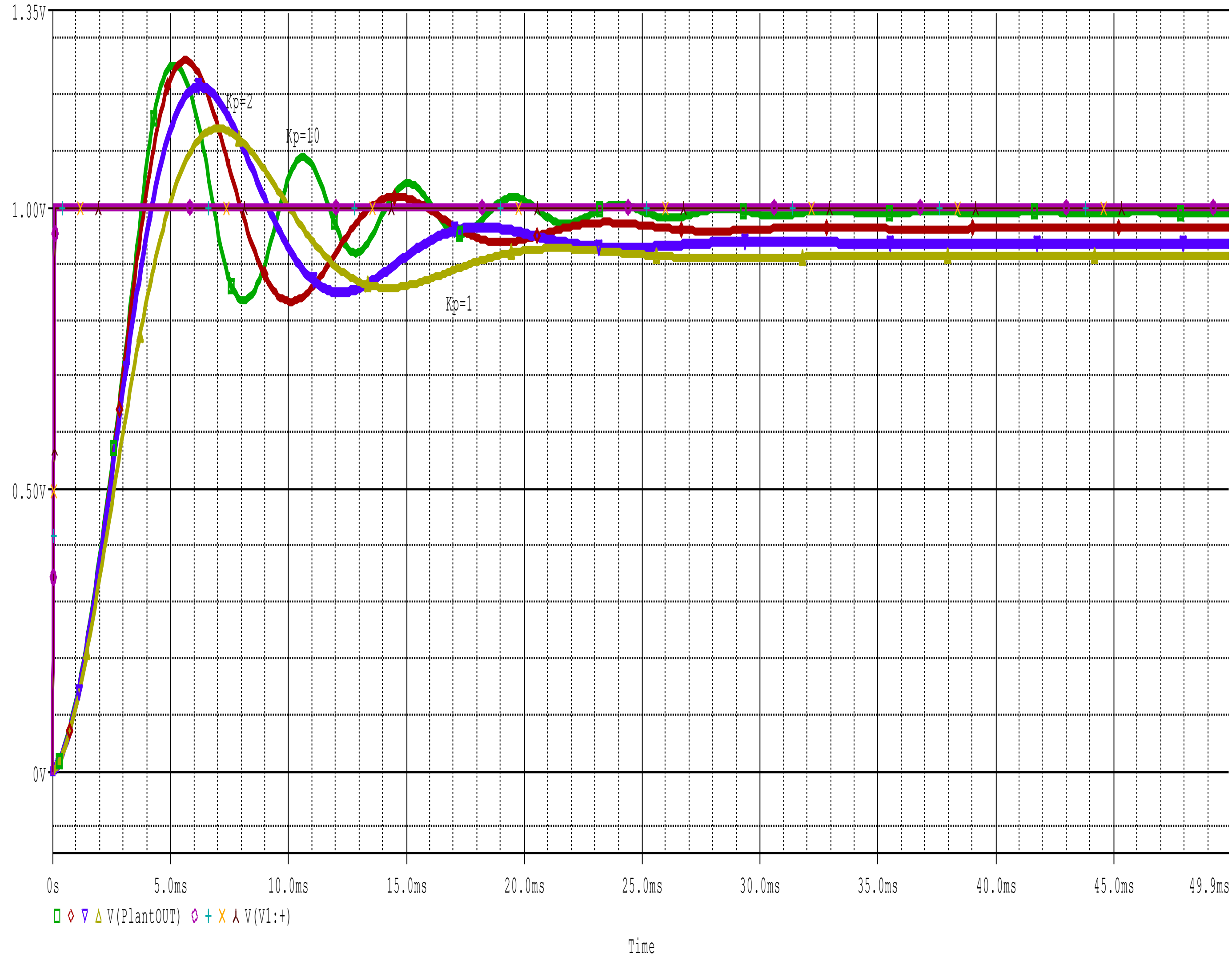
Step Response without PID:



We can see that the motor speed response is sluggish. The rise time is almost 150-200ms. So, the motor is clearly not working with precision.

Effect of the Proportionality Block:

Step response of the motor with Different Value of K_p:



To find the step response we do a parametric sweep on the value of resistance R1.

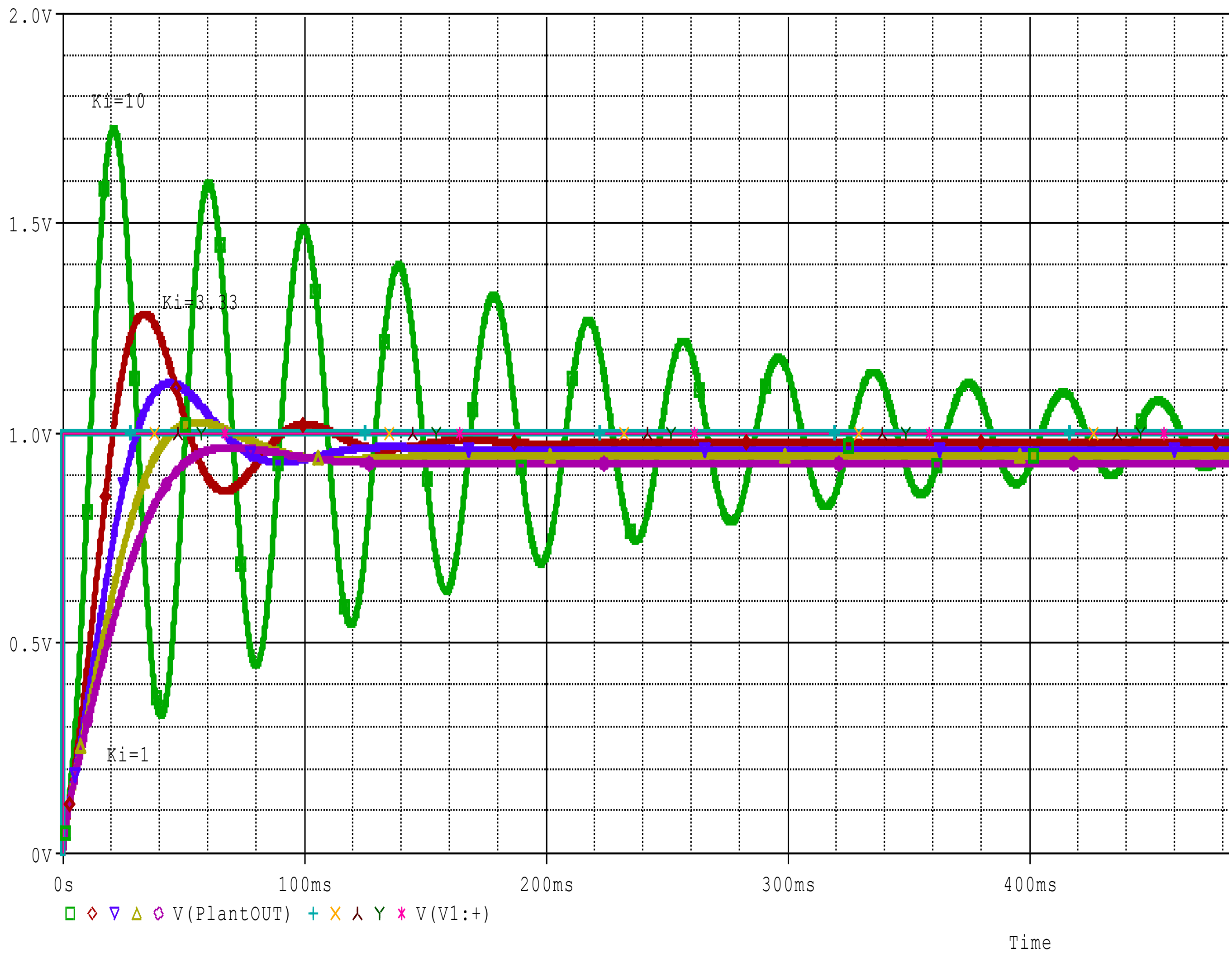
From the plot we can observe that as the value of K_p increases from 10 to 1,

