Model Predictive Control in Single Board Heater System using SCILAB

Pratik Behera
07002054

November 16, 2010

Contents

List of Figures

Chapter 1

The Project

1.1 Objective

- To implement Model Predictive Control (MPC) in Single Board Heater System using Scilab and perform experiments using it
- To perform experiments for various values of tuning parameters and study its effect on the system

1.2 Single Board Heater System

It is a single heater system, where in, 5cm x 2cm stainless steel blade acts as a plant, which is heated by a heating coil and cooled by a fan.

For Single Board Heater System (SBHS):

- Control variable: temperature
- Manipulated variable: heater
- Disturbance variable: fan

The heater element consists of Nichrome wire - of 0.7mm diameter, wound with 20 equally spaced helical turns into a coil of 5mm x 11mm. The heater element is kept at a distance of 3.5 mm from the steel blade.

Cooling is done by a computer fan, which is placed below the stainless steel blade.

Chapter 2

Model Predictive Control

An equivalent quadratic programming (QP) formulation for constrained DMC (as given in LQG_MPC_notes by Prof Sachin Patwardhan) is given as follows

$$min_{U_f} \frac{1}{2} U_f(k)^T H U_f(k) + F^T U_f(k) \tag{2.1}$$

Subject to

$$AU_f(k) \leq b \tag{2.2}$$

where

$$\tag{2.3}$$

$$A = \begin{bmatrix} I_{qm} \\ -I_{qm} \end{bmatrix}$$

$$b = \begin{bmatrix} U^H \\ -U_L \end{bmatrix}$$

Also, we have outputs and manipulated variables related to state variables by

$$x(k+1) = \Phi x(k) + \Gamma(k) + w(k) \tag{2.4}$$

$$y(k) = Cx(k) + v(k) \tag{2.5}$$

$$\tag{2.6}$$

ϕ is represented by matrix A in the code, Γ is represented as matrix B and C is represented as C matrix in the code.

Chapter 3

Implementing MPC

As mentioned earlier, MPC experiments were performed on SBHS 12 remotely. For this, the scilab codes uploaded on Moodle for Process Control SBHS assignments were used. The folder containing the codes, which was used to perform MPC experiments, have been included in the attached zip file in a folder named *codes*.

3.1 Working of codes

There are three main codes, which are being used for this experiment. *stepc.sce* is the code which opens the xcos window, wherein, we have step block for the set-point for temperature and the fan speed. Once the values have been entered into the xcos window and the simulation is started, the *scifunc* block of xcos calls the function *steptest.sci* after every sampling time. The *steptest.sci* in turn calls *mpc_run.sci* every time it is called by *scifunc* block. The *mpc_run.sci* code optimizes manipulated variable (heater) over control horizon and returns only the first manipulated variable (heater) value. This new heater value is then sent to the heater of the SBHS to control the temperature at the set point.

3.2 Procedure to implement MPC on SBHS

1. Download the folder named *codes* attached with the mail.
2. Copy the empty files (clientwrite and clientread files) from *Empty-files* folder to *Step-test-codes* folder.
3. Now open the folder named *Client-Java-latest18102010* and open VirtualLab-Client.
4. After entering the details and connecting to the alloted SBHS, open Scilab 5.2.2
5. Change the current directory to this folder, where steptest codes are present.
6. Open *stepc.sce* and *load in scilab*
7. The xcos window opens. Description of xcos window is mentioned below in the next section.

8. Enter the required step change values, if any, in the Temperature set point and/or fan block.
9. Enter the sampling time in the clock block as 1 second
10. Start the simulation from the xcos
11. After the experiment is over, the data files can be downloaded from the server.

3.3 Description of xcos

When *stepc.sce* is executed in scilab, an xcos window opens up. The xcos window has two step input blocks. The first step input block on the left side, is for the Temperature set point and the second step input block is for the fan (disturbance variable). Also the sampling time can be entered via clock block present on the xcos.
For all the experiments done for this project, sampling time of 1 second was used (entered via clock block of xcos).
Refer to the figure below for a clear picture of the xcos.

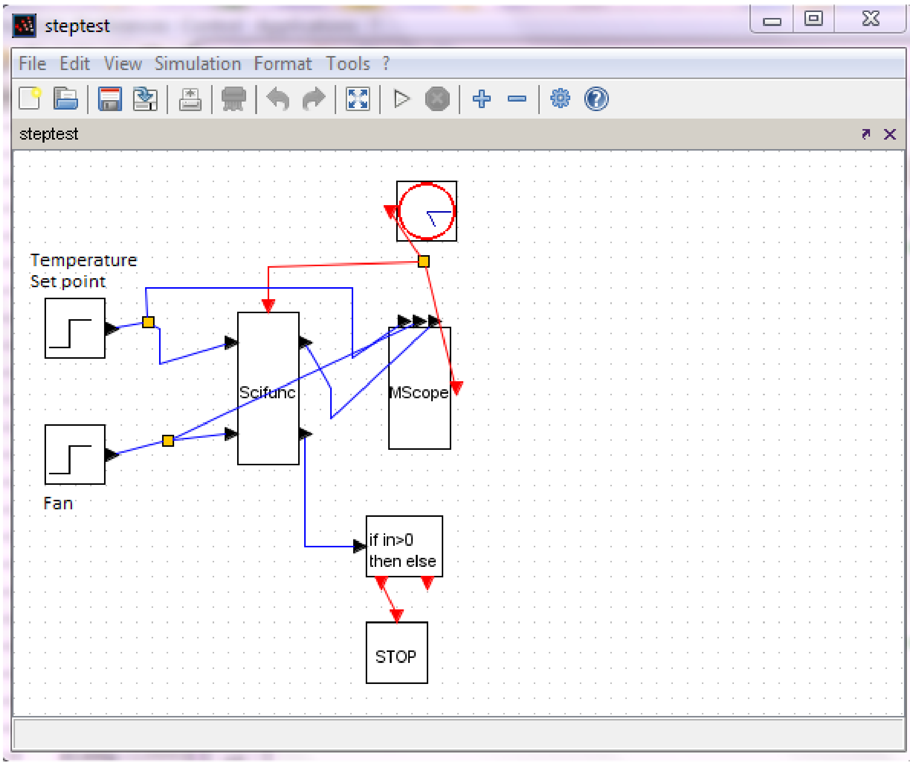


Figure 3.1: Screenshot of the xcos window with step input blocks labeled

After entering the values in the input step block, the simulation can be started. This opens up a graph, which shows the values of Temperature-set-point, fan and the actual temperature at each time instant during the simulation.

3.4 Code for MPC (mpc_run.sci)

The MPC code, implemented by me has been mentioned below. (Please note that this same code is present in the folder named *codes* → *Step-test-codes* attached with the mail.)

```
function [u_new] = mpc_run(T, u_prev, Tset)
global p q xk_old

A = [0.9780 0 0 0 0 0 0 0;
        1    0 0 0 0 0 0 0;
        0    1 0 0 0 0 0 0;
        0    0 1 0 0 0 0 0;
        0    0 0 1 0 0 0 0;
        0    0 0 0 1 0 0 0;
        0    0 0 0 0 1 0 0;
        0    0 0 0 0 0 1 0];
B = [1; 0; 0; 0; 0; 0; 0; 0];
C = [0 0 0 0 0 0 0 0.0079];

Tmax = 70;                  // Maximum Temperature
We = 100*eye(p,p);          // Error Weighting Matrix, We
Wu = 10*eye(q,q);           // Control Weighting Matrix, Wu
xk=A*xk_old+B*u_prev;

// Formation of Su Matrix for Quadtratic term of optimization
for i = 1:1:p
    for j = 1:1:q
        if i <= q
            if(i-j) >= 0
                Su(i,j)= C*A^(i-j)*B;
            else
                Su(i,j)= 0;
            end
        else
            if j < q
                Su(i,j)= C*A^(i-j)*B;
            else
                Su(i,j) = Su(i-1,j) + C*A^(i-j)*B;
            end
        end
    end
end

du_matrix=ones(q,q);

//  Lambda Matrix for Quadtratic term of optimization
for i = 1:1:q
    for j = 1:1:q
```

```
        if i == j
            du_matrix(i,j) = 1;
            if i >1
                du_matrix(i,j-1) = -1;
            end
        else
            du_matrix(i,j) = 0;
        end
    end
end

du_matrix_0 = eye(1,q);    // Declaration of Lambda_0 vector

// Formation of Sx Matrix for Linear term of optimization
for i = 1:1:p
    Sx(i,:) = C * A^i;
end

// Declaration of S_eta Matrix
S_eta = ones(1,p);

// Declaration of Set Point Vector
R = ones(1,p)*Tset;

// Temperature Prediction using information till previous instant
T_pred = C*xk;

// Measurement Error
eta = T - T_pred;

// Quadratic Term for Optimization
Su_t=Su';
du_matrix_t=du_matrix';
Q=2*((Su_t*We*Su)+(du_matrix_t*Wu*du_matrix));

// Linear Term in Optimization
R_t=R';
S_eta_t=S_eta';
du_matrix_0_t=du_matrix_0';
F_term1_t=(R_t-(Sx*xk)-(S_eta_t*eta))';
F=-2*((F_term1_t*We*Su)+((du_matrix_0_t*u_prev)'*Wu*du_matrix));

// Inequality Matrices and Vectors
A_ineq = [eye(q,q);-1*eye(q,q)];

b_ineq_term1_1=-Sx*xk-S_eta';
b_ineq_term1_2=(Tmax*ones(1,p))';
b_ineq_term1=(b_ineq_term1_1*eta+b_ineq_term1_2)';
b_ineq_term2_1=-Sx*xk-S_eta'*eta;
b_ineq_term2_2=(Tmax*ones(1,p))';
```

```
b_ineq_term2=-1*(b_ineq_term2_1+b_ineq_term2_2)';
b_ineq = [40*ones(1,q) zeros(1,q)];

me=0;
ci=zeros(q,1);
cs=40*ones(q,1);

[x,iact,iter,f]=qpsolve(Q,F,A_ineq,b_ineq');

u_new=x(1);
xk_old=xk;
endfunction
```

3.5 Other codes used

Other codes that were used for conducting this experiment are *stepc.sce* and *steptest.sci*. Both these codes were originally taken from Moodle (Process controls course for SBHS assignment). Please note that, both these codes were slightly modified to work with our MPC.
The only changes done in the original codes are:

- addition of global variables p,q and xk_old (p is the prediction horizon, q is the control horizon, xk_old represents the last value of an internal state)
- initialization of p, q and xk_old
- removal of some unnecessary lines (ie, lines not relevant to MPC implementation)

The modified codes are present in the *codes* → *Step-test-codes* folder attached with this mail.

3.5.1 stepc.sce code

```
// For scilab 5.1.1 or lower version users,
//use scicos command to open scicos diagrams instead of xcos

global fdfh fdt fncr fncw m err_count y p q xk_old

p = 40; //prediction horizon
q = 4;  // control horizon
xk_old = zeros(8,1);

fncr = 'clientread.sce';
fdt = mopen(fncr);
mseek(0);

err_count = 0; //initialising error count for network error
m =1;
exec ("steptest.sci");
```

```
exec("mpc_run.sci");

fdfh = mopen('clientwrite.sce');
mseek(0);
b = mgetl(fdfh,1);
a = mgetl(fdt,1);
mclose(fdfh);

if a ~= []
  if b~= []
    disp("ERROR!EMPTY THE CLIENTREAD AND CLIENTWRITE FILES");
    return
  else
    disp("ERROR!EMPTY THE CLIENTREAD FILE");
    return
  end
else
  if b~= []
    disp("ERROR!EMPTY THE CLIENTWRITE FILE");
    return
    end;
   A = [0.1,m,0,251];
    fdfh = file('open','clientwrite.sce','unknown');
    write(fdfh,A,'(7(e11.5,1x))');
    file('close', fdfh);
    sleep(1000);
    a = mgetl(fdt,1);
    mseek(0);
    if a~= [] //open xcos only if communication is through
       xcos('steptest.xcos');
    else
      disp("NO NETWORK CONNECTION!");
      return
end
```

3.5.2 steptest.sci code

```
function [y,stop] = steptest(Tsp,fan)
global fdfh fdt fncr fncw m err_count stop p q xk_old

fncr = 'clientread.sce';
fncw = 'clientwrite.sce';

a = mgetl(fdt,1);
b = evstr(a);
byte = mtell(fdt);
mseek(byte,fdt,'set');

if a~= []
    temps = b(1,$); heats = b(1,$-2);
```

```
    fans = b(1,$-1); y = temps;

    heat = mpc_run(y,heats,Tsp);

    if heat>40
      heat = 40;
    elseif heat<0
      heat = 0;
    end;

  A = [m,m,heat,fan];
  fdfh = file('open','clientwrite.sce','unknown');
  file('last', fdfh)
  write(fdfh,A,'(7(e11.5,1x))');
  file('close', fdfh);
  m = m+1;

  else
    y = 0;
    err_count = err_count + 1; //counts the no of times network error occurs
    if err_count > 300
      disp("NO NETWORK COMMUNICATION!");
     stop = 1;  // status set for stopping simulation
     end
  end

return
endfunction
```

Chapter 4

Experiments conducted to implement MPC

Experiments were performed as shown in table above for implementation of MPC. We carried out experiments in which both positive and negative step changes were given to Set point and Fan (disturbance variable) and the output response was obtained by application of MPC. We also have performed several experiments to study the effect of change in the values of q (control horizon) and tuning parameters - error and manipulated variable weighting factors.

The details of the experiments mentioned in this report has been tabulated in the table given in the next page. The first column of the table represents the experiment version (or number). For all the outputs and their figures, we have mentioned only their experiment version (or number) to tag them. Also note that the data files for these experiments are also named as per their experiment version number.

p and q mentioned in the table represents the prediction and control horizon respectively.

Please note: For all the above experiments and graphs, we adhered to:

- Scilab Version: 5.2.2
- SBHS number: 12 (remotely accessed)
- Sampling time: 1 second

For graphs: Until and unless mentioned, Graphic 1 represents the Temperature set point, Graphic 2 represents the Fan and Graphic 3 represents the Temperature.

Also, please note that there are two types of graphs. The first graph, containing Graphic 1, Graphic 2 and Graphic 3 were directly obtained via mscope of xcos. The graph following this in all the experiments is the temperature and heater value graphs, which were obtained from the data (from the text file downloaded from the server after each experiment).

Expt No	Temperature Set point			Fan			(p,q)	Weighing factor (We, Wu)
	T_initial (°C)	T_final (°C)	Time (s)	F_initial	F_final	Time (s)		
1.1	35	40	250	100	150	500	(40,4)	1,1
1.2	35	40	250	100	150	500	(40,4)	10,10
1.3	35	40	250	100	150	500	(40,4)	40,40
2.1	42	37	250	150	100	500	(40,4)	1,1
2.2	42	37	250	150	100	500	(40,4)	10,10
2.3	42	37	250	150	100	500	(40,4)	40,40
3.1	35	40	250	100	150	500	(40,2)	10,10
3.2	35	40	250	100	150	500	(40,3)	10,10
3.3	35	40	250	100	150	500	(40,4)	10,10
4.1	42	37	250	150	100	500	(40,2)	10,10
4.2	42	37	250	150	100	500	(40,3)	10,10
4.3	42	37	250	150	100	500	(40,4)	10,10
5.1	35	40	250	100	150	500	(40,4)	100,2
5.2	35	40	250	100	150	500	(40,4)	2,100
5.3	35	40	250	100	150	500	(40,4)	10,100
5.1	35	40	250	100	150	500	(40,4)	100,10

Figure 4.1: Experiments performed

All the experiments mentioned in this report has been labeled as shown in this table. This table is just a summary of all the parameters that was used for the corresponding experiment. Details on the inputs and a description of the output observed for each case has been mentioned in the corresponding section of each experiment.

Chapter 5

Sample run to implement MPC

5.1 Positive Step Change to Set Point and Fan

Let us consider experiment 1.1, wherein, a positive step change of 5° C (from 35° C to 40° C) was provided to set point at time t=250 s and a step change to fan was provided at t = 500 s, from 100 to 150.

