PID CONTROLLER

A thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF SCIENCE

by

SHRIMAN KESHRI

to the

School of Physical Sciences

National Institute of Science Education and Research

Bhubaneswar

Date October 10, 2021

ACKNOWLEDGEMENTS

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.

ABSTRACT

PID controllers are being widely used in industry due to their well-grounded established theory, simplicity, maintenance requirements, and ease of retuning online. In the past four decades, there are numerous papers dealing with tuning of PID controller. Designing a PID controller to meet gain and phase margin specification is a well-known design technique. If the gain and phase margin are not specified carefully then the design may not be optimum in the sense that could be large phase margin (more robust) that could give better performance. This paper studies the relationship between ISE performance index, gain margin, phase margin and compares two tuning technique, based on these three parameters. These tuning techniques are particularly useful in the context of adaptive control and auto-tuning, where the control parameters have to be calculated on-line.

In the first part, basics of various controllers, their working and importance of PID controller in reference to a practical system (thermal control system) is discussed.

In the latter part of the work, exhaustive study has been done on two different PID controller tuning techniques. A compromise between robustness and tracking performance of the system in presence of time delay is tried to achieve. Results of simulation, graph, plots, indicate the validity of the study.

Contents

List of Figures

Chapter 1

Introduction

In recent years, the control system has assumed an increasingly important role in developing and advancing modern civilization and technology. Practically every aspect of our day-to-day activities is affected by some control systems. Automatic control systems are abundant in all industry sectors, such as quality control of manufactured products, automated assembly lines, machine-tool control, space technology, weapon systems, computer control, transportation systems, power systems, robotics, etc. It is essential in such industrial operations as controlling pressure, temperature, humidity, and flow in the process industries.

Recent application of modern control theory includes such non-engineering systems as biological, biomedical, control of inventory, economic and socio-economic systems.

The basic ingredients of a control system can be described by:

- Objectives of control.
- Control system components.
- Results or output.

1.1 Automatic Controllers

An automatic controller is used to compare the actual value of plant result with reference command, determines the difference, and produces a control signal that will

reduce this difference to a negligible value. How the automatic controller has such a control signal is called the control action.

An industrial control system comprises an automatic controller, an actuator, a plant, and a sensor (measuring element). The controller detects the actuating error command, usually at a shallow power level, and amplifies it to a very high level. The output of the automatic controller is fed to an actuator, such as a hydraulic motor, an electric motor, or a pneumatic motor or valve (or any other sources of energy). The actuator is a power device that produces input to the plant according to the control signal so that the output signal will point to the reference input signal.

The sensor or the measuring element is a device that converts the output variable into another optimum variable, such as a displacement, pressure, or voltage, that can compare the output to the reference input command. This element is in a feedback path of the closed-loop system. The setpoint controller must be converted to reference input with the same unit as the feedback signal from the sensor element.

1.2 Classification of Industrial controllers

Industrial controllers may be classified according to their control action as:

- Two-position or on-off controllers
- Proportional controllers
- Integral controllers
- Proportional-plus-integral controllers
- Proportional-plus-derivative controllers
- Proportional-plus-integral-plus-derivative controllers

The type of controller to use must be decided depending upon the nature of the plant and the operating condition, including such consideration as safety, cost, availability, reliability, accuracy, weight, and size.

Two-position or on-off controllers:-

In a two-position control system, the actuating part has only two fixed positions, which are, in many simple cases, simply on and off. Due to its simplicity and inexpensiveness, it is very widely used in both industrial and domestic control system.

Let the output signal from the controller be u(t) and the actuating error signal be e(t). Then mathematically,

$$u(t) = U_1, \text{ for } e(t) > 0$$

$$= U_2, \text{ for } e(t) < 0$$

Where U_1 and U_2 are constants, and the minimum value of U_2 is usually either zero or - U1.

1.3 Proportional Control

A proportional control system is a type of linear feedback control system. Proportional control is how most drivers control the speed of a car. If the car is at target speed and the speed increases slightly, the power is reduced slightly, or in proportion to the error (the actual versus target speed), so that the car reduces speed gradually and reaches the target point with very little, if any, "overshoot," so the result is much smoother control than on-off control. In the proportional control algorithm, the controller output is proportional to the error signal, which is the difference between the setpoint and the process variable. In other words, the output of a proportional controller is the multiplication product of the error signal and the proportional gain. This can be mathematically expressed as

$$\mathrm{P_{out}} = \mathrm{K_p e(t)}$$

Where

P_{out} : Output of the proportional controller

K_p : Proportional gain

$e(t)$: Instantaneous process error at time ' t'.$e(t) = SP - PV$

SP : Set point

PV : Process variable

With increase in $\mathrm{K_p}$:

- Response speed of the system increases.
- Overshoot of the closed-loop system increases.
- Steady-state error decreases.

But with a high K_p value, the closed-loop system becomes unstable.

1.4 Integral Control

In a proportional control of a plant whose transfer function doesn't possess an integrator 1/s, there is a steady-state error or offset in response to a step input. Such an offset can be eliminated if the integral controller is included in the system.

In the integral control of a plant, the control signal, the output signal from the controller, at any instant, is the area under the actuating error signal curve up to that instant. But while removing the steady-state error, it may lead to an oscillatory response of slowly decreasing amplitude or even increasing amplitude, both of which is usually undesirable [5].

1.5 Proportional-plus-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 setpoint) 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

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

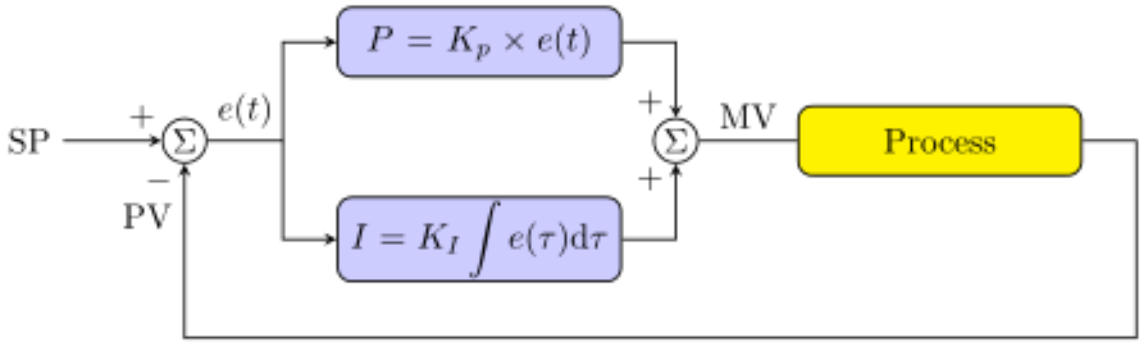


Figure 1.1: Basic block of PI Controller

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 K_p since roots of the characteristic eqn. It may have a 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.

1.6 Proportional-plus-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.

$$G_{pd}(s) = K_p + K_d s \text{ or}$$

$$= K_p (1 + T_d s)$$

With the increase of T_d

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

1.7 Proportional-plus-integral-plus-derivative controllers

The PID controller was first placed on the market in 1939 and has remained the most widely used controller in process control until today. An investigation performed in 1989 in Japan indicated that more than 90% of the controllers used in process indus-

tries are PID controllers and advanced versions of the PID controller. PI controllers are fairly common since derivative action is sensitive to measurement noise

"PID control" is the feedback control method that uses the PID controller as the main tool. The basic structure of conventional feedback control systems is shown below, using a block diagram representation. In this figure, the process is the object to be controlled. The purpose of control is to make the process variable y follow the setpoint value r. To achieve this purpose, the manipulated variable u is changed at the command of the controller. As an example of processes, consider a heating tank where some liquid is heated to the desired temperature by burning fuel gas. The process variable y is the temperature of the liquid, and the manipulated variable u is the flow of the fuel gas. The "disturbance" is any factor other than the manipulated variable that influences the process variable. The figure below assumes that only one disturbance is added to the manipulated variable. However, in some applications, a major disturbance enters the process differently, or plural disturbances need to be considered. The error e is defined by $e = r - y$. The compensator $C(s)$ is the computational rule that determines the manipulated variable u based on its input data, which is the error e in the case of the figure. The last thing to notice about the figure is that the process variable y is assumed to be measured by the detector, which is not shown explicitly here, with sufficient accuracy instantaneously that the input to the controller can be regarded as being exactly equal to y.

