Exam Assignment

Modified Four Tank System

02619 Model Predictive Control

Haotian Gao(s192232)

Kongens Lyngby 2021

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Contents

CHAPTER 1

Control Structure

1.1 Problem 1.1

Question: What are the states, x, the measurement, y, the manipulated variables (MVs), u, the measured disturbance variables (DVs), and the controlled variables (CVs) for this system?

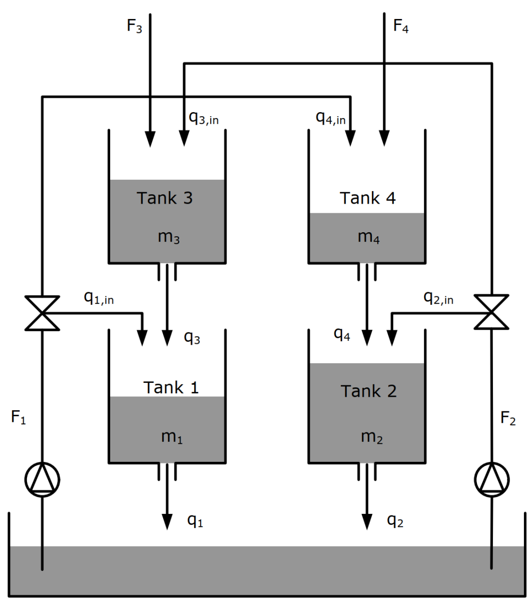

Figure 1.1: The Modified Four Tank System.

Figure 1.1 is a modified model of four water tanks, where states x is the mass of water in the four water tanks, $[m1, m2, m3, m4]$; Measurement y is the height of the water in the four water tanks, $[h1, h2, h3, h4]$; The manipulated variables (MVs), u is Flow rate $[F1, F2$; The The measured disturbance variables (DVs) are Flow rate $[F3, F4]$; The controlled variables (CVs) are the heights of the water in the two tanks below $[h1, h2]$.

1.2 Problem 1.2

Question: Draw a block-diagram of the Modified Four Tank System with an MPC system. The MPC block should illustrate both the state-estimator and regulator of the MPC.

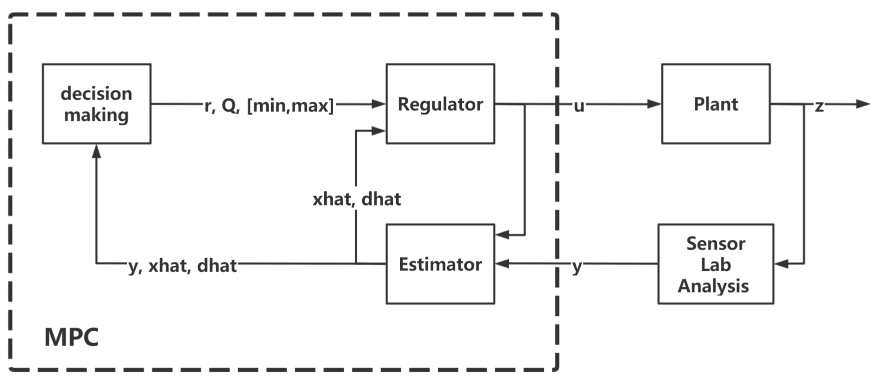

Figure 1.2: Block-diagram of the Modified Four Tank System with an MPC system.

Where r is the target value of CVs, Q is the weight value of regulator, $[min, max]$ is the constraint value of regulator, xbar is the estimated state, dbar is the estimated disturbance

CHAPTER 2

Nonlinear Modeling

In the following you must select one or several set of parameters for the Modified Four Tank System. You should explain and discuss the properties of the model for the parameters that you select. E.g. select a parameter set that make the system a minimum phase system and a parameter set that makes the system a non-minimum phase system. You should select an operating point.

The parameters of the model determine the open-loop transfer function, and then determine whether the closed-loop system is the minimum phase system. Generally speaking, when the open-loop transfer function has no positive poles, positive zeros or time delay, the closed-loop system is the minimum phase system. The existence of water tank 3 and water tank 4 in the model makes Flow rate F1 have a time delay effect on water tank 2, and Flow rate F2 has a time delay effect on water tank 1. The degree of time delay is related to $a3/\sqrt{A3}$, $a4/\sqrt{A4}$ and γ_1, γ_2, in which a3, a4 is the cross sectional area of the pipe, A3, A4 is the cross sectional area of the tank and γ_1, γ_2 are flow distribution constants of Valve 1 and Valve 2.

Through testing, it is found that increasing a3, a4 or decreasing A3, A4 can reduce or even eliminate the time delay, making the non-minimum phase system into the minimum phase system. Two sets of parameters are selected to make the closed-loop system a minimum phase system and a non-minimum phase system. But all calculations and simulations in this report are based on the selected non-minimum phase system

Non-minimum phase system:

Area of outlet pipe1 $[cm^2]$	$a_1 = 1.2272$
Area of outlet pipe2 $[cm^2]$	$a_2 = 1.2272$
Area of outlet pipe3 $[cm^2]$	$a_3 = 1.2272$
Area of outlet pipe4 $[cm^2]$	$a_4 = 1.2272$
Cross sectional area of tank1 $[cm^2]$	$A_1 = 380.1327$
Cross sectional area of tank2 $[cm^2]$	$A_2 = 380.1327$
Cross sectional area of tank3 $[cm^2]$	$A_3 = 380.1327$
Cross sectional area of tank4 $[cm^2]$	$A_4 = 380.1327$
Flow distribution constant of Valve1	$\gamma_1 = 0.45$
Flow distribution constant of Valve2	$\gamma_2 = 0.40$
The acceleration of gravity $[cm/s^2]$	$g = 981$
Density of water $[g/cm^3]$	$\rho = 1.00$

Table 2.1: System parameters of non-minimum phase system.

Minimum phase system:

Area of outlet pipe1 $[cm^2]$	$a_1 = 1.2272$
Area of outlet pipe2 $[cm^2]$	$a_2 = 1.2272$
Area of outlet pipe3 $[cm^2]$	$a_3 = 8.0$
Area of outlet pipe4 $[cm^2]$	$a_4 = 8.0$
Cross sectional area of tank1 $[cm^2]$	$A_1 = 380.1327$
Cross sectional area of tank2 $[cm^2]$	$A_2 = 380.1327$
Cross sectional area of tank3 $[cm^2]$	$A_3 = 380.1327$
Cross sectional area of tank4 $[cm^2]$	$A_4 = 380.1327$
Flow distribution constant of Valve1	$\gamma_1 = 0.45$
Flow distribution constant of Valve2	$\gamma_2 = 0.40$
The acceleration of gravity $[cm/s^2]$	$g = 981$
Density of water $[g/cm^3]$	$\rho = 1.00$

Table 2.2: System parameters of minimum phase system.

Operating point:

Flow rates in pumps1 $[cm^3/s]$	$F_{1ss} = 300$
Flow rates in pumps2 $[cm^3/s]$	$F_{2ss} = 300$
Flow rates into tank3(disturbance) $[cm^3/s]$	$F_{3ss} = 250$
Flow rates into tank4(disturbance) $[cm^3/s]$	$F_{4ss} = 250$

Table 2.3: Operating point.

2.1 Deterministic Nonlinear Model

2.1.1 Problem 2.1.1

Question: Develop a deterministic mathematical model for the dynamics of the system. It should be in the form $\dot{x}(t) = f(x(t), u(t), d(t), p)$.

