Bangladesh University of Engineering & Technology

Department of Electrical and Electronic Engineering

Course Number : EEE 208
Course Title : Electronic Circuits II Laboratory

Project Title : PID Controller
Group Number : 8

Submitted By :

Himel Kundu
ID: 1906014

Md. Rowatul Jannat Rafi
ID: 1906016

Submitted To :

Dr. Md. Shafiqul Islam
Professor
Department of Electrical and Electronic Engineering, BUET

Shafin Bin Hamid
Lecturer
Department of Electrical and Electronic Engineering, BUET

Date of Submission : 04/09/2022

OBJECTIVE

Designing controllers for various everyday practical processes has proven to be a difficult task for designers. When designing a controller, various factors like stability, steady state error, the impact of parameter variations, speed, etc. must be taken into account. This project implements an arbitrary second order plant with a highly oscillatory response. The system stability is then improved by using a PID controller.

Introduction

PID control is an established technique for moving a system toward a desired position or level. It is used in numerous chemical and scientific processes, as well as automation, and is essentially ubiquitous as a means of controlling temperature. To keep a process's actual output as close as possible to the target or setpoint output, PID control employs closed-loop control feedback.

Background Theory

A proportional integral derivative (PID) controller can be used as a means of controlling temperature, pressure, flow and other process variables. As its name suggests, a PID controller combines proportional control with additional integral and derivative adjustments which help the unit automatically compensate for changes in the system. The purpose of a PID controller is to force feedback to match a setpoint.

The working principle behind a PID controller is that the proportional, integral and derivative terms must be individually adjusted or "tuned." Based on the difference between these values, a correction factor is calculated and applied to the input. For example, if an oven is cooler than required, the heat will be increased.

The ability of the PID controller to apply precise and effective control using the three control terms of proportional, integral, and derivative influence on the controller output makes it unique. The fundamentals of how these terms are created and used are illustrated in the block diagram on the right. It depicts a

PID controller that continuously calculates an error value e (t) as the difference between a desired setpoint SP = r(t) and a measured process variable PV=y(t), that is, e(t) = r(t)- y(t) and then applies a correction based on adjusting the control variable u(t) in an effort to reduce the error over time.

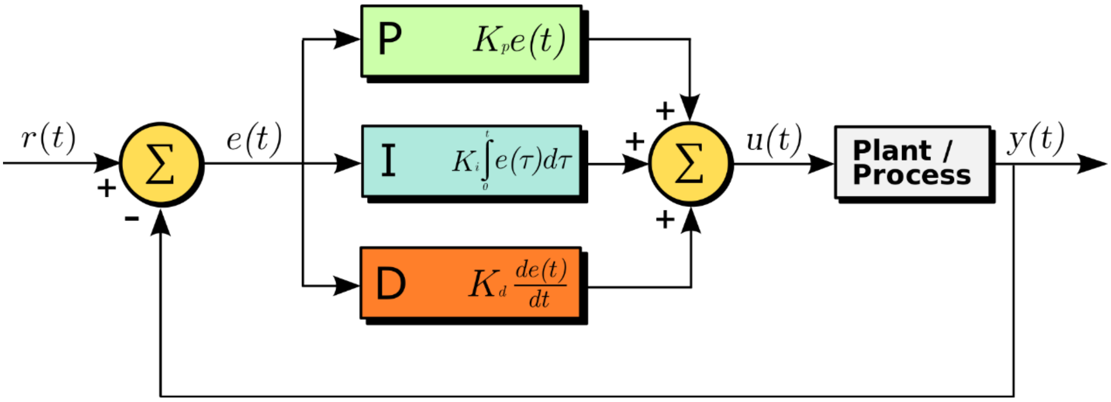


Figure: A Block Diagram of PID Controller in a feedback loop

The control variable u(t) can be expressed mathematically by the equation,

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

where,
K_p is the proportional gain, a tuning parameter
K_i is the integral gain, a tuning parameter
K_p is the derivative gain, a tuning parameter
e(t) is the error (difference between setpoint r(t) and the process variable y(t))
u(t) is the controller output

Tuning parameters:

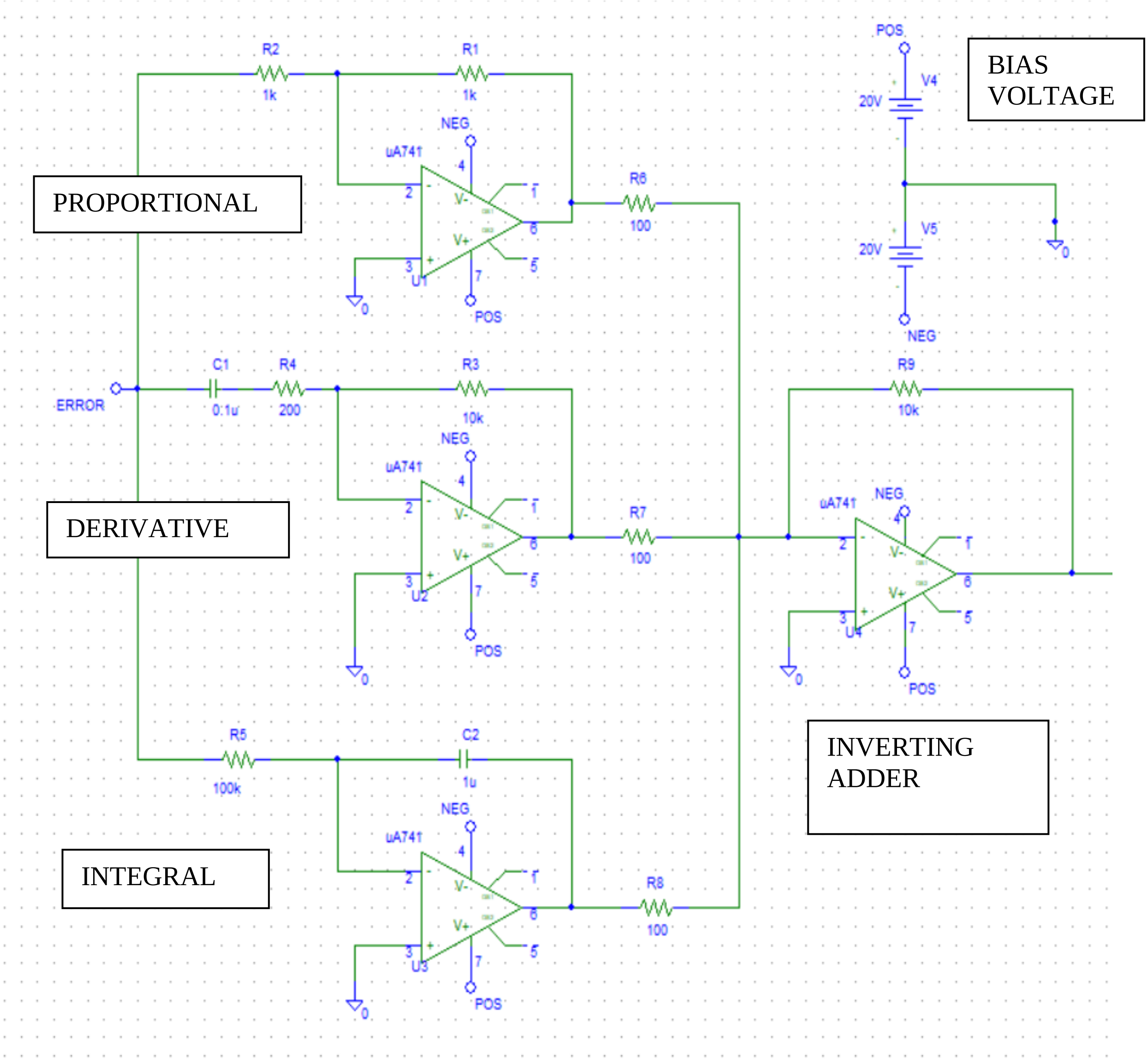
Proportional Term: The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p, called the proportional gain constant. For a given change in the error, a high proportional gain causes a significant change in the output. The system may become unstable if the proportional gain is too high. A small gain, on the other hand, results in a controller that is less sensitive or responsive and has a small output response to a large input error. When responding to system disturbances, if the proportional gain is too low, the control action may be insufficient. The proportional term should be primarily responsible for the change in output.

