CH5120 - Modern Control Theory
Course Project 2
Group 8

Team Members:

1. **Abhishek Yadav - ME20B008**
2. **Cecil Jacob Thomas - ME20B050**
3. **Nilesh Balu - ME20B121**

Synopsis:

The program consists of 3 functions(kf, mpc_constraint_MIMO, mpcgain_MIMO) along with the main function. To run the program, we set the variables Cc and C_controller according to which heights are being measured and which one's controlled. We also need to set the setpoint on a case by case basis. Nc and Np indicate the control and prediction horizons respectively and are values that have been tuned to obtain optimized results.

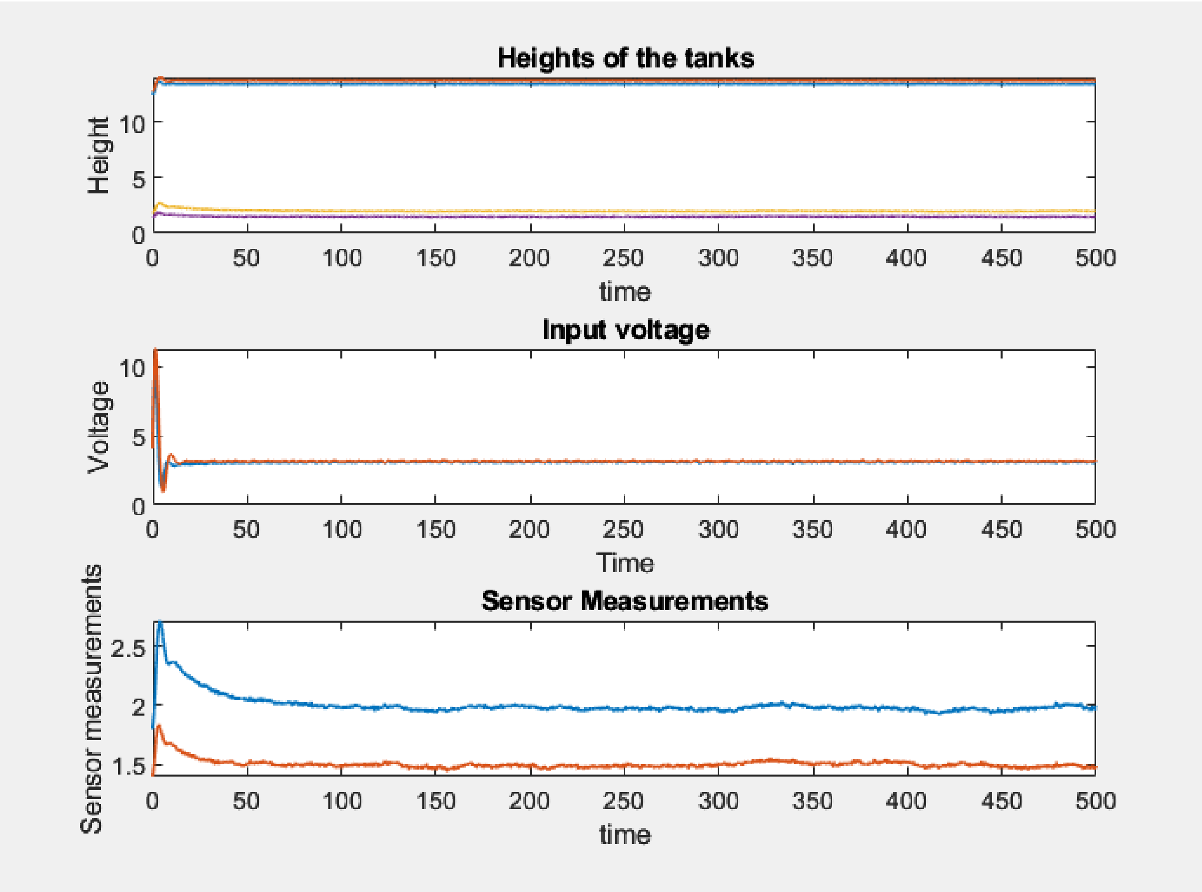
Results:

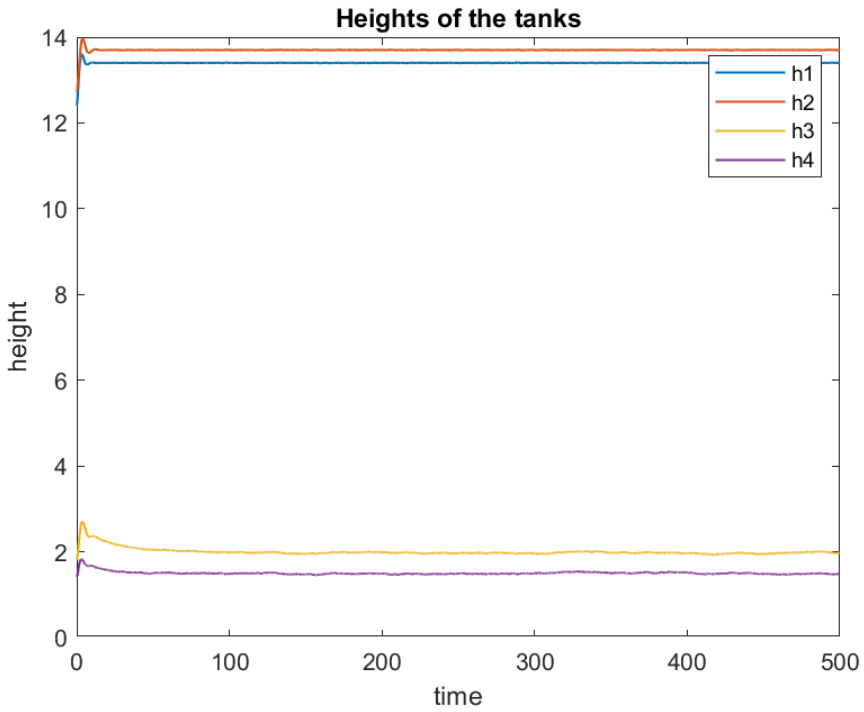
Part A - Constraints on the input and the rate of change of input

Case 1:

Measured variables: h3 and h4

Controlled Variables: h1 and h2



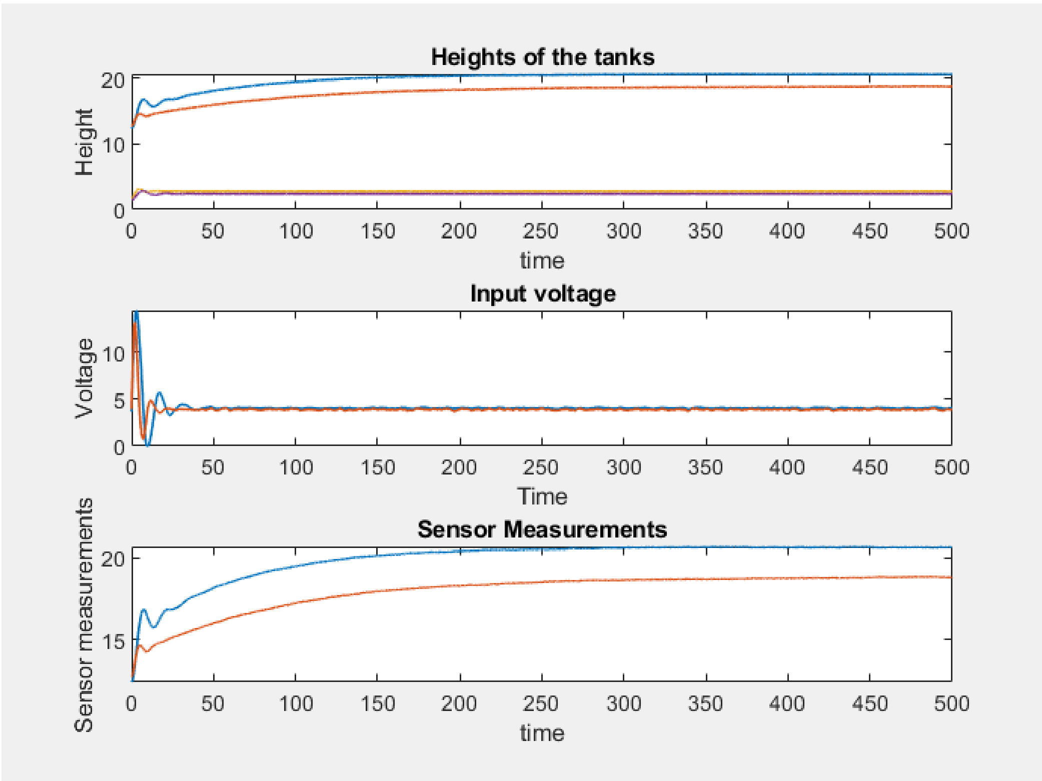


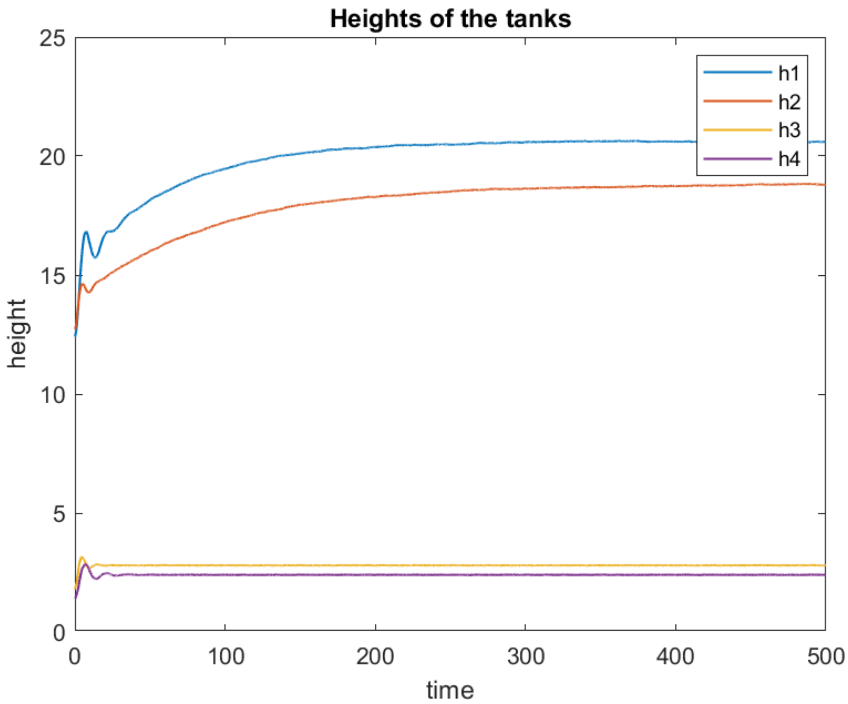
The graphs indicate that the controlled variables reach thier set point(h1 = 13.4 , h2 = 13.7) with a slight margin of error(0.22% for h1 and 0.16% for h2) at t = 500 sec. All tank heights seem to converge for this configuration of measured and controlled variables. The controller seems to only require about 40-50 seconds to converge for all variables. Accuracy for control variables also is pretty high.

Case 2:

Measured variables: h1 and h2

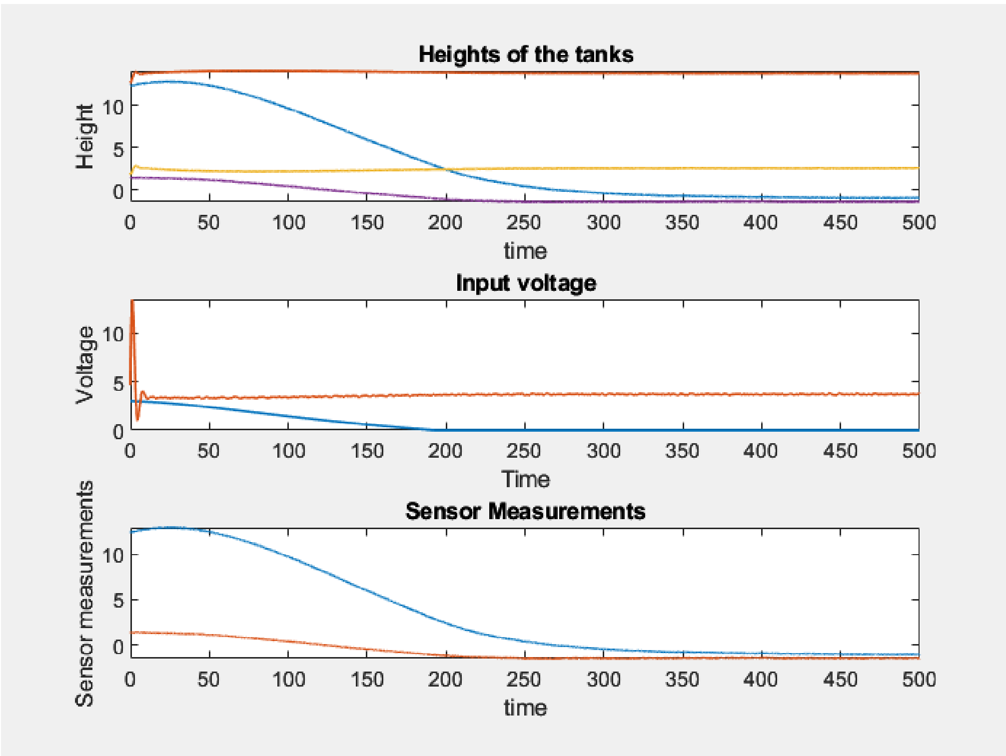
Controlled Variables: h3 and h4

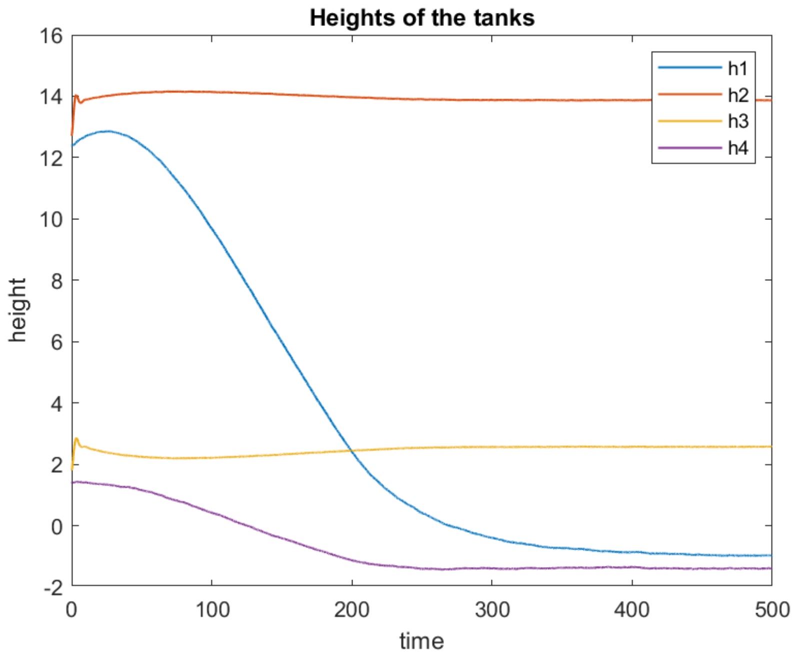




The graphs indicate that the controlled variables reach their set point(h3 = 2.8 and h4 = 2.4) with a slight margin of error(0.35% for h3 and 2.91% for h4) at t = 500 sec. The controlled variables h3 and h4 converge at the set point. It's also observed that it takes about 250 seconds for the measured variables h1 and h2 to attain steady-state values. Accuracy for control variables is acceptable