The graph obtained has been attached below:

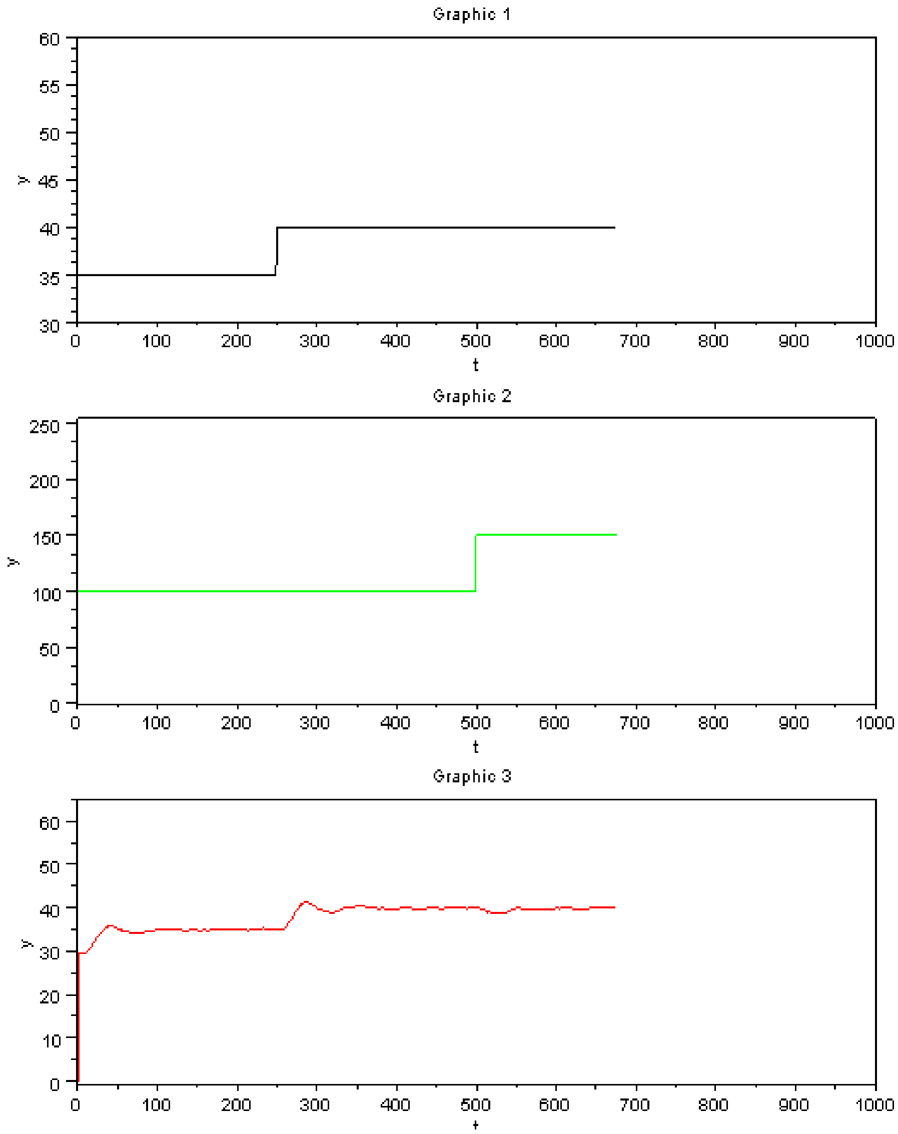


Figure 5.1: Expt 1.1

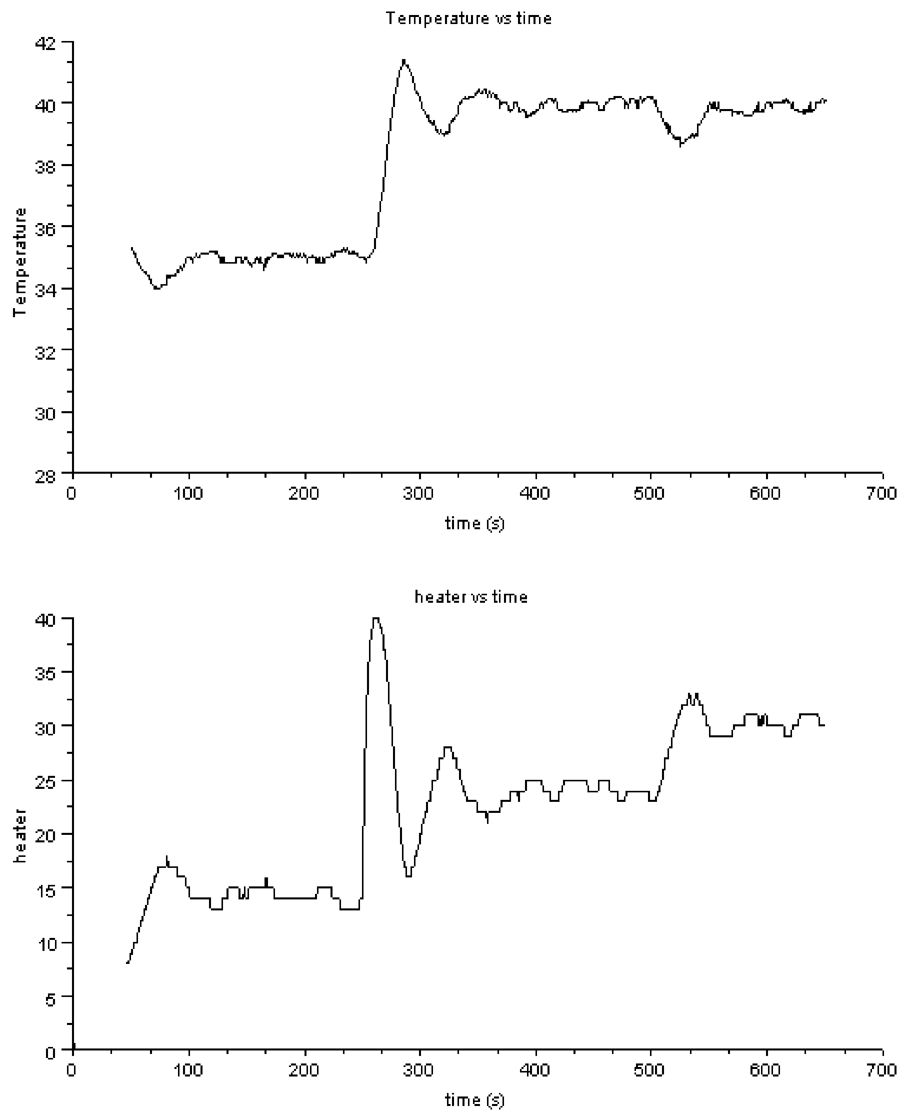


Figure 5.2: Expt 1.1

As can, be seen above, when, the temperature set point is raised to 40 from 30, at t=250 s, the value of the heater increases, so that it can heat up the plant upto the required set point. Similarly, when the fan speed is increased at t=500s, the heater value increases yet again to maintain the same constant temperature of the SBHS blade.

5.2 Negative Step Change to Set Point and Fan

Let us consider experiment 2.1, wherein, a negative step change of 5°C (from 42°C to 37°C) was provided to set point at time t=250 s and a step change to fan was provided at t = 500 s, from 150 to 100.
The graph obtained has been attached below:

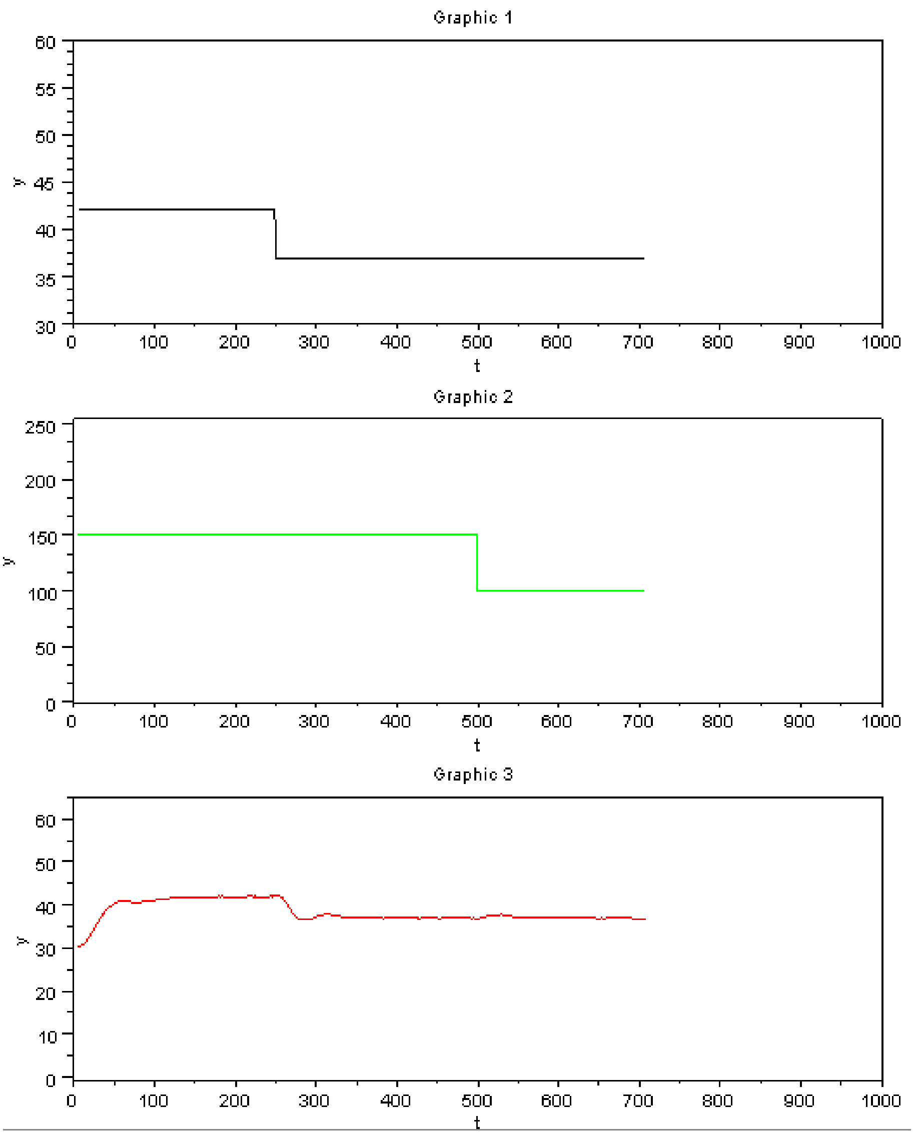


Figure 5.3: Expt 2.1

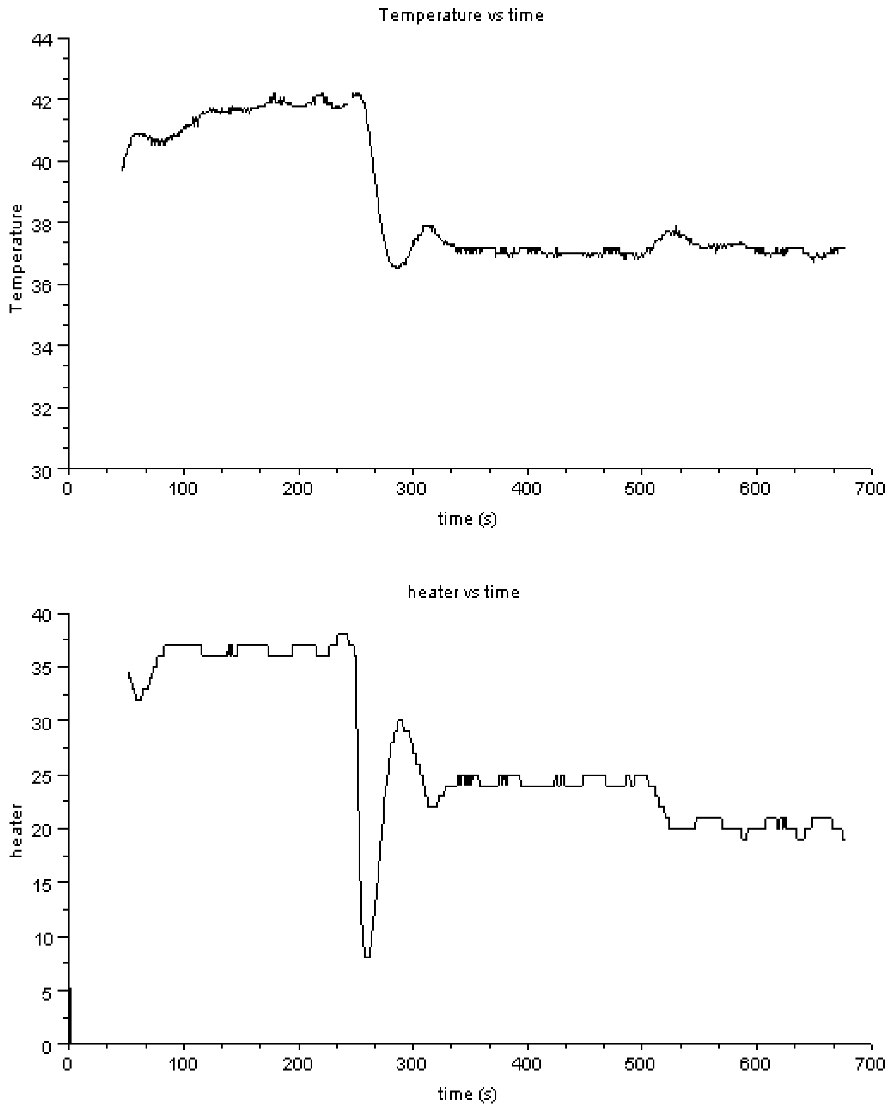


Figure 5.4: Expt 2.1

As can be seen from the graphs above, when the temperature set point drops at t=250 s, the value of the heater too falls, so that the plant (SBHS blade) can cool down to the required set point. Similarly, when the fan speed was decreased at t=500s, the heater value decreased yet again to maintain the same constant temperature of the SBHS blade.

Chapter 6

Effect of Tuning parameters: Weighting factors, We and Wu

We also, conducted several experiments in order the study the effect of the value of Weighting factors (both error,We and manipulated variable,Wu). We used weighting factors to be 1, 10 and 40 for both positive and negative step changes to both set point and fan (as has been summarized in Table 1). Also, experiments were done for different values of We and Wu. The results have been shown in the following graph.

6.1 For same factor of We and Wu

6.1.1 Positive Step Change and (We, Wu)=(1,1) (Expt 1.1)

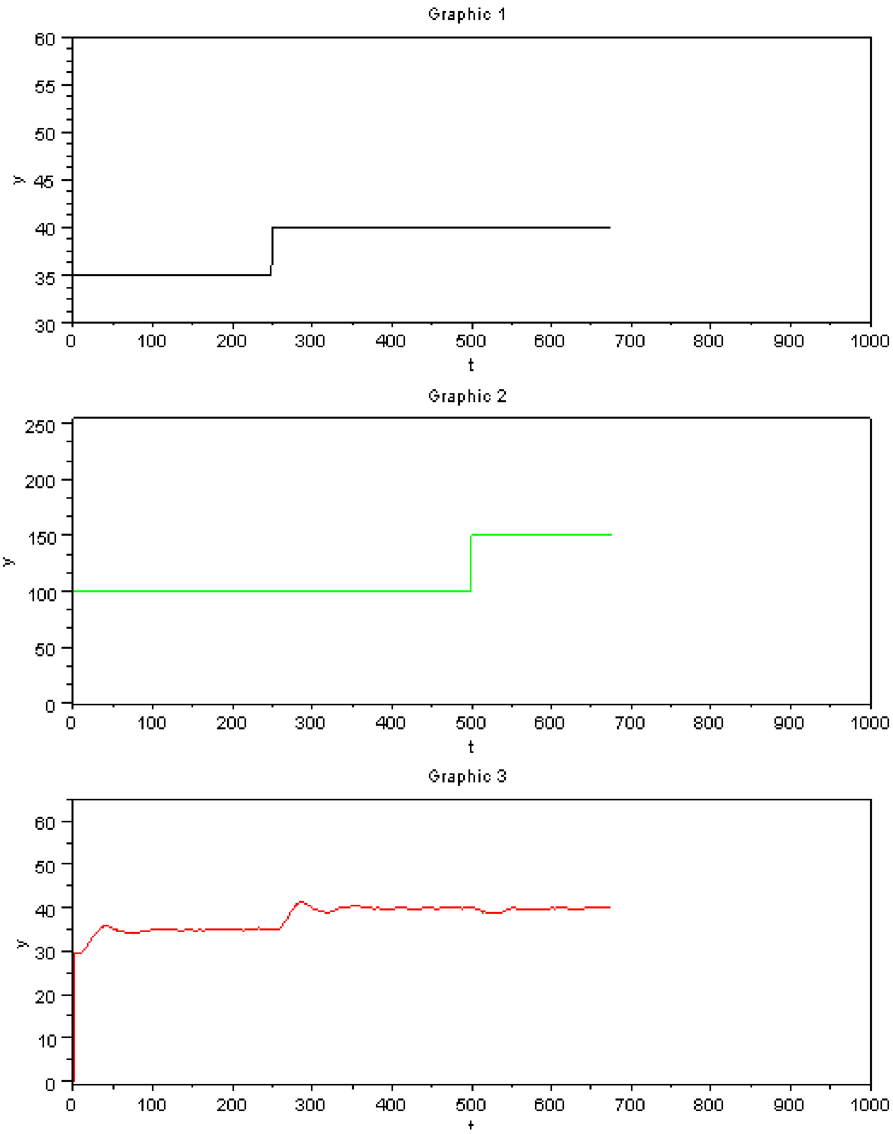


Figure 6.1: Expt 1.1

Here we can clearly see the expected output. Providing a positive step to temperature set point at 250 seconds, increased heater value as per the control effort put in by MPC. A positive step in fan at 500 seconds, decreased the temperature below its set point and hence heater value increased to take the temperature close to its setpoint.

This will be clear from the heater graph attached next.

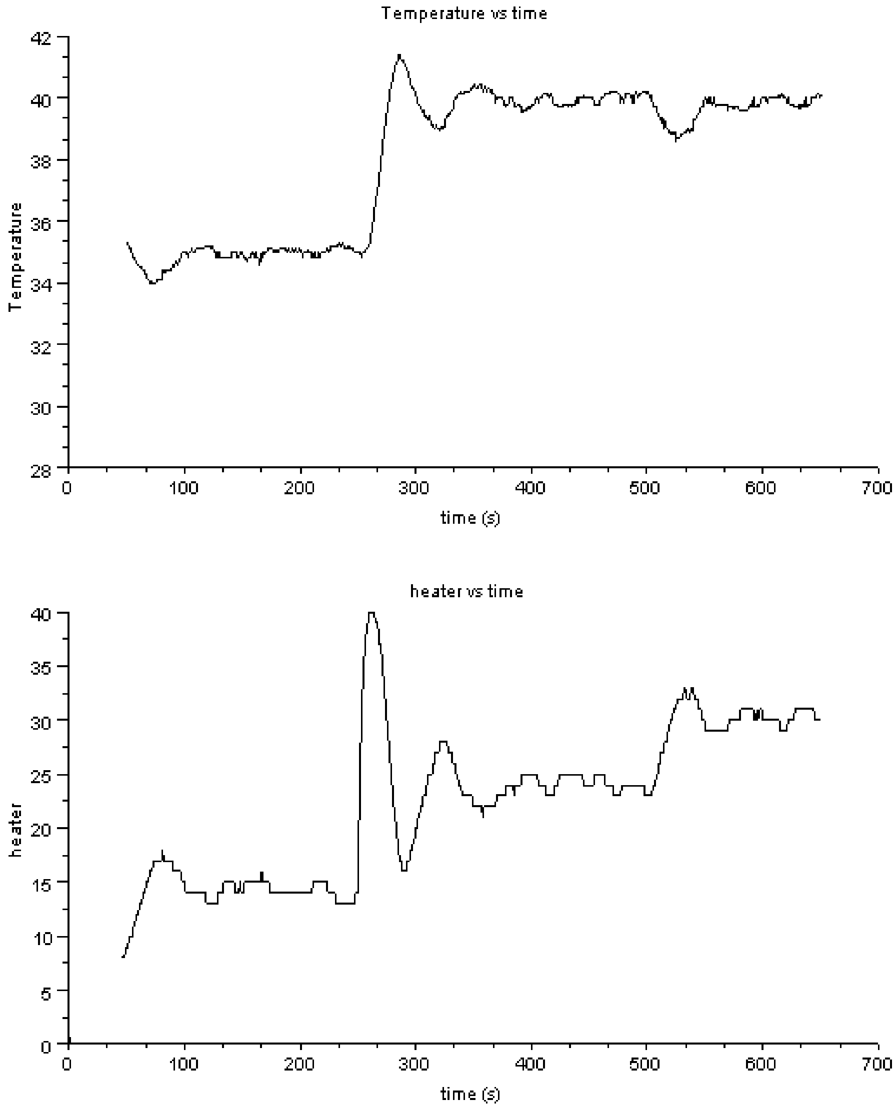


Figure 6.2: Expt 1.1

As can be clearly seen, the heater graph follows the expected trend that we talked of in the last page. Also, note that the temperature variation can be clearly seen from this graph.
This graph shows the result for the case, where we had same weighting factors for both error and manipulated variables (We and Wu). We will now see if changing both of these is going to have any effect on the control behavior.
So, we now try an experiment with both We and Wu increased to 10.