Integral Term: The integral term's contribution is proportional to both the magnitude and duration of the error. In a PID controller, the integral is the sum of the instantaneous error over time, giving the accumulated offset that should have been corrected previously. After that, the accumulated error is multiplied by the integral gain (Ki) and added to the controller output. The integral term speeds up the process's movement toward the setpoint and eliminates the residual steady-state error that occurs only with a proportional controller. However, because the integral term responds to past errors, it can cause the present value to overshoot the setpoint value.

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

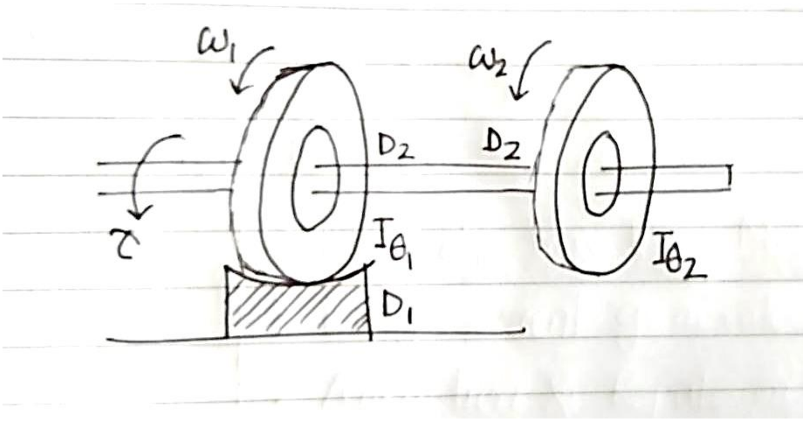
Simulation Setup

The schematic of the PID controller unit is given below.



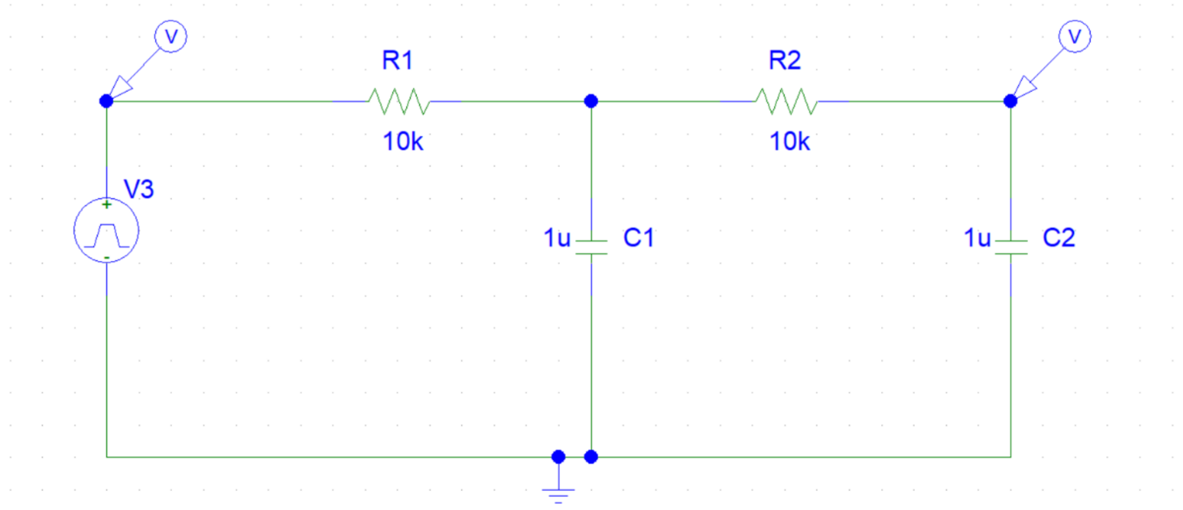
The PID controller implemented above has 5 blocks in total. The proportional block is an inverting operational amplifier. The derivative and integral block has also been implemented by using capacitors, resistors and op amps. The output of the proportional, integral and derivative block is then summed up by an inverting adder which is also implemented by a simple op-amp and resistors. All the operational amplifier used in this project is a basic uA741 amplifier. The whole unit serves as a controller, whose output will directly control the process or the plant in consideration.