Case 3:
Measured variables: h1 and h4
Controlled Variables: h2 and h3



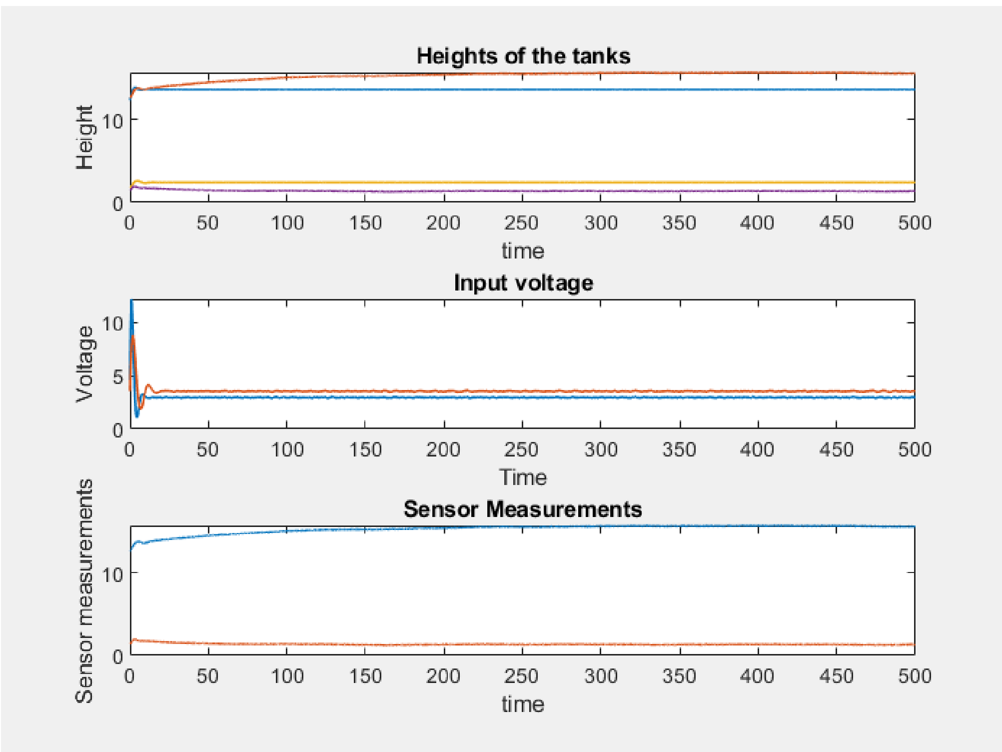


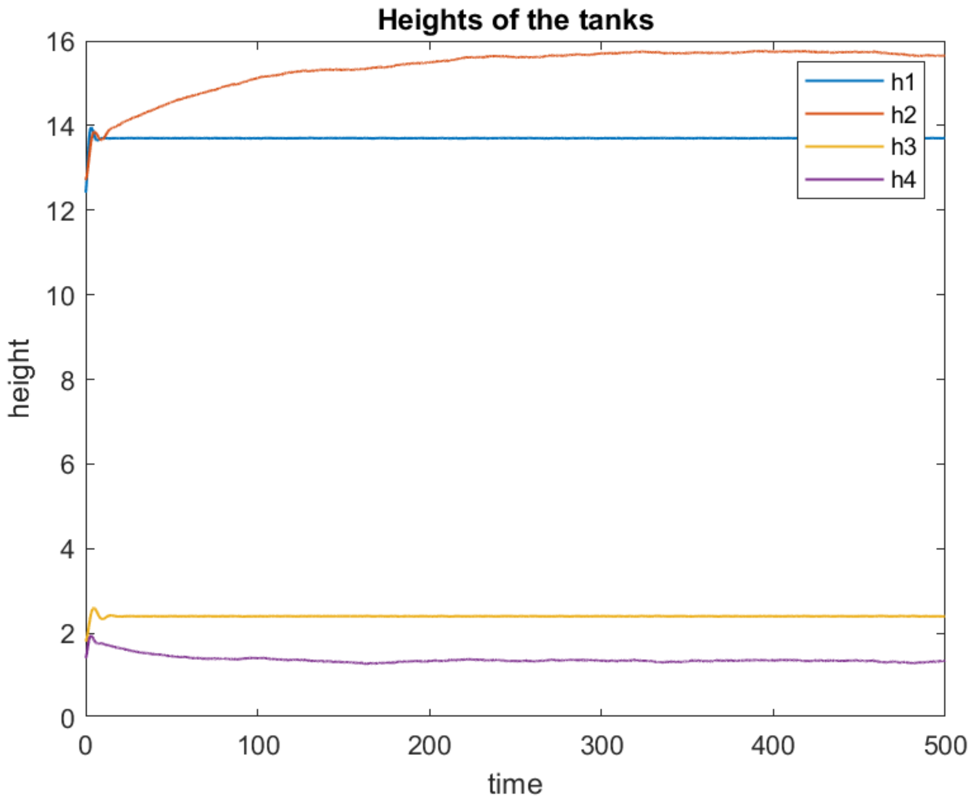
The graphs indicate that the controlled variables reach their set point(h2 = 13.7, h3 = 2.8) with a slight margin of error(0.94% for h2 and 7.78% for h3) at t = 500 sec. All tank heights seem to converge(h1 = -1.12, h2 = 13.85 ,h3 = 2.58, h4 = -1.46) for this configuration of measured and controlled variables. Since we haven't implemented any constraints, the tank heights h1 and h4 come out to be negative, which is physically impossible but possible as far as the equations are concerned. Accuracy for the control variable h3 is pretty bad in this case

Case 4:

Measured variables: h2 and h4

Controlled Variables: h1 and h3





The graphs indicate that the controlled variables reach their set point [13.7, 2.4] with a slight margin of error(0.072% for h1 and 1.125% for h3) with h1 = 13.69 and h3 = 2.427 for t = 500 sec. The tank heights seem to be taking about 350-400 seconds to converge. The accuracy of the control variable at the steady state is pretty good.