Regarding the mass balance of the four water tanks

$$\frac{dm_1}{dt}(t) = \rho q_{1,in}(t) + \rho q_{3,out}(t) - \rho q_{1,out}(t) \qquad m_1(t_0) = m_{1,0} \tag{2.1.1}$$

$$\frac{dm_2}{dt}(t) = \rho q_{2,in}(t) + \rho q_{4,out}(t) - \rho q_{2,out}(t) \qquad m_2(t_0) = m_{2,0} \tag{2.1.2}$$

$$\frac{dm_3}{dt}(t) = \rho q_{3,in}(t) - \rho q_{3,out}(t) + \rho F_3(t) \qquad m_3(t_0) = m_{3,0} \tag{2.1.3}$$

$$\frac{dm_4}{dt}(t) = \rho q_{4,in}(t) - \rho q_{4,out}(t) + \rho F_4(t) \qquad m_4(t_0) = m_{4,0} \tag{2.1.4}$$

Regarding the inflows of the four water tanks

$$q_{1,in}(t) = \gamma_1 F_1(t) \tag{2.1.5}$$

$$q_{2,in}(t) = \gamma_2 F_2(t) \tag{2.1.6}$$

$$q_{3,in}(t) = (1 - \gamma_2) F_2(t) \tag{2.1.7}$$

$$q_{4,in}(t) = (1 - \gamma_1) F_1(t) \tag{2.1.8}$$

Regarding the outflows of the four water tanks

$$q_{1,out}(t) = a_1\sqrt{2gh_1(t)} \qquad h_1(t) = \frac{m_1(t)}{\rho A_1} \tag{2.1.9}$$

$$q_{2,out}(t) = a_2\sqrt{2gh_2(t)} \qquad h_2(t) = \frac{m_2(t)}{\rho A_2} \tag{2.1.10}$$

$$q_{3,out}(t) = a_3\sqrt{2gh_3(t)} \qquad h_3(t) = \frac{m_3(t)}{\rho A_3} \tag{2.1.11}$$

$$q_{4,out}(t) = a_4\sqrt{2gh_4(t)} \qquad h_4(t) = \frac{m_4(t)}{\rho A_4} \tag{2.1.12}$$

A deterministic mathematical model for the dynamics of system is expressed in the form of System of ordinary differential equations

$$\dot{x}(t) = f(x(t), u(t), d(t), p) \qquad x(t_0) = x_0$$

Where

$$x = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \quad u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad d = \begin{bmatrix} F_3 \\ F_4 \end{bmatrix}$$

$$p = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & A_1 & A_2 & A_3 & A_4 & \gamma_1 & \gamma_2 & g & \rho \end{bmatrix}$$

2.1.2 Problem 2.1.2

Question: Develop a mathematical model for the sensors (measurements). It should be in the form $y(t) = g(x(t))$.

From the equations (2.1.9), (2.1.10), (2.1.11) and (2.1.12), A deterministic mathematical model for the sensors (measurements) is expressed in the form of System of ordinary differential equations

$$y(t) = g(x(t))$$

Where

$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

2.1.3 Problem 2.1.3

Question: Develop a mathematical model for the outputs, $z(t) = h(x(t))$.

From the equations (2.1.9), (2.1.10), (2.1.11) and (2.1.12), A deterministic mathematical model for the outputs is expressed in the form of System of ordinary differential equations

$$z(t) = h(x(t))$$

Where

$$z = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

2.2 Stochastic Nonlinear Model

Question: Assume that the disturbances F3 and F4 are stochastic variables but piecewise constant.

2.2.1 Problem 2.2.1

Question: Develop a deterministic mathematical model for the dynamics of the system. It should bein the form $\dot{x}(t) = f(x(t), u(t), d(t), p)$ with $d(t) = dk$ for $t_k \leq t < t_k + 1$.

A deterministic mathematical model for the dynamics of system is expressed in the form of System of ordinary differential equations

$$\dot{x}(t) = f(x(t), u(t), d_k, p) \qquad t_k \leq t < t_k + 1 \qquad x(t_k) = x_k$$

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} F3 \\ F4 \end{bmatrix}, \begin{bmatrix} Q1^2 & 0 \\ 0 & Q2^2 \end{bmatrix} \right)$$

2.2.2 Problem 2.2.2

Question: Develop a mathematical model for the sensors (measurements). It should be in the form $y(t) = g(x(t)) + v(t)$ with $v(t) \sim N(0, R_{vv})$.

A deterministic mathematical model for the sensors (measurements) is expressed in the form of System of ordinary differential equations

$$y(t) = g(x(t)) + v(t)$$

$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \qquad \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} R1^2 & 0 & 0 & 0 \\ 0 & R2^2 & 0 & 0 \\ 0 & 0 & R3^2 & 0 \\ 0 & 0 & 0 & R4^2 \end{bmatrix} \right)$$

2.2.3 Problem 2.2.3

Question: Develop a mathematical model for the outputs, $z(t) = h(x(t))$.

A deterministic mathematical model for the outputs is expressed in the form of System of ordinary differential equations

$$z(t) = h(x(t))$$

Where

$$z = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

2.3 Stochastic Nonlinear Model (SDE)

Question: Assume that the disturbances F3 and F4 can be modeled as Brownian motion with some variance.

2.3.1 Problem 2.3.1

Question: Develop a mathematical model (stochastic differential equation system) for the dynamics of the system. It should be in the form

$$d\mathbf{x}(t) = f(\mathbf{x}(t), u(t), p)dt + \sigma d\omega(t)$$

where ω models the unmeasured disturbances F3 and F4

In the previous subsection we derived the deterministic mathematical model for the dynamics of the system.

$$\dot{x}(t) = f(x(t), u(t), d(t), p) \qquad x(t_0) = x_0$$

when the disturbances are treated as a Brownian motion with some variance, we can write

$$\frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{x}(t), u(t), p) + \sigma \boldsymbol{w}(t)$$

However, this construction is not well defined. We extend the approximation of a deterministic differential equation to the stochastic case

$$\boldsymbol{x}(t+\delta t) - \boldsymbol{x}(t) = f(\boldsymbol{x}(t), u(t), p)\delta t + \sigma[\boldsymbol{w}(t+\delta t) - \boldsymbol{w}(t)] + o(\delta t)$$

We let

$$\Delta \boldsymbol{w}(t) = [\boldsymbol{w}(t+\delta t) - \boldsymbol{w}(t)] \sim N_{iid}(0, I\delta t) = \sqrt{\delta t} N_{iid}(0, I)$$

When $\delta t \to 0$

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t), u(t), p)dt + \sigma d\boldsymbol{w}(t)$$

Because the disturbances F3 and F4 can be modeled as Brownian motion with some variance

- $\boldsymbol{w}(t)$ is normally distributed.
- $\boldsymbol{w}(t)$ is independent of $\boldsymbol{w}(s)$ for all $s \neq t$.
- $E\{\boldsymbol{w}(t)\} = 0$
- $E\left\{d\boldsymbol{w}(t)d\boldsymbol{w}(t)'\right\} = Idt$

2.3.2 Problem 2.3.2

Question: Develop a mathematical model for the sensors (measurements). It should be in the form $y(t) = g(x(t)) + v(t)$ with $v(t) \sim N(0, R_{vv})$.

A deterministic mathematical model for the sensors (measurements) is expressed in the form of System of ordinary differential equations

$$y(t) = g(x(t)) + v(t)$$

$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \qquad \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} R1^2 & 0 & 0 & 0 \\ 0 & R2^2 & 0 & 0 \\ 0 & 0 & R3^2 & 0 \\ 0 & 0 & 0 & R4^2 \end{bmatrix}\right)$$

2.3.3 Problem 2.3.3

Question: Develop a mathematical model for the outputs, $z(t) = h(x(t))$.

A deterministic mathematical model for the outputs is expressed in the form of System of ordinary differential equations

$$z(t) = h(x(t))$$

Where

$$z = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

CHAPTER 3

Nonlinear Simulation

3.1 Problem 3.1

Question: Simulate the step responses for 10%, 25% and 50% steps in the manipulated variables. Do this for the deterministic model.

The sampling time is 4s, and the total simulation time is 1200s. The operating point of the manipulated variables is set to$u_{1ss} = 300, u_{2ss} = 300$, so the step signal of 10%, 25%, and 50% of the test is

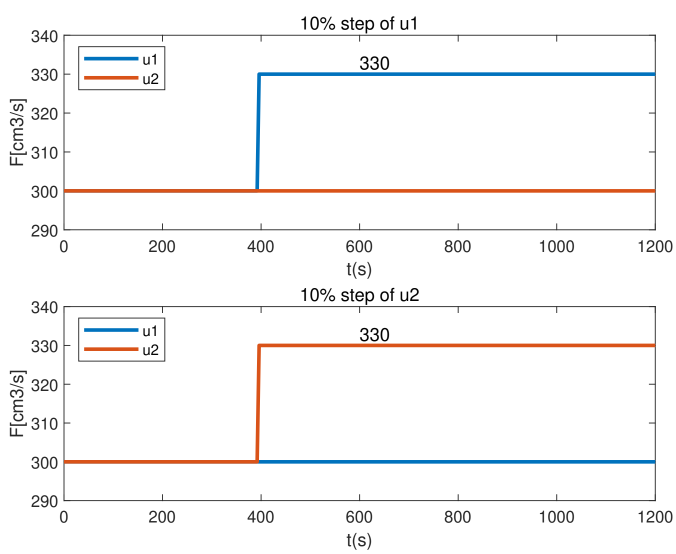

Figure 3.1: The step input of 10% step for deterministic model.