- The rise time decreases
- The setting time increases

- The overshoot increases
- Steady State error decreases

Effect of the Integral Block:

Step response of the motor with Different Value of K_i:

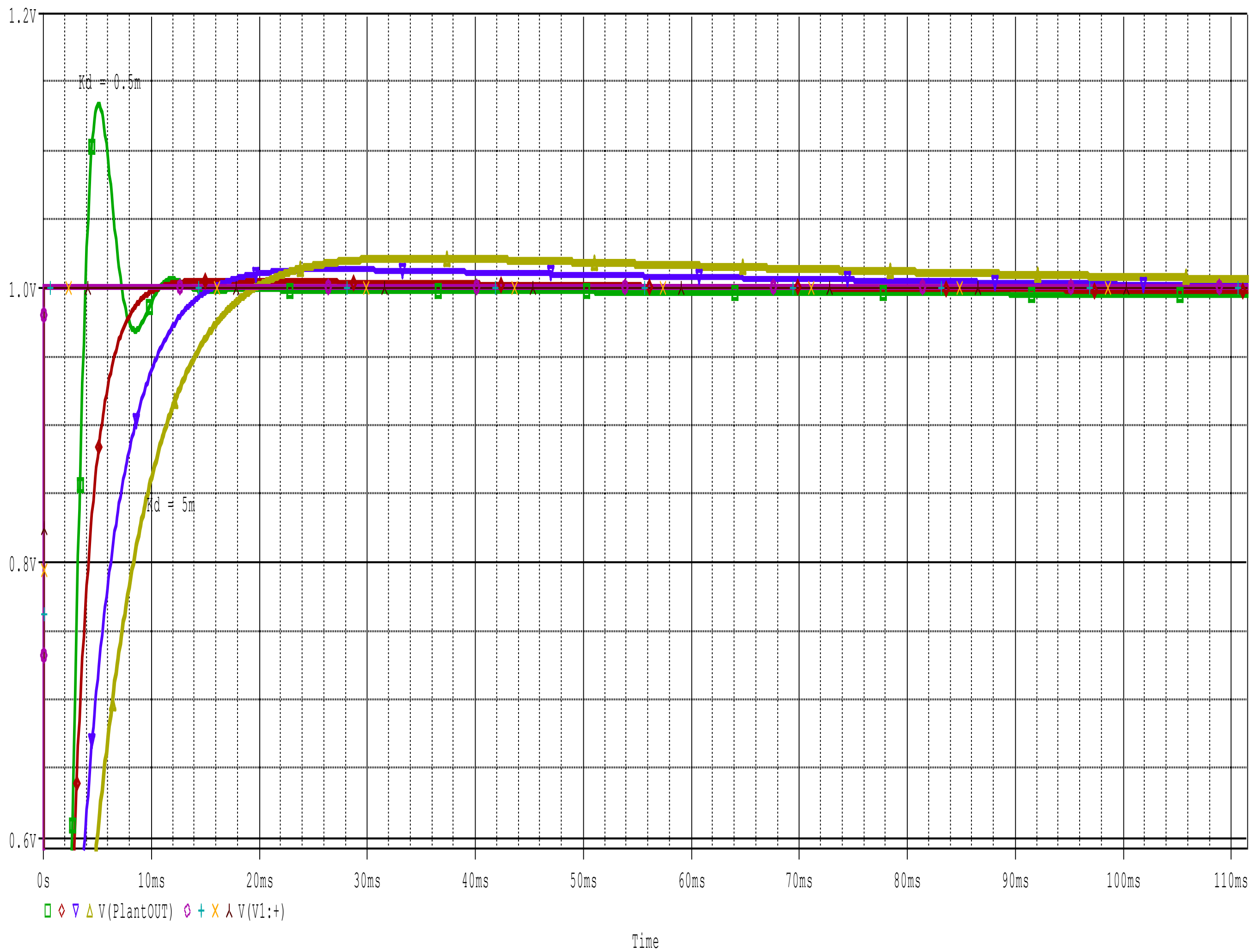


Performing the parametric sweep on the value of R3, we can observe that with increasing value of K_i,

- The rise time decreases
- The setting time increases
- The overshoot increases
- Steady State error eliminates (tends to zero)

Effect of the Derivative Block:

Step response of the motor with Different Value of KD:



Performing the parametric sweep on the value of R4, we can observe that with increasing value of K_d,

- The rise time increases
- The setting decreases
- The overshoot decreases
- Steady State has no effect

The summary of the simulation outcomes of step response is given in the following table-

Table: of Increasing Gain Factors Independently

Parameter	Rise Time	Overshoot	Setting Time	Steady State Error
Kp	Decrease	Increase	Increase	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Increase	Decrease	Decrease	No change

TUNING METHOD:

There are different tuning processes for PID tuning. In this project to control the process variable of the plant we will use "Manual Tuning Process".