Overall, case 1 provides the fastest convergence along with very good accuracy.

<u>Questions:</u>

Q1:

All cases seem to achieve the required set point. From the graphs it is evident that Case 1 is the most optimal in terms of both time taken to converge and accuracy. Details about accuracy are values detailed in the respective case descriptions.

Q2:

Out of the 4 cases, case 1 gave the best performance in terms of low latency and high accuracy. Case 3 did not give physically realisable values after exceeding a certain number of iterations.

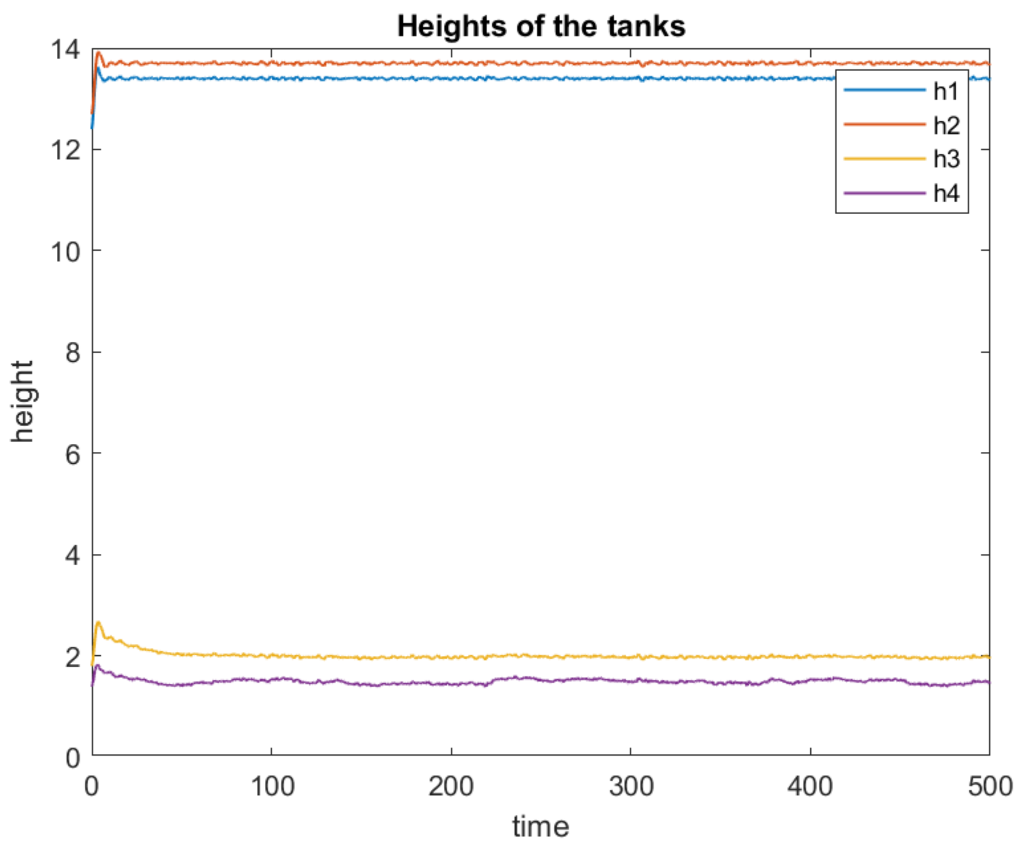
The problem lies in the fact that h2 and h3 are directly controlled by pump 2, whereas h1 and h4 are indirectly controlled by pump 1. As a result, we get accurate estimates of how pump 1 should be controlled but don't get accurate estimates as to how pump 2 should be controlled.

In other words, the system is not observable. Rank(Observability matrix) = 3, which is not full rank. So the problem lies in estimating the next state, as we can't properly map the sensor outputs to the hidden states.

Q3:
We experimented with running this code for different values of Np, Nc, Q and R.
The MPC part can be tuned by changing Np and Nc. Increasing Nc and decreasing Np would make the system more aggressive. We found the case where Np = 15 and Nc = 10 to be the most stable.
We can also tune the Kalman filter by changing Q and R. We found that Q = 0.01 and R = 0.2 gave good results.

Q4:
Upon increasing Np from 15 to 30, the peak overshoot decreased, making the system more stable. Similar results were obtained when we decreased the value of Nc from 10 to 1. In both these cases, there was a slight increase in the number of iterations to reach a steady state value.