6.1.2 Positive Step Change and (We, Wu)=(10,10) (Expt 1.2)

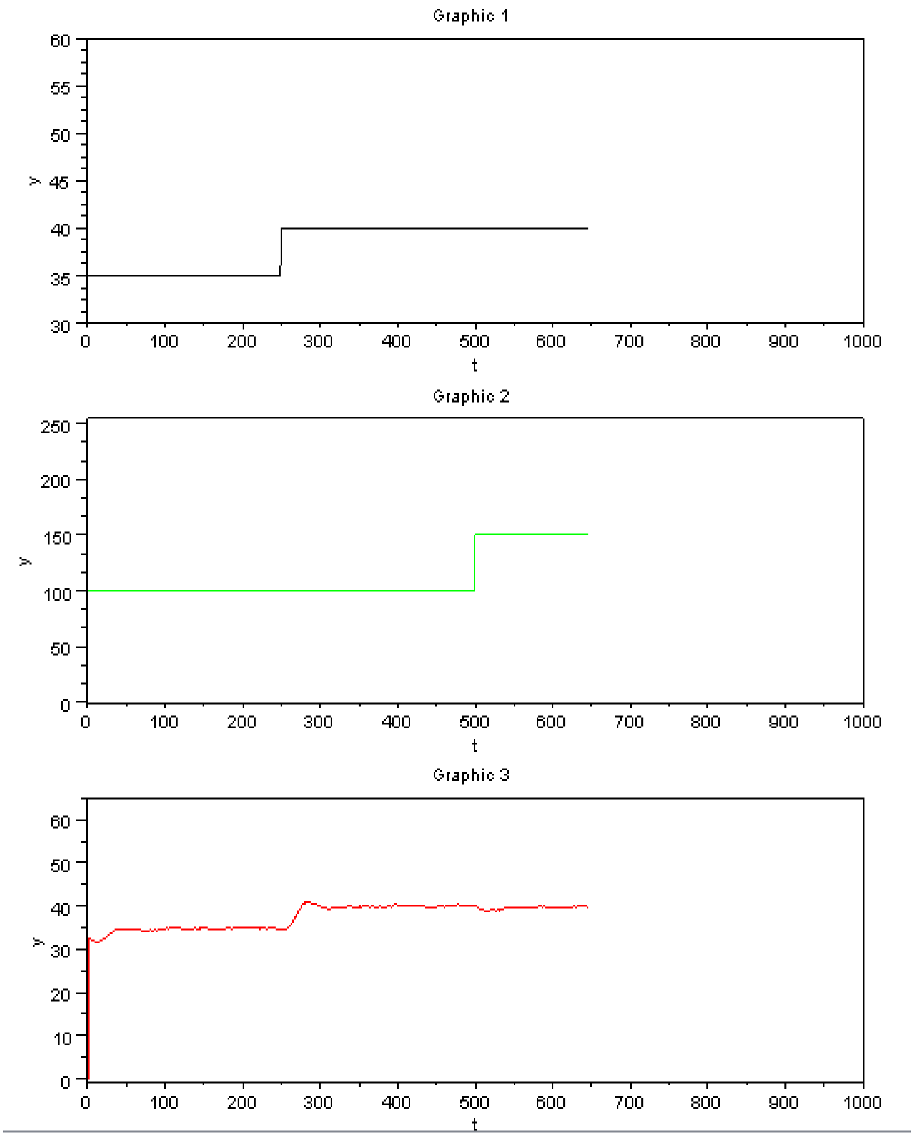


Figure 6.3: Expt 1.2

Using the same logic as has been explained in the last section, we expected to see similar temperature and heater value profiles for the positive step change in temperature set point and the fan. (Heater graph is shown in the next page along with the temperature on an expanded scale).
In this experiment, we increased We and Wu both to 10 from 1 and wish to observe if this changes the response of the plant.

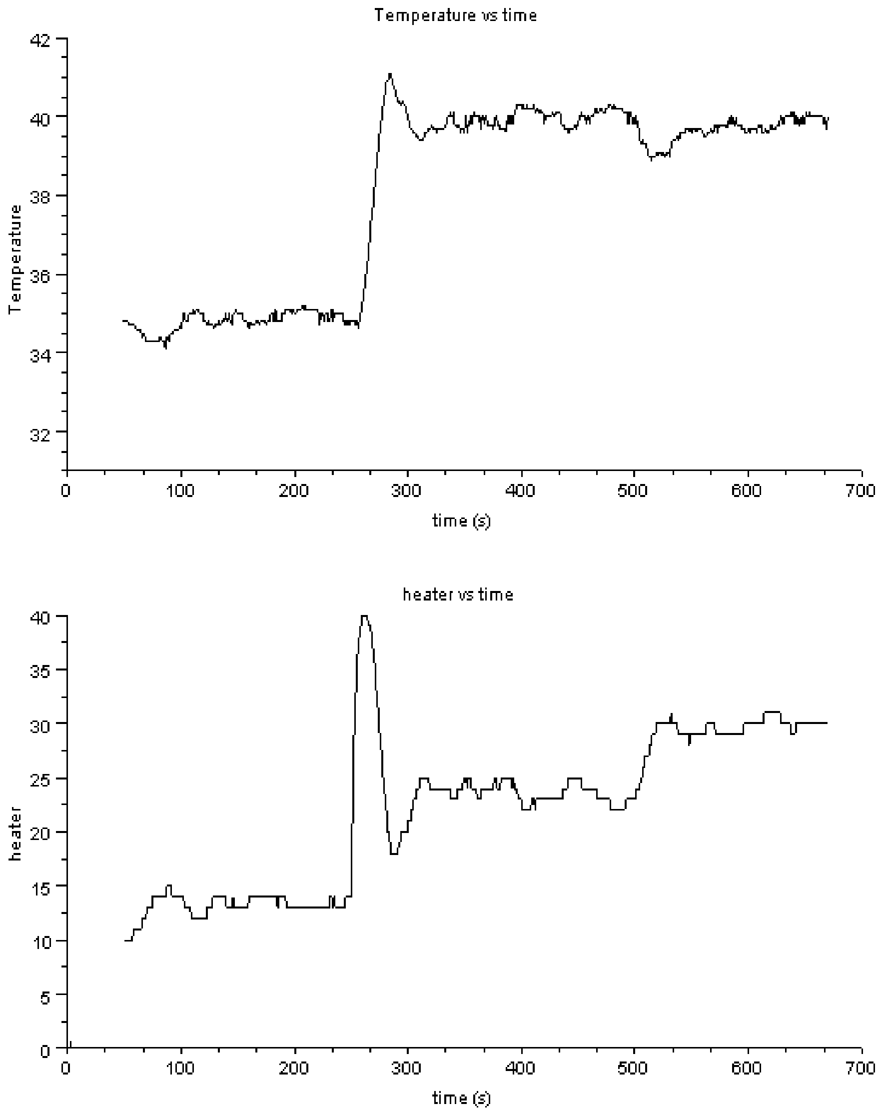


Figure 6.4: Expt 1.2

The results here are almost the same as that mentioned in the last section (where We and Wu both were 1). So, we can for the time being keep in mind that We and Wu isn't actually much affected the output. We now will carry out the experiment for even higher We and Wu (say 40) and see if it really does affect the output much.

6.1.3 Positive Step Change and (We, Wu)=(40,40) (Expt 1.3)

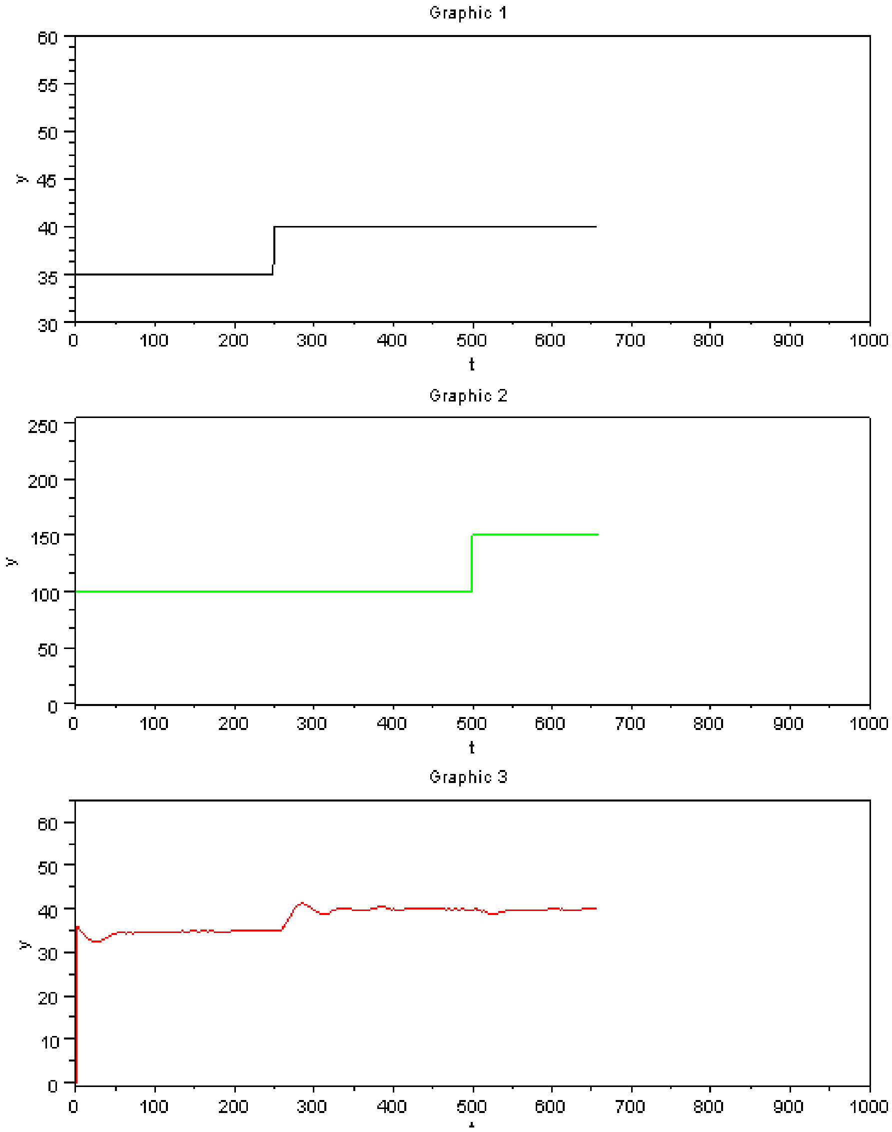


Figure 6.5: Expt 1.3

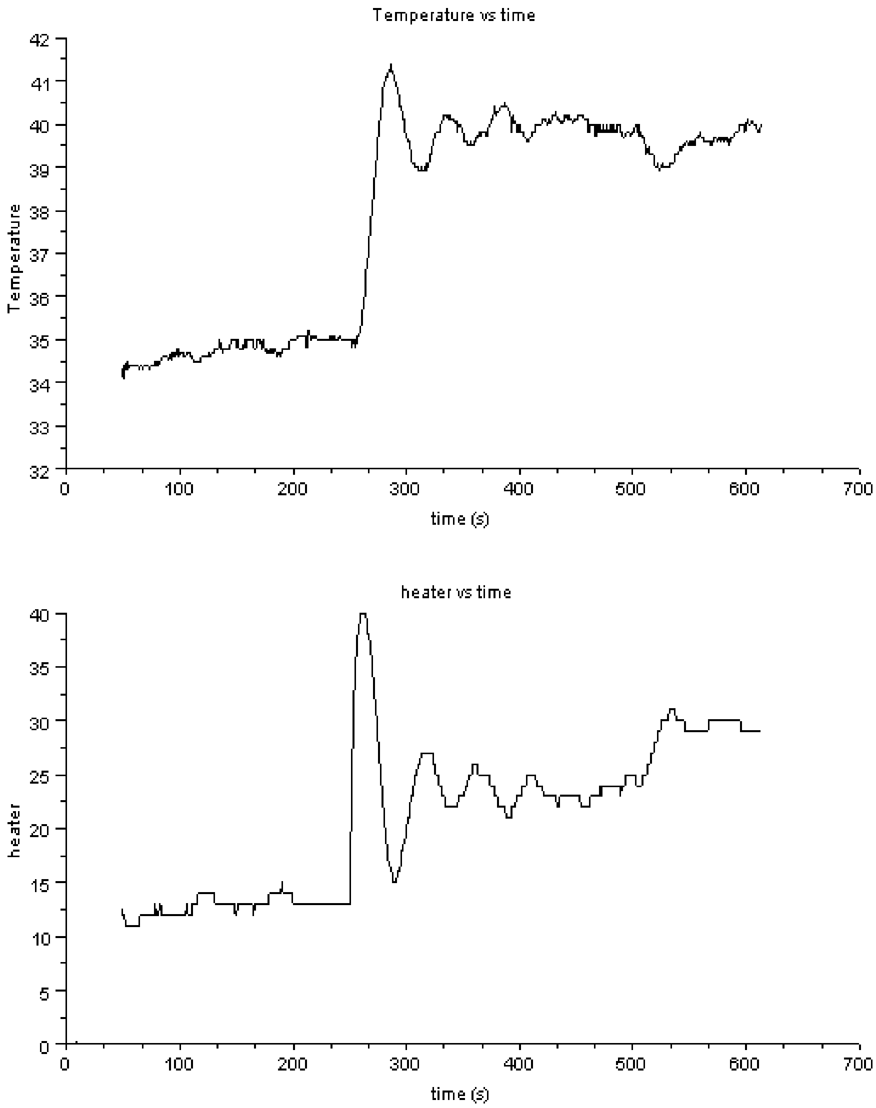


Figure 6.6: Expt 1.3

Even the results with We and Wu as 40 doesn't show much difference. They are more or less similar looking as the last two experiment's results.

6.1.4 Negative Step Change and (We,Wu)=(1,1) (Expt 2.1)

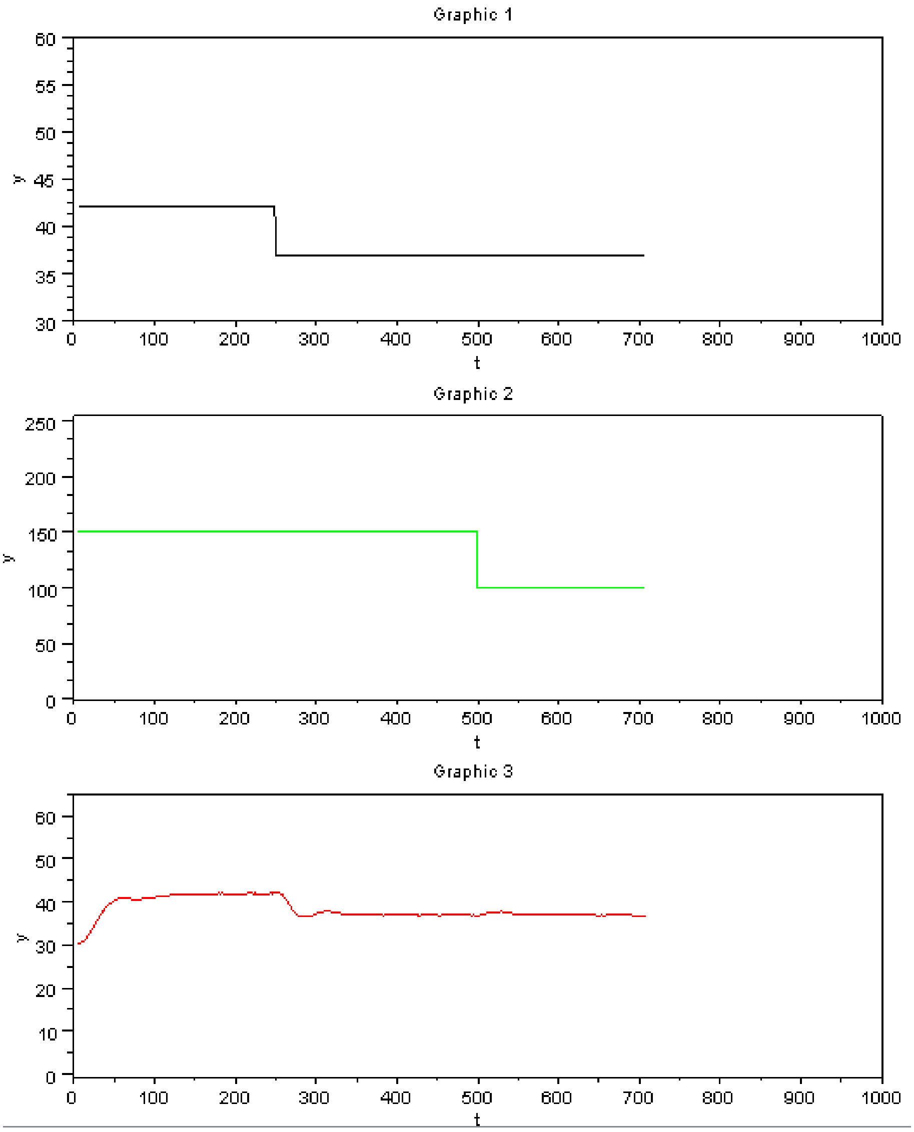


Figure 6.7: Expt 2.1

Here we expect somewhat similar results as was the case with positive step in temperature set point.