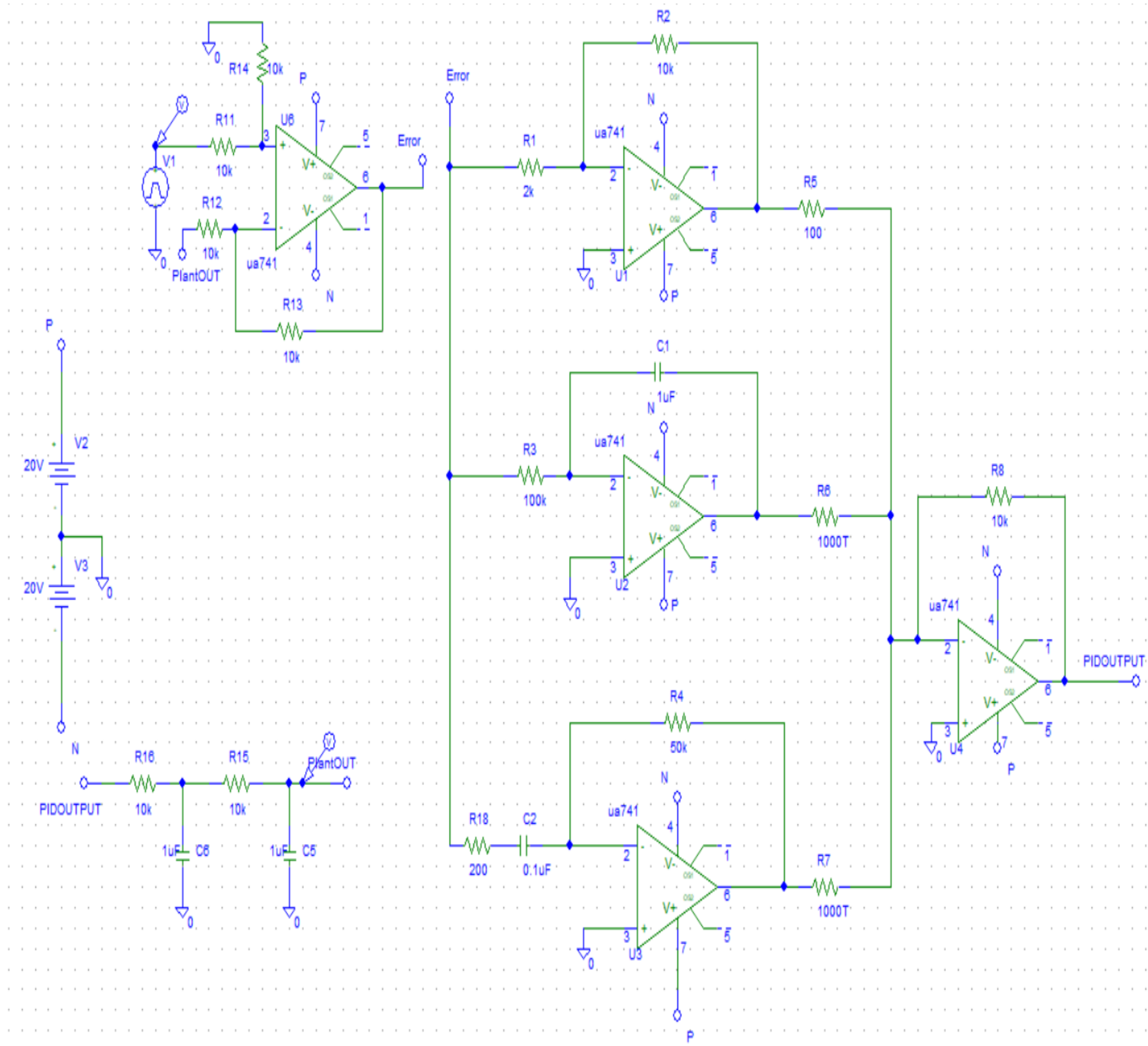
In manually tuning process we tune the gain factors (K_p, K_i, K_d) manually by varying responsible circuit parameter based on the empirical data and expertise. There are other tuning processes as well which require rigorous mathematical analysis which was out of our scope. Therefore, we use manual tuning to show the application of operational amplifier in implementing PID controller,

Now we will tune our PID controller gradually and observe how our output wave changes to follow the reference value.

STEP 1: Varying K_p While Keeping the Values of K_i and K_d Close to Zero

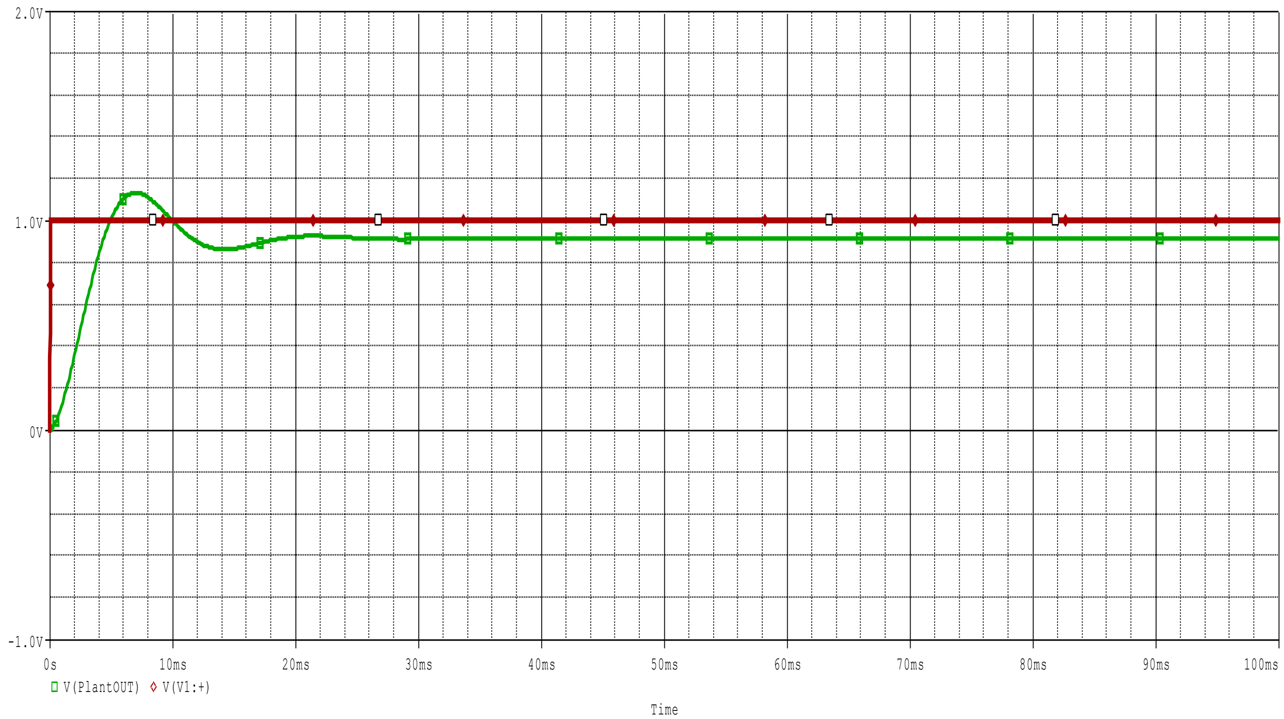
We will continue do it until the output starts to getting close to the desired output and starts to oscillate increasingly.

