MECH 420

MODEL PREDICTIVE CONTROL
SPRING 2018

PROJECT REPORT

-Pranay Desai

-Saravana Kumar Phillip Nagarajan

TABLE OF CONTENT:

1. Abstract

This document shows the use of three different controllers such as PI controller, Optimal controller and MPC for a Surge tank to control the height of the tank. Later all these controllers are compared based on specific performance requirements. Here our goal is to get the tank height between 10m - 29m by controlling the outflow of tank, we cannot measure the input that is coming in the tank, so we are designing observers to estimate the Q_{in}**.** We have also designed observer and Kalman filter to estimate the state of the system. Here our system is partially controllable and fully observable.

Key Words: MPC, Kalman Filter, Observer, Optimal.

2. Introduction:

The schematic diagram of tank with level controller is given in the figure 2.1.

Parameter	**Variable used**	**Units**
Area of the tank	A	ft^2
Tank inlet flow	Q_{in}	ft^3/sec
Tank Outlet flow	Q_{out}	ft^3/sec
Tank level	h	ft
Tank level setpoint	h_s	ft

The controller is implemented in discrete-time, so we are using the discrete model. The equation of motion is given by,

$$A\left[h(i+1) - h(i)\right] = T\left[Q_{in}(i) - Q_{out}(i)\right]$$

$$h(i+1) = h(i) + \frac{T}{A}Q_{in}(i) - \frac{T}{A}Q_{out}(i)$$

Assumption: The inlet flow Q_{in} is considered as a disturbance and is not measured.

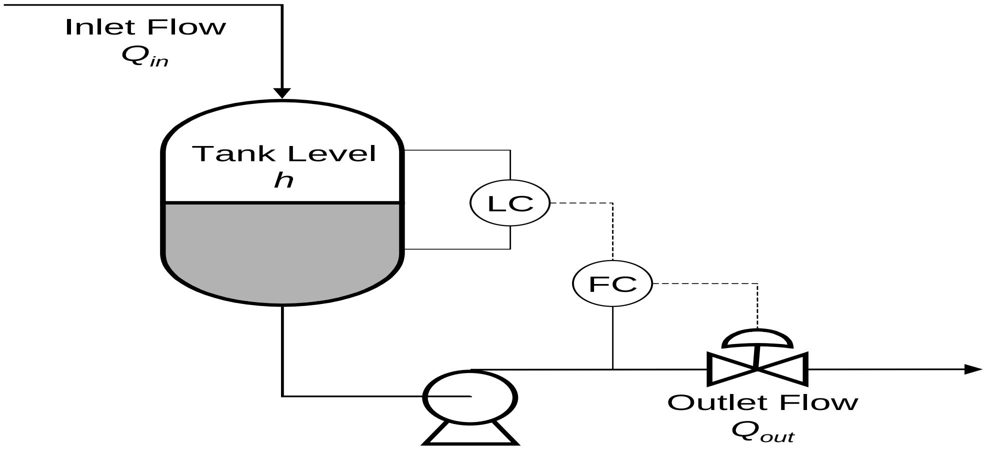


Figure 1. Schematic of a tank with level controller.

3. Objective:

The objective is to regulate the tank level, *h*, by manipulating the outlet flow Q_{out}.

For this we need,

- To simulate the system using Matlab and/or Simulink.
- To design an observer to estimate the input flow rate.
- To design a Kalman filter to estimate the input flow rate.
- To design an optimal controller to minimize the tank level variability assuming the outlet flow variation is minimal.
- To design a PI controller to control the level.
- To design a MPC.
- To compare these three controllers in regards to the required performance.