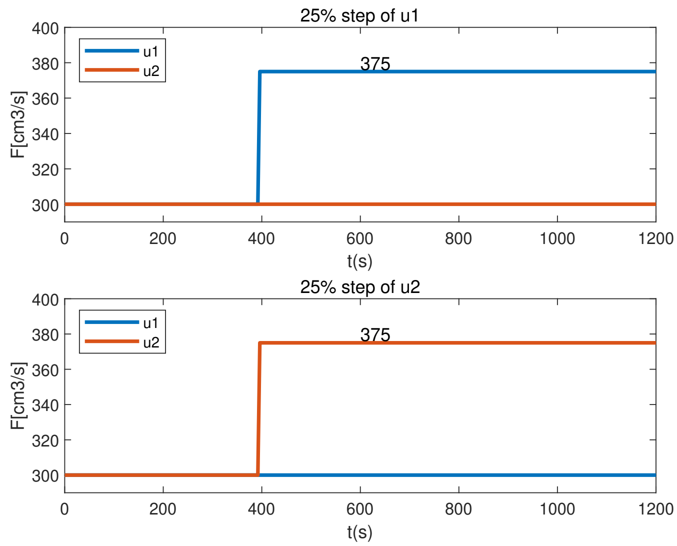

Figure 3.2: The step input of 25% step for deterministic model.

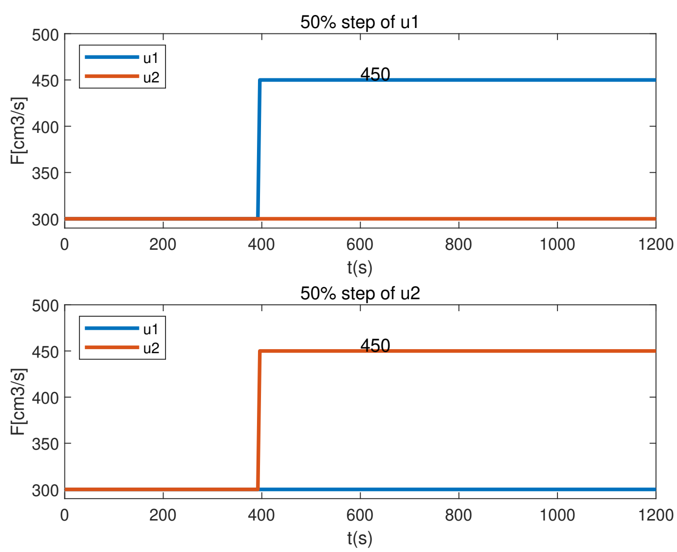

Figure 3.3: The step input of 50% step for deterministic model.

The step response of 10% step of u1 for deterministic model is

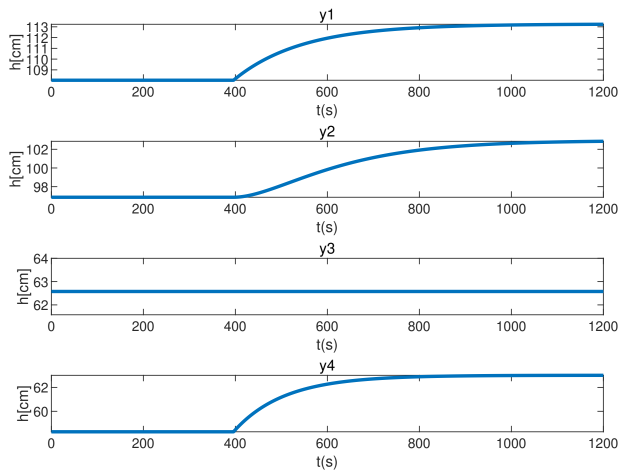

Figure 3.4: The step response of 10% step of u1 for deterministic model.

The step response of 10% step of u2 for deterministic model is

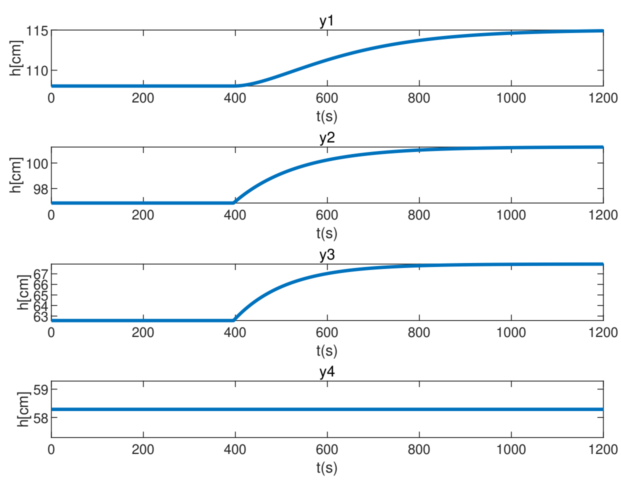

Figure 3.5: The step response of 10% step of u2 for deterministic model.

The step response of 25% step of u1 for deterministic model is

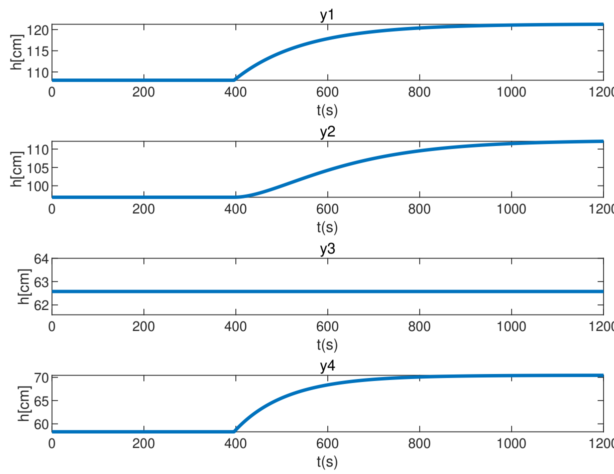

Figure 3.6: The step response of 25% step of u1 for deterministic model.

The step response of 25% step of u2 for deterministic model is

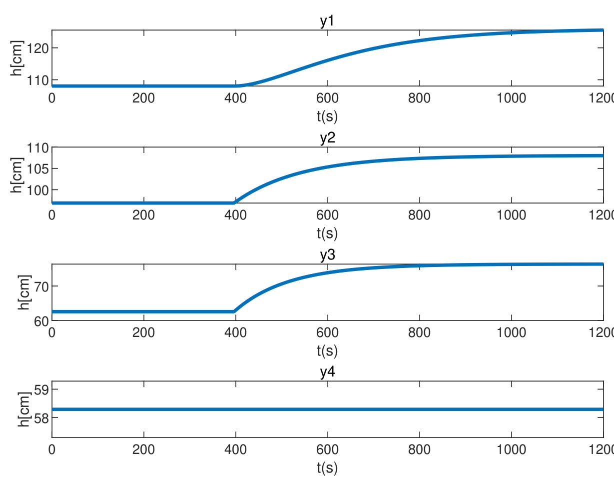

Figure 3.7: The step response of 25% step of u2 for deterministic model.

The step response of 50% step of u1 for deterministic model is

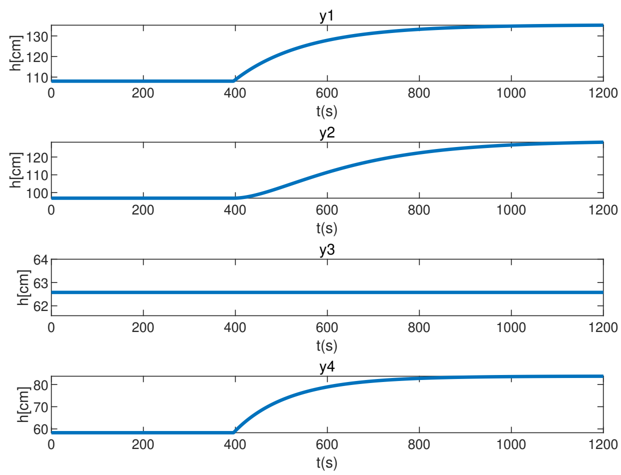

Figure 3.8: The step response of 50% step of u1 for deterministic model.