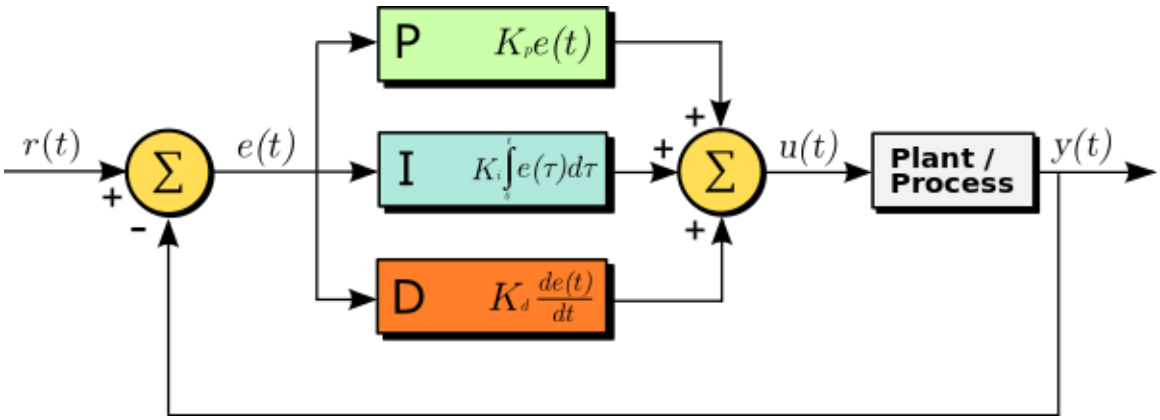


Figure 1.2: courtesy

When used in this manner, the three-element of PID produces outputs with the following nature:

- *P* element: proportional to the error at the instant t, this is the "present" error.
- *I* element: proportional to the integral of the error up to the instant t, which can be interpreted as the accumulation of the "past" error.
- *D* element: proportional to the derivative of the error at the instant t, which can be interpreted as predicting the "future" error.

Thus, the PID controller can be understood as a controller that considers the present, the past, and the future of the error. The transfer function $\mathrm{G_c}(s)$ of the PID controller is :

$$\begin{aligned} \mathrm{G_c(s)} &= \mathrm{K_p}\left(1 + \frac{1}{\mathrm{ST_i}} + \mathrm{T_d s}\right) \\ &= \mathrm{K_p} + \frac{\mathrm{K_i}}{\mathrm{S}} + \mathrm{K_d s} \end{aligned}$$

1.8 Application

In the early history of automatic process control, the PID controller was implemented as a mechanical device. These mechanical controllers used a lever, spring, and mass

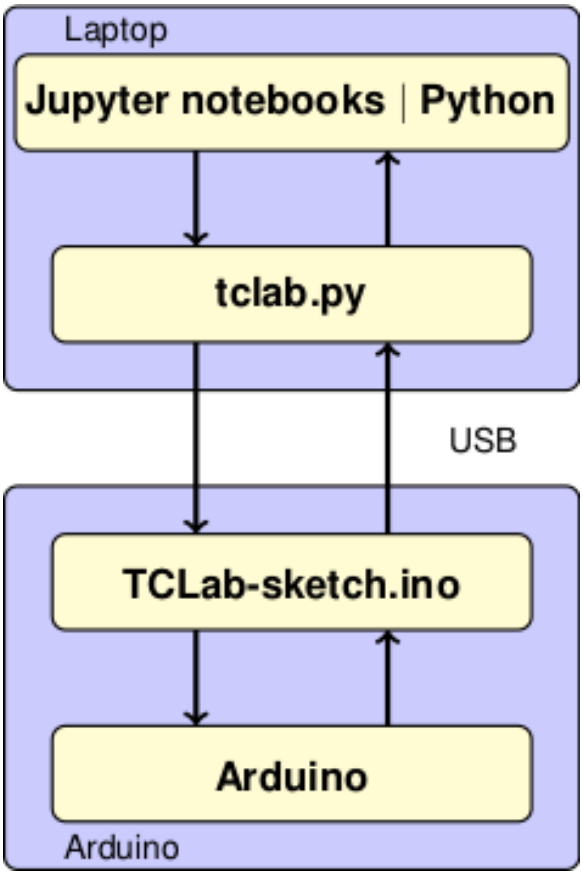
and were often energized by compressed air. These pneumatic controllers were once the industry standard . Electronic analog controllers can be made from a solid-state or tube amplifier, a capacitor, and a resistance. Electronic analog PID control loops were often found within more complex electronic systems, for example, the head positioning of a disk drive, the power conditioning of a power supply, or even the movement-detection circuit of a modern seismometer. Nowadays, electronic controllers have largely been replaced by digital controllers implemented with microcontrollers or FPGAs. Most modern PID controllers in the industry are implemented in programmable logic controllers (PLCs) or as a panel-mounted digital controller. Software implementations have the advantages that they are relatively cheap and are flexible with respect to the implementation of the PID algorithm [5]

Chapter 2

PID controller using python

2.1 package: TCLab

The Arduino Temperature Control Lab is a modular, portable, and inexpensive solution for hands-on process control learning. Heat output is adjusted by modulating current flow to each of two transistors. Thermistors measure the temperatures. Energy from the transistor output is transferred by conduction and convection to the temperature sensor. The dynamics of heat transfer provide rich opportunities to implement single and multivariable control systems. The lab is integrated into a small PCB shield which can be mounted to any Arduino or Arduino compatible microcontroller.



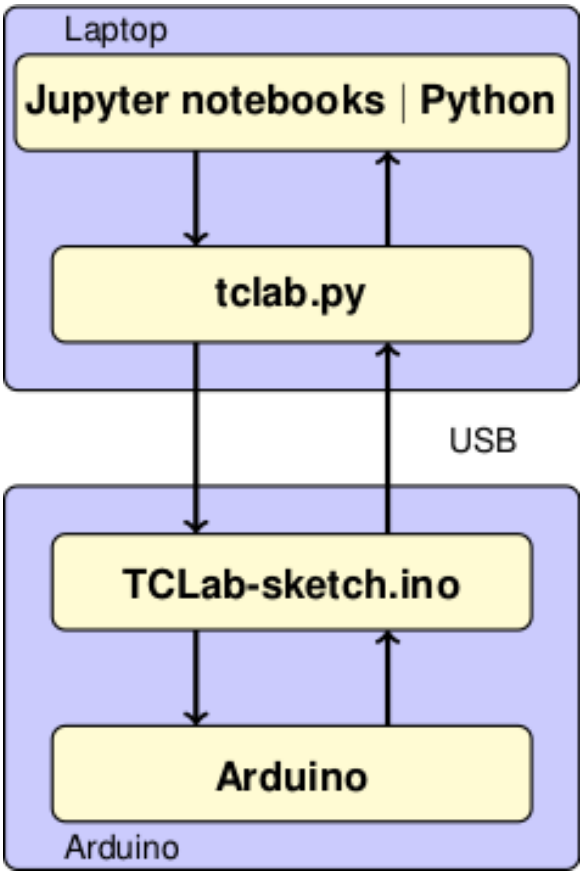
2.1.1 TCLab Overview

The tclab package provides a set of Python tools for interfacing with the BYU Temperature Control Laboratory. The Temperature Control Laboratory consists of two

heaters and two temperature sensors mounted on an Arduino microcontroller board. Together, the tclab package and the Temperature Control Laboratory provide a low-cost experimental platform for implementing algorithms commonly used for process control.

2.1.2 TCLab Architecture

The tclab package is intended to be used as a teaching tool. The package provides high-level access to sensors, heaters, a pseudo-realtime clock. The package includes the following Python classes and functions:



- TCLab() : providing access to the Temperature Control Laboratory hardware.
- TCLabModel() : providing access to a simulation of the Temperature Control Laboratory hardware.
- clock : for synchronizing with a real time clock.
- Historian : for data logging.
- Plotter : for realtime plotting.

Using these Python tools, students can create Jupyter notebooks and python codes covering a wide range of topics in process control.

tclab.py: A Python package providing high-level access to sensors, heaters, a pseudo-realtime clock. The package includes TCLab() providing access to the device, clock for synchronizing with a real time clock, Historian for data logging, and Plotter for realtime plotting.

TCLab-sketch.ino: Firmware for the intrisically safe operation of the Arduino board and shield. The sketch is available at https://github.com/jckantor/TCLab-sketch.

Arduino: Hardware platform for the Temperature Control Laboratory. TCLab is compatiable with Arduino Uno, Arduino Leonardo, and compatible clones.