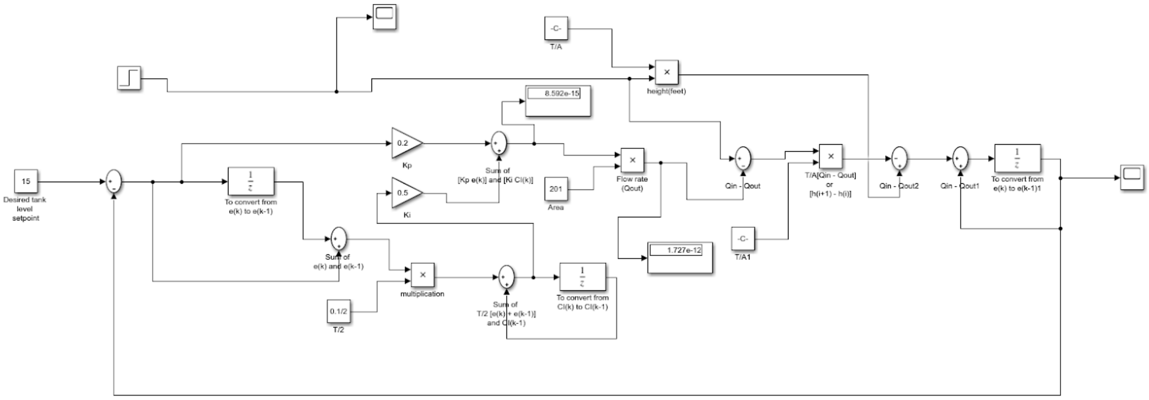
Two control objectives are:

- First, the rate of change of the outlet flow should be smooth to avoid upsetting downstream equipment.
- Second, level constraints must not be exceeded. Also, in the case of cascade tanks, it is important to minimize outlet flow overshoot, since this peak can be amplified as it moves down the cascade.

4. PI CONTROLLER:

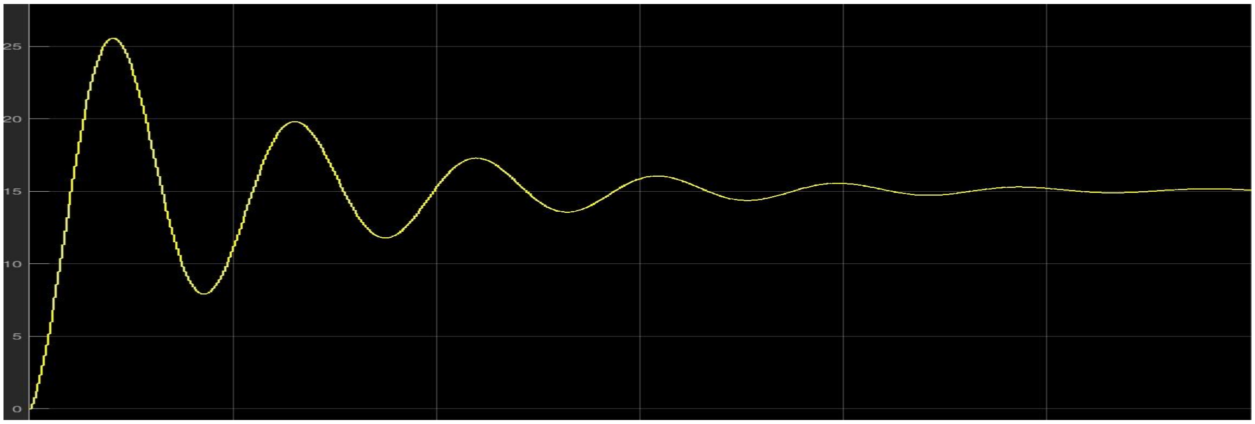
- A PI Controller (proportional-integral controller) is a special case of the PID controller in which the derivative (D) of the error is not used.
- In this project, PI controller is used to manipulate the height of the tank to a desired set point at 15 feet.
- Here,
  i) proportional controller $K_p = 0.2$
  ii) Integral controller $K_i = 0.5$
- Thus, the gain values (K_p and K_i) are used to drive our system from unstable values to more stabilized system.

Simulink model:



15
Desired tank level setpoint
1/z
To convert from e(k) to e(k-1)
0.2
Kp
0.5
Ki
Sum of [Kp e(k)] and [Ki CI(k)]
8.592e-15
-C-
T/A
height(feet)
201
Area
Flow rate (Qout)
Sum of e(k) and e(k-1)
multiplication
0.1/2
T/2
Sum of T/2 [e(k) + e(k-1)] and CI(k-1)
To convert from CI(k) to CI(k-1)
1.727e-12
Qin - Qout
T/A[Qin - Qout] or [h(i+1) - h(i)]
-C-
T/A1
Qin - Qout2
Qin - Qout1
To convert from e(k) to e(k-1)1