Model of the plant



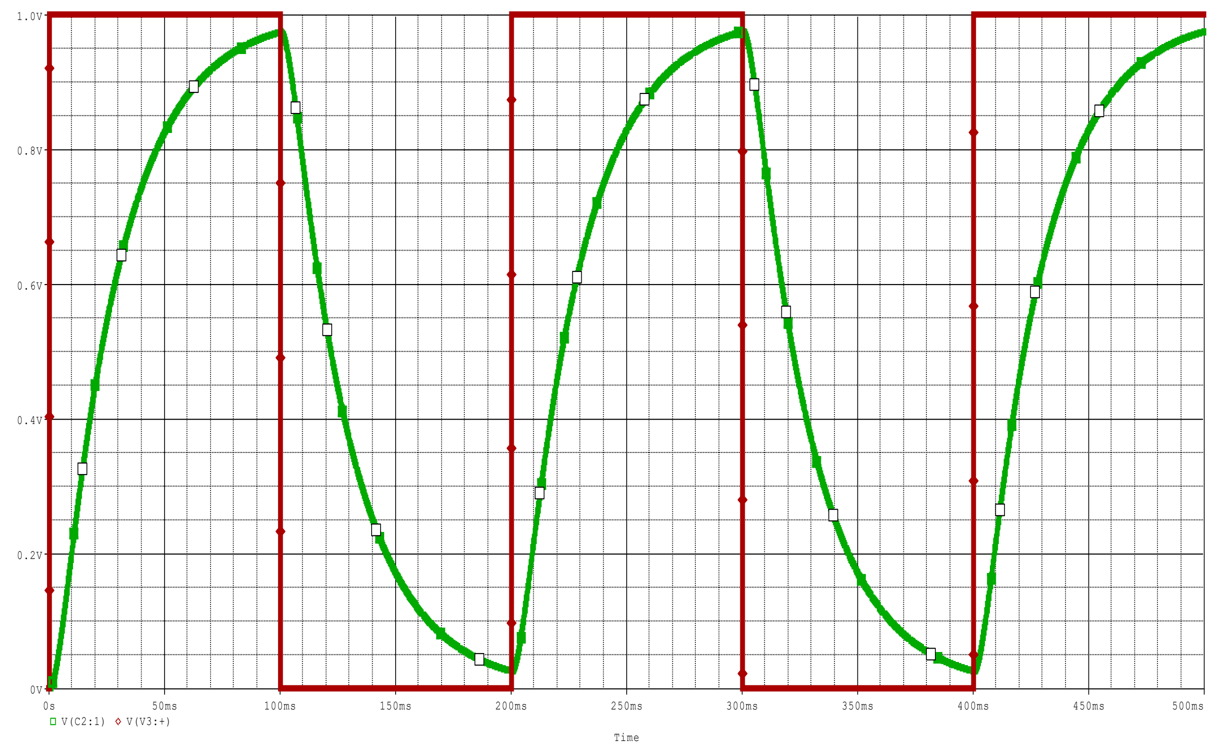
The mechanical model used as a plant in this project is a two-wheeler. The left wheel has a damper attached to it. As a result, for a given applied torque, the speed of the two wheels is not synchronized. The wheel on the right side has higher speed compared to the left wheel. So, it slows down after a delay. This is not an acceptable phenomenon in the automobile manufacture industry. A PID can be used to synchronize the speed of a two-wheeler.

Schematic Diagram of the plant (Analogous to a mechanical system)



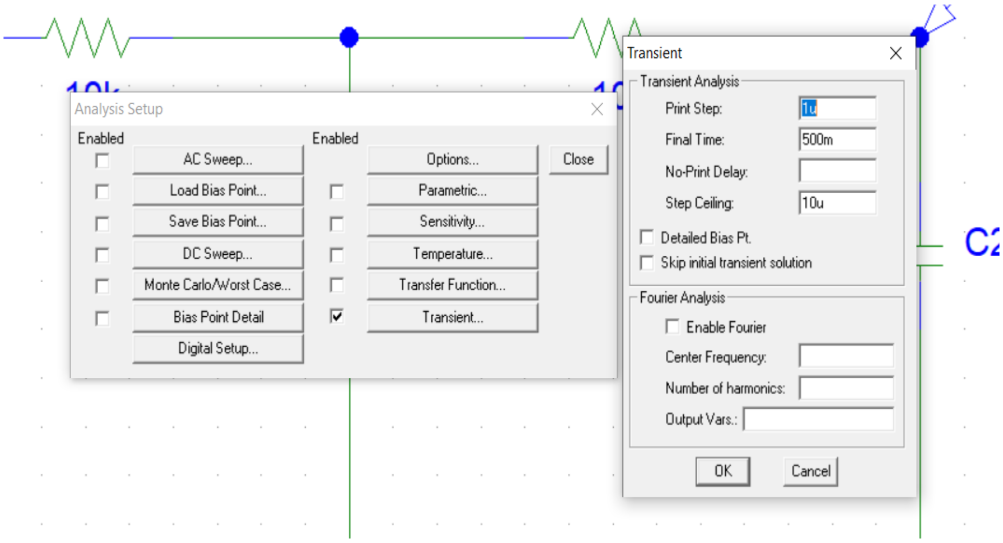
An analogous electrical system has been implemented to study the response of the mechanical system.

The output Response of the plant without the PID controller



The output response has been observed for a square wave or pulsating wave. The result is to some extent disappointing. It almost took 100 milli seconds to follow the setpoint for the output. The rise time is also not satisfactory. Such distorted output response has a detrimental effect to the plant and decreases the stability with respect to the varying setpoint. A controller is needed to make the output follow the command with a very little rising time and match the magnitude of the setpoint without any delay or oscillations. Hence, a PID controller is designed to give the desired output.

To observe the output in PSPICE, a transient analysis has been performed. The attributes chosen for the simulation is depicted here.



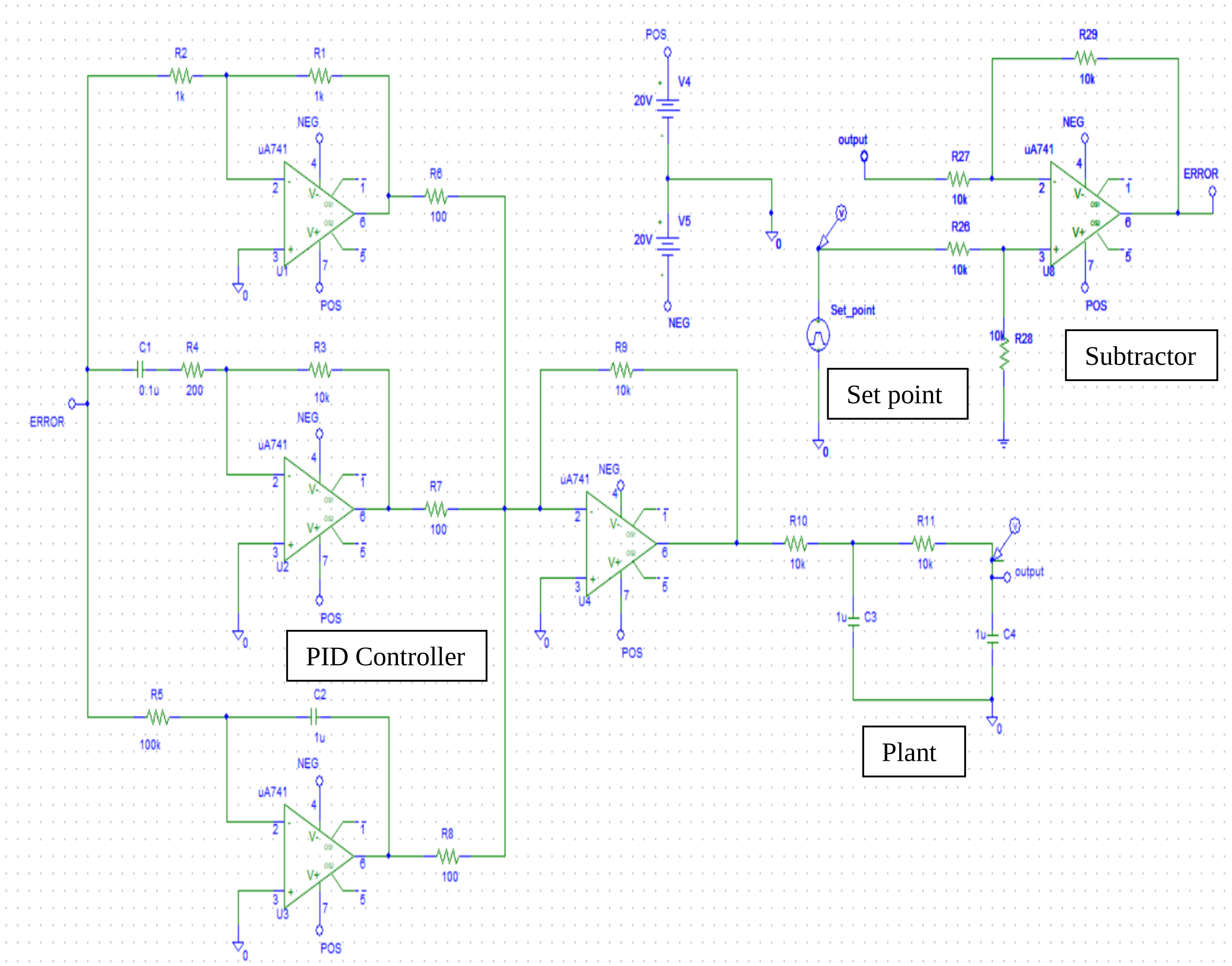
☐ AC Sweep...
☐ Load Bias Point...
☐ Save Bias Point...
☐ DC Sweep...
☐ Monte Carlo/Worst Case...
☐ Bias Point Detail
Digital Setup...
Options...
☐ Parametric...
☐ Sensitivity...
☐ Temperature...
☐ Transfer Function...
☑ Transient...