Output Modification By Proportional Component:



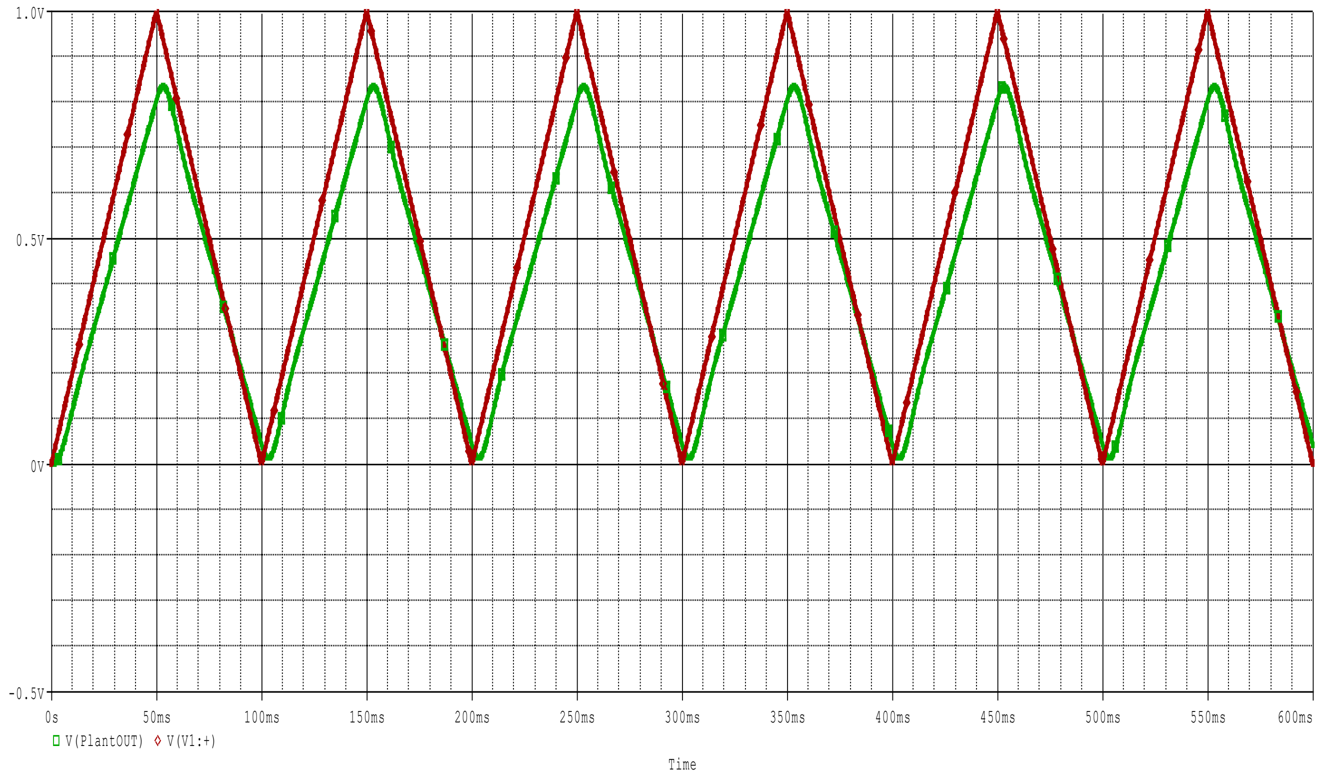
Output Plot:

For step response:



For Rectangular Pulse:

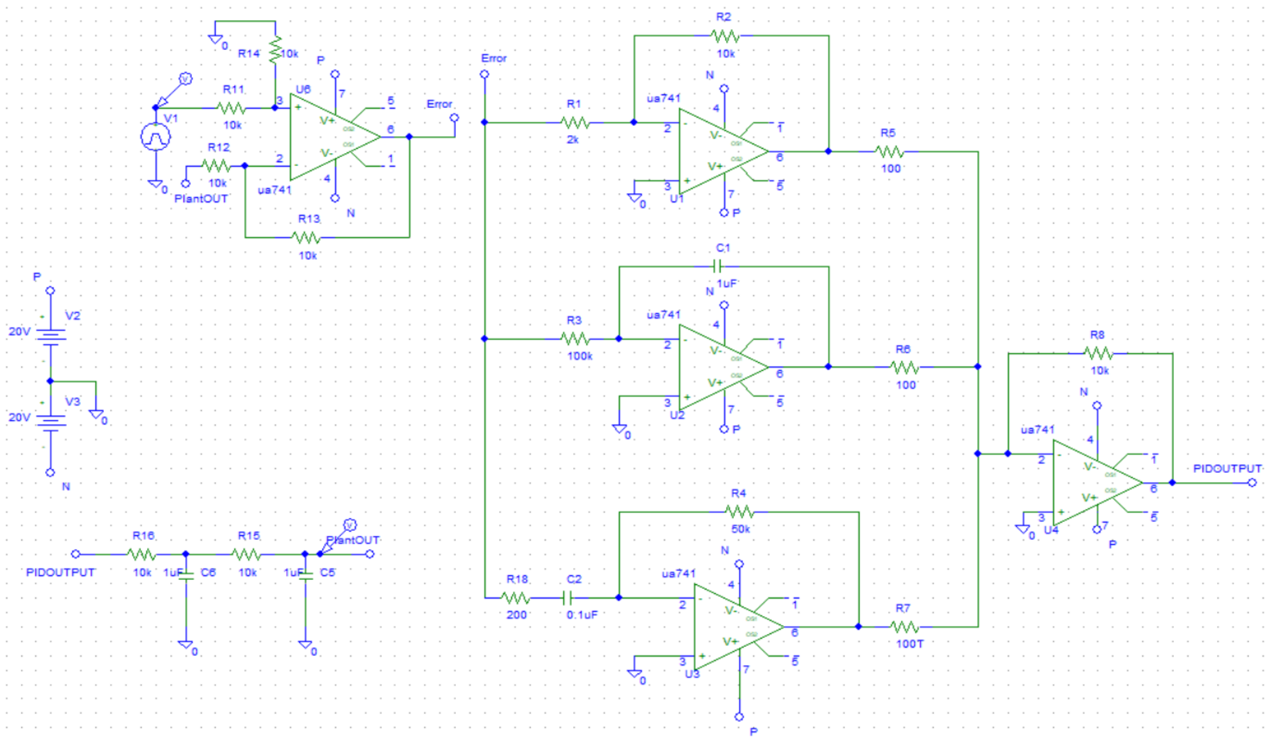
For Triangular Wave:



Proportional block multiplies the error signal with K_p to reduces the error to follow the reference signal. As the signal amplifies with the necessary gain factor to get improved, the rising time decreases to approximately 8ms in case of step response and the steady state error also decreases. But proportional block cannot entirely minimize the error as we can see, there is steady state error in every case. It is due to the fact that as the process variable approaches the set points error signal gets smaller and smaller as a result proportional block then cannot increase the signal appreciably anymore. Also, the proportional block cannot predict future so when the error is high, it increases the input signal so rapidly that the signal crosses the set points and then again as error get negative it reduces the signal output to follow the set points as a result, overshoots increases and finally the output gets set a lower value then the expected one.

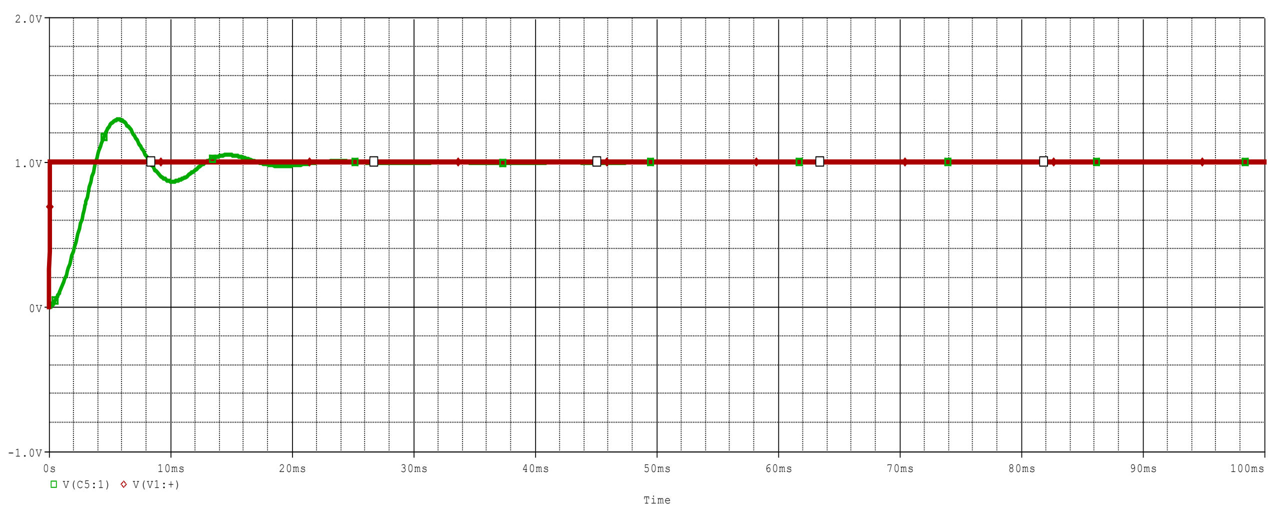
Step 2: Varying K_i holding K_p constant and K_d close to zero

We continue it until the steady state error is eliminated within the sufficient time for the process