2.2 code

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
```

```
                    else:
                        M = np.concatenate((M, VT), 1)

                    VT = np.zeros((n*m,1), int)

        return M
```

```
    def PID(Kp, Ki, Kd, MV_bar=0):
    # initialize stored data
    e_prev = 0
    t_prev = -100
    I = 0

    # initial control
    MV = MV_bar

    while True:
        # yield MV, wait for new t, PV, SP
        t, PV, SP = yield MV

        # PID calculations
        e = SP - PV

        P = Kp*e
        I = I + Ki*e*(t - t_prev)
        D = Kd*(e - e_prev)/(t - t_prev)

        MV = MV_bar + P + I + D

        # update stored data for next iteration
        e_prev = e
        t_prev = t
```

```
    %matplotlib inline
    from tclab import clock, setup, Historian, Plotter

    TCLab = setup(connected=False, speedup=10)

```

```
    controller = PID(2, 0.1, 2)         # create pid control
    controller.send(None)               # initialize

    tfinal = 800

    with TCLab() as lab:
        h = Historian([
            ('SP', lambda: SP),
            ('T1', lambda: lab.T1),
            ('MV', lambda: MV),
            ('Q1', lab.Q1)])
        p = Plotter(h, tfinal)
        T1 = lab.T1
        for t in clock(tfinal, 2):
            SP = T1 if t < 50 else 50           # get setpoint
            PV = lab.T1                         # get measurement
            MV = controller.send([t, PV, SP])   # compute manipulated variable
            lab.U1 = MV                         # apply
            p.update(t)                         # update information display
```

2.3 Observation

Seting the values of $K_p = 2$, $K_i = 0.1$ $K_d = 2$ we get the plot.

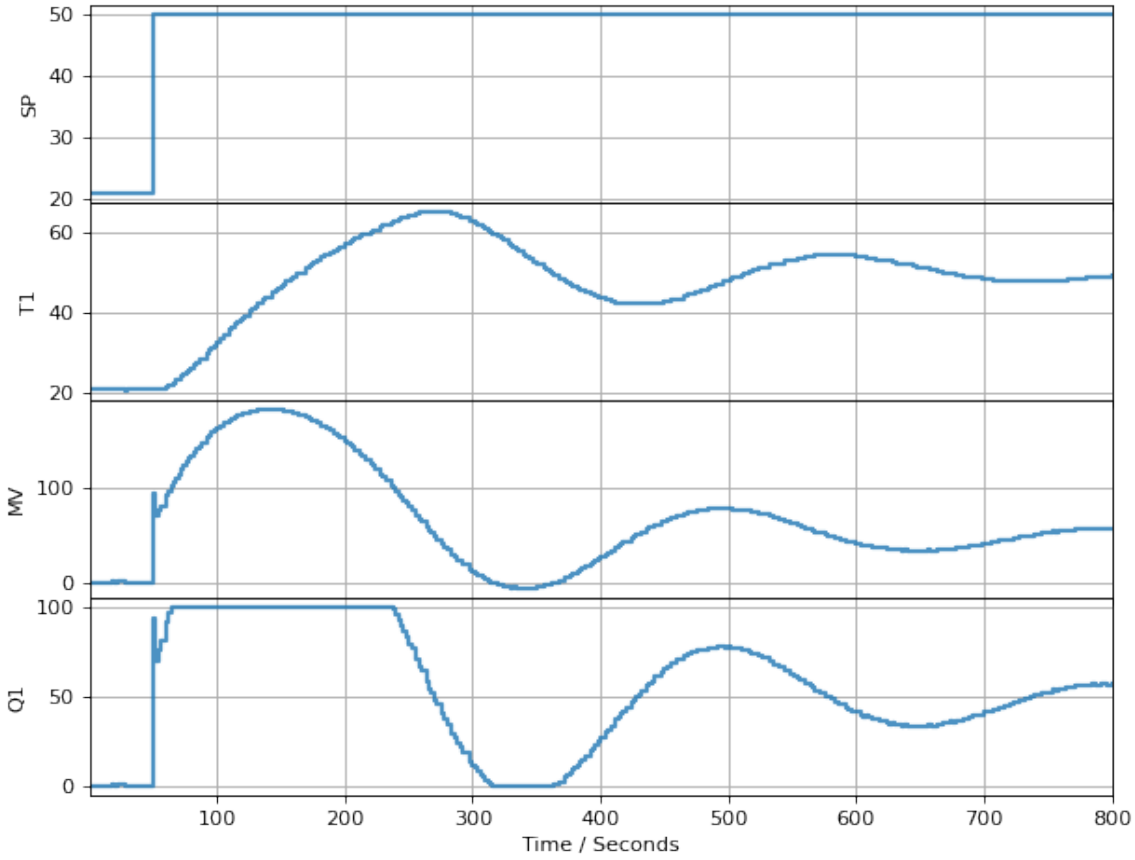


Figure 2.1: PID(2,0.1,2)
SP Target temperature. **T1** Current temperature. **MV** Calculated power by PID. **Q1** Power on the heater coil.

Seting the values of $K_p = 0.2$, $K_i = 0.1$ $K_d = 2$ we get the plot.

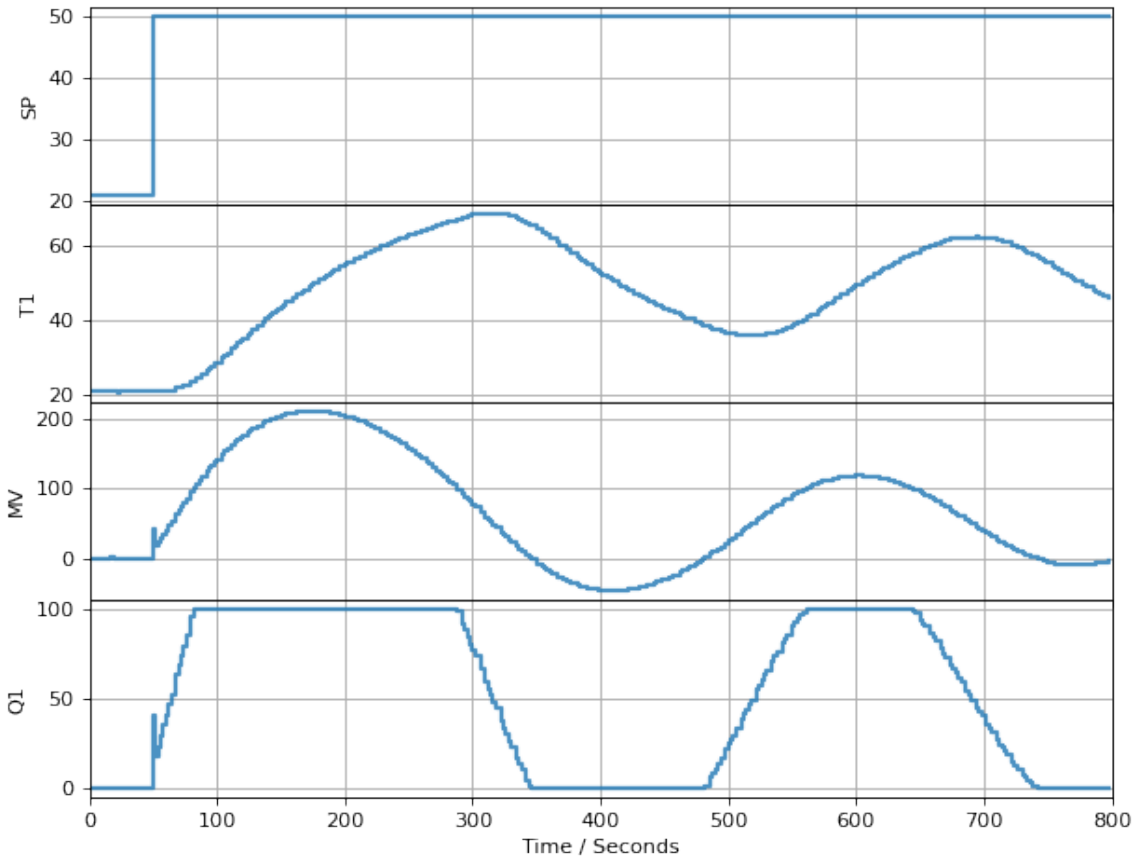


Figure 2.2: PID(0.2,0.1,2)
SP Target temperature. **T1** Current temperature. **MV** Calculated power by PID. **Q1** Power on the heater coil.

Seting the values of $K_p = 2$, $K_i = 2$ $K_d = 2$ we get the plot.

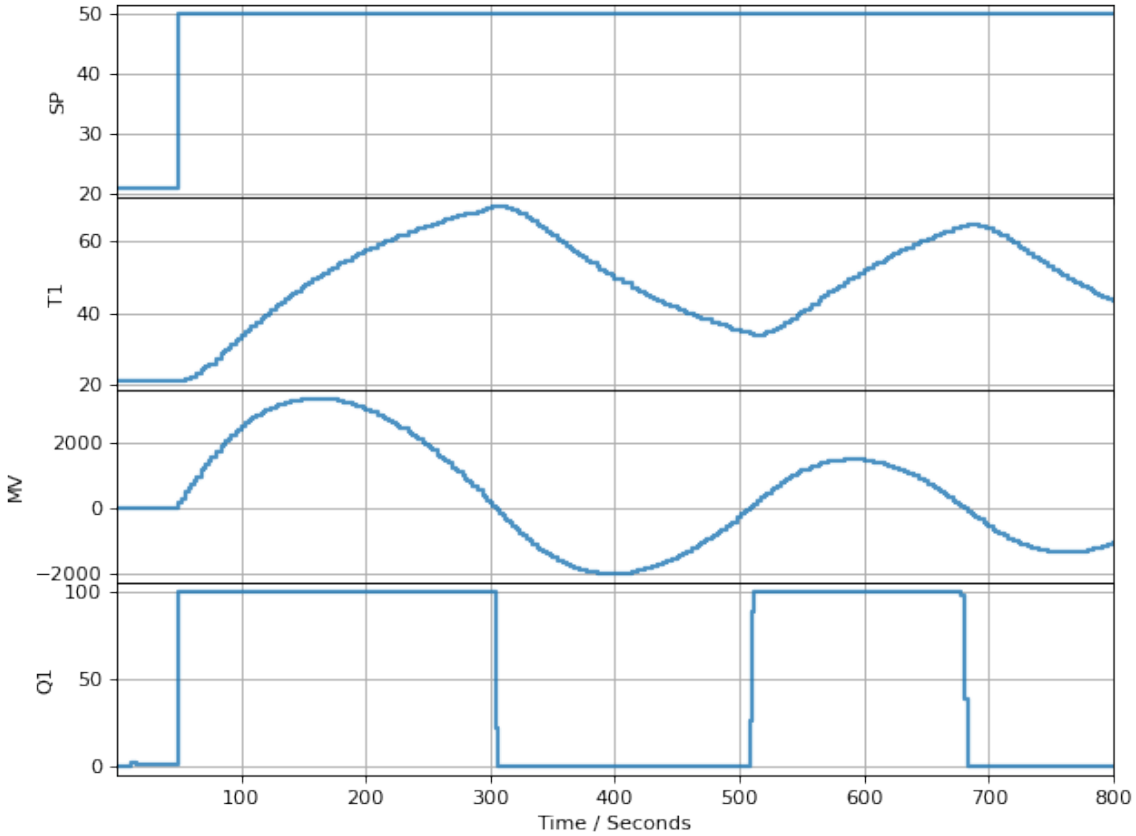


Figure 2.3: PID(2,2,2)
SP Target temperature. **T1** Current temperature. **MV** Calculated power by PID. **Q1** Power on the heater coil.

Seting the values of $K_p = 2$, $K_i = 20$ $K_d = 2$ we get the plot.