The step response of 50% step of u2 for deterministic model is

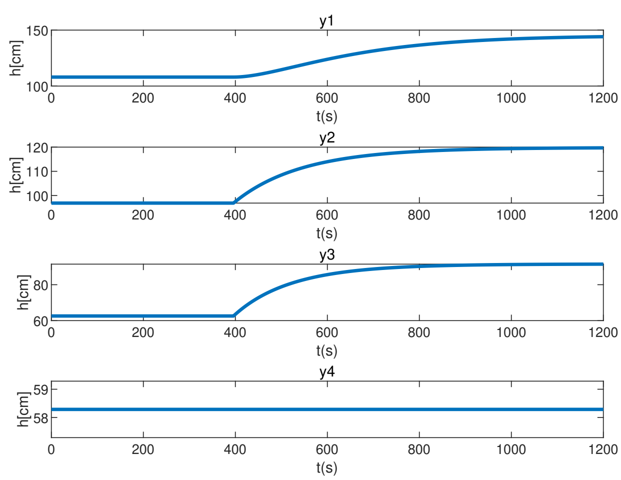

Figure 3.9: The step response of 50% step of u2 for deterministic model.

3.2 Problem 3.2

Question: Simulate the step responses for 10%, 25% and 50% steps in the manipulated variables. In this case you should include measurement noise. Try 3 different noise levels (low noise, medium noise, and high noise).

3 different noise levels of measurement noise is set. The deterministic model with low measurement noise is

$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} v_{1,low} \\ v_{2,low} \\ v_{3,low} \\ v_{4,low} \end{bmatrix} \qquad \begin{bmatrix} v_{1,low} \\ v_{2,low} \\ v_{3,low} \\ v_{4,low} \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)$$

The step response of 10% step of u1 for deterministic model with low measurement noise is

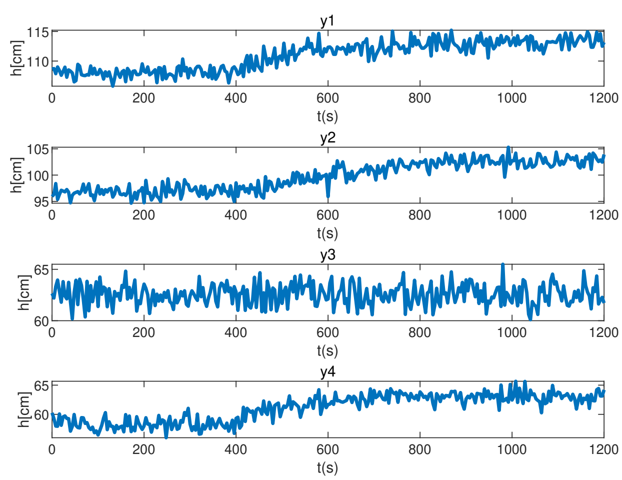

Figure 3.10: The step response of 10% step of u1 with low measurement noise.

The step response of 10% step of u2 for deterministic model with measurement noise is

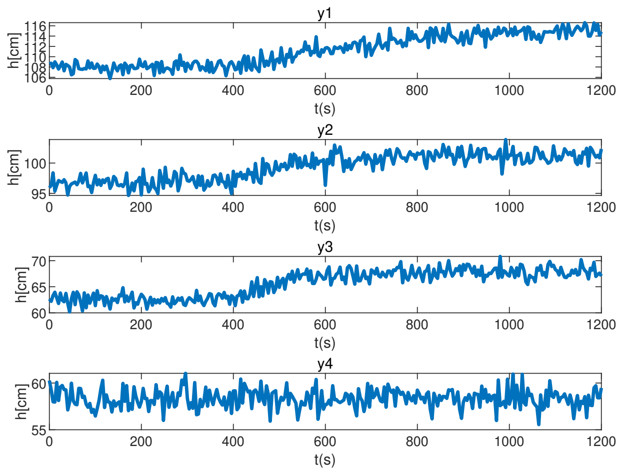

Figure 3.11: The step response of 10% step of u2 with low measurement noise.

The step response of 25% step of u1 for deterministic model with low measurement noise is

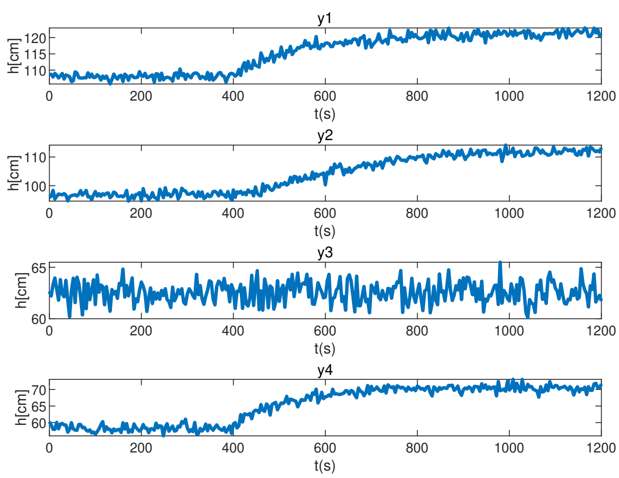

Figure 3.12: The step response of 25% step of u1 with low measurement noise.

The step response of 25% step of u2 for deterministic model with low measurement noise is

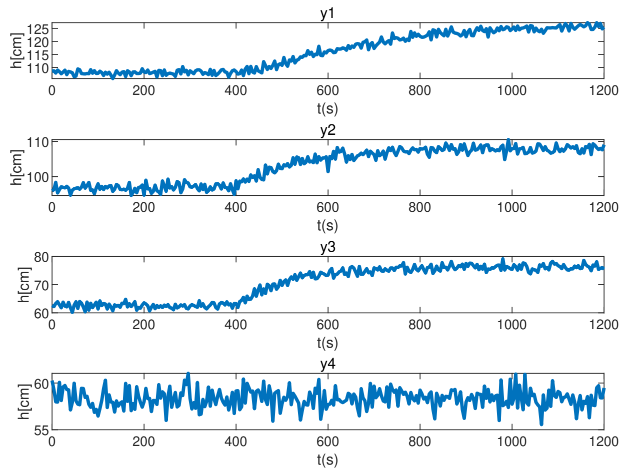

Figure 3.13: The step response of 25% step of u2 with low measurement noise.

The step response of 50% step of u1 for deterministic model with low measurement noise is

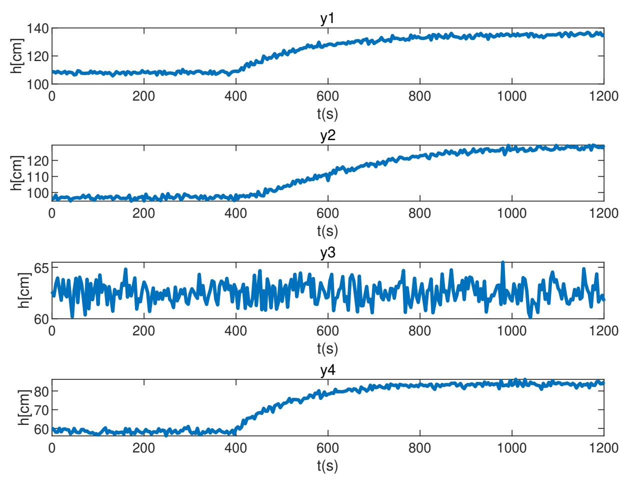

Figure 3.14: The step response of 50% step of u1 with low measurement noise.

The step response of 50% step of u2 for deterministic model with low measurement noise is

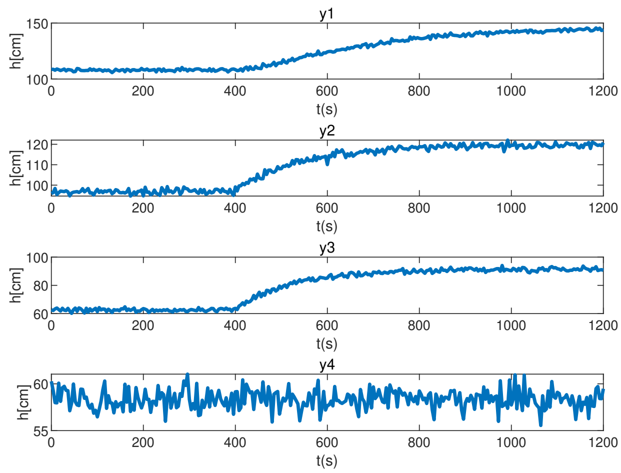

Figure 3.15: The step response of 50% step of u2 for deterministic model with low measurement noise.