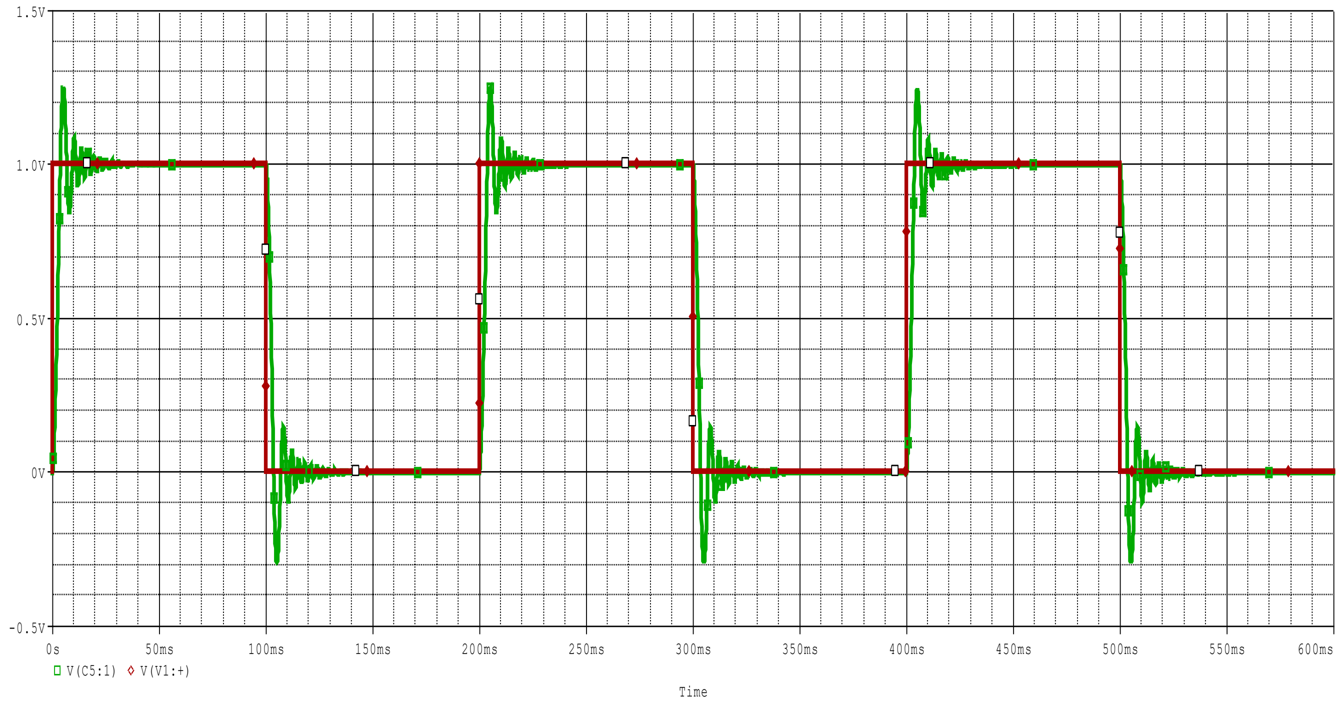


Output Plot:

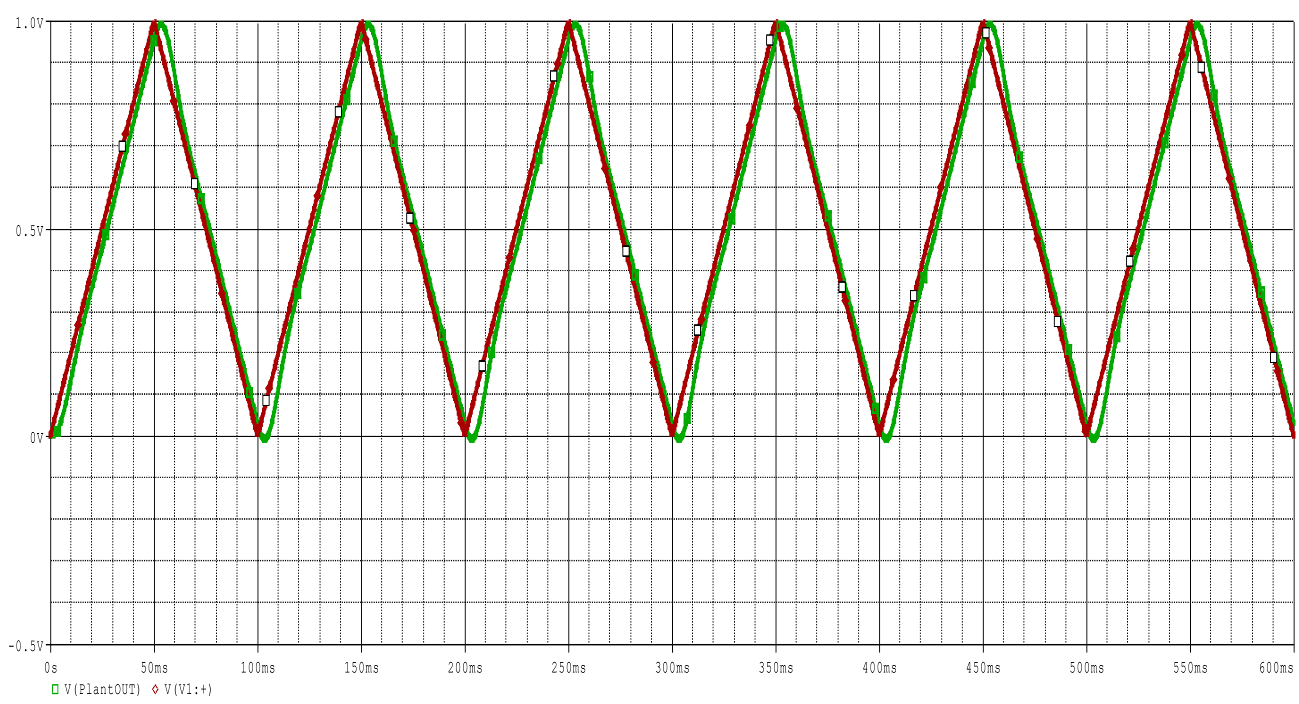
For step response:



For Rectangular Pulse:

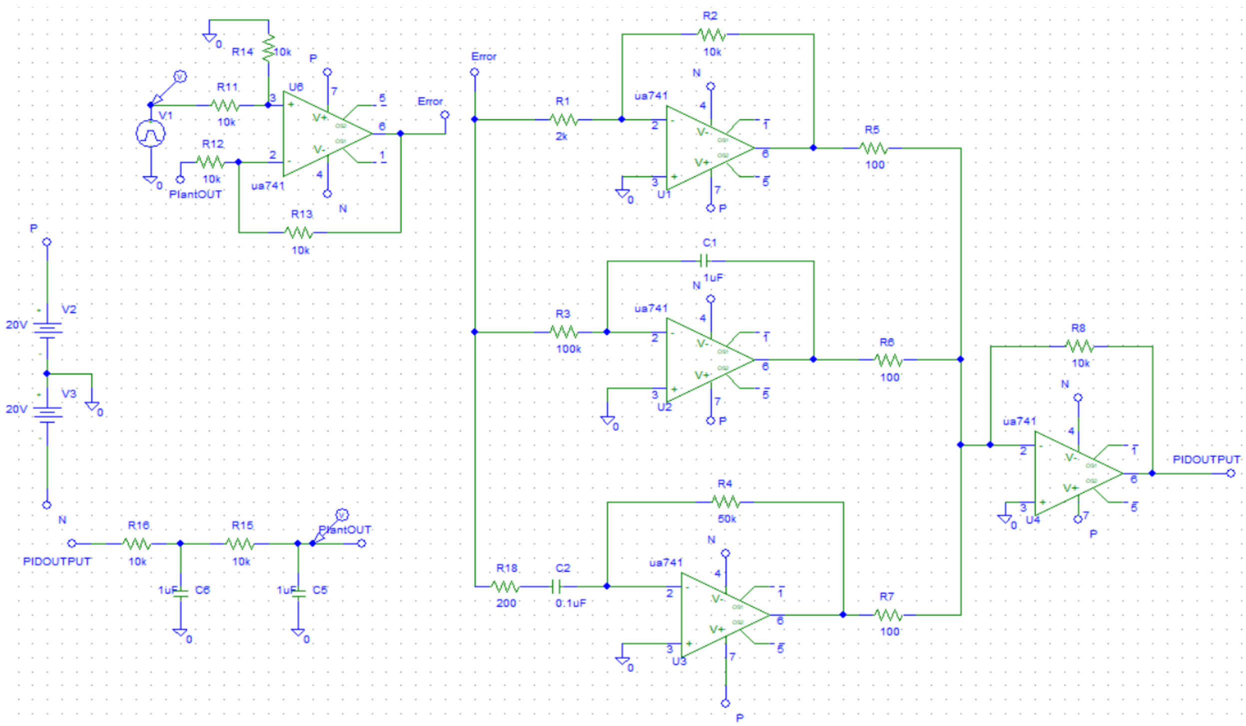


For Triangular Wave:



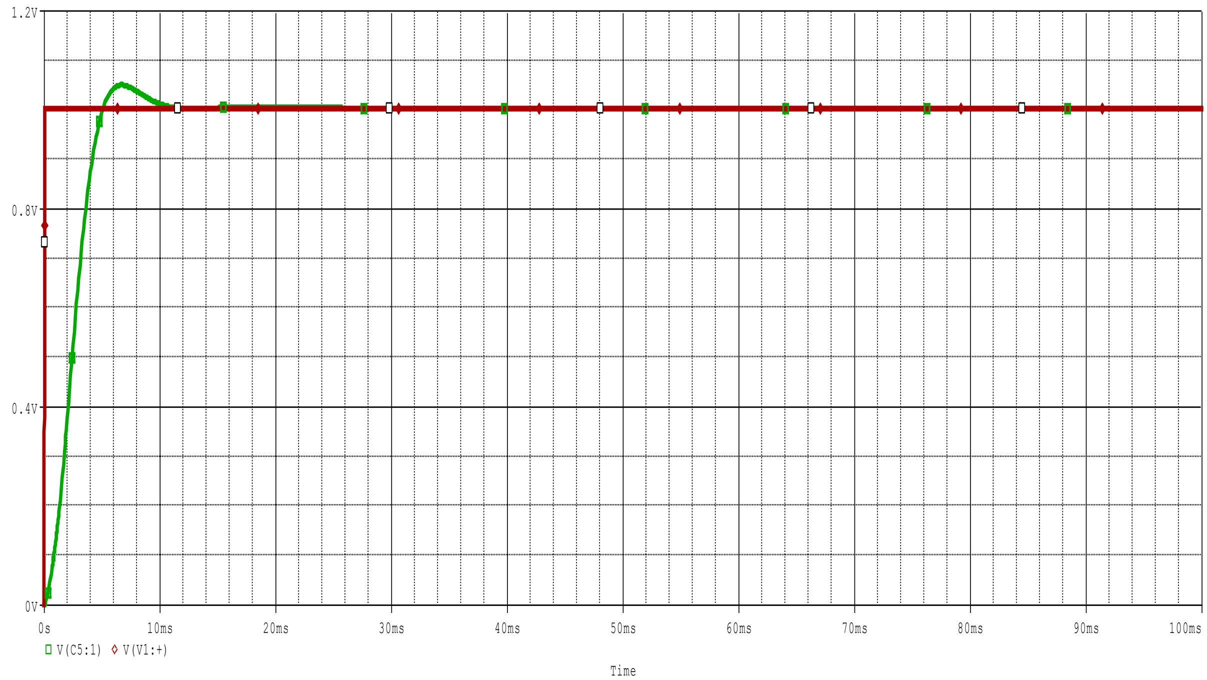
We can clearly notice that with the involvement of integrator, the steady state error gets eliminated as expected. Though the proportional block minimizes the steady state error, it cannot totally eliminates the steady state error as it doesn't have any memory. Proportional blocks increase or decreases output depending on present error value. As a result, proportional block can never match the desired output value, the output is always less than or greater than the desired value. But integrator block can act as a memory device. It accumulates the past error and try to compensate for it. The total summation of residual error helps the integral block to increase the output signal rapidly, ensuring small rise time. However, sometimes accumulation of residual error is too large and so overshoots happens because of crossing the set points from up and down. Therefore, we can see in all three cases increasing value of Ki eliminates the steady state error and reduces rise time.