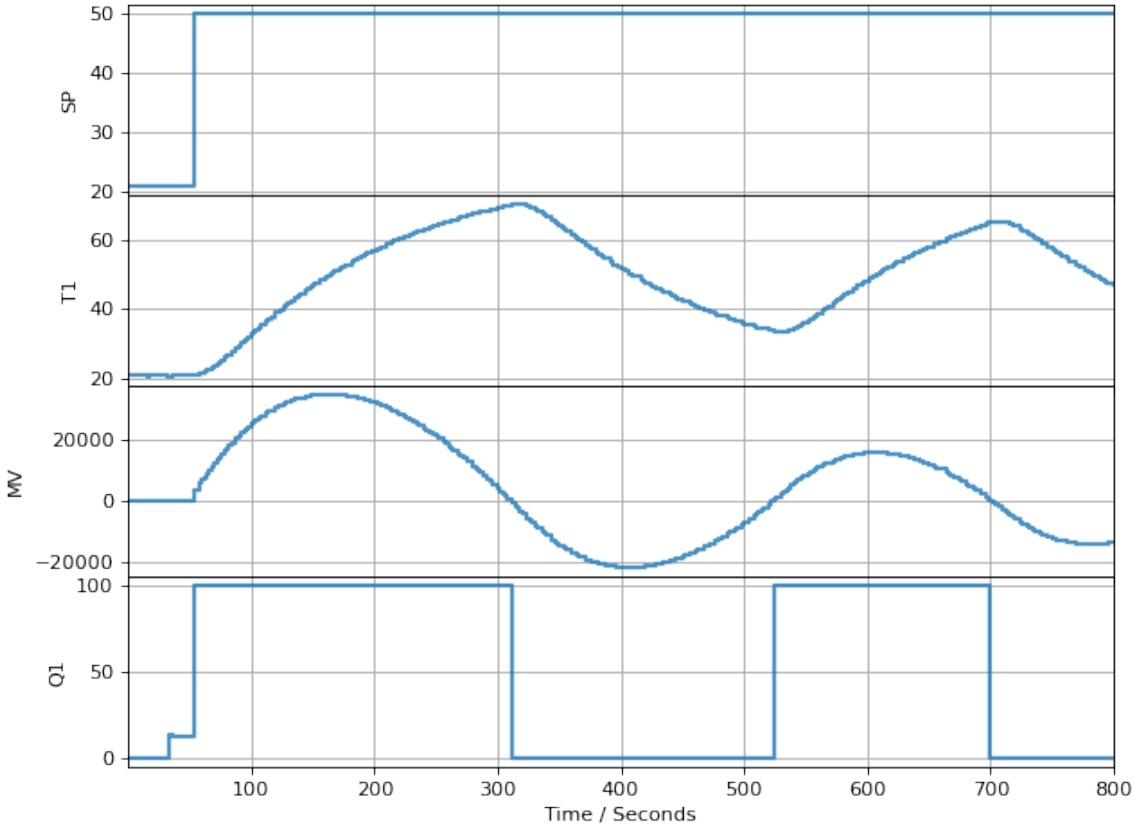


Figure 2.4: PID(2,20,2)
SP Target temperature. **T1** Current temperature. **MV** Calculated power by PID. **Q1** Power on the heater coil.

Seting the values of $K_p = 10$, $K_i = 0.1$ $K_d = 2$ we get the plot.

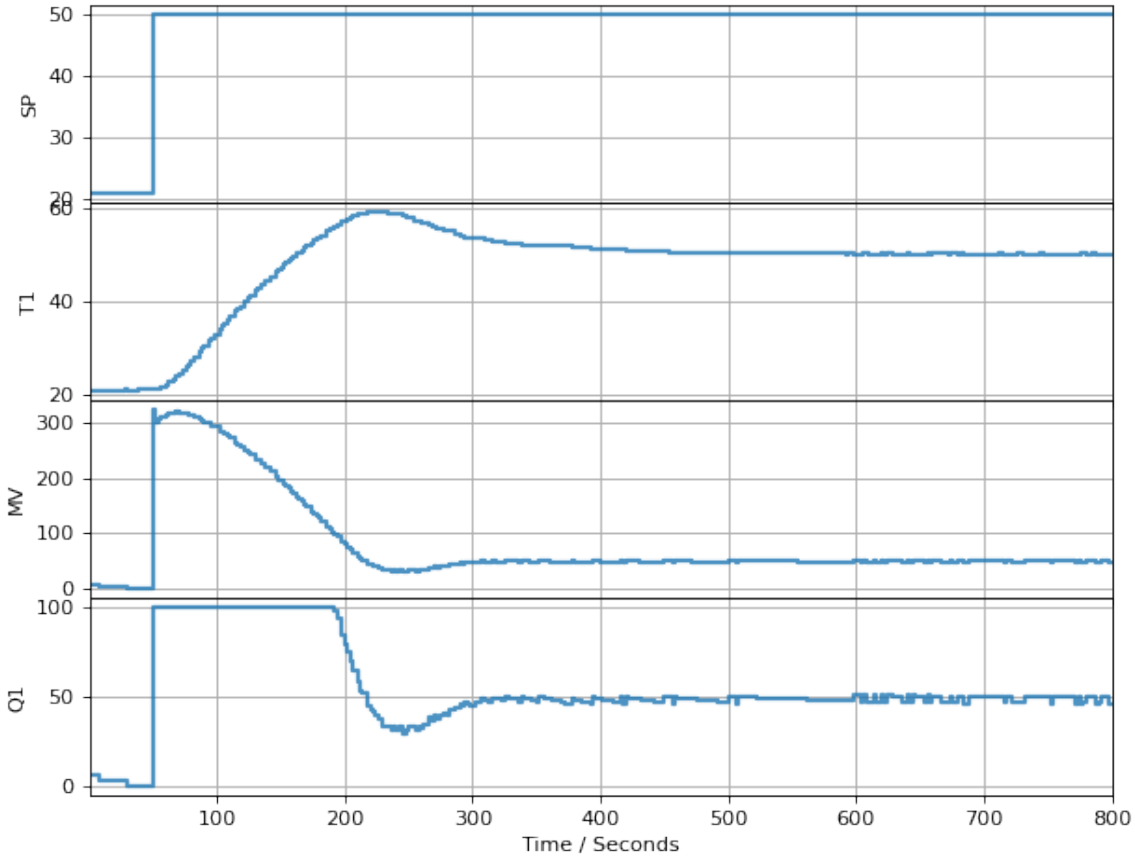


Figure 2.5: PID(10,0.1,2)
SP Target temperature. **T1** Current temperature. **MV** Calculated power by PID. **Q1** Power on the heater coil.

Seting the values of $K_p = 20$, $K_i = 0.1$ $K_d = 2$ we get the plot.

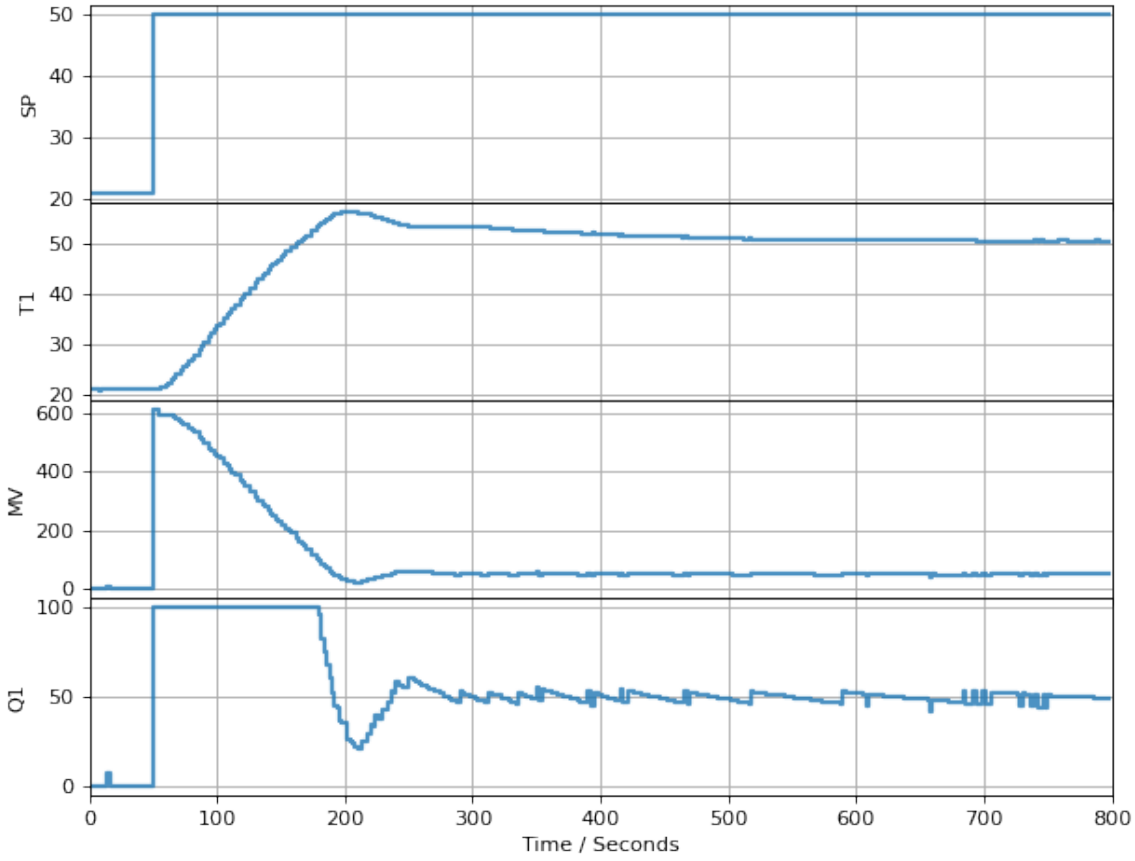


Figure 2.6: PID(20,0.1,2)
SP Target temperature. **T1** Current temperature. **MV** Calculated power by PID. **Q1** Power on the heater coil.

Chapter 3

PID Controller using Multisim

3.1 Multisim

NI Multisim (formerly MultiSIM) is an electronic schematic capture and simulation program which is part of a suite of circuit design programs, along with NI Ultiboard. Multisim is one of the few circuit design programs to employ the original Berkeley SPICE based software simulation. Multisim was originally created by a company named Electronics Workbench Group, which is now a division of National Instruments. Multisim includes microcontroller simulation (formerly known as MultiMCU), as well as integrated import and export features to the printed circuit board layout software in the suite, NI Ultiboard.

Multisim is widely used in academia and industry for circuits education, electronic schematic design and SPICE simulation.

Multisim has desktop-based and browser-based editing environment. I used the browser-based editing environment to build the PID controller.

3.2 Circuit design

In this simulation we used 6 OPAMP. Three are used for integration differentiation and proportional amplification. One is used to calculate the difference between current value and the target value. Using one OPAMP wish you let a heater(this is a OPAMP as an amplifier.) At last we used one OPAMP for amplifying the sumed signal from all the integration differentiation and proportional circuit.

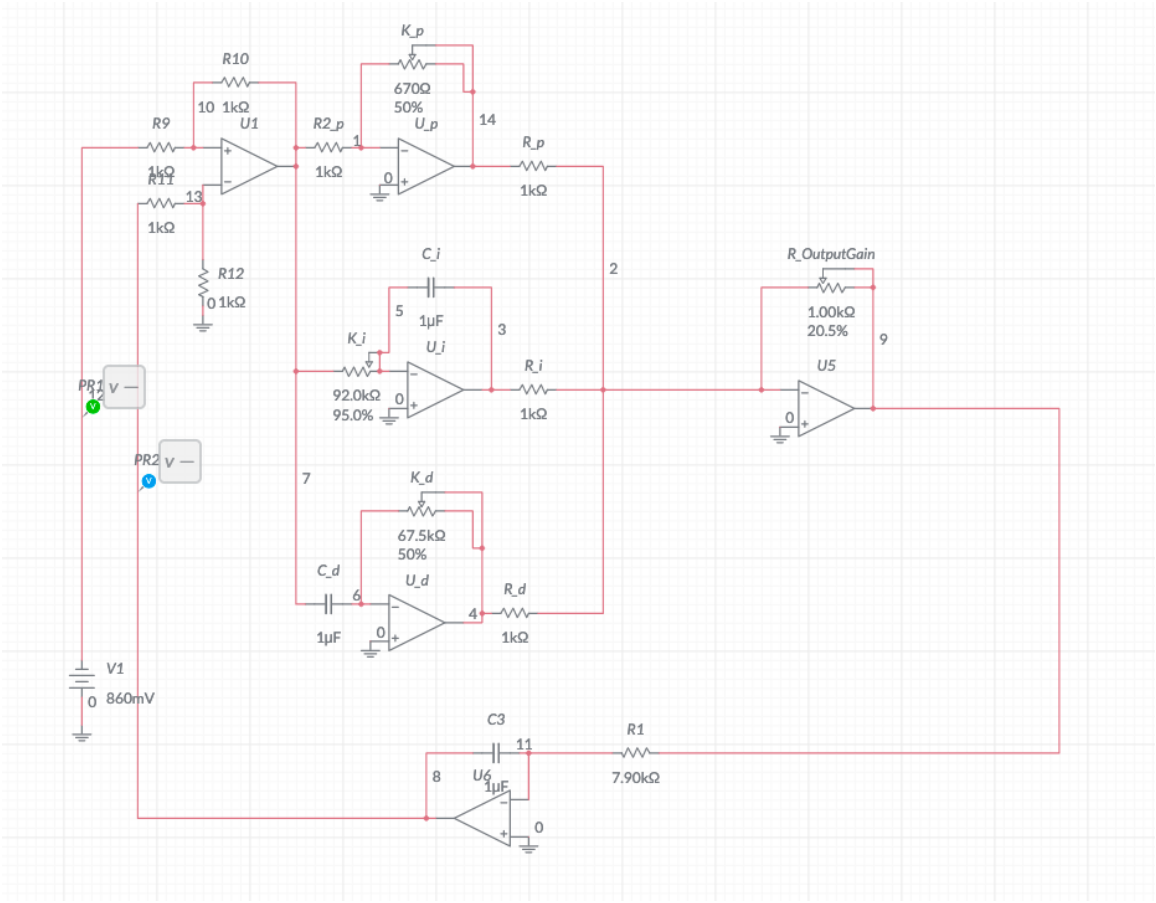


Figure 3.1: Circuit diagram drawn in multisim

3.2.1 Difference amplifier

the 0PAMP we used for this is labled U1 and has resistance $R_9 = 1K\Omega$ $R_10 = 1K\Omega$. There for its amplification is one, in other words it's just calculating the difference between the signals. The signals we are feeding in this difference in the fire is the target signal and the current value. The output of this is going in integration circuit, differentiation circuit and proportional circuit.