Transient Analysis
Print Step: 1u
Final Time: 500m
No-Print Delay:
Step Ceiling: 10u
☐ Detailed Bias Pt.
☐ Skip initial transient solution

Fourier Analysis
☐ Enable Fourier
Center Frequency:
Number of harmonics:
Output Vars.:

Schematic of the PID

controller with the plant (After tuning)

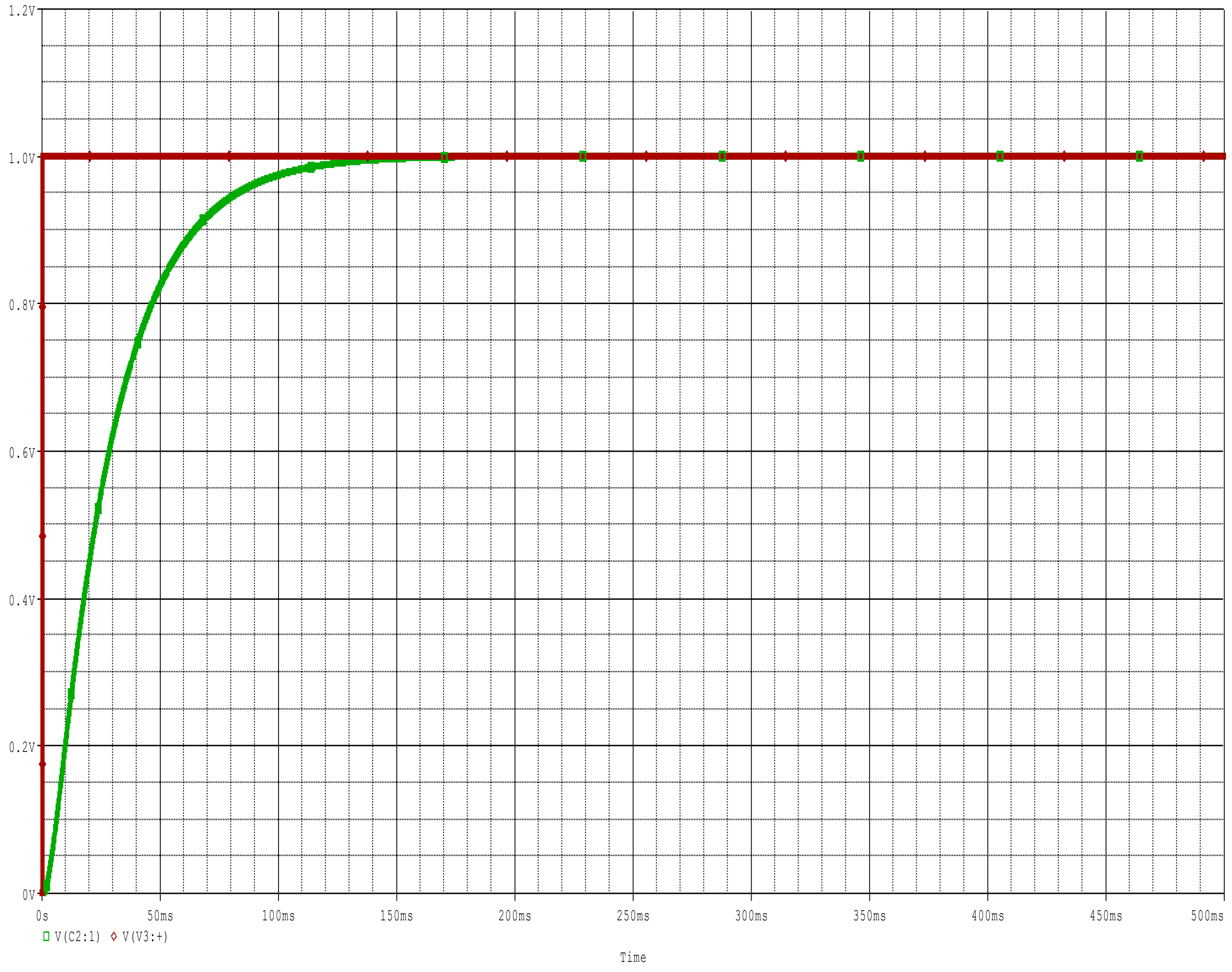


The overall circuit with the controller, mixer and plant has been shown in the above schematic. PID is embedded in the negative feedback. The mixer is implemented with a subtractor designed as a differential amplifier. Again, a uA741 operational amplifier is used to subtract the feedback signal from the setpoint, resulting in an error value which serves as the input of the controller. The output of the controller serves as the input of the plant. The output is again fed to the mixer to generate error. Thus, negative feedback ensures the desired output with the help of the controller. Again, a transient analysis has been done to observe the output.