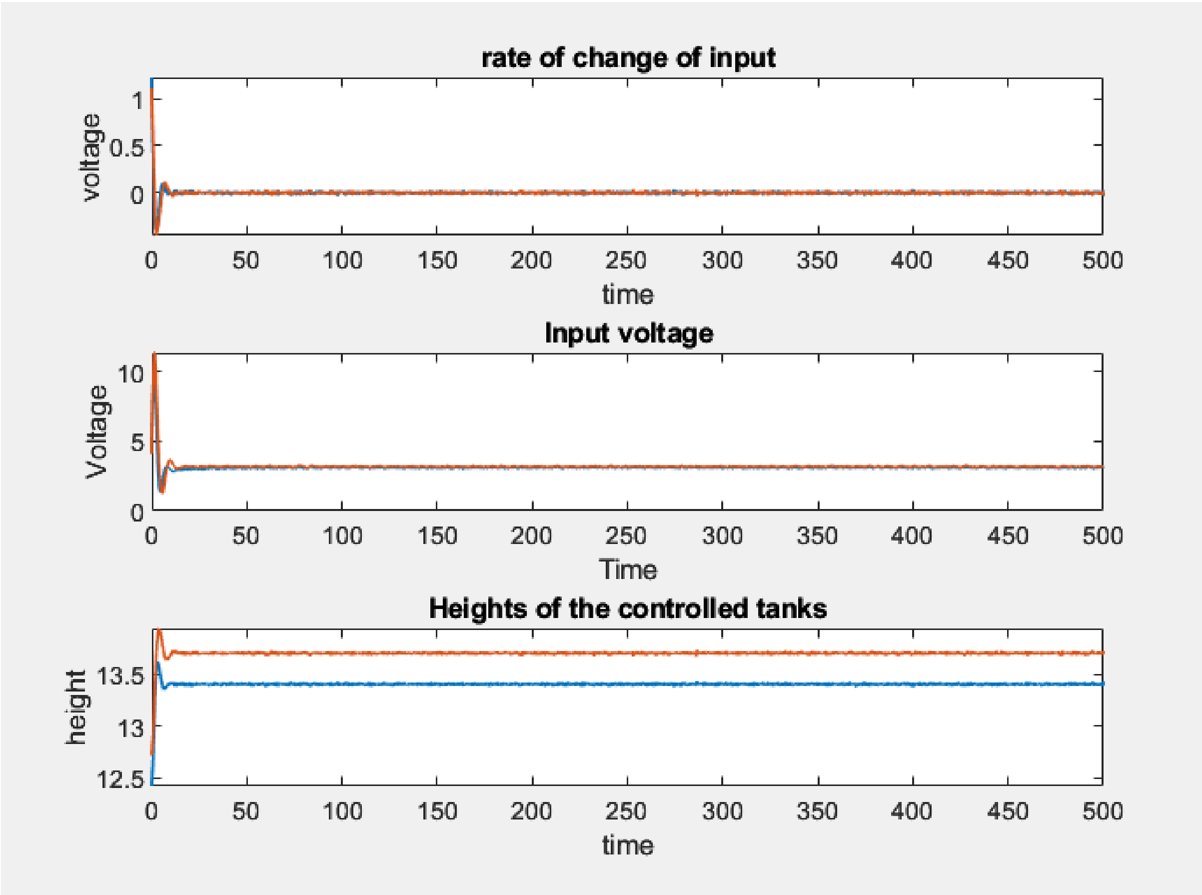
<u>Part B: - With constraints on output:</u>



Questions:

Q1:

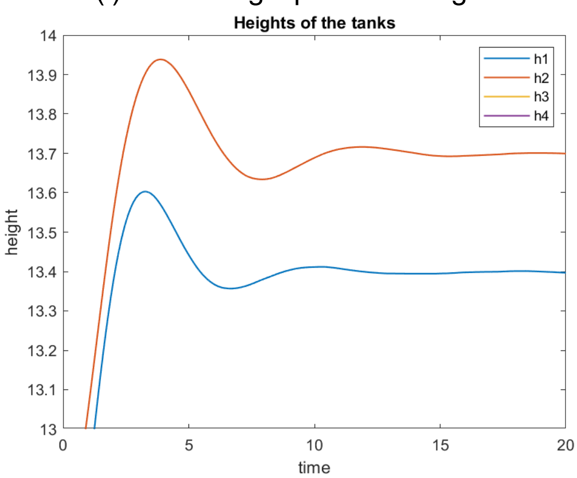
Similar to case 1 in part A, the setpoint is achieved in under 50 seconds.
As can be seen from the first figure, all the 3 constraints are met.