The deterministic model with medium measurement noise is

$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} v_{1,medium} \\ v_{2,medium} \\ v_{3,medium} \\ v_{4,medium} \end{bmatrix} \qquad \begin{bmatrix} v_{1,medium} \\ v_{2,medium} \\ v_{3,medium} \\ v_{4,medium} \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}\right)$$

The step response of 10% step of u1 for deterministic model with medium measurement noise is

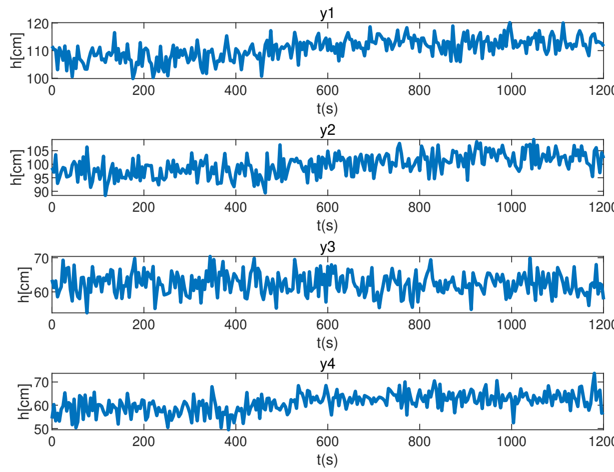

Figure 3.16: The step response of 10% step of u1 with medium measurement noise.

The step response of 10% step of u2 for deterministic model with medium measurement noise is

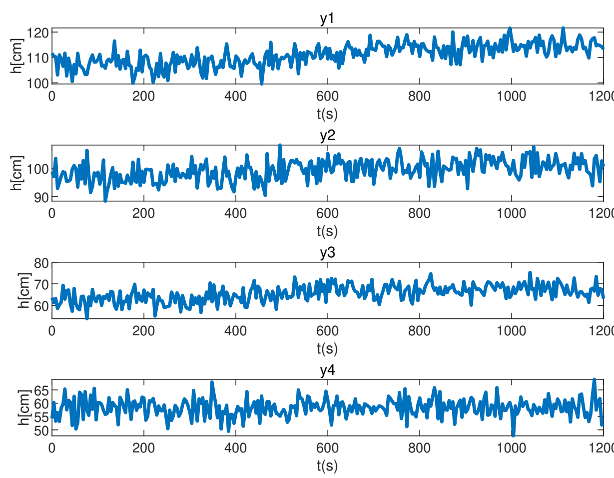

Figure 3.17: The step response of 10% step of u2 with medium measurement noise.

The step response of 25% step of u1 for deterministic model with medium measurement noise is

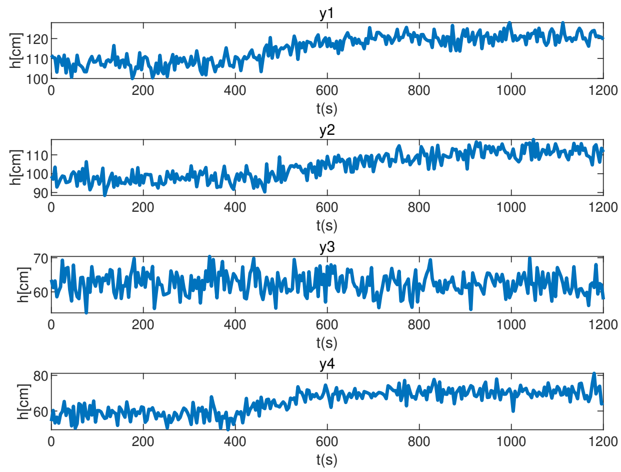

Figure 3.18: The step response of 25% step of u1 with medium measurement noise.

The step response of 25% step of u2 for deterministic model with medium measurement noise is

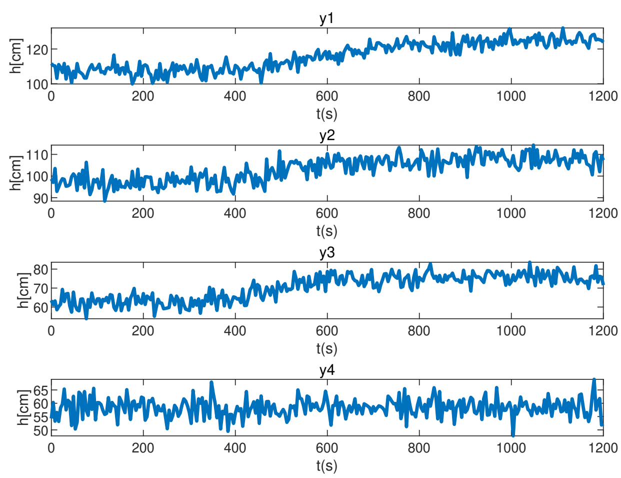

Figure 3.19: The step response of 25% step of u2 with medium measurement noise.

The step response of 50% step of u1 for deterministic model with medium measurement noise is

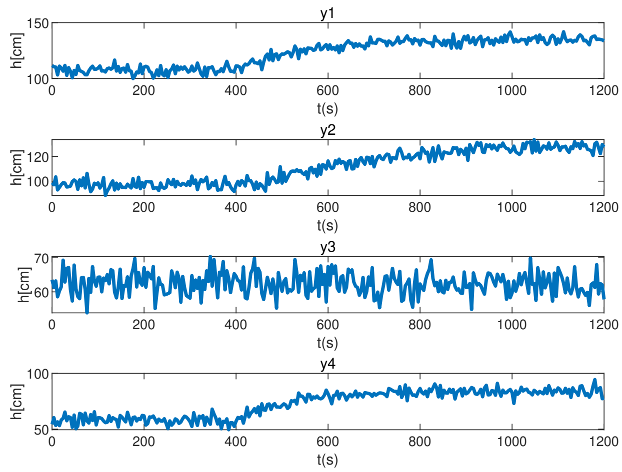

Figure 3.20: The step response of 50% step of u1 with medium measurement noise.

The step response of 50% step of u2 for deterministic model with medium measurement noise is

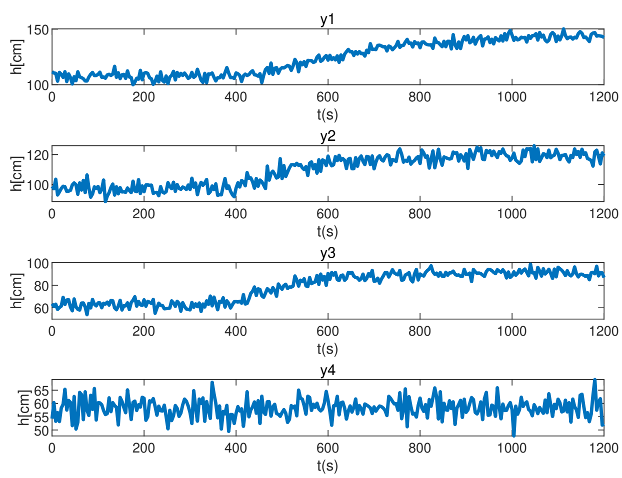

Figure 3.21: The step response of 50% step of u2 for deterministic model with medium measurement noise.

The deterministic model with high measurement noise is

$$y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} v_{1,high} \\ v_{2,high} \\ v_{3,high} \\ v_{4,high} \end{bmatrix} \qquad \begin{bmatrix} v_{1,high} \\ v_{2,high} \\ v_{3,high} \\ v_{4,high} \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix} \right)$$

The step response of 10% step of u1 for deterministic model with high measurement noise is

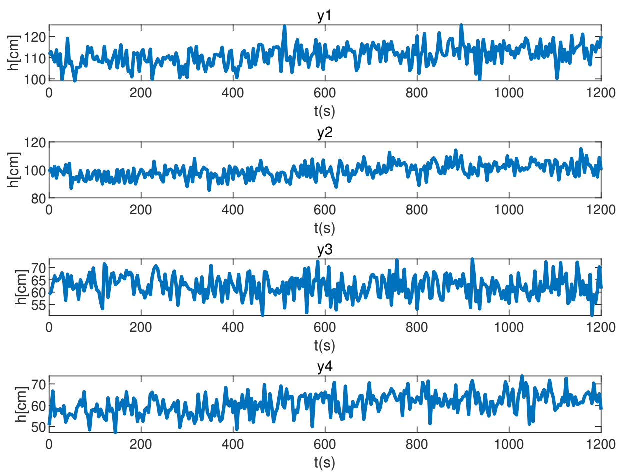

Figure 3.22: The step response of 10% step of u1 with high measurement noise.