Results

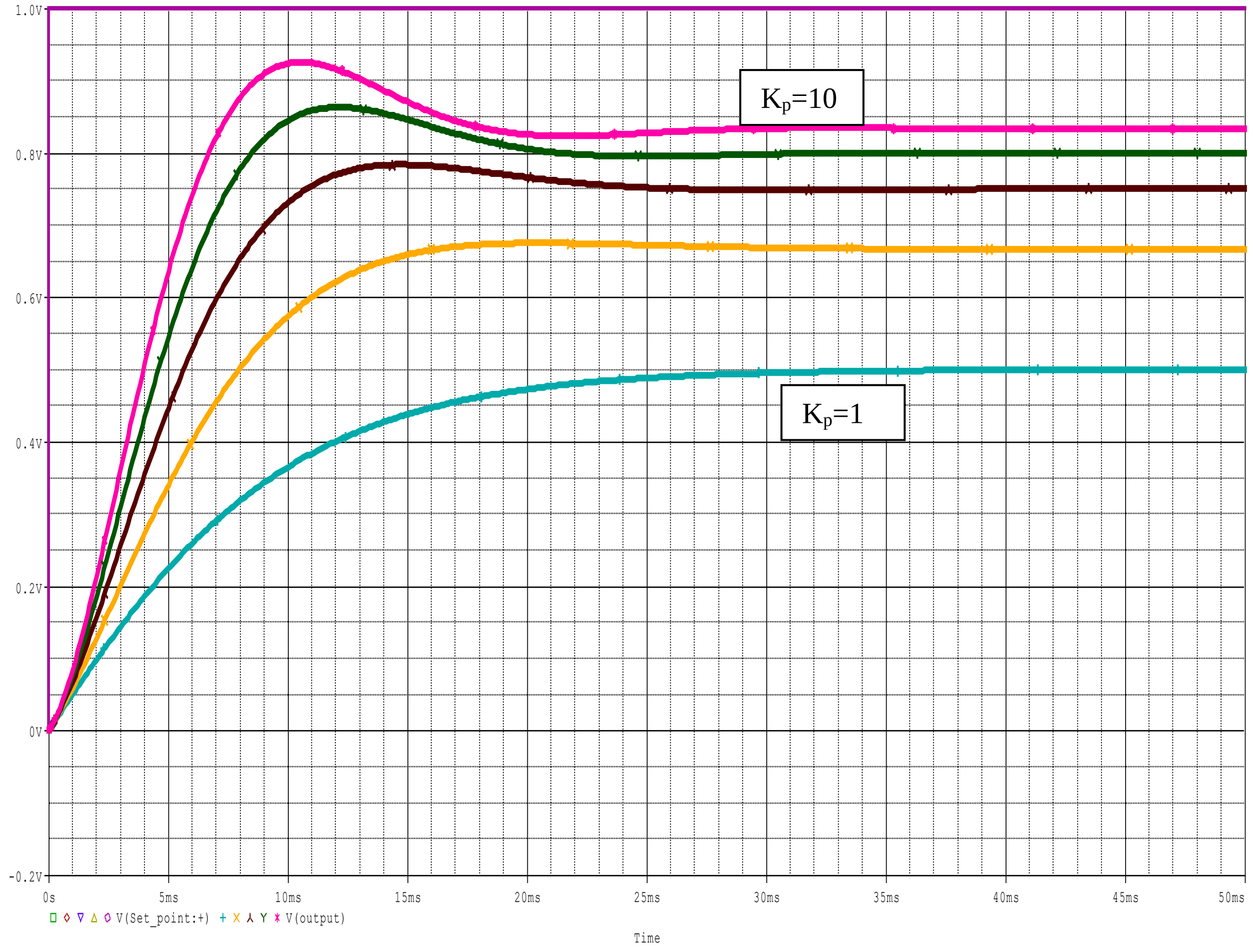
Effects of Proportional, Integral and Differential Block individually

Step Response without PID



Without any controller, the plant considered in this project has output response like the green curve for the step input or setpoint. It takes almost 150ms to follow the setpoint for the output signal. The rising time is pretty much higher as well. A PID controller with proper tuning can eliminate such discrepancies from the output, so that, the output follows the input fast without any distortion and overshoots or oscillations.

The effects of proportional gain solely

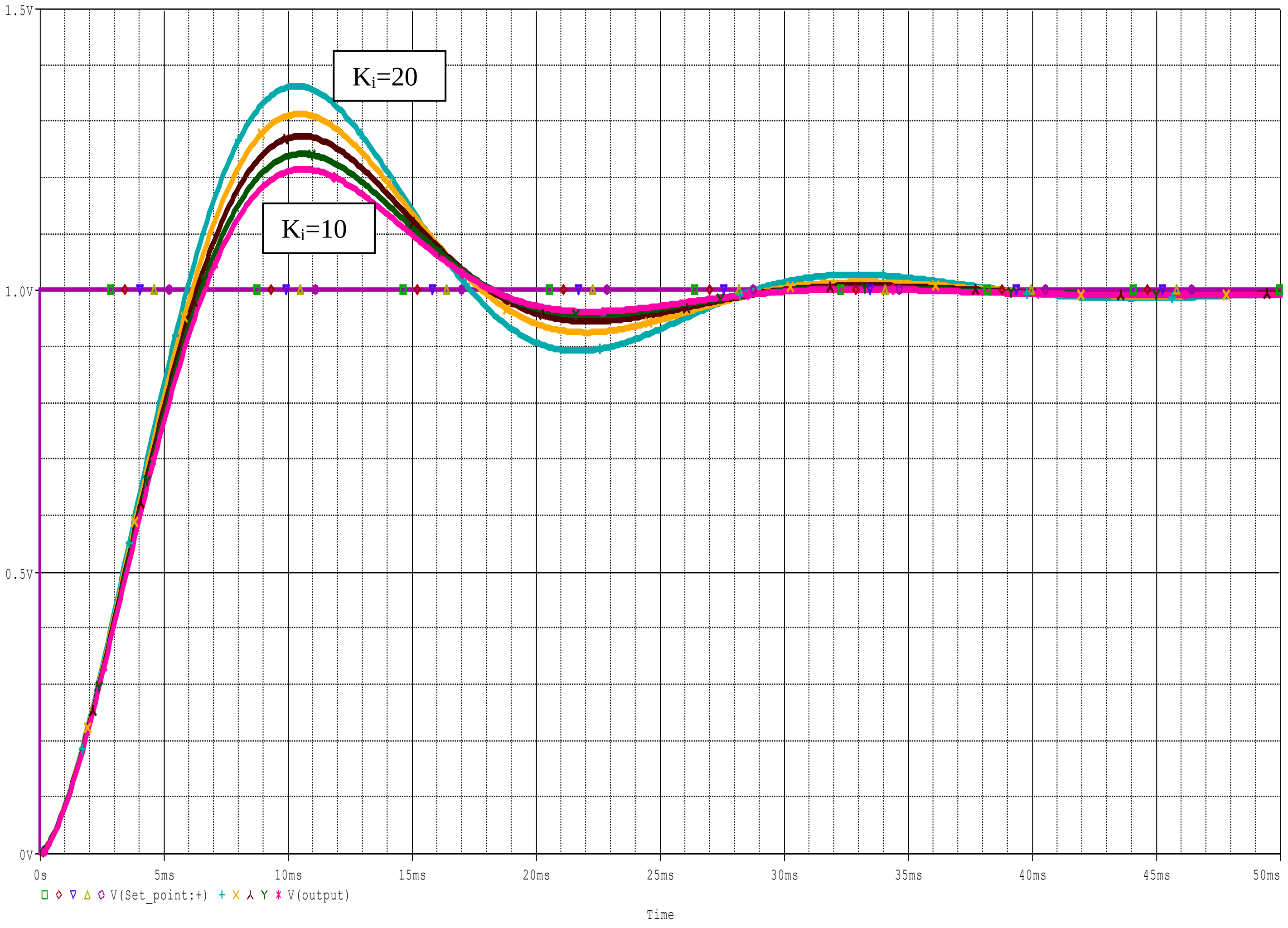


Observation:

The proportional block has been implemented by an inverting adder. By, performing a parametric sweep on the feedback resistor of this block, we got a family of curves. If we increase K_p from 1 to 10, varying the feedback resistor of the op amp of the proportional block of the PID controller, we can see

- The steady state error decreases
- The rise time decreases
- The overshoot increases
- The setting time increases

The effects of integral gain solely

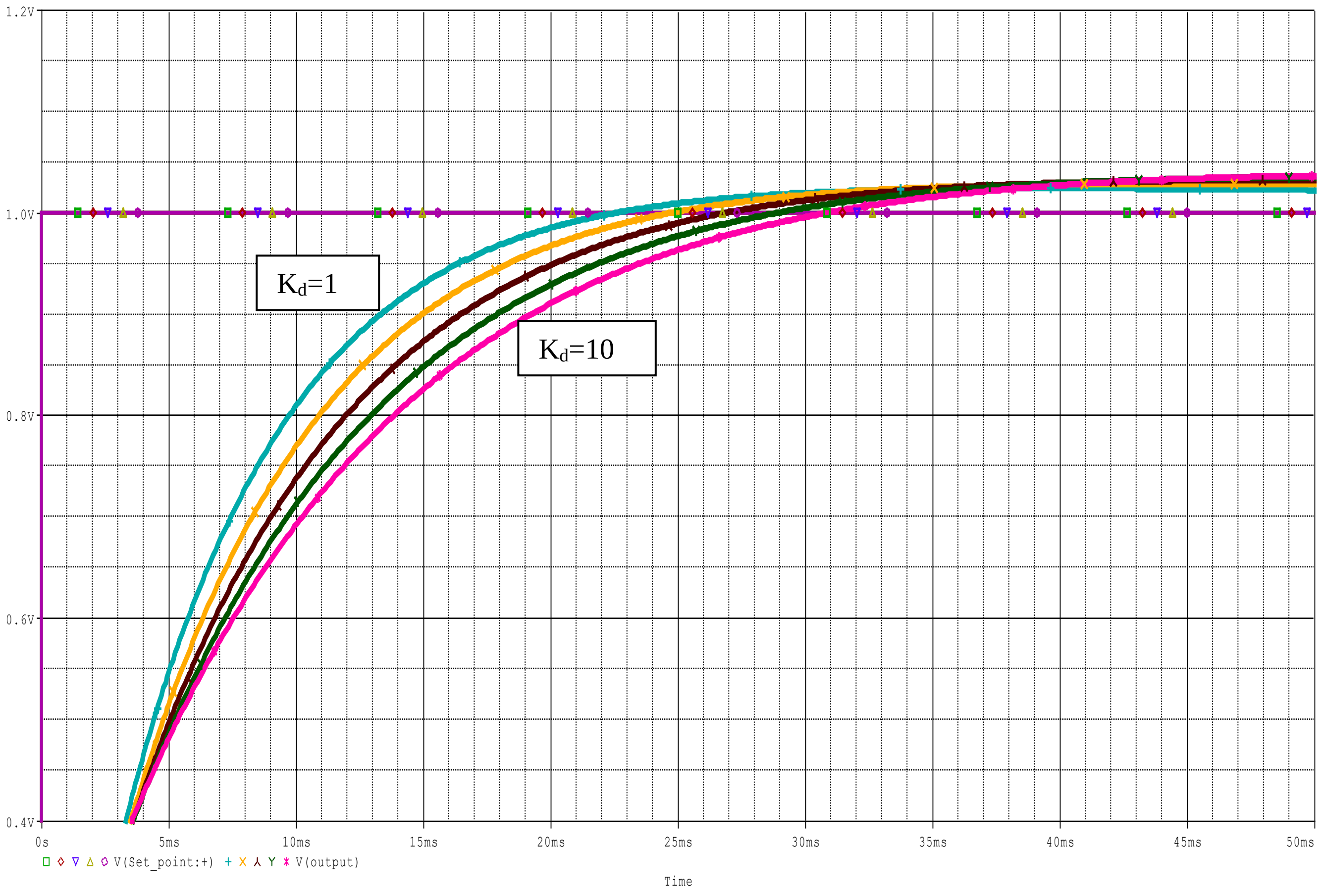


Observation:

The integral block has been designed by an operational amplifier, a capacitor and a resistor. By performing the parametric sweep on the resistor R5, we can control the integral gain K_i. If we increase KI from 10 to 20, by varying the resistance R5 and holding K_d and K_p constant, we can observe

- The rise time decreases
- The setting time increases
- The steady state error tends to 0
- The overshoot increases

The effects of differential gain solely



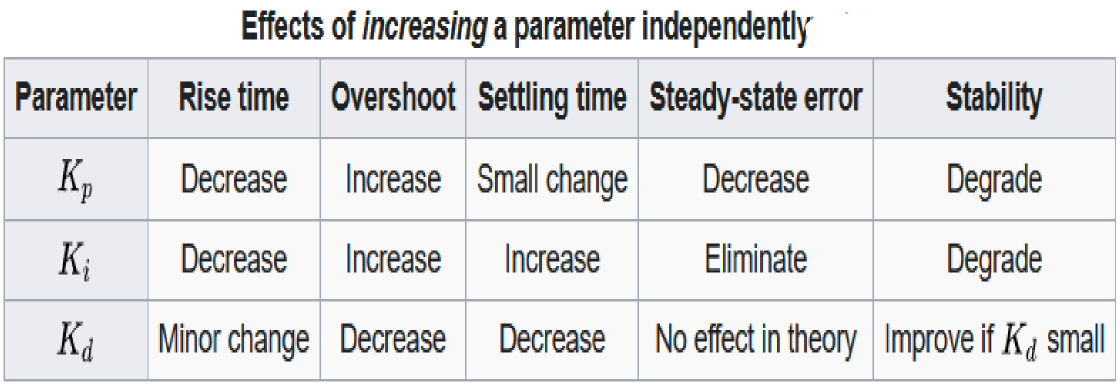
Observation:

The differential block has been designed by an operational amplifier, a capacitor and a resistor. A 200-ohm resistance has been added in series with the capacitor so that, at high frequencies, when the capacitor gets shorted, the differential block acts like an inverting amplifier or proportional block, thus, maintaining the stability of the controller.

If we increase the value of K_d, by varying the resistance R3 and holding K_i and k_p constant, we can observe

- The rise time increases
- The setting decreases
- The overshoot decreases (no overshoots)
- Steady state error remains unchanged

Summary:



Effects of *increasing* a parameter independently

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	Decrease	Increase	Small change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No effect in theory	Improve if K_d small

Manual Tuning:

Though there are numerous methods of tuning PID controller, we have employed the simplest of all tuning methods, that is, manual tuning. We just have to adjust the potentiometers to get the best results. No rigorous algorithm is needed. The manual tuning can be performed by three steps.