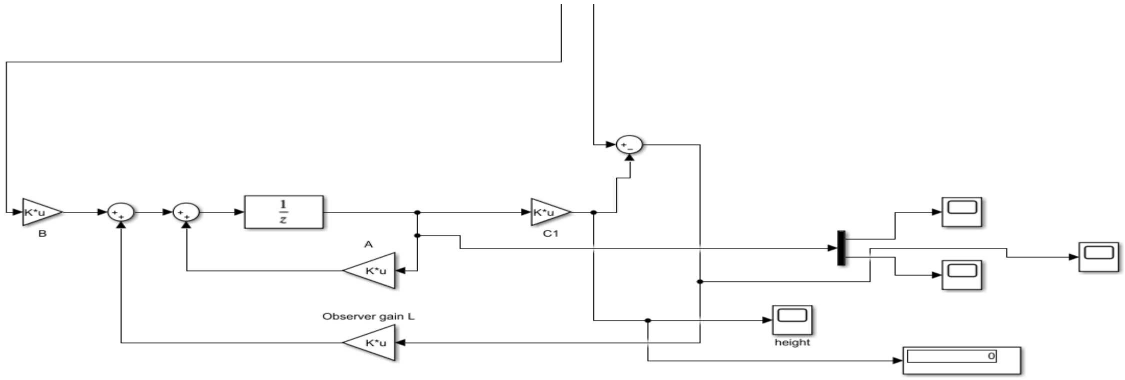
Graph:



5. OBSERVER:

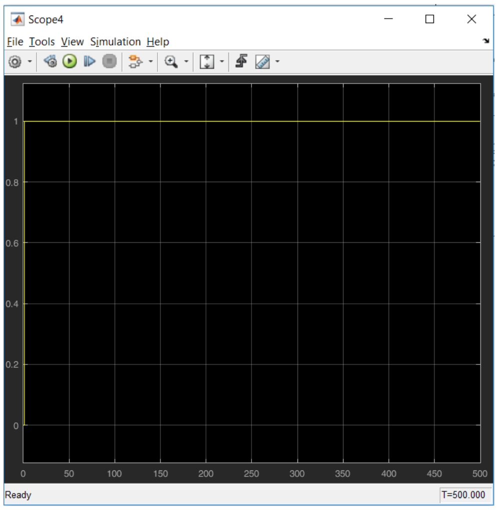
- As Q_{in} is considered as disturbance, it is considered as step disturbance. And the actual Q_{in} graph is shown below.
- Now, we can have information about over Qin
- Observer helps us to determine the state of the system by "looking" at the measurable output with the help of system model.

Simulink model:

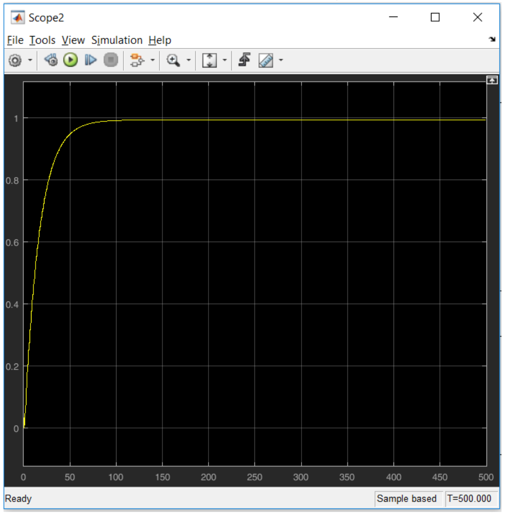


Graph:

Actual Q_{in}



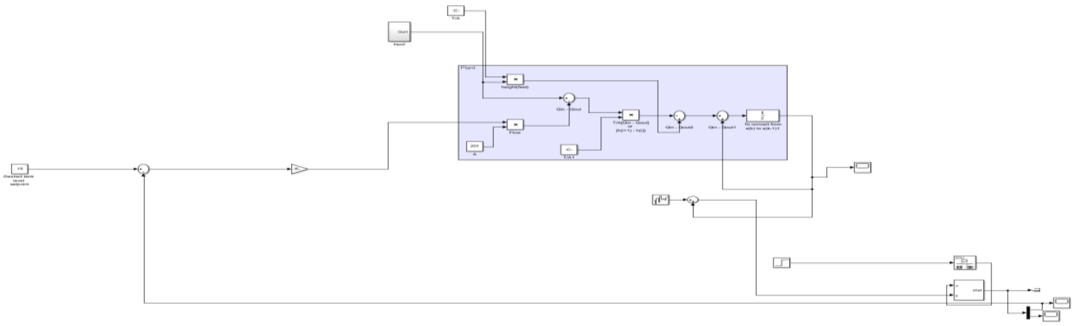
Observed Q_{in}



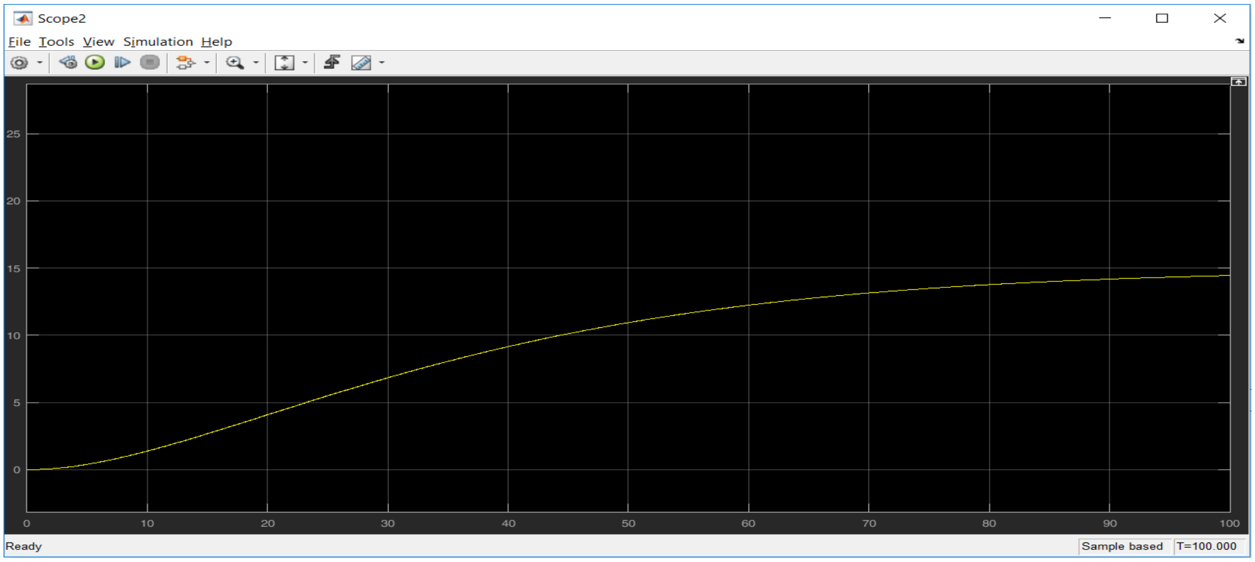
6. Optimal Control:

- Here, we have used LQR method to get the gain for our optimal controller, here it should be noted that when we implement the controller it behaves very much like a proportional controller.
- The advantage being that for the determined cost function, we have the optimal gain.
- While in PI we have to tune gains manually to meet the requirement, here we define the cost function according to the our needs and have calculated gain K.
- We have used kalman filter to get rid of the noise. Here our Q is 10*I and R is 1.

Simulink Model:



Graph:



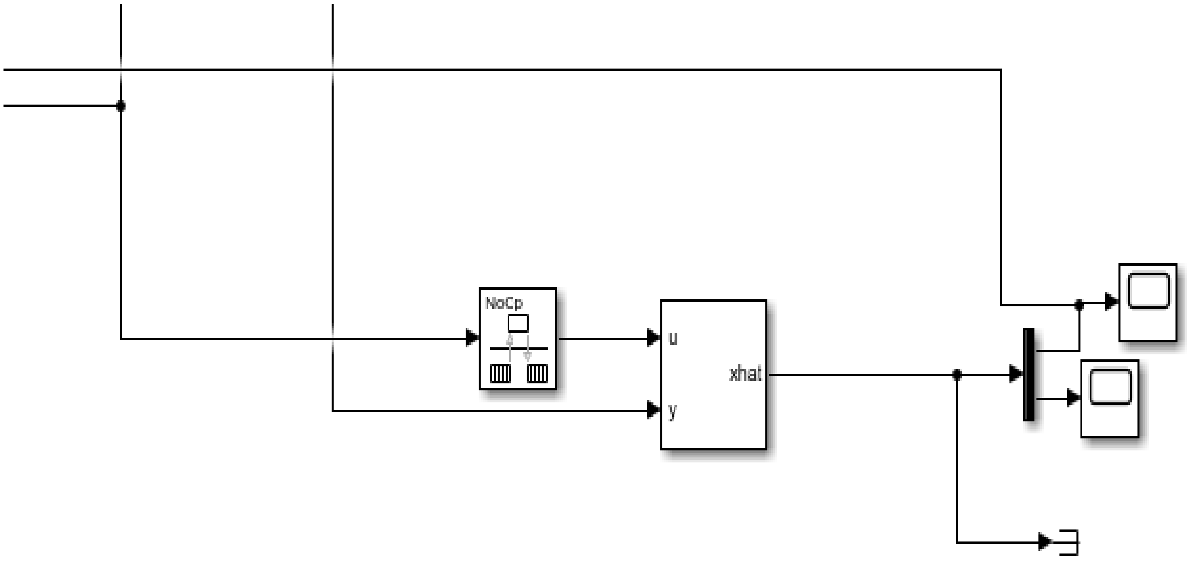
7. **Kalman Filter:**

- Here we have first used observer to observer the flow/disturbance coming in the system as we can see in the input because of the noise it’s hard for observer to give us interpretable output values.

- Now, we will move on to Kalman filter which is also an observer, but it removes noise.

- Kalman filter removes noise by looking at model and measurement both and based on how we define the covariance matrix, it will give us the output with reduced noise, generally this is will have significant improvement over the observer.

Simulink Model:



- Here, u is over control effort and y is our measurement which contains noise.

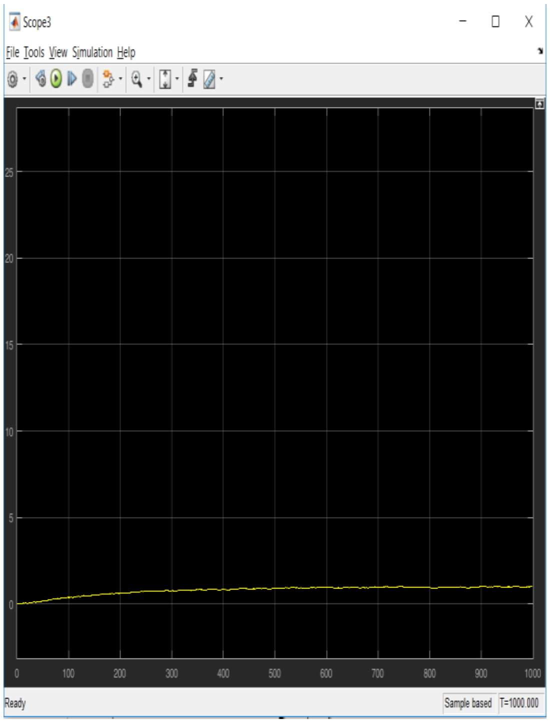
- Below is the whole model which contain both the observer and Kalman filter. We have implemented both here and we are not taking estimated state from Kalman filter from feedback

- Here’s the output from our Kalman Filter for Q_{in} and just below the output for Q_{in} for observer is given.