3.2.2 Integration amplifier

The OPAMP we used for this is labled with U_i and has resistance $K_i = 92K\Omega$ and camapciter $C_i = 1\mu F$ it has amplification of $K_i = 1.08E1$.

3.2.3 Differentiation amplifier

The OPAMP we used for this is labled with U_d and has resistance $K_d = 67.5K\Omega$ and camapciter $C_i = 1\mu F$ it has amplification of $K_i = -6.75E - 2$.

3.2.4 Proportional amplifier

The OPAMP we used for this is labled with U_p and has resistance $K_p = 670K\Omega$ and $R_p = 1K\Omega$ it has amplification of $K_p = 0.670$.

3.2.5 Amplification over sum

The OPAMP we used for this is labled with U_5 and has resistance $R_{OutputGain} = 1.0K\Omega$ and $R = 1K\Omega$ it has amplification of one.

3.2.6 Heater circuit simulation

The OPAMP we used for this is labled with U_6 and has resistance $R_1 = 7.9K\Omega$ and camapciter $C_i = 1\mu F$ it has amplification of $K_i = 1.26E2$.

3.3 Op-amp as Integrator

Operational amplifiers can be used for mathematical applications such as Integration and Differentiation by implementing specific op-amp configurations.

When the feedback path is made through a capacitor instead of a resistance , an RC Network has been established across the operational amplifiers' negative feedback path. This kind of circuit configuration producing helps in implementing mathematical operation, specifically integration, and this operational amplifier circuit is known as an Operational amplifier Integrator circuit. The output of the circuit is the integration of the applied input voltage with time.

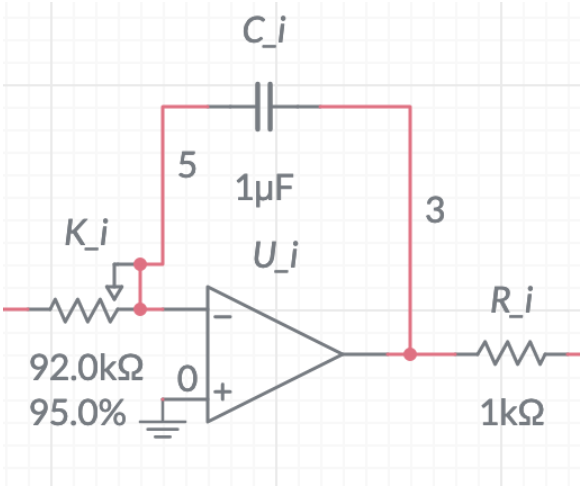


Figure 3.2: OPAMP as integrator

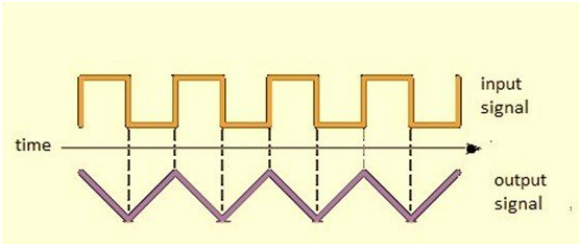


Figure 3.3: input and output signal through OPAMP has an integrator.

Integrator circuits are basically inverting operational amplifiers (they work in in-

verting op-amp configuration, with suitable capacitors and resistors), which generally produce a triangular wave output from a square wave input. Hence, they are also used for creating triangular pulses.

The current in the feedback path is involved in the charging and discharging of the capacitor; therefore, the magnitude of the output signal is dependent on the amount of time a voltage is present (applied) at the input terminal of the circuit.

3.3.1 Derivation of Op-amp as integrator