The step response of 10% step of u2 for deterministic model with high measurement noise is

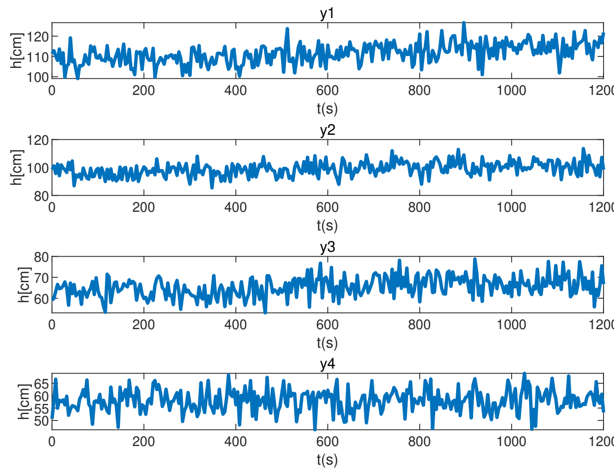

Figure 3.23: The step response of 10% step of u2 with high measurement noise.

The step response of 25% step of u1 for deterministic model with high measurement noise is

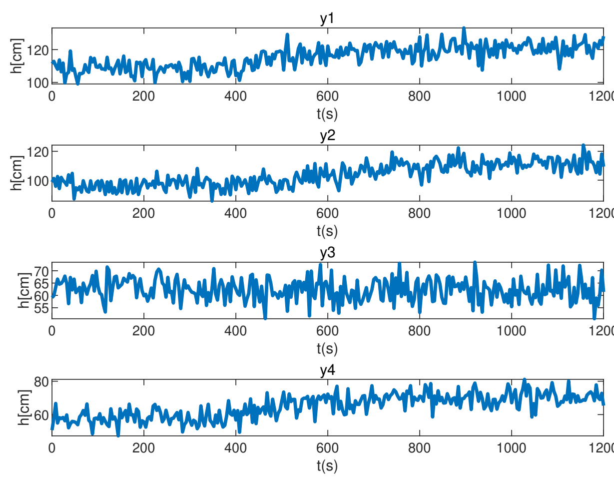

Figure 3.24: The step response of 25% step of u1 with high measurement noise.

The step response of 25% step of u2 for deterministic model with high measurement noise is

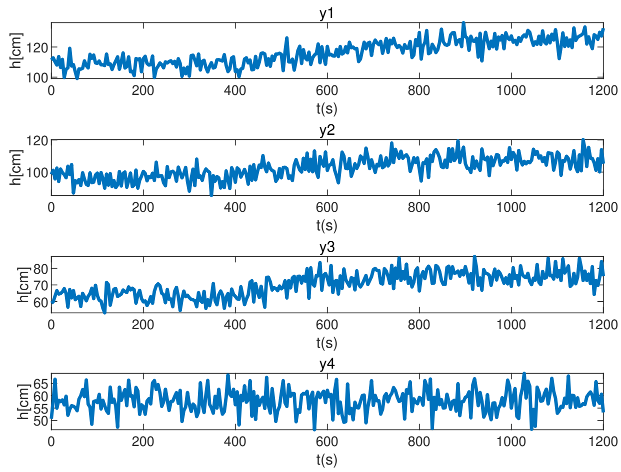

Figure 3.25: The step response of 25% step of u2 with high measurement noise.

The step response of 50% step of u1 for deterministic model with high measurement noise is

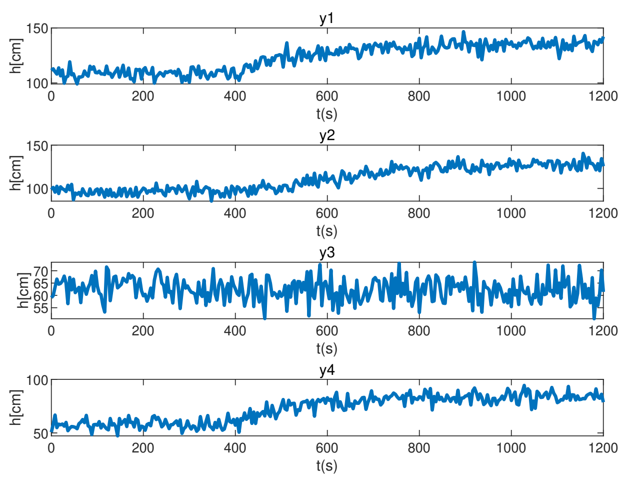

Figure 3.26: The step response of 50% step of u1 with high measurement noise.

The step response of 50% step of u2 for deterministic model with high measurement noise is

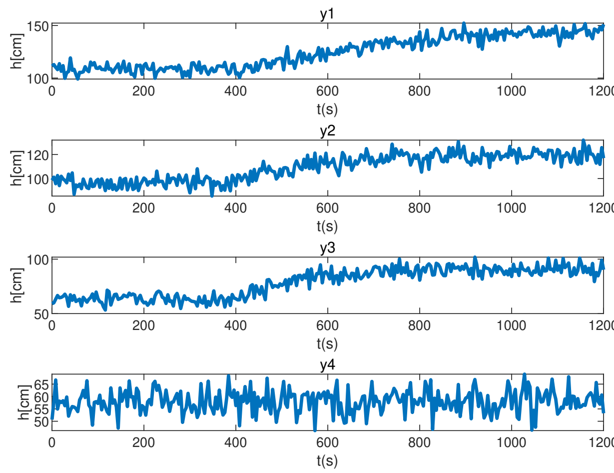

Figure 3.27: The step response of 50% step of u2 for deterministic model with high measurement noise.

3.3 Problem 3.3&3.4

Question: Do this for the deterministic model, in all cases compute and plot (in appropriate plots) the normalized steps

The formula for normalization is expressed as

$$S_{i,j}(t) = (y_i(t) - y_{s,i}) = (u_j(t) - u_{s,j})$$

For the part before the change of the step signal, considering that the denominator cannot be 0, the value of this part is directly set to be the same as the value at the moment of change.

The normalized step response of 10% step of u1 for deterministic model is

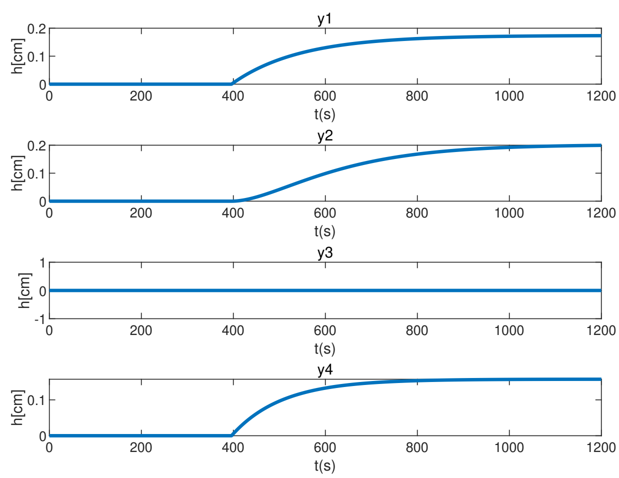

Figure 3.28: The normalized step response of 10% step of u1.

The normalized step response of 10% step of u2 for deterministic model is

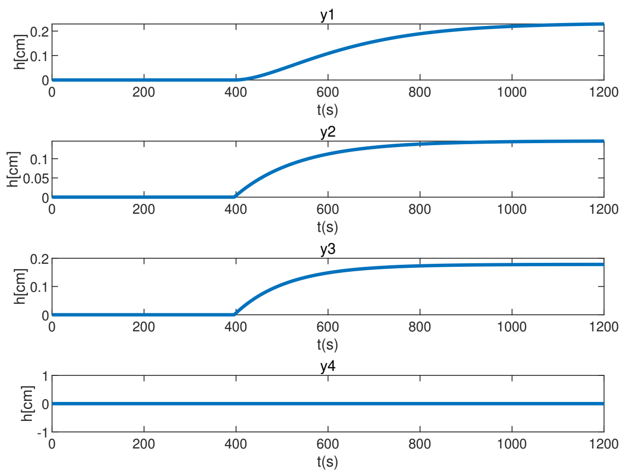

Figure 3.29: The normalized step response of 10% step of u2.