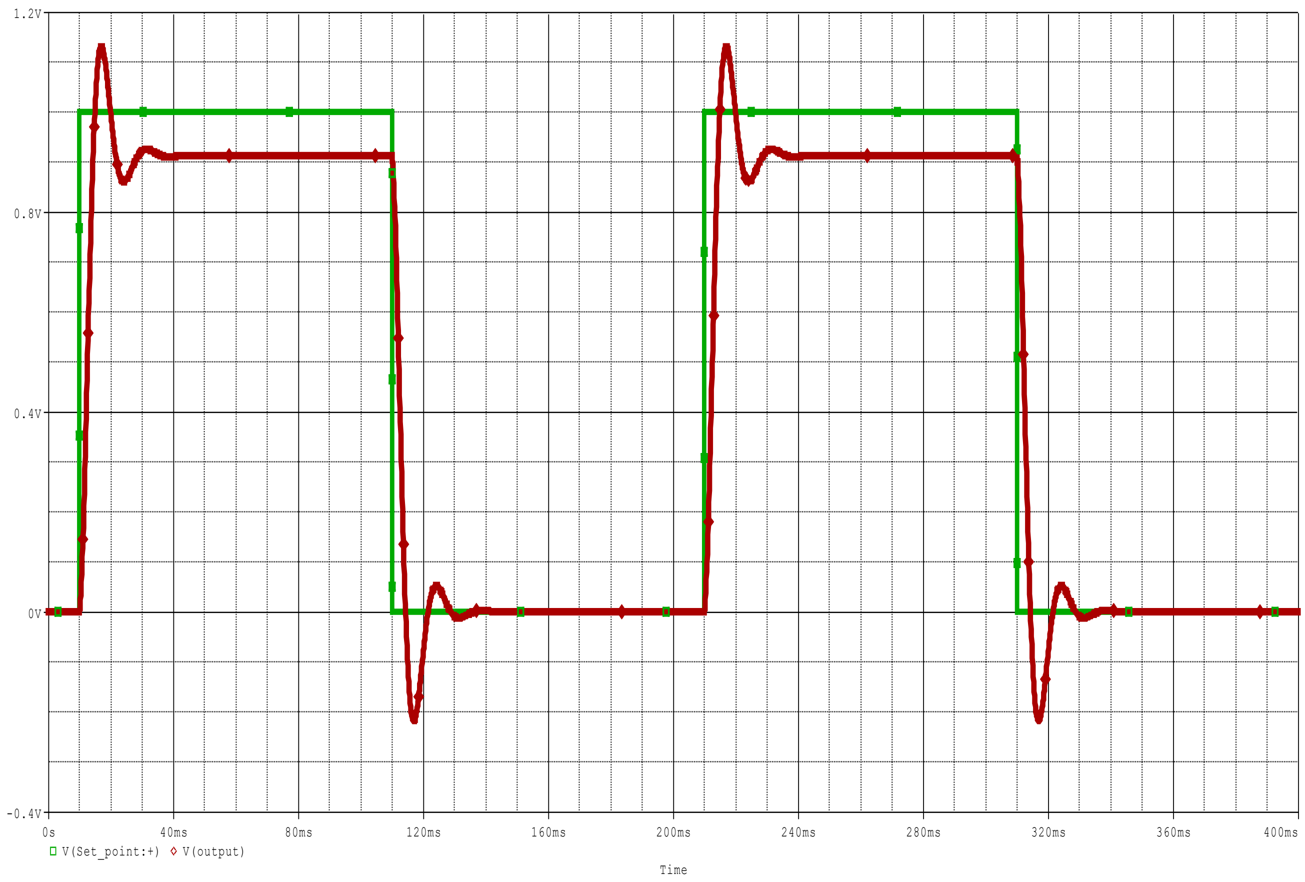
Step:1 varying K_p (Making K_d and K_i zero)
Step:2 varying K_i (Holding K_p constant and K_d = 0)
Step:3 varying K_d (Holding K_p and K_i constant)

Finally, we can get our desired output.

Each step has been described sequentially below.

Step:1 varying K_p (Making K_d and K_i zero)

The proportional gain K_p can be varied by adjusting the potentiometer R1. The resistors R7 and R8 needs to be set to 100T to block the output from the integral and differential block. The impact of tuning the proportional gain is evident in the graph given below.

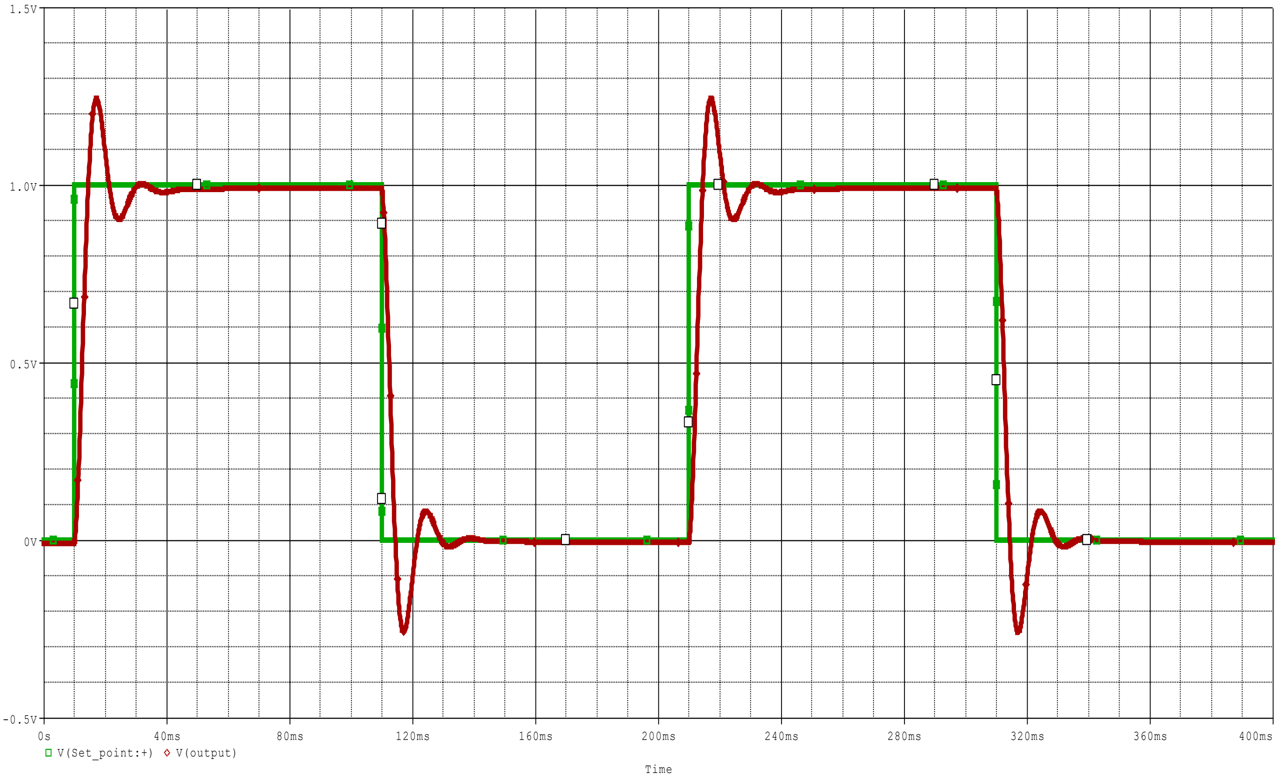


Observation:

By tuning the proportional gain, the steady state can not be avoided. However, the rising time has been decreased at an incredible scale, from 150ms to 15ms. There is an overshoot, resulted from the higher value of K_p. The steady state error in this step remains at 0.1V and the peak of the overshoot exceeds 1.1V. The proportional gain has a value of K_p=1 for the value of the potentiometer R1=1K. The settling time is 30ms.