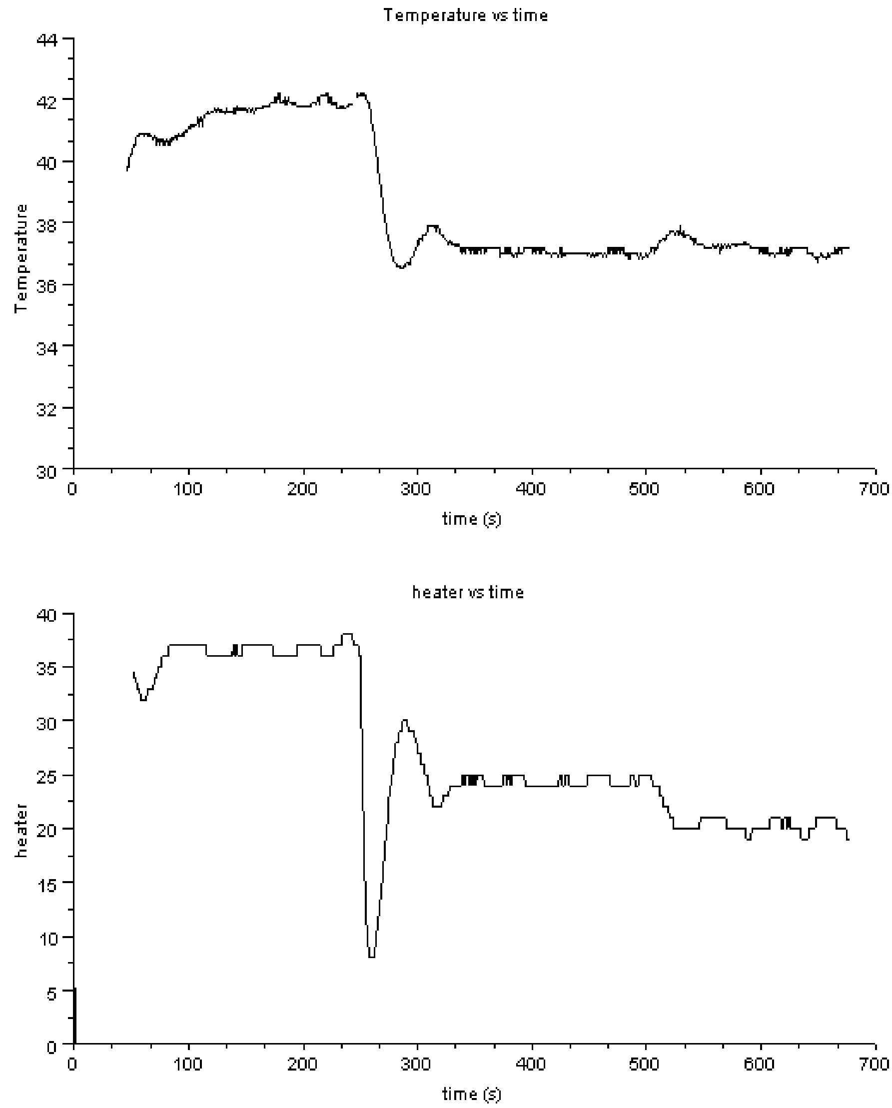


Figure 6.8: Expt 2.1

We can very clearly make out that the results follow the trends as was explained for the negative step input in the section 5.2

6.1.5 Negative Step Change and (We, Wu)=(10,10) (Expt 2.2)

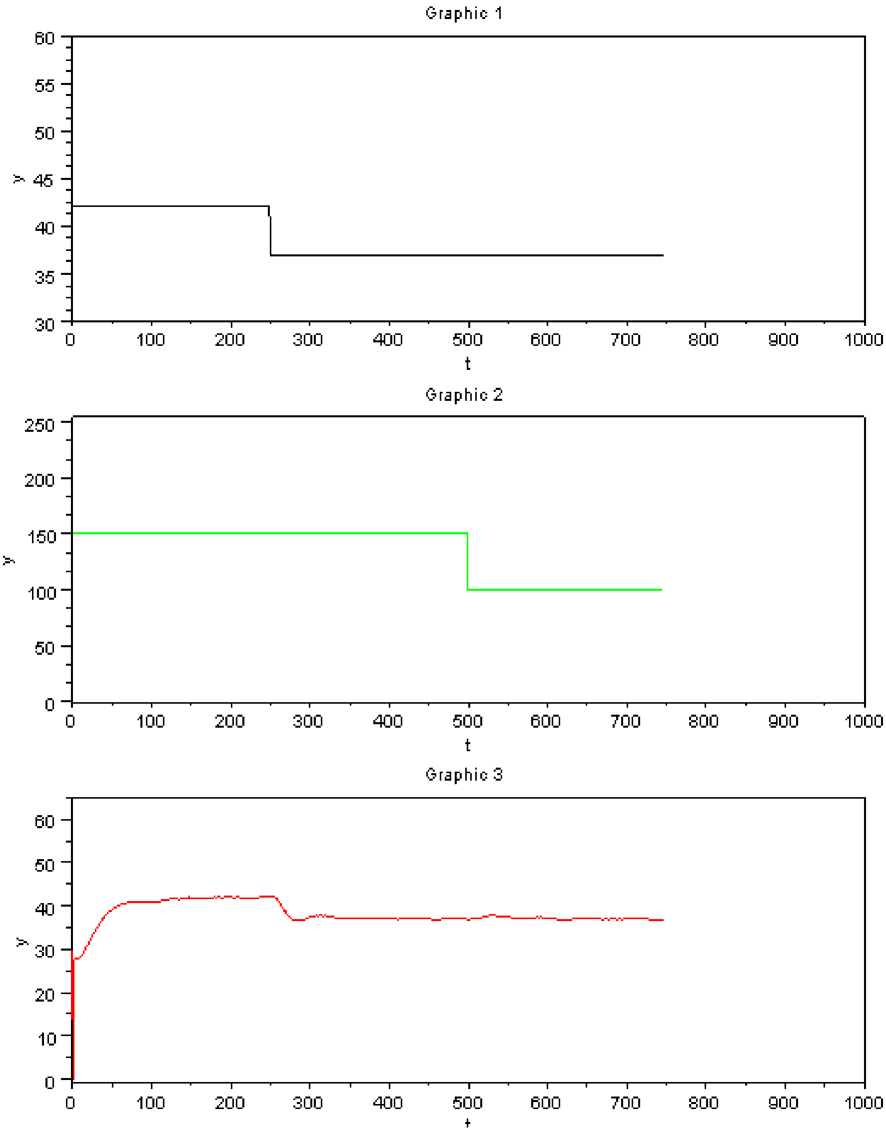


Figure 6.9: Expt 2.2

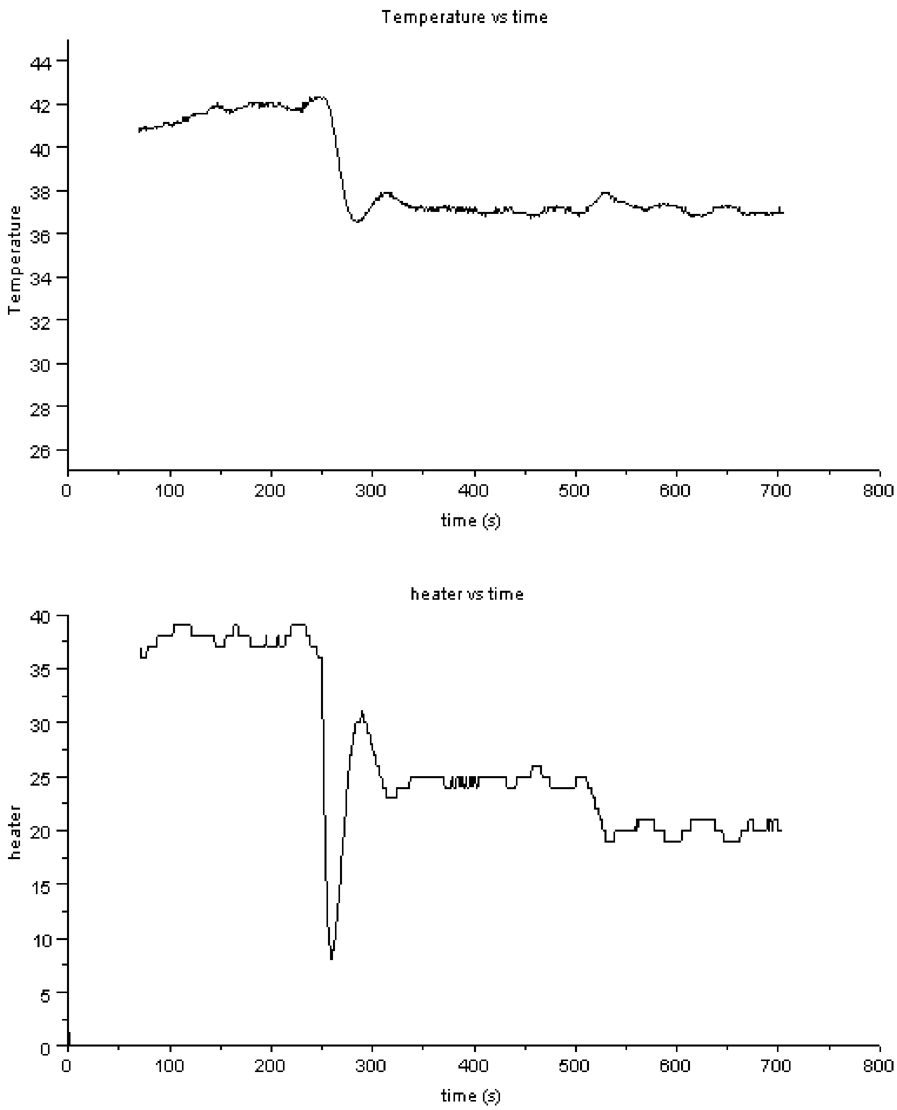


Figure 6.10: Expt 2.2

6.1.6 Negative Step Change and (We, Wu)=(40,40) (Expt 2.3)

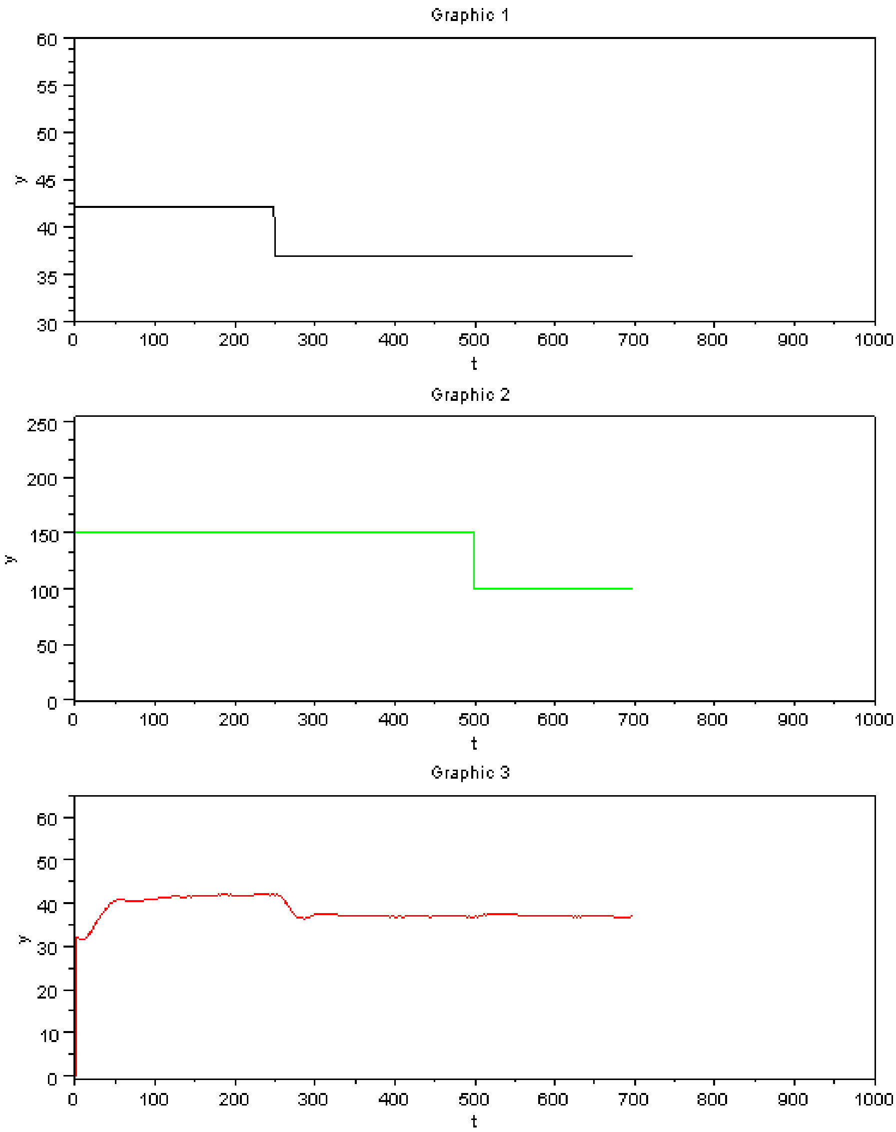


Figure 6.11: Expt 2.3

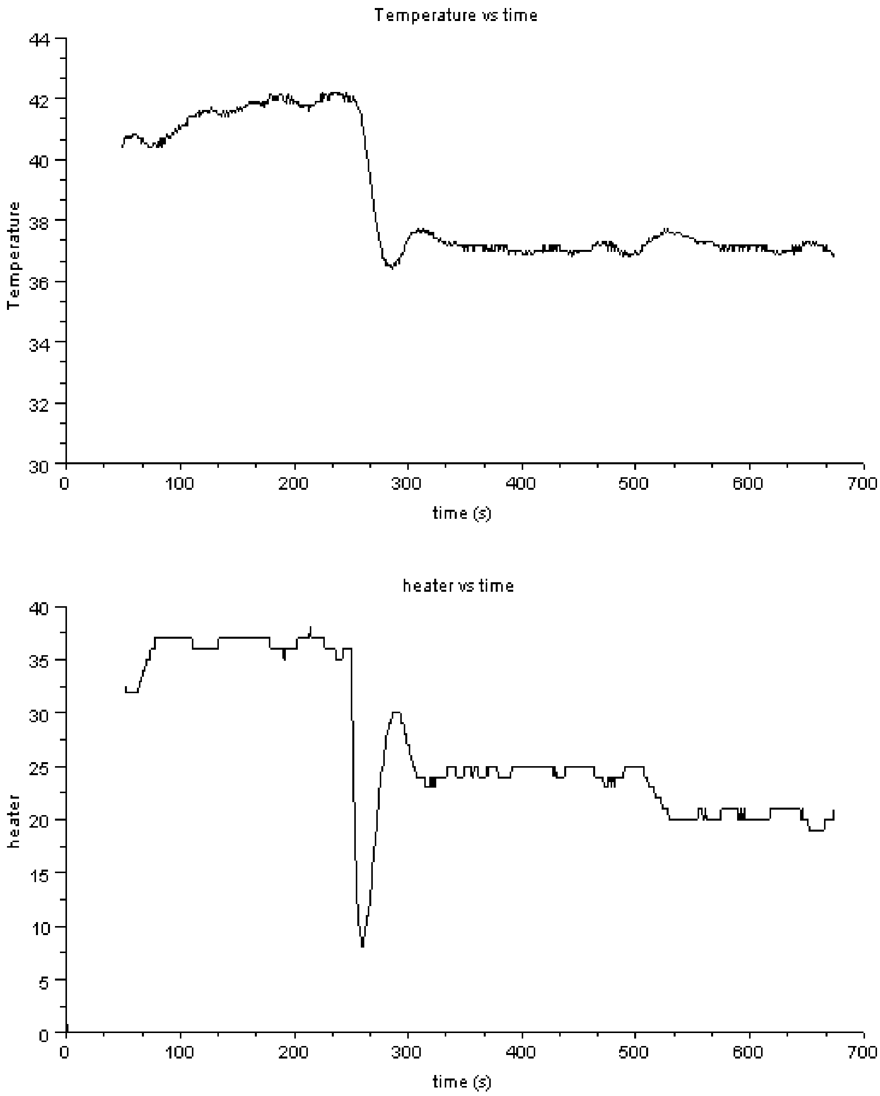


Figure 6.12: Expt 2.3

6.2 For different We and Wu factors

We very clearly see that using the same values of We and Wu is not making much difference in the control response. So, will now be trying different values for We and Wu.