As we know from the virtual ground concept, the voltage at point 1 is 0 V. Hence, the capacitor is present between the terminals, one having zero potential and other at potential V_0. When a constant voltage is applied at the input, it outcomes in a linearly increasing voltage (positive or negative as per the sign of the input signal) at the output whose rate of change is proportional to the value of the applied input voltage. From the above circuitry it is observed, $V_1 = V_2 = 0$ The input current as:

$$I = \frac{V_i - V_1}{R_1} = \frac{V_i}{R_1}$$

Due to the op-amp characteristics (the input impedance of the op-amp is infinite) as the input current to the input of an op-amp is ideally zero. Therefore the current passing from the input resistor by applied input voltage V_i has flown along the feedback path into the capacitor C_1. Therefore the current from the output side can also be expressed as:

$$I = C_1 \frac{d(V_1 - V_0)}{dt} = -C_1 \frac{d}{dt}(V_0)$$

Equating the above equations we get,

$$\frac{V_i}{R_1} = -C_1 \frac{d}{dt}(V_0)$$

Therefore the op-amp output of this integrator circuit is:

$$V_0 = -\frac{1}{R_1 C_1} \int_0^t V_i dt$$

As a consequence the circuit has a gain constant of $-1/RC$. The negative sign point toward an 180° phase shift.

3.4 Op-amp as a Differentiator

As we have studied earlier in the integrator circuit, op-amps can be used for implementing different mathematical applications. Here we will be studying the differential op-amp configuration in detail. The differentiator amplifier is also used for creating wave shapes and also in frequency modulators.

An operational amplifier differentiator basically works as a high pass filter and, the amplitude of the output voltage produced by the differentiator is proportionate to the change of the applied input voltage.

When the input resistance in the inverting terminal is replaced by a capacitor, an RC Network has been established across the operational amplifiers' negative feedback path. This kind of circuit configuration helps in implementing differentiation of the input voltage, and this operational amplifier circuit configuration is known as an Operational amplifier differentiator circuit.

In a differentiating op-amp circuit, the output of the circuit is the differentiation of the input voltage applied to the op-amp with respect to time. Therefore the op-amp differentiator works in an inverting amplifier configuration, which causes the output to be 180 degrees out of phase with the input. Differentiating op-amp configuration generally responds to triangular or rectangular input waveforms.