The normalized step response of 25% step of u1 for deterministic model is

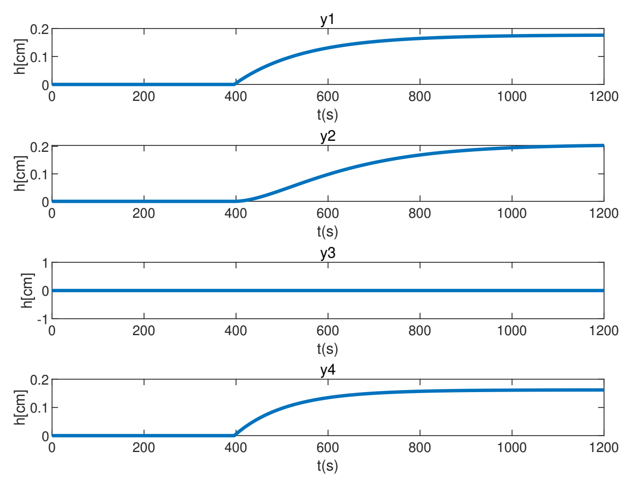

Figure 3.30: The normalized step response of 25% step of u1.

The normalized step response of 25% step of u2 for deterministic model is

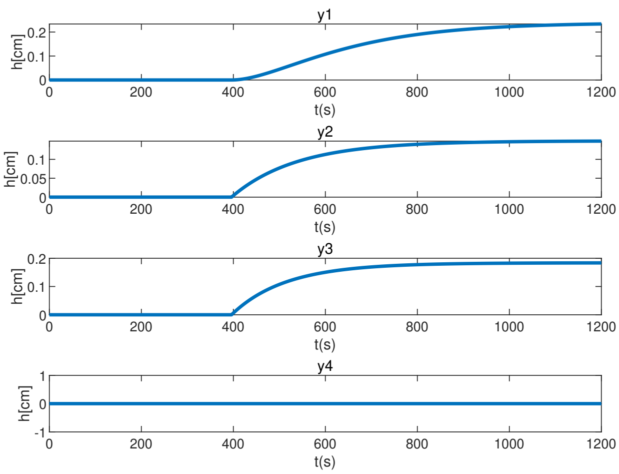

Figure 3.31: The normalized step response of 25% step of u2.

The normalized step response of 50% step of u1 for deterministic model is

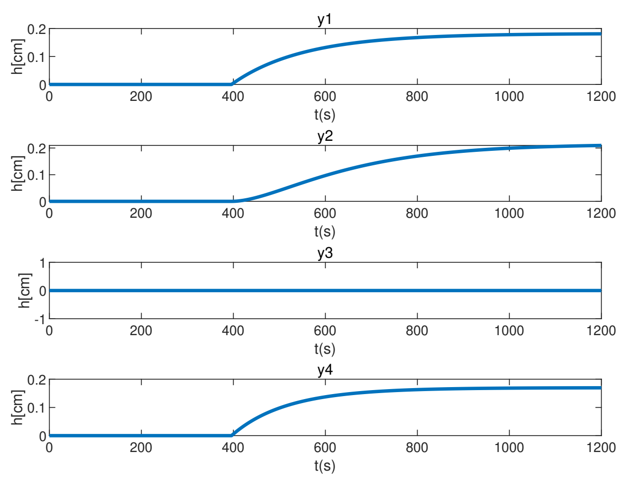

Figure 3.32: The normalized step response of 50% step of u1.

The normalized step response of 50% step of u2 for deterministic model is

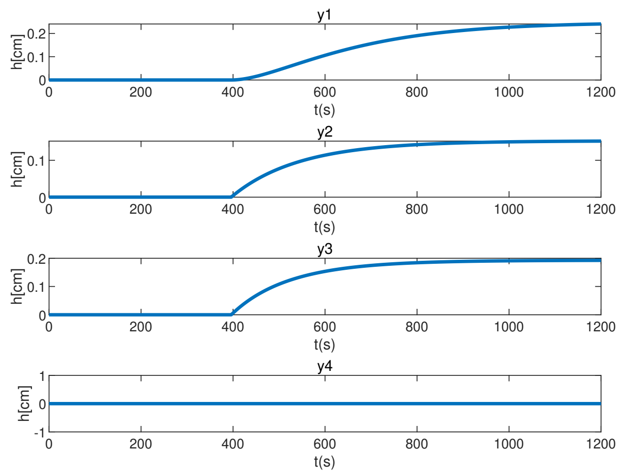

Figure 3.33: The normalized step response of 50% step of u2.

3.4 Problem 3.5

Question: From the normalized steps, identify a transfer function for the four tank system (transfer function from u to y).

The System Identification Toolbox in matlab is used to identify the transfer function for the four tank system (u to y), where The normalized step response of 10% step of u1 and u2 for deterministic model is used.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} g_{u_1,y_1} & g_{u_2,y_1} \\ g_{u_1,y_2} & g_{u_2,y_2} \\ g_{u_1,y_3} & g_{u_2,y_3} \\ g_{u_1,y_4} & g_{u_2,y_4} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The model of the identified transfer function consists of two poles, a zero and a time delay term

$$g_{ij}(s) = \frac{K_{ij}\left(\beta_{ij}s+1\right)}{\left(\tau_{1,ij}s+1\right)\left(\tau_{2,ij}s+1\right)}e^{-\tau_{d,ij}s}$$

The identified discrete-time transfer function of u1 to y1 is

$$g_{u_1,y_1}(s) = \frac{K_{u_1,y_1}\left(\beta_{u_1,y_1}s+1\right)}{\left(\tau_{1,u_1y_1}s+1\right)\left(\tau_{2,u_1y_1}s+1\right)}e^{-\tau_{d,u_1y_1}s}$$

$$K_{u_1,y_1} = 0.17415 \qquad \tau_{1,u_1y_1} = 148.96 \qquad \tau_{2,u_1y_1} = 73.054$$
$$\tau_{d,u_1y_1} = 0 \qquad \beta_{u_1,y_1} = 73.908$$

The following is the original step response and the step response of the identified transfer function. It can be seen that the two curves are almost identical

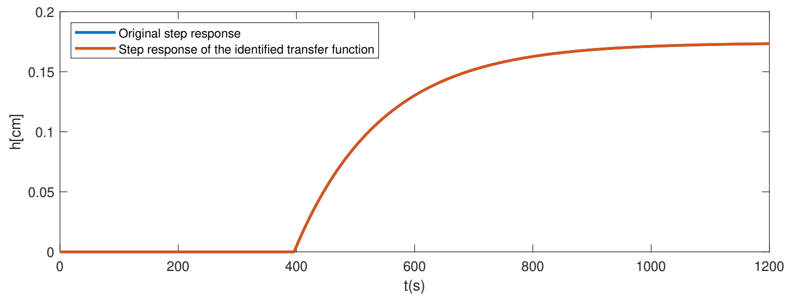

Figure 3.34: Verification of identification results of u1 to y1.

The identified discrete-time transfer function of u1 to y2 is

$$g_{u_1,y_2}(s) = \frac{K_{u_1,y_2}\left(\beta_{u_1,y_2}s+1\right)}{\left(\tau_{1,u_1y_2}s+1\right)\left(\tau_{2,u_1y_2}s+1\right)}e^{-\tau_{d,u_1y_2}s}$$

$$K_{u_1,y_2} = 0.20236 \qquad \tau_{1,u_1y_2} = 148.56 \qquad \tau_{2,u_1y_2} = 102.81$$
$$\tau_{d,u_1y_2} = 3.336 \qquad \beta_{u_1,y_2} = 4.0165$$

The following is the original step response and the step response of the identified transfer function. It can be seen that the two curves are almost identical

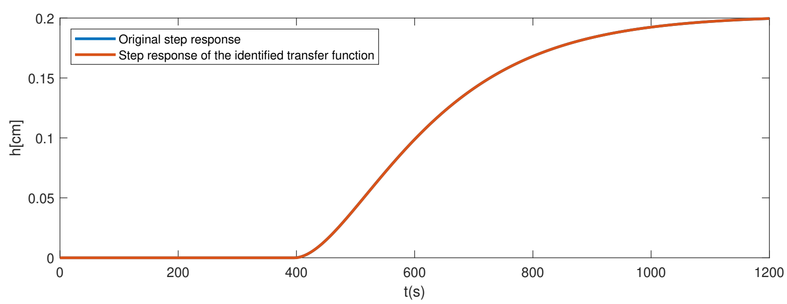

Figure 3.35: Verification of identification results of u1 to y2.