6.2.1 We =100 and Wu = 2 (Expt 5.1)

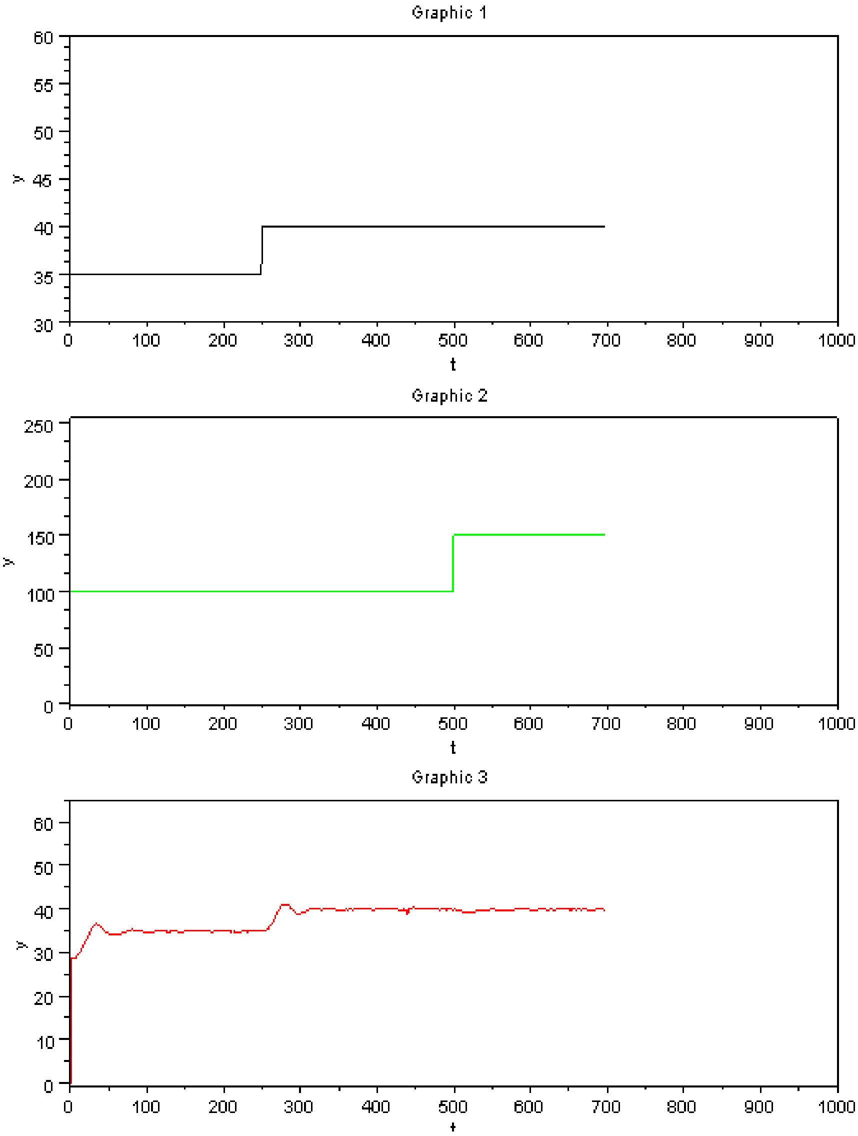
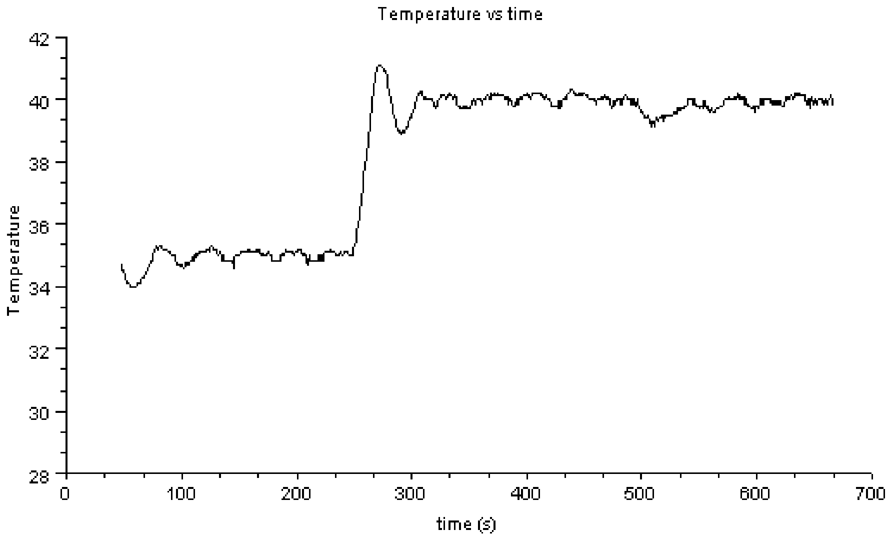


Figure 6.13: Expt 5.1

Here, we have used We as 100 and Wu as 2. The response after the positive step in temperature set point is slightly oscillatory. The temperature very well stabilzes at the required setpoint. The settling time observed is fairly low.



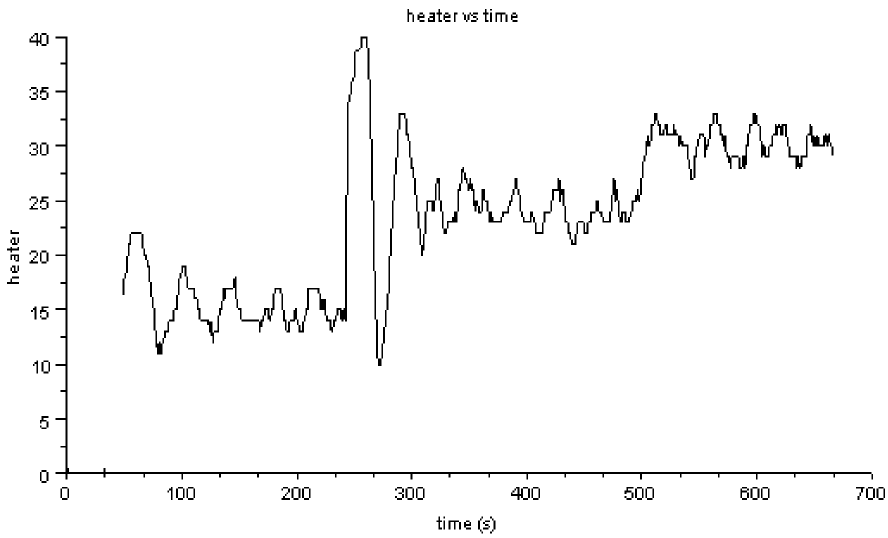


Figure 6.14: Expt 5.1

Now having seen the results of this experiment, we would like to check the possible effect of reversing the values of We and Wu. So, we conduct the next experiment, in which we have We as 2 and Wu as 100.

6.2.2 We =2 and Wu = 100 (Expt 5.2)

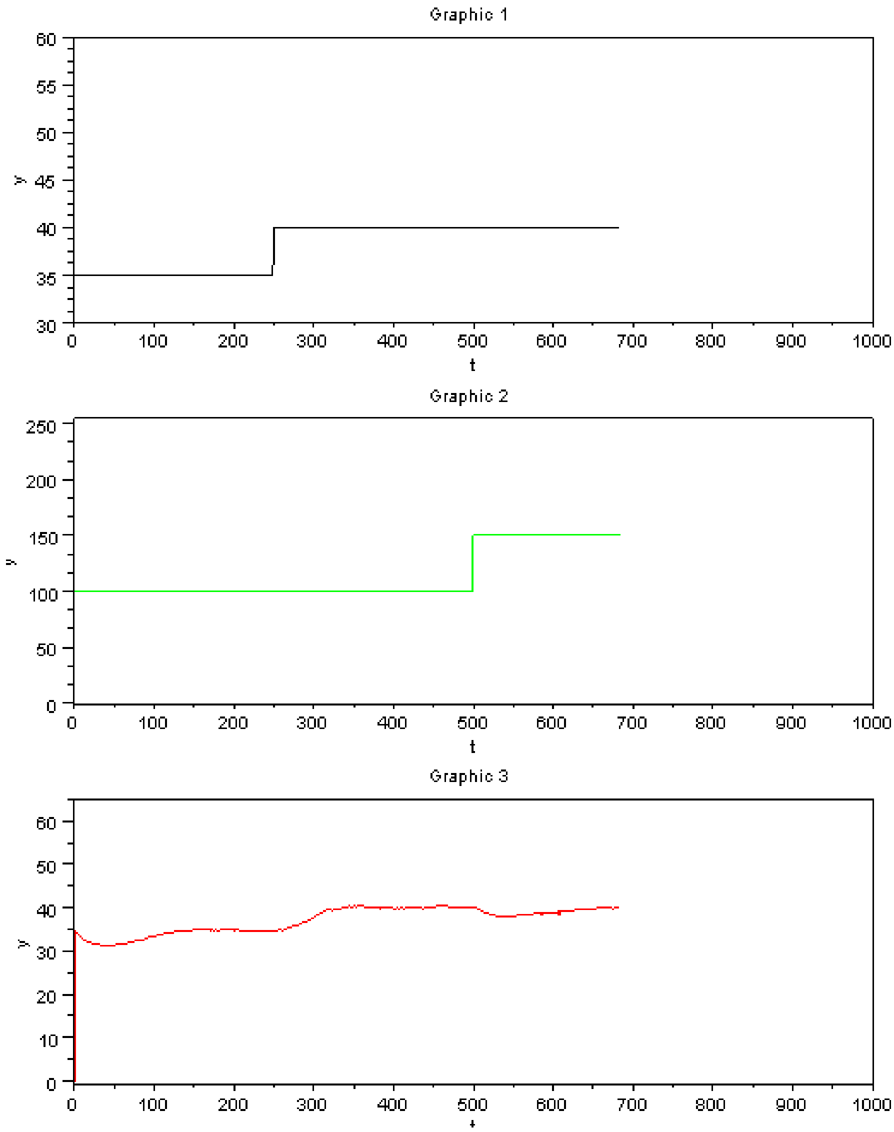


Figure 6.15: Expt 5.2

With increase in Wu, we observe that the temperature stabilzes at the required ste-point, but the settling time for reaching that setpoint increases. Also, the response is not oscillatory. This result can be very clearly seen in the following temperature and heater graph.

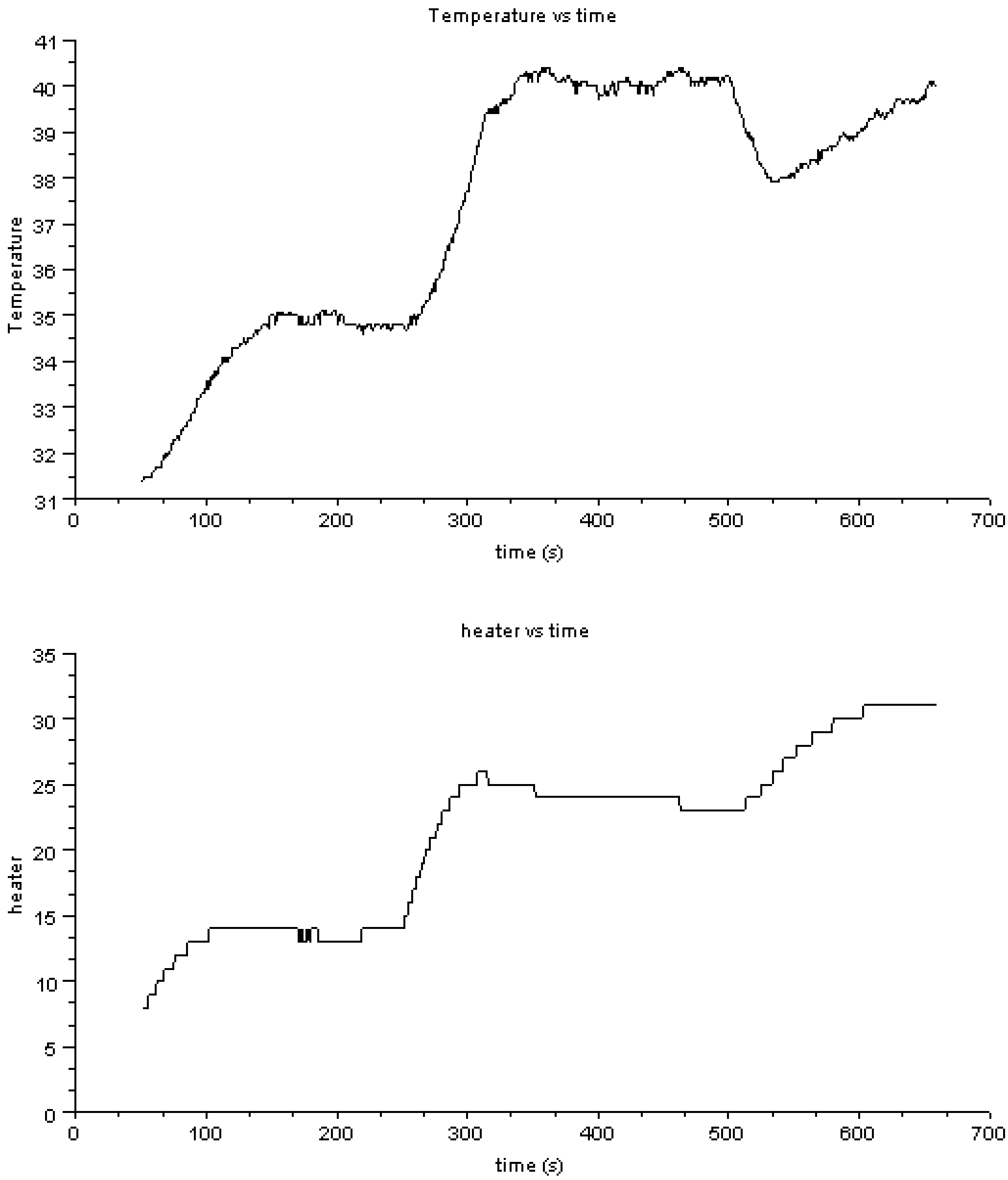


Figure 6.16: Expt 5.2

6.2.3 We =10 and Wu = 100 (Expt 5.3)

Having seen the effect of low We and high Wu (in the last section), we would like to see what happens if We is slightly increased keeping Wu the same. For this we increase the value of We to 10 and keep Wu at constant 100.

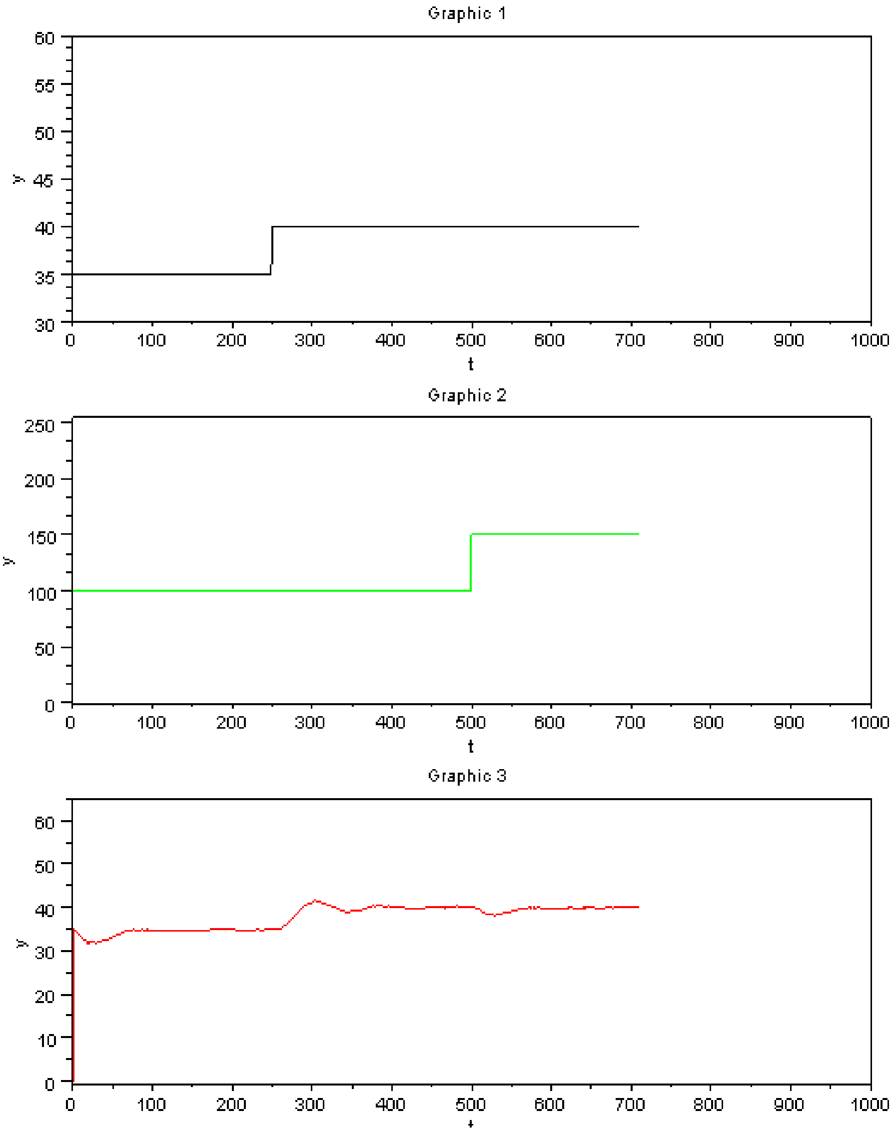


Figure 6.17: Expt 5.3

We observe that this experiments performs better than in the last section (where We was 2). It is slighly oscillatory and also, the settling time decreased much as compared to last experiment.