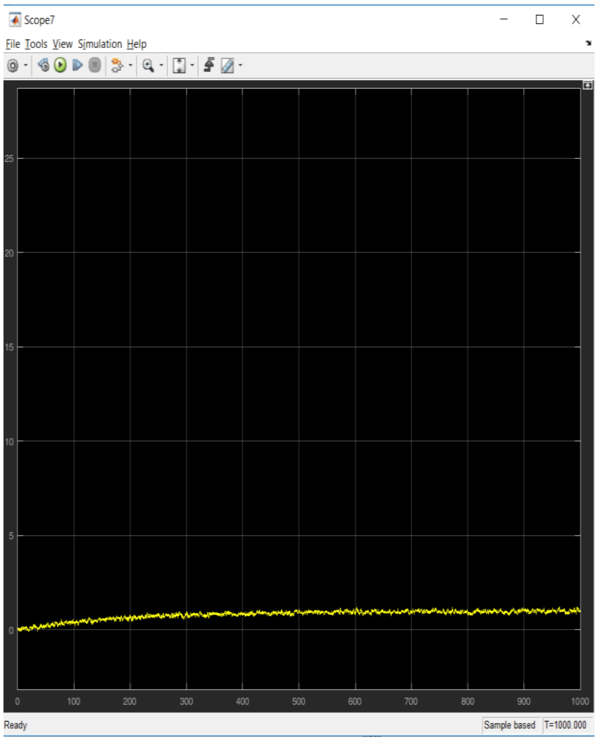
- As you can see the noise difference is clear also, you can see the same for height.

Graph:

Without Noise



With Noise

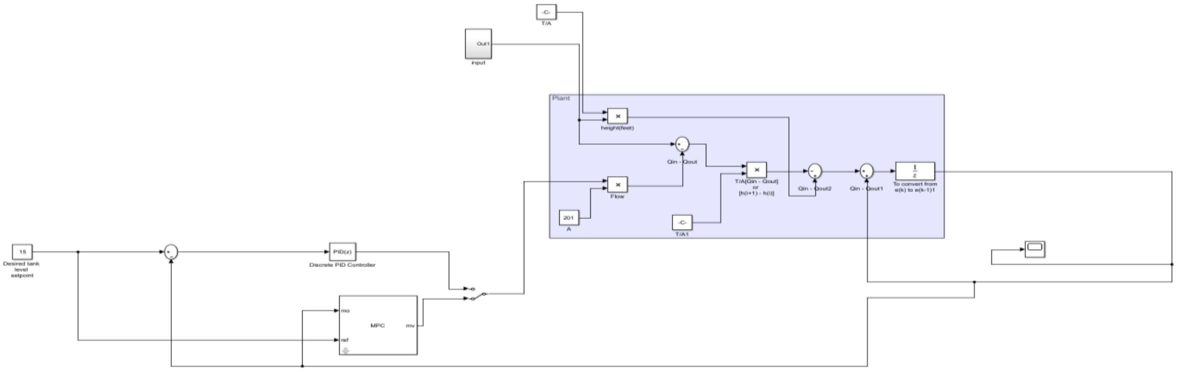


8. **Model Predictive Control:**

- Model predictive control as here we can see gives the best output and also handles the constraint limits very well compared to others. It can be seen from the diagram.

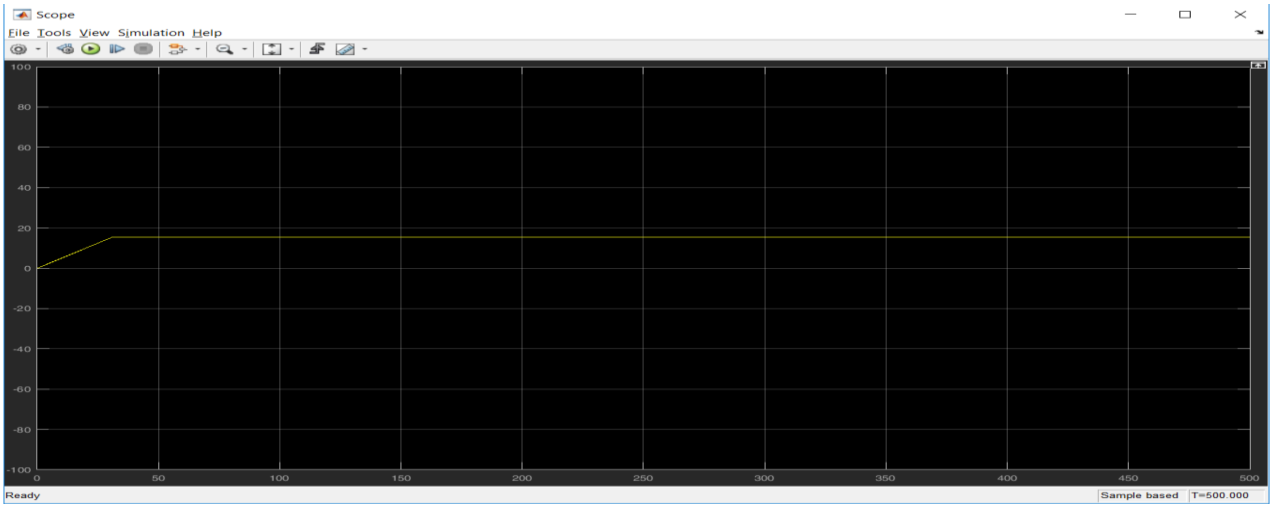
- Also, this controller handles noise very well, in the input noise is given with step input with we can see that in the output there is no to zero noise present here.