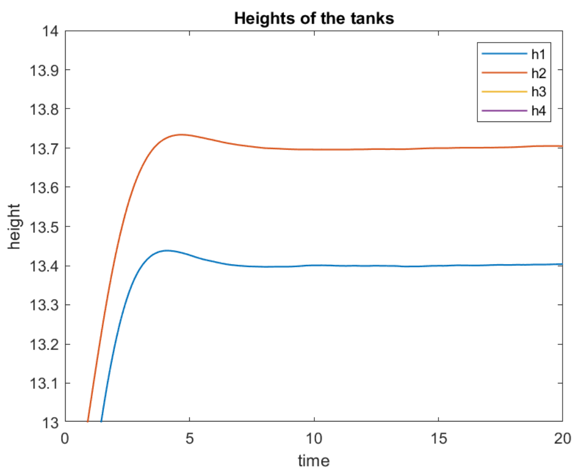


Q2:

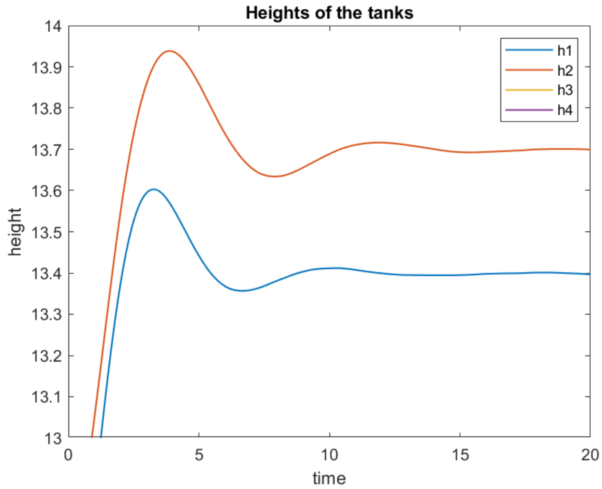
We reduced the overshoot by changing the following:

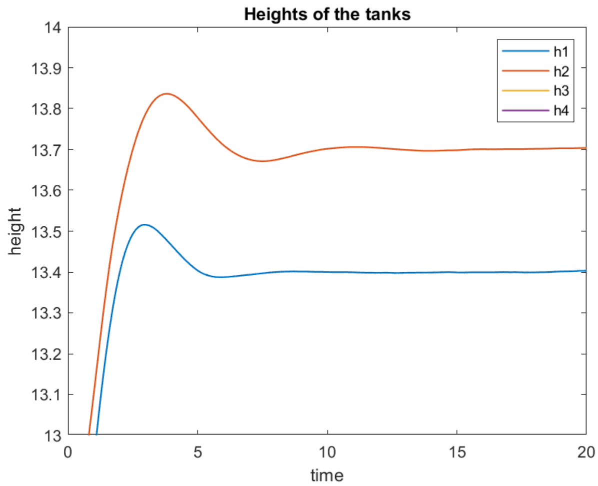
(i) Increasing Np - We changed it from 15 to 30





(ii) Decreasing Nc - We changed it from 10 to 1





Q3:
When the Ymin exceeds 12.4 cm, the quadratic programming problem becomes infeasible. So, for a feasible solution, Ymin < 12.4 cm.
This result makes sense because the initial value itself is 12.4, and there is a constraint put on the change rate of the input. Hence a feasible solution wouldn't exist.

<u>Contributions:</u>
Nilesh and Cecil: Sat together and worked on Part A of the project. Wrote the code from scratch and wrote most of the report for Part A.
Abhishek: Wrote the code and report for Part B completely.

All of us sat together and worked on answering the questions given in Part A, with each of us giving our take. We discussed the possible causes of the results observed and how to troubleshoot errors.
(It's difficult to provide a concise contribution of each member as it was a synergistic effort)

<u>Attachments:</u>
All code written by our team is attached in the google drive link shared below. Different sets of code were used in Part A and Part B of our project.
https://drive.google.com/drive/folders/1pqt_FXQROHEc13-VM0YQR8cSy7L9uRZ2?usp=sharing