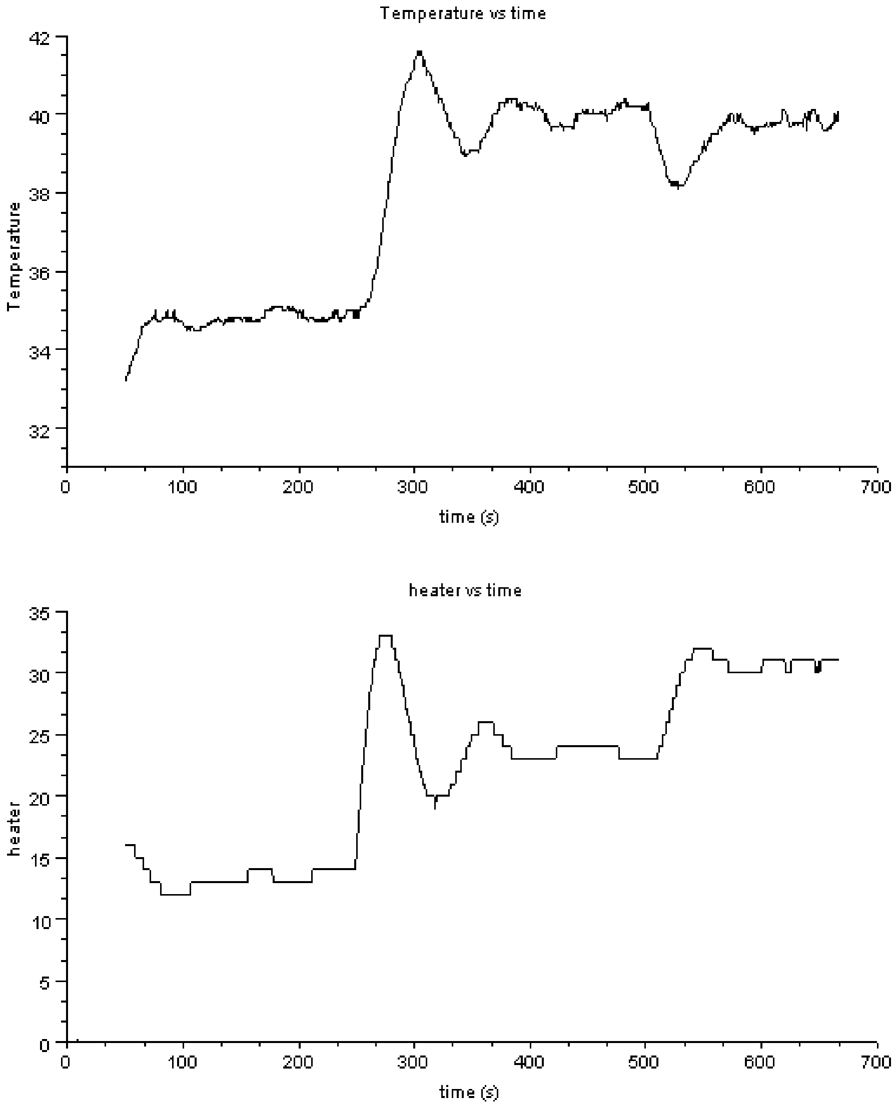


Figure 6.18: Expt 5.3

6.2.4 We =100 and Wu = 10 (Expt 5.4)

We now do a similar study for the case of Wu. We increase the value of Wu to 10, keeping We constant at 100.

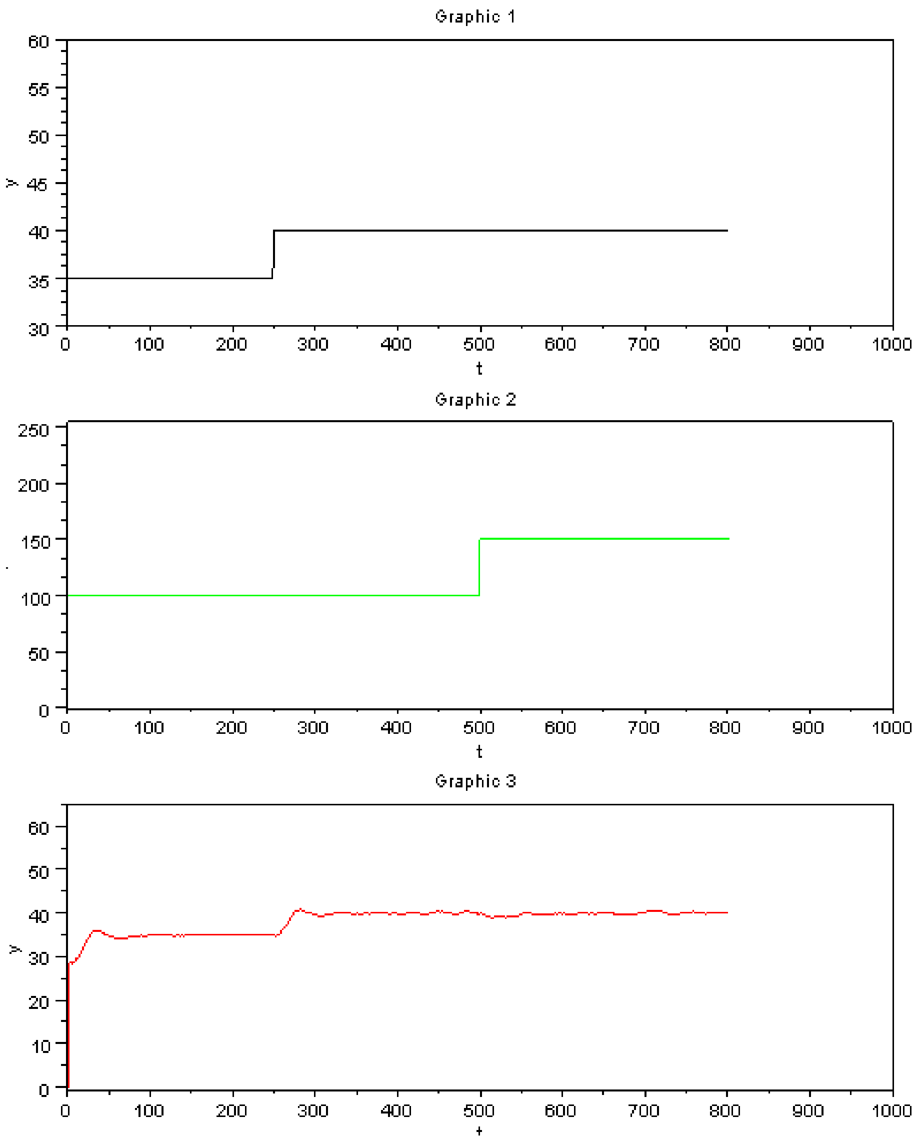


Figure 6.19: Expt 5.4

As is clear from the figure, we see slightly lesser oscillations compared to the case when We was 100 and Wu was 2. Settling time more or less remained the same.

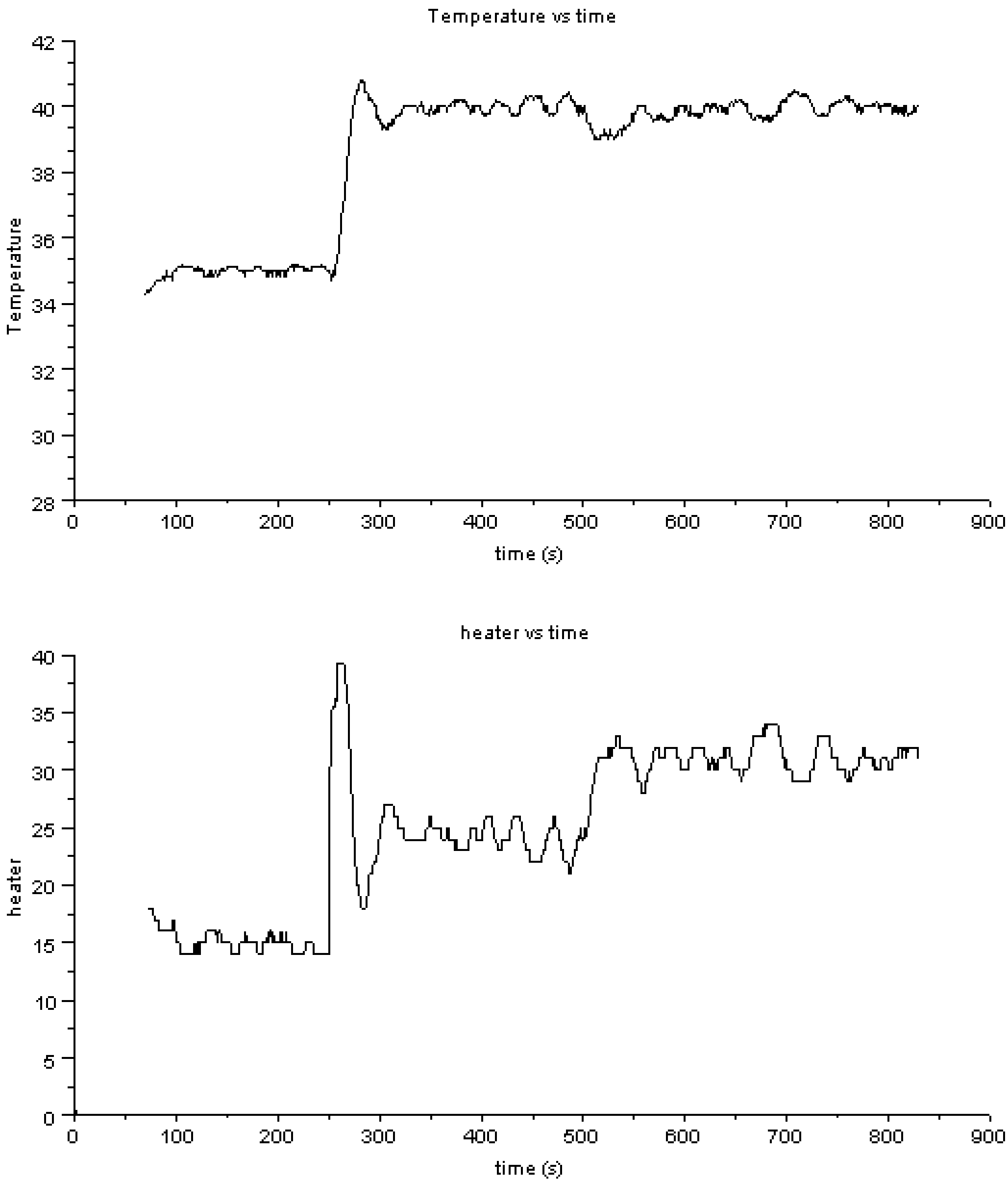


Figure 6.20: Expt 5.4

6.3 Conclusion on Weighting factor experiments

For experiments with same values of We and Wu:

- Not much difference was seen in heater value trends and temperature value trends for all the experiments performed above.
- Reason for this will be clear from the discussion on the trends mentioned below (for experiments with different values of We and Wu)

For experiments with different values of We and Wu:

- Keeping We as large (around 100) and Wu as small (2) shows better performance as compared to the case when the values are kept the other way around.
- With We very small (say around 1-2), oscillations are less, and the settling time observed was found to be more.
- With increase in We, the oscillations were observed to increase and the settling time was found to reduce and hence, better control was observed.
- So, with increase in We, any error is quickly dealt with, because with increase in We, we are actually increasing the significance of change of temperature in deciding the control action.
- With increase in Wu, oscillations reduced and the settling time was found to increase and hence, less preferred.

So, the best performance is obtained for the cases with high We and low Wu.

Chapter 7

Effect of Control Horizon Paramter, q

We also tried to study the effect of change of control horizon (q) on the response of the SBHS to step change in Setpoint and disturbance variable. Generally the value of q (control parameter) is taken somewhere between 2 to 5. So, we performed our SBHS experiment for values of q as 2, 3 and 4 (as suggested by Mr Prashant Gupta).

Both positive and negative step change experiments for temperature set point and disturbance variable (fan) was performed for the sake of completeness. The results obtained thereby has been mentioned in form of graphs in this section. The overall conclusion over these experiments has been mentioned in the conclusion of this part.