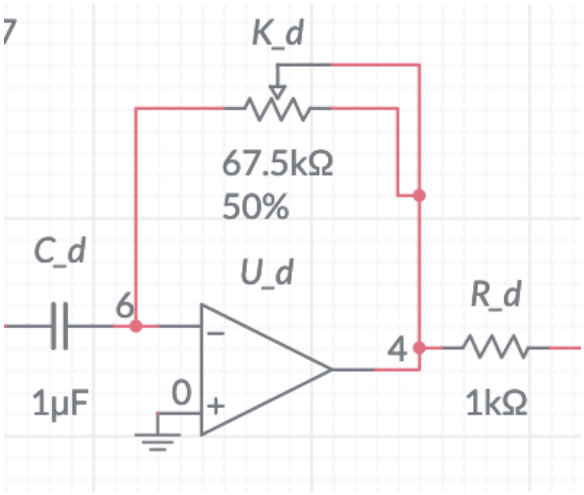


Figure 3.4: OPAMP as a differentiator

3.4.1 Derivation of Op-amp as Differentiator

As shown in the figure, a connection of capacitor in series with the input voltage source has been made. The input capacitor C_1 is initially uncharged and hence operate as an open-circuit. The non-inverting terminal of the amplifier is connected to the ground, whereas the inverting input terminal is through the negative feedback resistor R_f and connected to output terminal.

Due to the ideal op-amp characteristics (the input impedance of the op-amp is infinite) as the input current, I to the input of an op-amp is ideally zero. Therefore the current flowing through the capacitor (in this configuration, the input resistance is replaced by a capacitor) due to the applied input voltage V_{in} flows along the feedback path through the feedback resistor R_f. As observed from the figure, point X is virtually grounded (according to the virtual ground concept) because the non-inverting input terminal is grounded (point Y is at ground potential i.e., 0 V). Consequently, Vx = Vy = 0 With respect to the input side capacitor, the current

carrying through the capacitor can be written as:

$$\mathrm{I} = \mathrm{c}_1 \frac{\mathrm{d}\,(\mathrm{V_{in}} - \mathrm{V_X})}{\mathrm{dt}} = \frac{\mathrm{c}_1\ \mathrm{d}\,(\ \mathrm{V_{in}}\ )}{\mathrm{dt}}$$

With respect to the output side feedback resistor, the current flowing through it can be represented as:

$$1 = \frac{V_x - V_0}{R_f} = -\frac{V_0}{R_f}$$

From the above equations when we equate the currents in both the results we get,

$$\frac{c_1 d\,(v_{in})}{dt} = -\frac{v_0}{R_f}$$

$$V_0 = -C_1 R_f \frac{d\,(V_{in})}{dt}$$

The differentiating amplifier circuit requires a very small time constant for its application (differentiation), and hence it is one of its main advantages. The product value $C_1 R_f$ is known as differentiator's time constant, and output of the differentiator is $C_1 R_f$ times the differentiation of $\mathrm{V_{in}}$ signal. The -ve sign in the equation refers that the output is 180° difference in phase with reference to the input. When we apply a constant voltage with one step change at t = 0 like a step signal in the input terminal of the differentiator, the output should be ideally zero as the differentiation of constant is zero. But in practice, the output is not exactly zero because the constant input wave takes some amount of time to step from 0 volts to some $\mathrm{V_{max}}$ volts. Therefore the output waveform appears to have a spike at time t = 0.