Simulink model:



Output and Simulink model are shown here, we have put a switch here so that we can quickly change between 2 most popular controller and compare them.

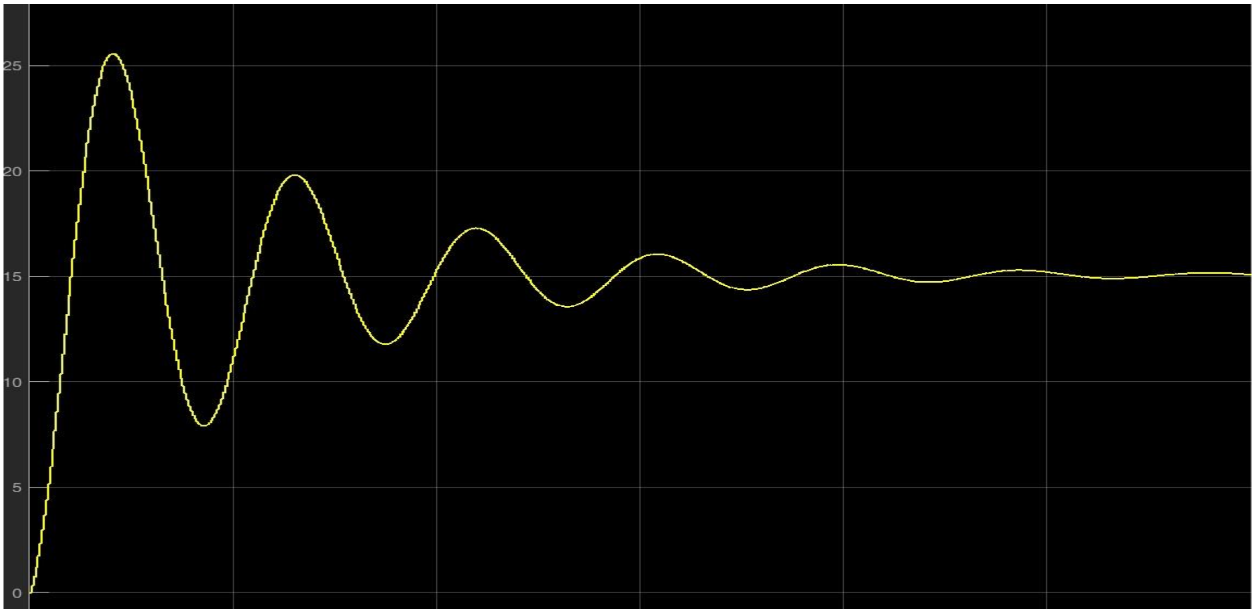
Graph:



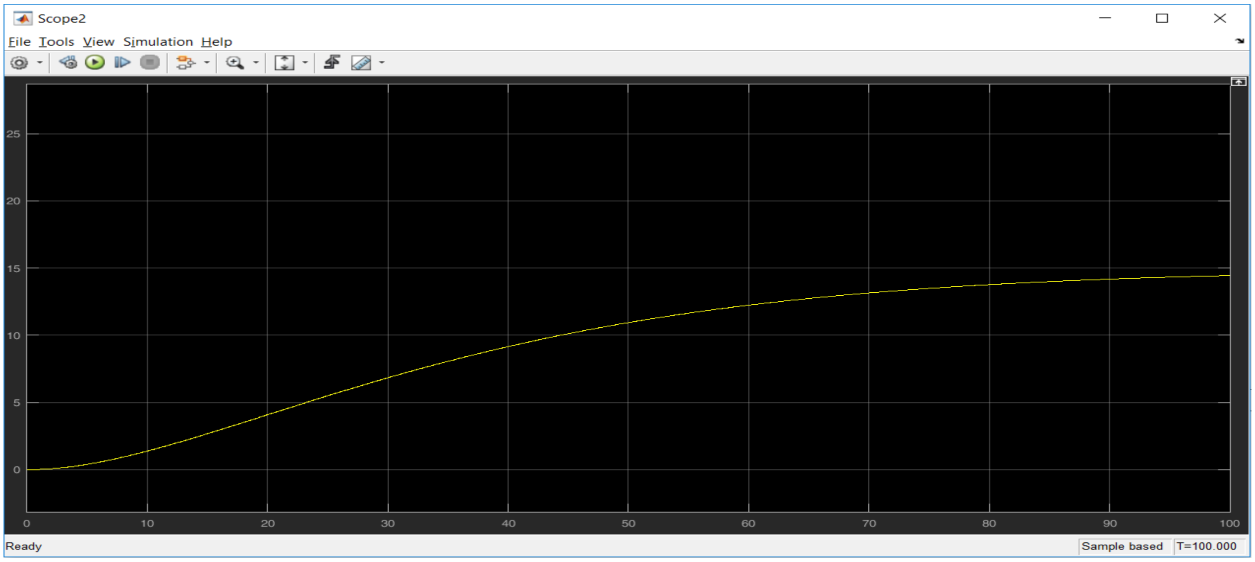
9. Analysis and Comparison:

- In PI controller, we have to check for the constraints while tuning our gain. Although we get the solution it is not optimal.
- In Optimal controller, we will get an optimal solution according to our cost function. This controller closely resembles "P" controller and there is non to less overshoot while PI has considerable.
- For MPC, after defining constraints for input and output and linearizing our model we will get the best solution which is also most robust one.
- We have attached graph here, so that comparison can be visualized.

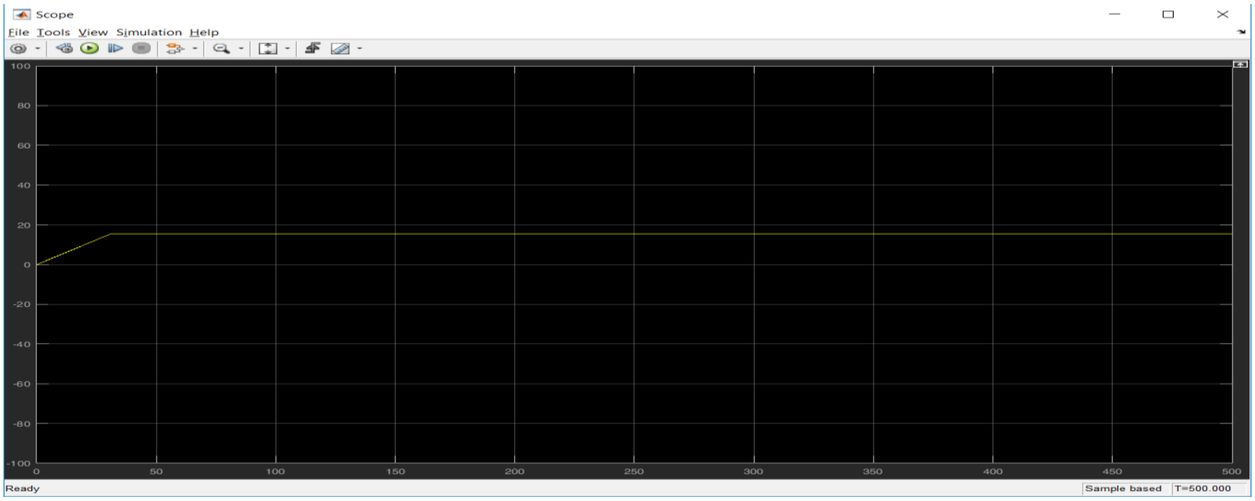
PI Graph:



Optimal controller:



MPC:



Conclusion:

After comparing, all three controllers exhaustively, we came to the conclusion that MPC provides the best results when we have to work within certain constraints which are model based.

For certain applications, PI and optimal controller may give overshoot which can be harmful to the plant. In conditions like this such as the tank that we have modelled here, MPC works best.