7.1 For positive step change in Set point and Fan speed

7.1.1 For q =2 (Expt 3.1)

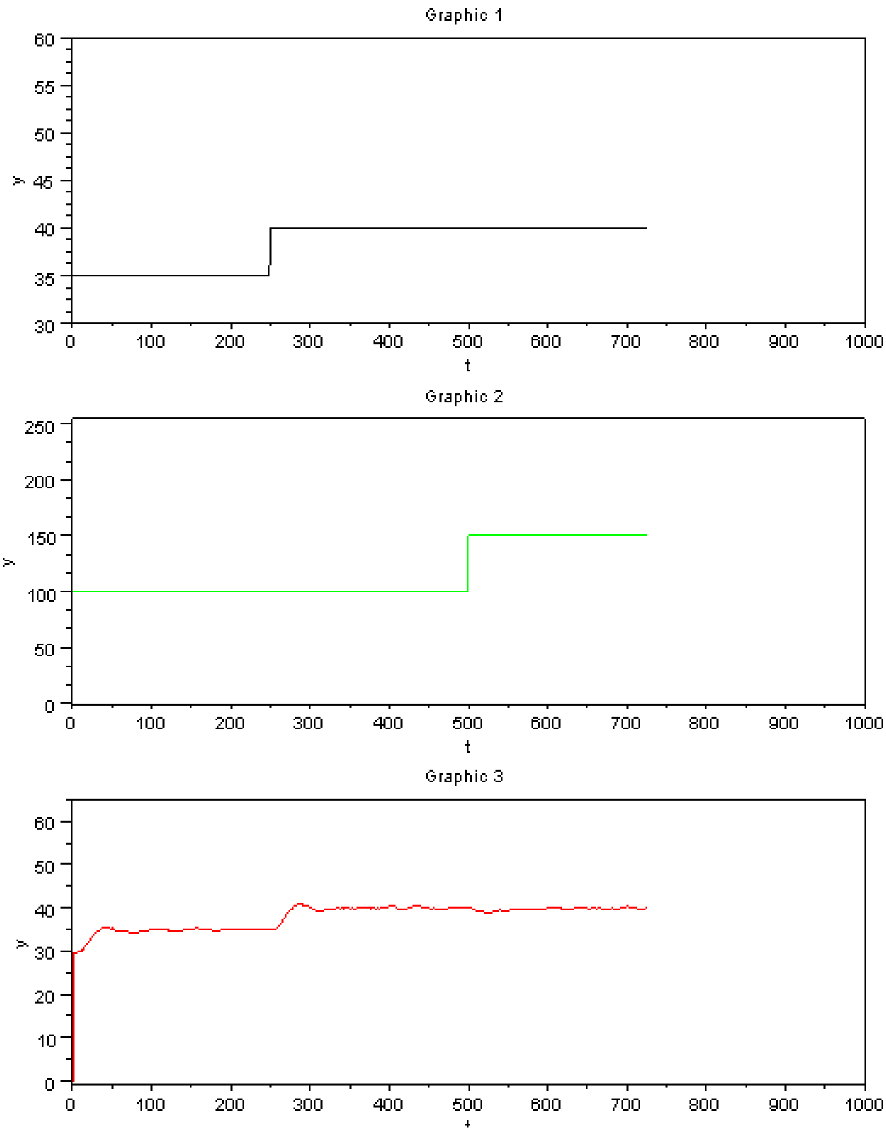


Figure 7.1: Expt 3.1

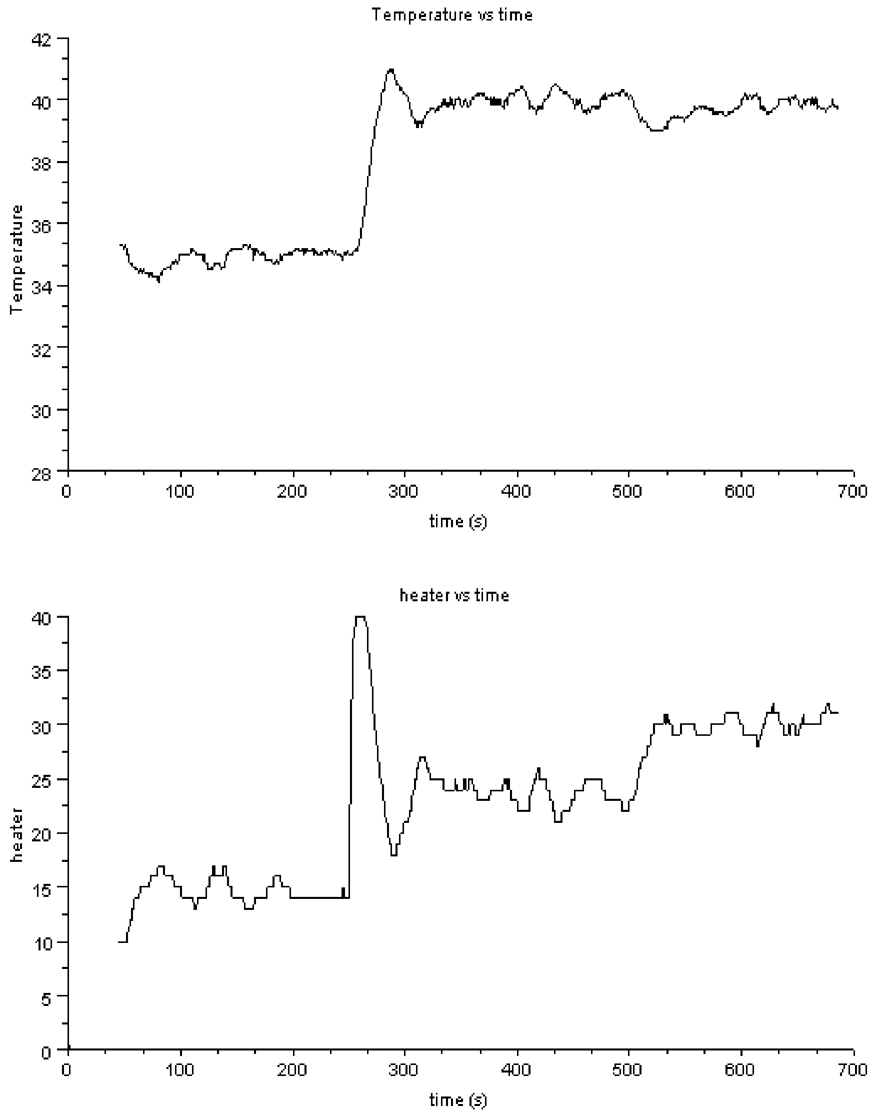


Figure 7.2: Expt 3.1

7.1.2 For q =3 (Expt 3.2)

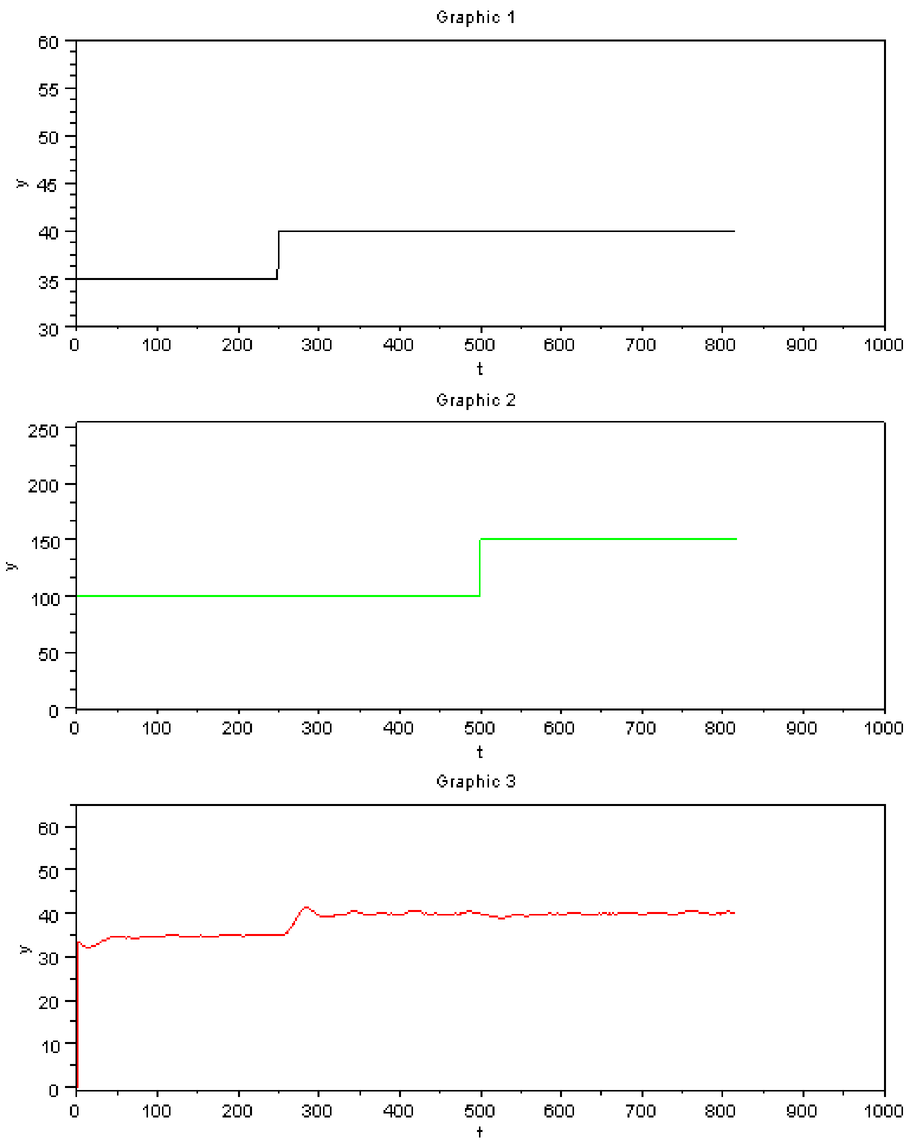


Figure 7.3: Expt 3.2

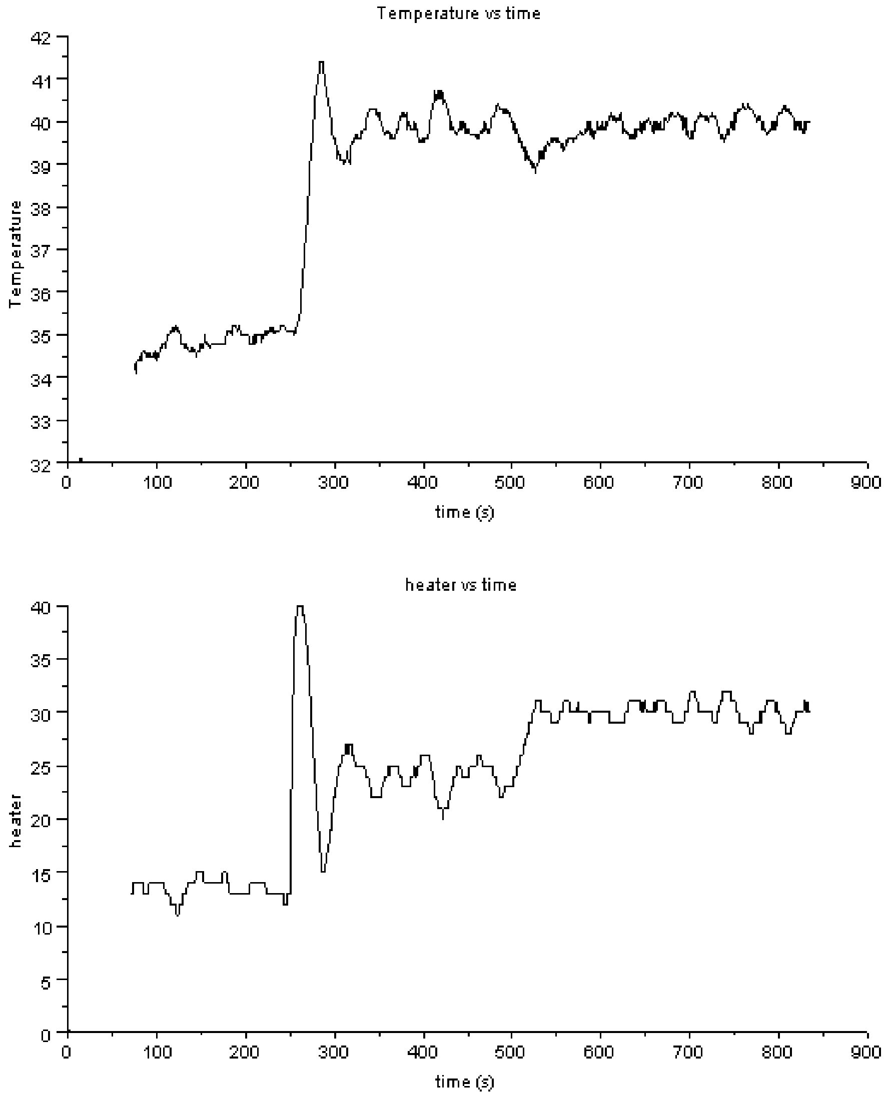


Figure 7.4: Expt 3.2

7.1.3 For q =4 (Expt 3.3)

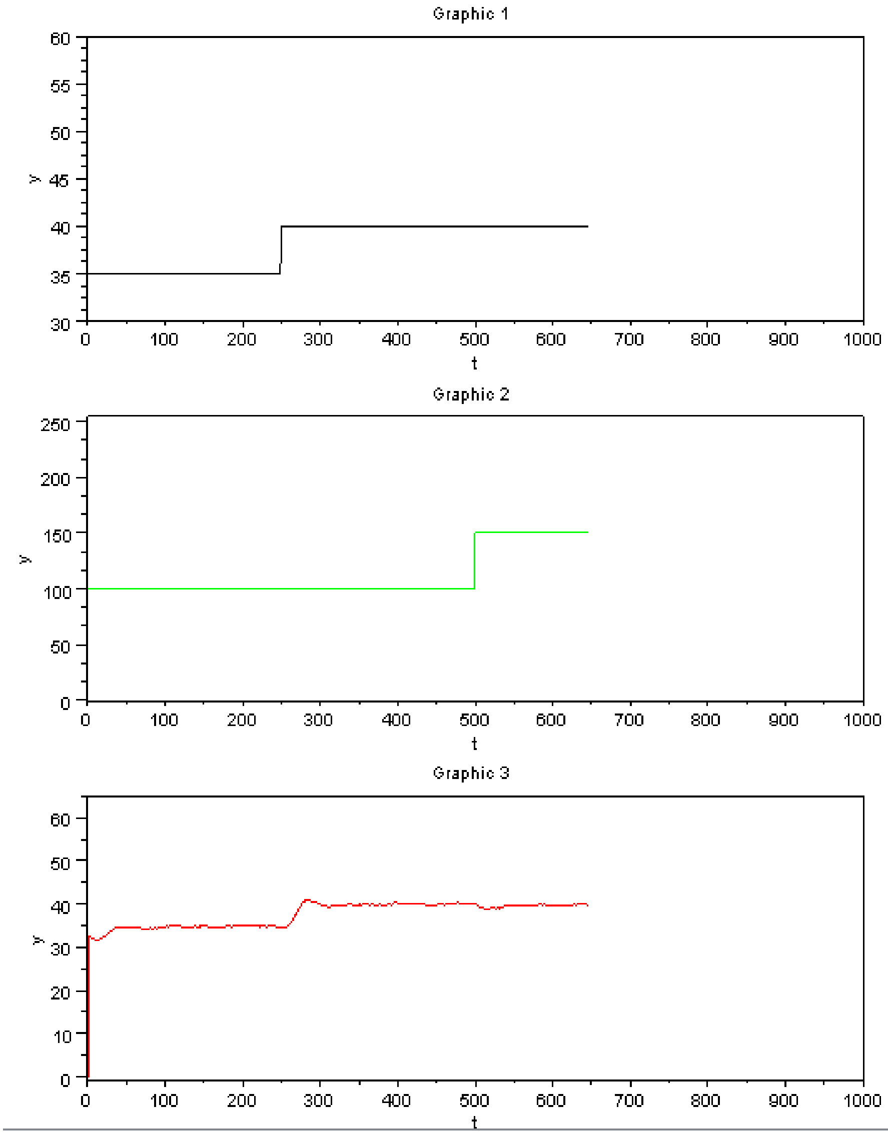


Figure 7.5: Expt 3.3

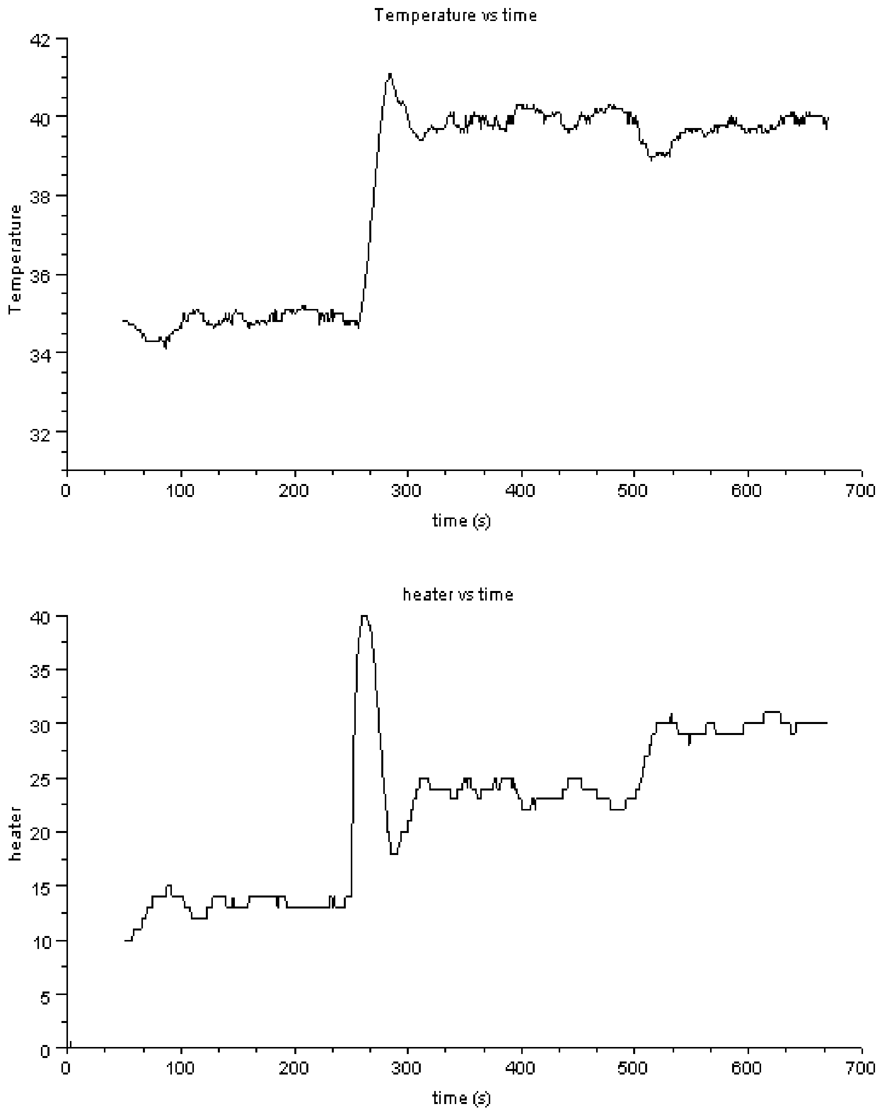


Figure 7.6: Expt 3.3

7.2 For negative step change in Set point and Fan speed

7.2.1 For q =2 (Expt 4.1)

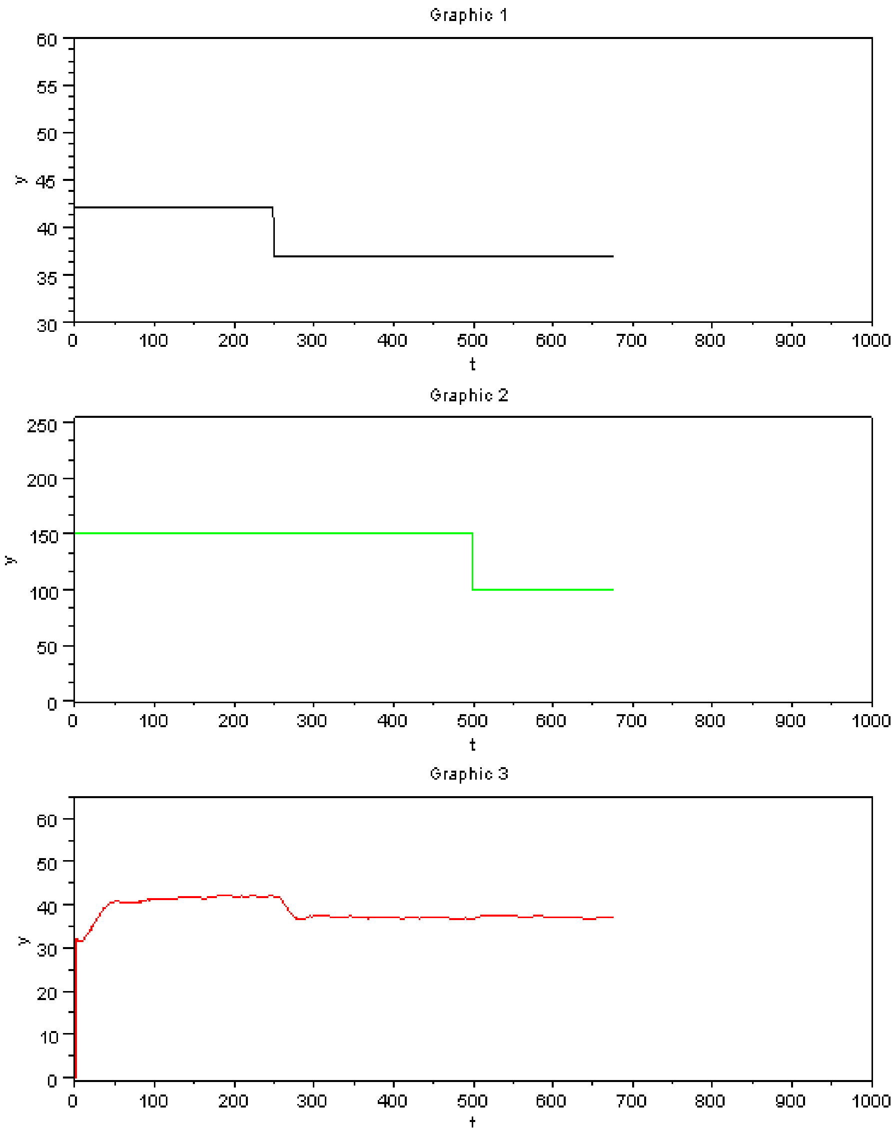


Figure 7.7: Expt 4.1

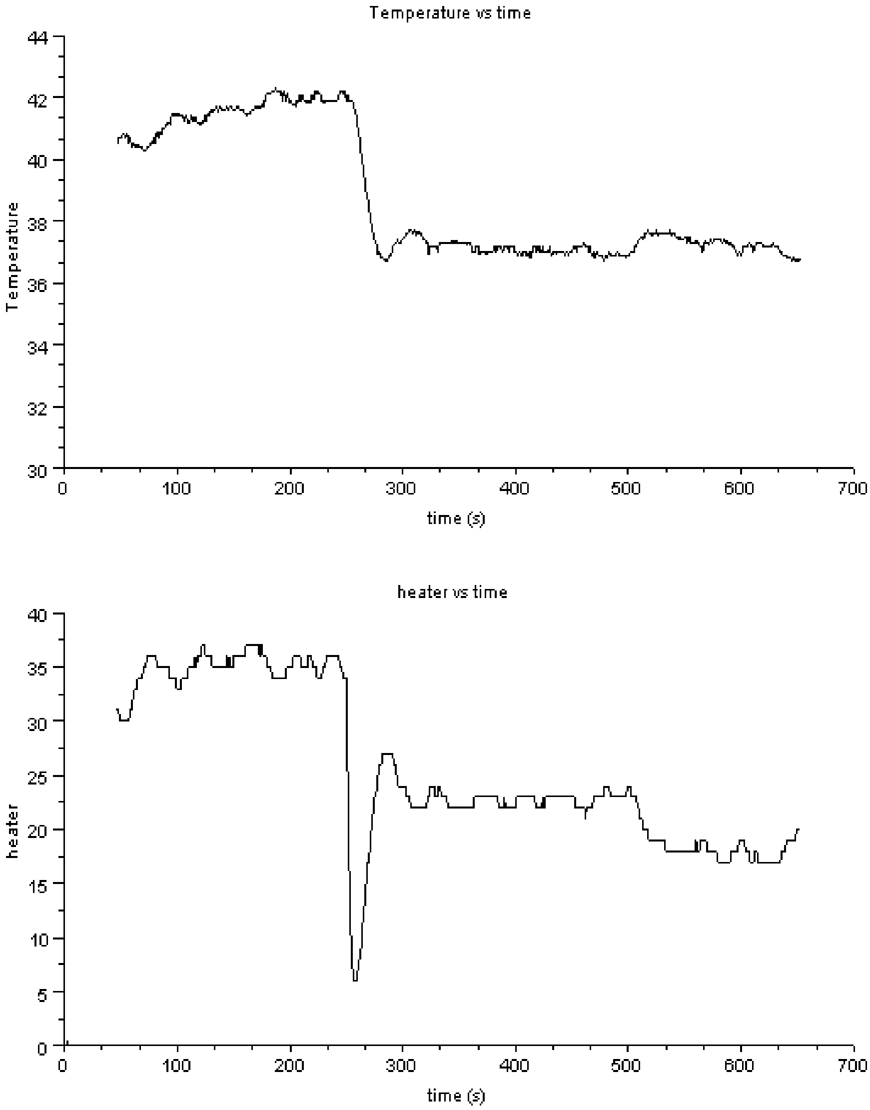


Figure 7.8: Expt 4.1

7.2.2 For q =3 (Expt 4.2)

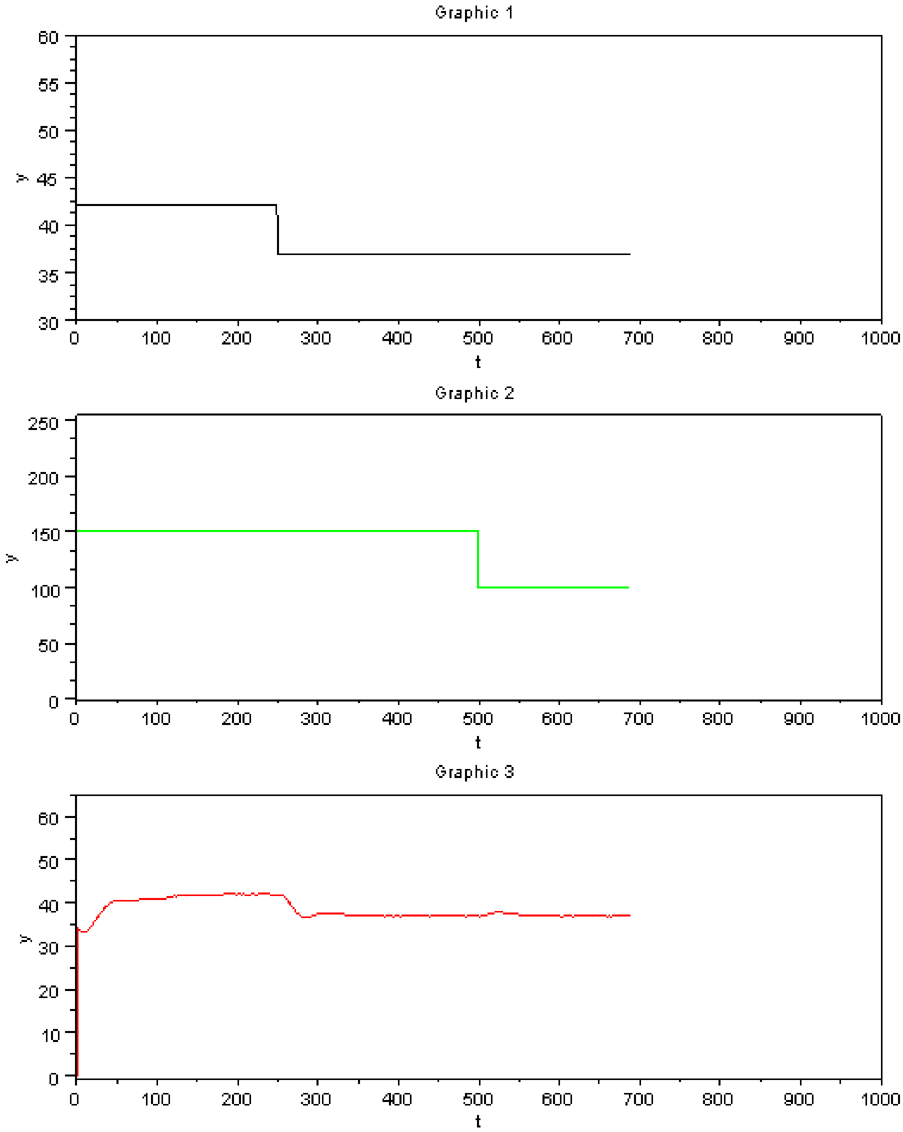


Figure 7.9: Expt 4.2

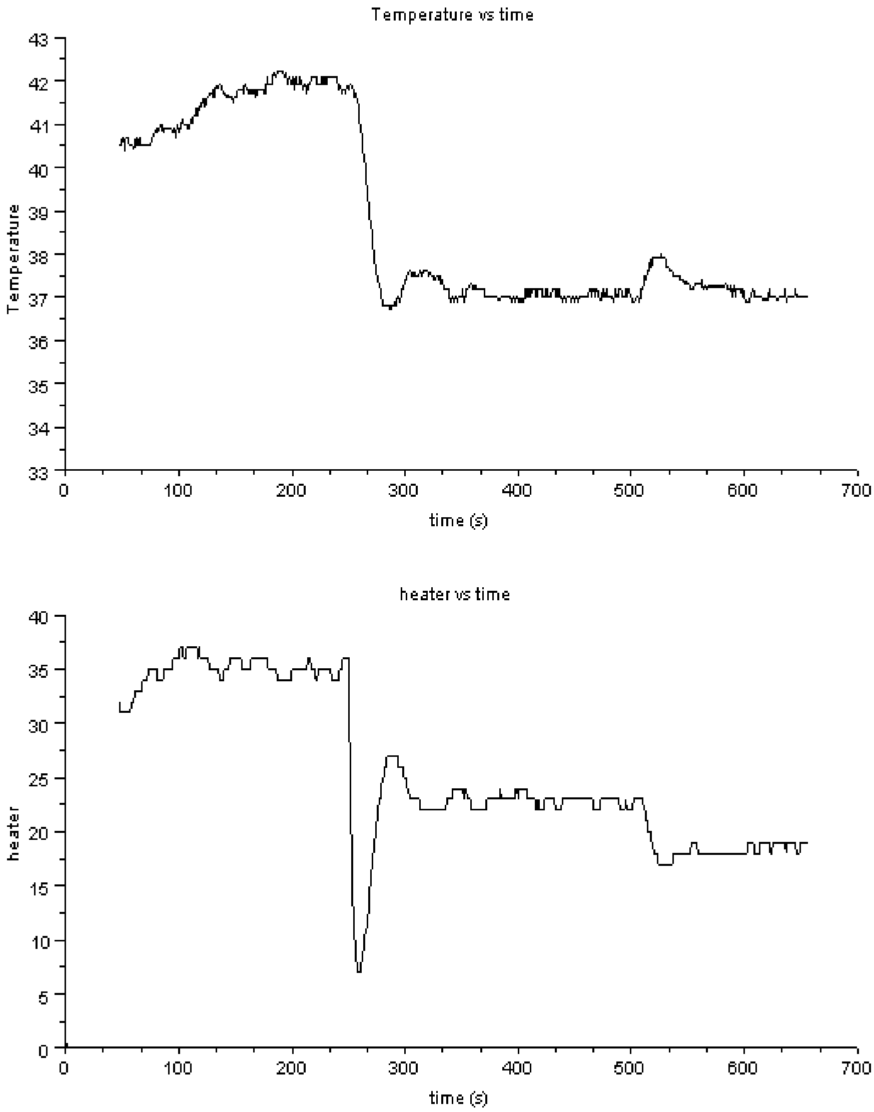


Figure 7.10: Expt 4.2

7.2.3 For q =4 (Expt 4.3)

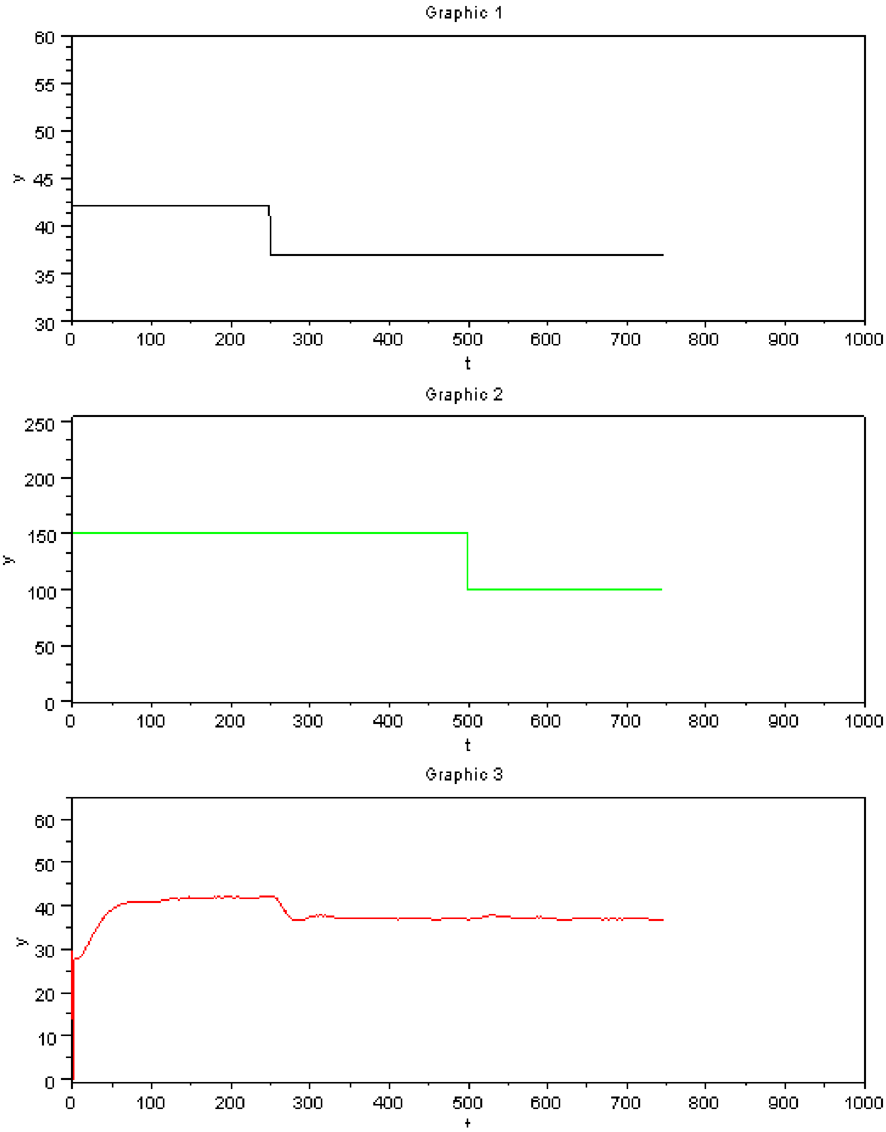


Figure 7.11: Expt 4.3

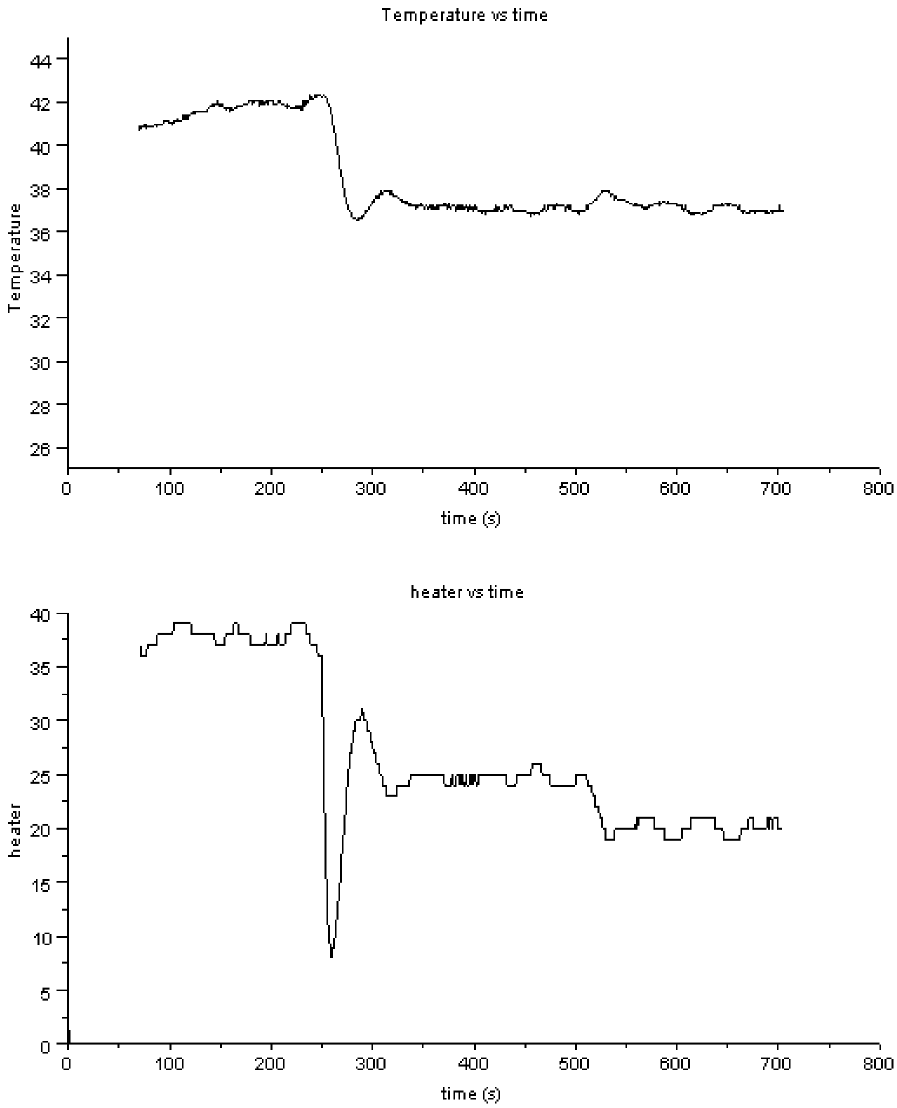


Figure 7.12: Expt 4.3

7.3 Conclusion on the effect of Control Horizon parameter

- The effect of change in q isn't very distinct in the experiments performed.
- While we are calculating the optimized value of manipulated variable at a time, the number of manipulated input moves is increasing as we are increasing the q value.
- But, only the first value of the optimized manipulated variable vector is used for control.
- Increase in q is only increasing the length of the manipulated variable vector which is to be optimized.
- Since, only the first value of manipulated variable vector is used, which itself lies in some specified range, the effect of changing q isn't very significant for SBHS.
- Also, SBHS system is a simple system with very few variables (as compared to real life industrial systems).
- Ideally, the value of q is to be maintained at 3 or 4.

Chapter 8

Conclusion for MPC project

The objective of this project, ie, implementing Model Predictive Control in Single Board Heater System using Scilab was successfully achieved. Several experiments were successfully performed using the developed SCILAB MPC algorithm for both positive and negative step changes in both temperature-set-point and the disturbance variable (fan).

In addition to the above objective, we also tried studying the effect of weighting factors (tuning parameter) and control horizon parameter. We observed and concluded that increase in values of We (error weighting factor), increases oscillations and decreases settling time, while decrease in We leads to opposite effect. Wu (manipulated variable weighting factor), on the other hand has an opposite effect. It decreases oscillations and increases settling time with increase in its value. Hence, better control is obtained for high value of We and low value of Wu.

Thus, with this project, we were able to implement MPC successfully and also were able to comment on the general preferred tuning parameters (weighting factors for error and manipulated variable).

Chapter 9

Acknowledgement