Step:2 varying K_i (Holding K_p constant and $K_d = 0$)

The previous value of K_p =1 is kept constant in this step. The output of the derivative block has been stopped by taking R7=100T. The gain of the integral block has been varied by varying the potentiometer R5. After tuning, we get the following graph. The value of R5 is set to 100K.

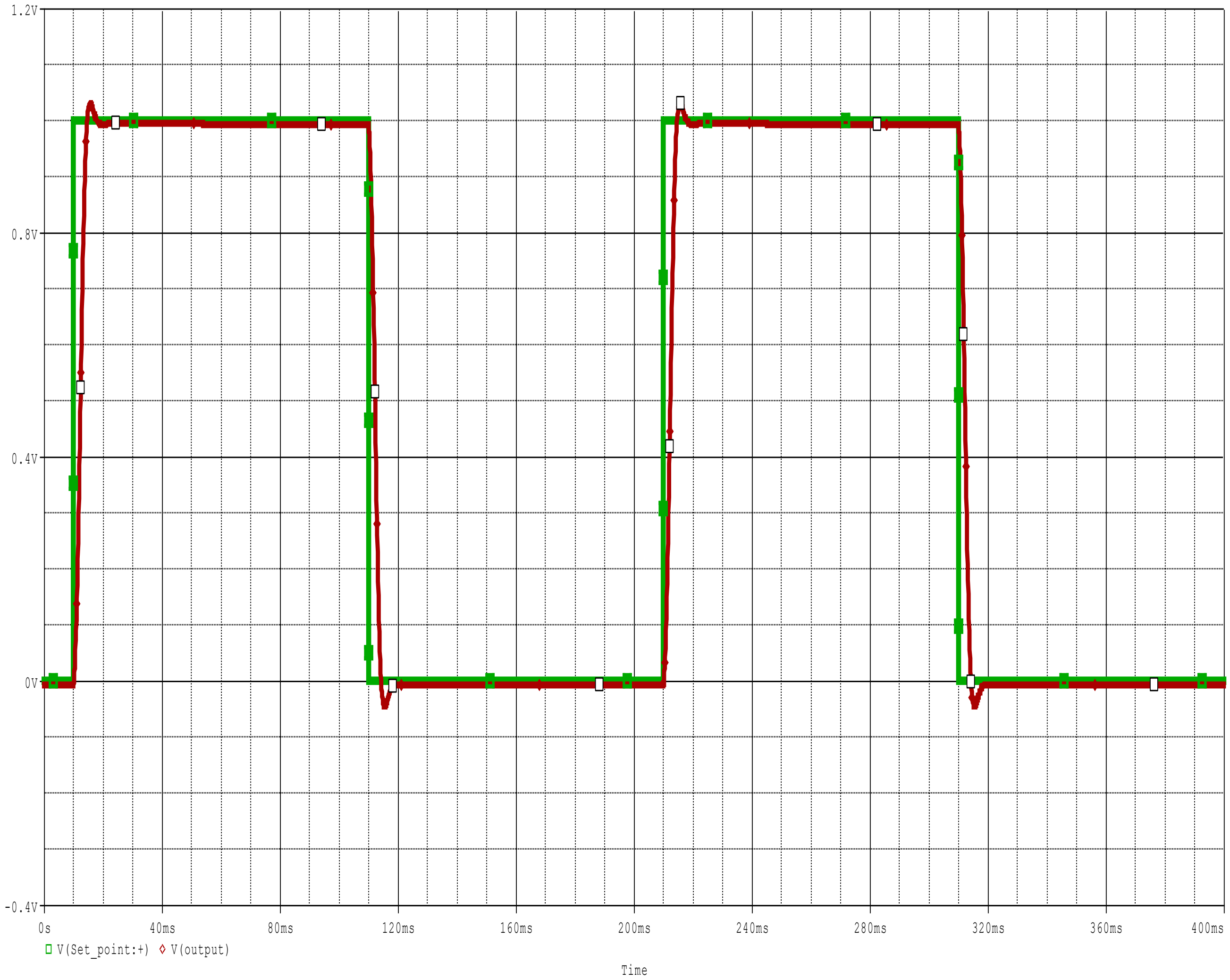


Observation:

By tuning the integral block of the PID controller, the steady state error has been completely made to 0. The rising time also decreases a bit, though not significant enough. However, the magnitude of overshoot is higher this time. It exceeds 1.2V which is quite dissatisfactory. The settling time has also been increased to 40ms from 30ms. This issue can be solved by the derivative block. The integral gain has been tuned to K_i= 10.

Step:3 varying K_d (Holding K_p and K_i constant)

The previous value of K_p=1 and K_i=10 has been kept constant. The gain of the derivative block has been varied by adjusting the resistor R3. After tuning, the value of R3 has been set to 10k. The final graph after tuning is given below.



Observation:

Finally, we get a satisfactory result. There is no steady state error. The overshot is minimized to a great extent. The rising time decreases to 5ms. The settling time has been reduced to 10ms. Thus, using a PID controller, we can get the desired output.

Limitations:

- Our project PID controller is basic implementation of the model of PID control system
- We manually tuned the controller for our created plant thus it will be not that well functional for other plants
- For complex real-life plants, it needs to be tuned using furthermore complex tuning techniques apart from manual tuning
- Our setpoint signal has no distortion so we didn't model the controller for unusual behavior of setpoints
- The controller can't be modeled instantaneously for a given gain, overshoot percentage and setting time. It has to be modeled manually for each given case
- Its performance is degraded in a non-linear and asymmetric system

Future Work:

Using complex tuning method, we can improve the efficiency of the controller. Thus, we can fit it to various complex plants and get the highest amount of efficiency. We can also make it work in such a way so that we can control the gain, overshoot and setting time according to our need. PID controller has paved the way for us to get to know control theory which will help us getting to know further type of controllers. We are yet to tackle real life problems with our modeled PID controller. Trying it in the real-life examples will introduce us to the scopes of improvement.

Conclusion:

Here in this project, we tried to implement the basic model of a PID controller. We had a fairly good result. Our controller managed to reduce the delay that we forcefully implemented very well. This project helped us to understand the basic principle of control loop feedback mechanism. We thus with the help of our project vision to understand more in the field and work with more techniques in the future.