3.5 Observation and Calculations

Seting the values of $K_p \Longrightarrow K_p$, $K_i \rightarrow K_i$ $K_d \rightarrow K_d$ we get the plot.

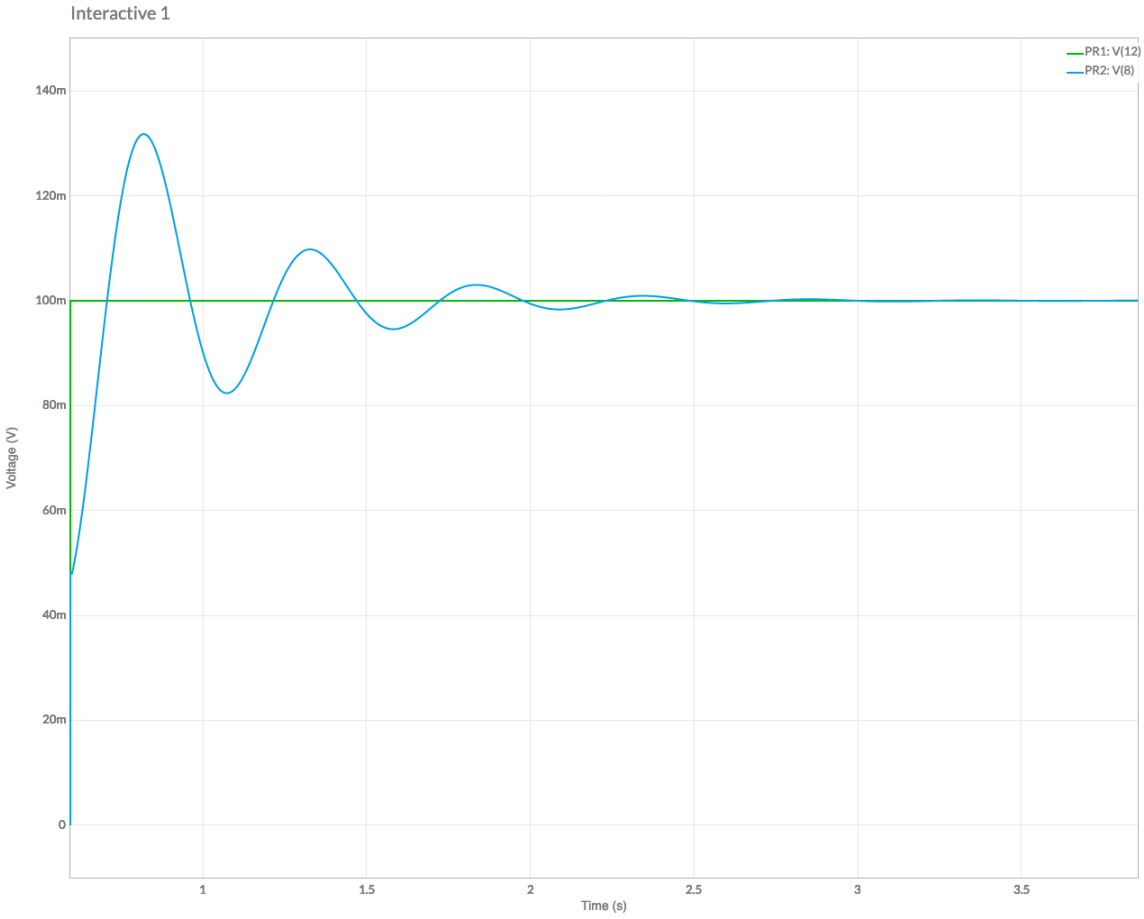


Figure 3.5: PID(,,)

Seting the values of $K_p =\rightarrow$ 2xK_p , $K_i \rightarrow K_i$ $K_d \rightarrow K_d$ we get the plot.

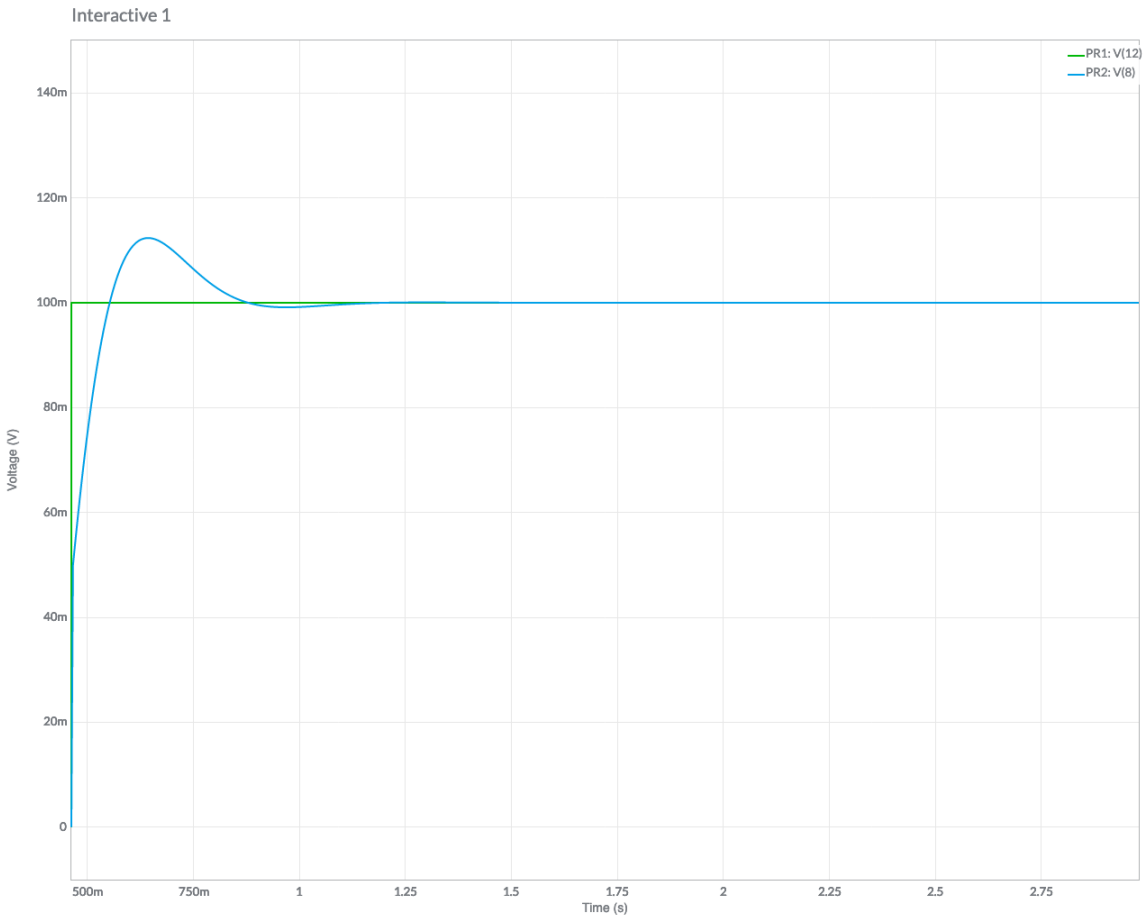


Figure 3.6: PID(2x,,)

Seting the values of $K_p =\rightarrow K_p$, $K_i \rightarrow$ 2xK_i $K_d \rightarrow K_d$ we get the plot.

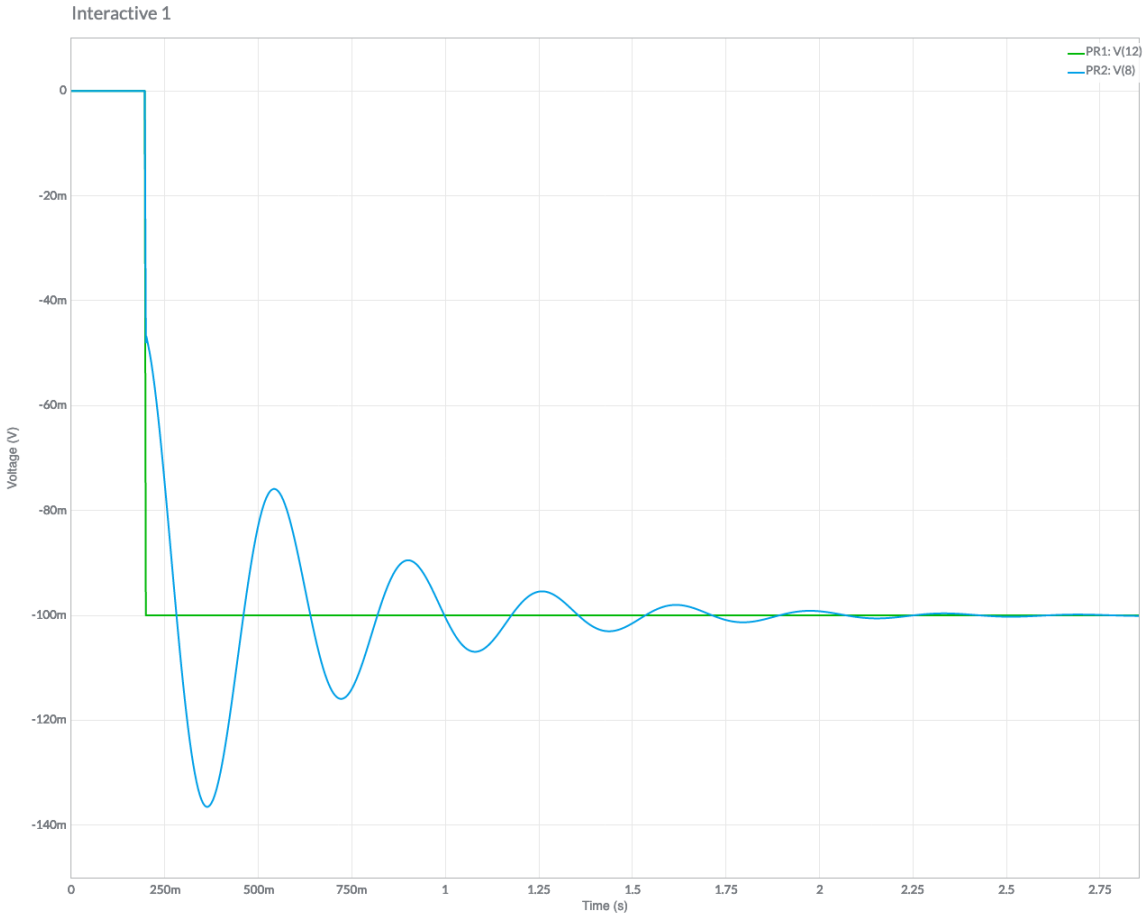


Figure 3.7: PID(,2x,)

Seting the values of $K_p =\rightarrow K_p$, $K_i \rightarrow K_i$ $K_d \rightarrow$ 2xK_d we get the plot.

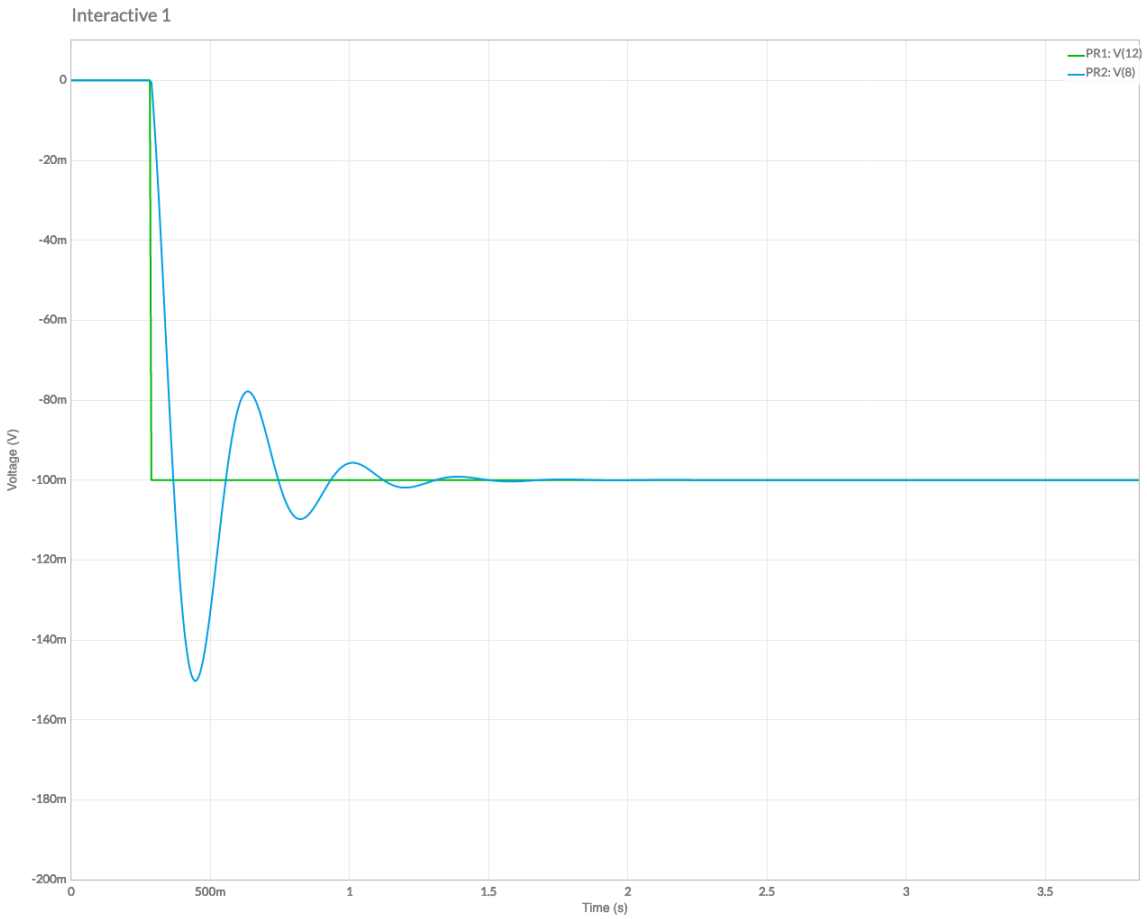


Figure 3.8: PID(,,2x)

Chapter 4

Summary and Conclusions

1. We learned how the PID controller works.
2. We studied the PID controllers theoretical explanations.
3. We first implement the PID controller using the TCLab in python.
4. We observe the changes in how fast or slow the graph reach to its target with changing parameters.
5. We learned how multiSim works.
6. We implemented the PID controller in multiSim.
7. We observe the change in time to reach the target with changing the value of resistance in the circuit in multiSim.

P-I-D control and its variations are commonly used in the industry. They have so many applications. 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. In almost all cases fast transient response and zero steady state error is desired for a closed loop system. Usually, these two specifications conflict with each other which makes the design harder. The reason why P-I-D is preferred is that it provides both of these features at the same time. In this recitation, it was aimed to explain how one can successfully use P-I-D controllers in their prospective projects. Being prospective control engineers, we feel lucky to give a presentation on the P-I-D

subject. Finally, we encourage prospective control engineers to use P-I-D controllers wherever necessary, especially, when a great controller is required.

Chapter 5

Reference

1. Wikipedia.

2. NISER lab report.

3. TClab documentation.

4. Multisim documentation.