Firstly, I would like to thank Prof Moudgalya Kannan, for giving me this opportunity to undertake MPC project on SBHS. This project, which involved implementing Model Predictive Control in SBHS using SCILAB, was very interesting and provided an excellent learning opportunity. For developing the MPC algorithm, lecture notes on Model Predictive Control by Prof Sachin Patwardhan too were extremely helpful. Also, I got to learn a lot from the speaking tutorials of SCILAB and LaTeX, which had to be referred to for the completion of this project. Over and above this, it was very encouraging to see the experiments working perfectly with the developed Model Predictive Control algorithm.

I would also like to sincerely thank Mr Prashant Gupta, without whom, this project would not have been splendidly completed. I would like to thank him for the time he spent explaining the concepts, clearing the doubts and suggestions for the experiments to implement MPC.

Chapter 10

Appendix

10.1 Appendix 1: General Information on Experiments for this Project

All the experiments for this project was performed remotely on SBHS 12, using a sampling time of 1 second. Basic codes (stepc.sce and steptest.sci) was taken from moodle for this course. Code for implementing MPC was written in scilab and has been mentioned in the report.

Scilab Version used: 5.2.2
SBHS number: 12 (remotely used)
Sampling time: 1 second

For graphs: Until and unless mentioned, Graphic 1 represents the Temperature set point, Graphic 2 represents the Fan and Graphic 3 represents the Temperature.

10.2 Appendix 2: Values of State Space matrices

Initially, open loop experiment was performed, and Plant Transfer function was obtained. For the open loop experiment, a step change in heater from 15 to 25 units at t =200 seconds was provided (sampling time 1s). The response data was fitted to a first order transfer function with a time delay and the following was observed: Kp=0.37, time constant = 45s and delay = 7s.

Using the above, we obtained the plant transfer function:

$$G_p = \frac{0.37}{1+45s}e^{-7s} \tag{10.1}$$

Scilab Method to calculate State Space matrices
State space matrices for a transfer function can be calculated as follows using Scilab:

```
s=poly(0,'s');
```

```
TFcont=syslin('c',[kp*(1-0.5*D)/(tau*s+1)/(1+0.5*D)]);
SScont=tf2ss(TFcont);
```

SScont (in the last line above), has the value of the required State Space matrices. (Please note: Time delays can not be directly handled in Scilab. So, for systems with delays, we will have to use alternate approach. Pade's approximation for time delay being one of the approach.)
The transfer function which we dervied for our SBHS was very close to the transfer function derived by Mr Prashant Gupta. So, using the values of A, B and C which were already calculated by him previously, we obtain the following exact values:

$$A = \begin{bmatrix} 0.9780 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0079 \end{bmatrix}$$

10.3 Appendix 3: Attachments and Contact Information

Attachments

- Folder named *codes*, which contains all the codes used for the experiments. The codes in this folder should be used to reproduce MPC control experiments mentioned in this report.
- Folder named *data files*, which contains data files for all the experiments performed
- MPC Report

Contact Information

My details:
Pratik Behera (07002054)
pratik_behera@iitb.ac.in
+91 99871 54061