Step 3: Varying K_d while keeping values of K_p and K_i constant

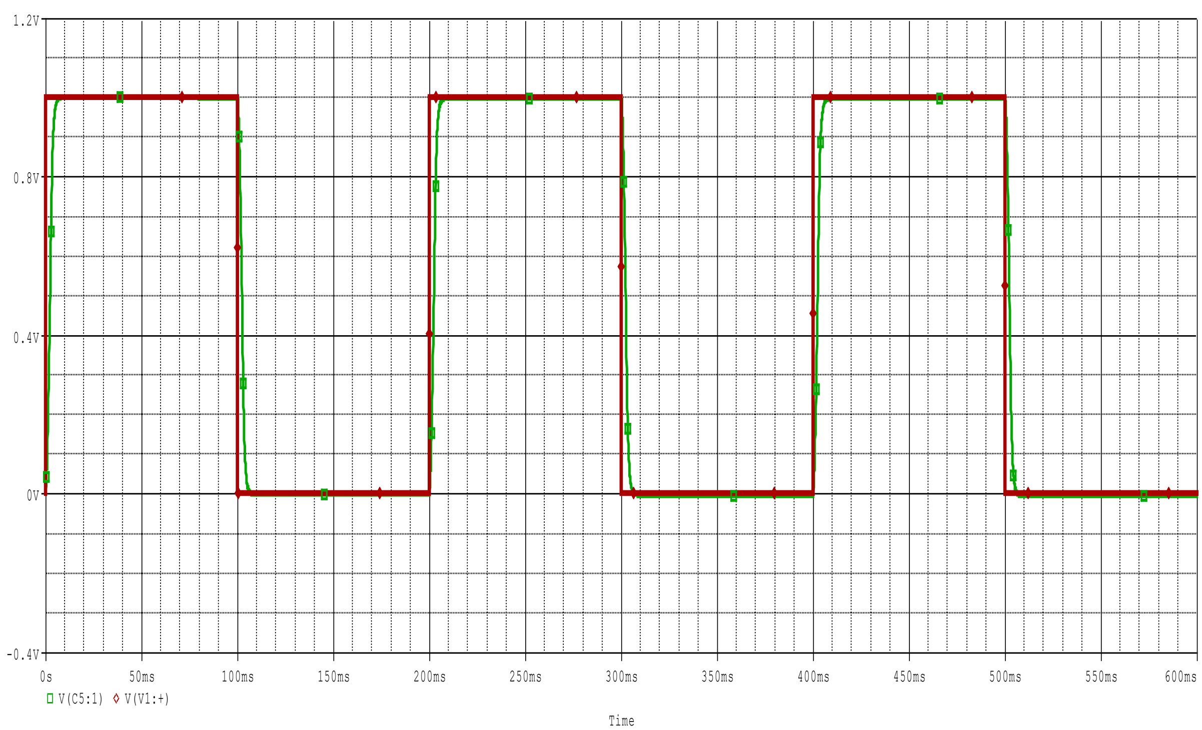


Output Plot:

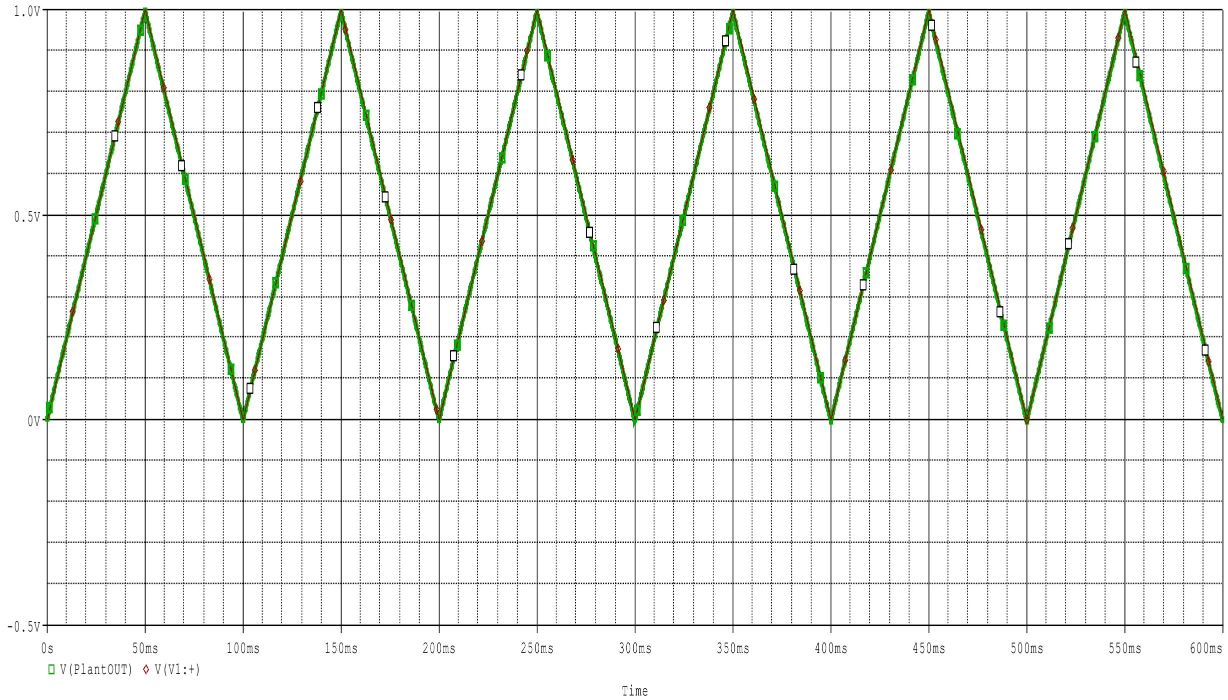
For step response:



For Rectangular Pulse:



For Triangular Wave:



We can observe that the differential block decreases the overshoot. Though potential gain can reduce the rise time and steady state error by rapidly amplifying the error signal to reach the set points and integral gain totally eliminates the error during this process, the overshoots get increased. Differential block can predict future by controlling the rate of change of the output, at first when the error was high and proportional block was increasing rapidly the rate of change was higher and derivate blocks also helps to increase rapidly. When the process variable approaches to the set point the rate of change of error voltage start to reduce with time. So, the output of the differential block becomes negative to control the output not to overtake the set point and create overshoot. Since the output approaches the set points more oriented manner without fluctuation, we can get an oscillation free output, but in this procedure, the rising time increases significantly.

Gain Factors Value at Tuned Condition:

By following this process, we can see that we are almost approaching the reference point. We can tune the PID more for even better output which entirely follows our set point. However, as we can see the rise time is now only 5ms compared to the without PID 100ms and overshoot is nearly eliminated, and the steady state value is also zero. So, the level of accuracy we attain is adequate for the practical purpose of our plant.

Here at the tuned condition,

$K_p = 5$

$K_i = 10$

$K_d = 0.005$

Limitations:

In this project we tune the PID controller manually. Manual tuning is time consuming and sometimes takes a lot of trial to reach the set points. There are some well developed algorithms which is more efficient and can lead us to perfect condition at which we desire to operate. Such as- Ziegler-Nichols method, Tyreus-Luyben method etc.

Another important shortcoming is the noise in derivative block. Derivative block amplifies the higher frequency process noise that can cause a large amount of change in the output. Besides, in higher frequency the capacitor in the derivative block gets short and the op amp can act uncertainly. So, to recover that we already use a resistance in series with that. In higher frequency thus it will not be unstable rather work as an inverting amplifier.

Scope for Future Work:

- We can implement a low pass filter with derivative block to eliminate the higher frequency noise and make the PID controller more efficient and stable
- We can work to find a better algorithm to reach the tuned condition more easily and effectively
- We can work to implement the whole system using a smaller number of operational amplifiers to make it cost efficient in large scale.

Conclusion:

PID Control is an important and effective form of control mechanism used extensively in industrial and scientific fields. Our project was to implement such a mechanism using µa-741 Operational Amplifiers in the light of our curriculum. We have designed a PID control system and explored the shortcomings of this system as well as manually attempted to solve them. We have been successful in yielding satisfactory enough results implementing our knowledge and also gained experience regarding future potentials and ways of improving this project.