The identified discrete-time transfer function of u1 to y3 is

$$g_{u_1,y_2}(s) = 0$$

The identified discrete-time transfer function of u1 to y4 is

$$g_{u_1,y_4}(s) = \frac{K_{u_1,y_4}\left(\beta_{u_1,y_4}s+1\right)}{\left(\tau_{1,u_1y_4}s+1\right)\left(\tau_{2,u_1y_4}s+1\right)}e^{-\tau_{d,u_1y_4}s}$$

$$K_{u_1,y_4} = 0.15756 \qquad \tau_{1,u_1y_4} = 111.13 \qquad \tau_{2,u_1y_4} = 53.974$$
$$\tau_{d,u_1y_4} = 0 \qquad \beta_{u_1,y_4} = 55.011$$

The following is the original step response and the step response of the identified transfer function. It can be seen that the two curves are almost identical

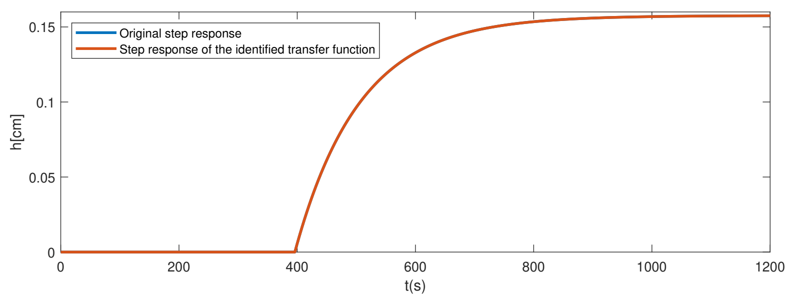

Figure 3.36: Verification of identification results of u1 to y4.

The identified discrete-time transfer function of u2 to y1 is

$$g_{u_2,y_1}(s) = \frac{K_{u_2,y_1}\left(\beta_{u_2,y_1}s+1\right)}{\left(\tau_{1,u_2y_1}s+1\right)\left(\tau_{2,u_2y_1}s+1\right)}e^{-\tau_{d,u_2y_1}s}$$

$$K_{u_2,y_1} = 0.23328 \qquad \tau_{1,u_2y_1} = 157.1 \qquad \tau_{2,u_2y_1} = 106.48$$
$$\tau_{d,u_2y_1} = 3.444 \qquad \beta_{u_2,y_1} = 4.1492$$

The following is the original step response and the step response of the identified transfer function. It can be seen that the two curves are almost identical

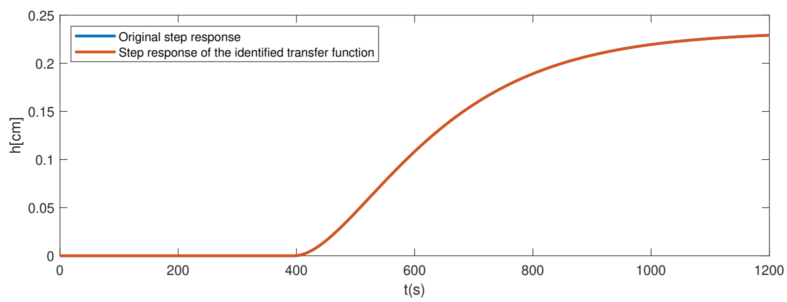

Figure 3.37: Verification of identification results of u2 to y1.

The identified discrete-time transfer function of u2 to y2 is

$$g_{u_2,y_2}(s) = \frac{K_{u_2,y_2}\left(\beta_{u_2,y_2}s+1\right)}{\left(\tau_{1,u_2y_2}s+1\right)\left(\tau_{2,u_2y_2}s+1\right)}e^{-\tau_{d,u_2y_2}s}$$

$$K_{u_2,y_2} = 0.14647 \qquad \tau_{1,u_2y_2} = 140.86 \qquad \tau_{2,u_2y_2} = 69.184$$
$$\tau_{d,u_2y_2} = 0 \qquad \beta_{u_2,y_2} = 69.944$$

The following is the original step response and the step response of the identified transfer function. It can be seen that the two curves are almost identical

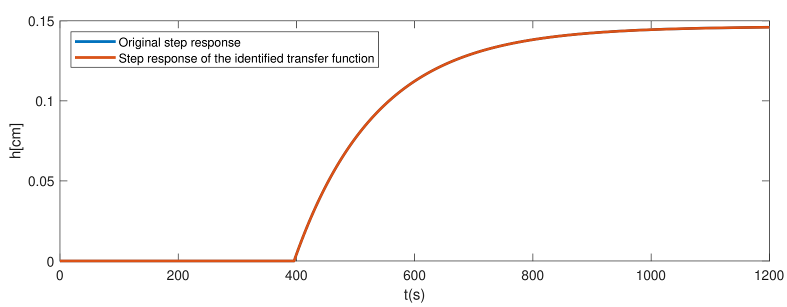

Figure 3.38: Verification of identification results of u2 to y2.

The identified discrete-time transfer function of u2 to y3 is

$$g_{u_2,y_3}(s) = \frac{K_{u_2,y_3}\left(\beta_{u_2,y_3}s+1\right)}{\left(\tau_{1,u_2y_3}s+1\right)\left(\tau_{2,u_2y_3}s+1\right)}e^{-\tau_{d,u_2y_3}s}$$

$$K_{u_2,y_3} = 0.17829 \qquad \tau_{1,u_2y_3} = 115.38 \qquad \tau_{2,u_2y_3} = 55.938$$
$$\tau_{d,u_2y_3} = 0 \qquad \beta_{u_2,y_3} = 57.068$$

The following is the original step response and the step response of the identified transfer function. It can be seen that the two curves are almost identical

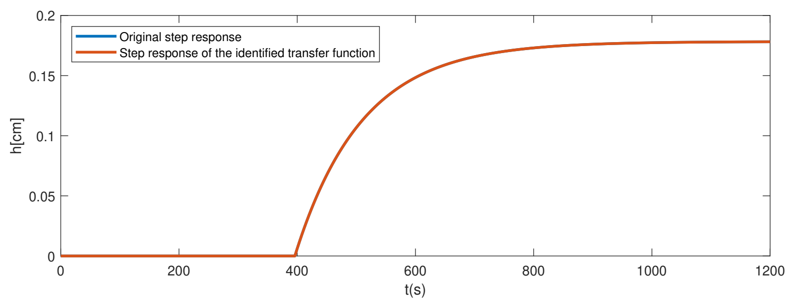

Figure 3.39: Verification of identification results of u2 to y3.

The identified discrete-time transfer function of u2 to y4 is

$$g_{u_2,y_4}(s) = 0$$

3.5 Problem 3.6

Question: Compute an estimate of the noise (mean and variance) if we assume $Y = GU + E$.

The mean value of noise can be obtained by

$$E = Y_{noise} - G_{identify}U$$

Where Y_{noise} is the normalized step response with noise, and y2 is the normalized step response of the identified transfer function. The normalized signal will not change the mean value of the noise, but it will change the variance of the noise, so the variance of the noise cannot be obtained from it. However, the transfer function of u1 to y3, u2 to y4 is 0, which means that the mean value of u1 to y3 and u2 to y4 is always 0, no normalization is required, and the variance value of the noise can be obtained directly.

The step response of 10% step of u1 and u2 for deterministic model is used and three levels of noise are estimated. Through the previous simulation of three levels of measurement noise, it can be seen that when the noise level becomes higher, the step response of 10% is greatly affected. Therefore, when measuring three levels of noise, use step response of 10% to estimate low noise, step response of 25% to estimate medium noise, and step response of 50%to estimate high noise.

About the low noise

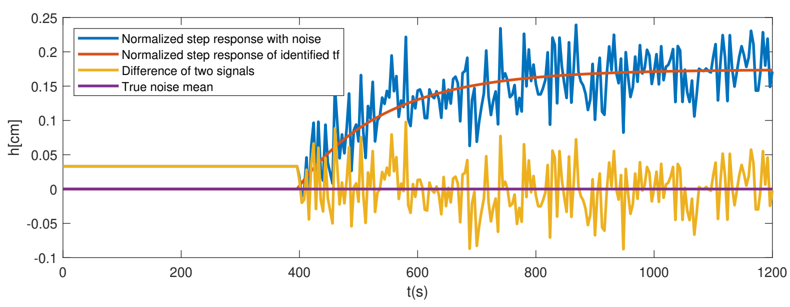

Figure 3.40: Mean of low noise.
The true mean of low noise is 0, and the calculated mean is 5.6089e-04.

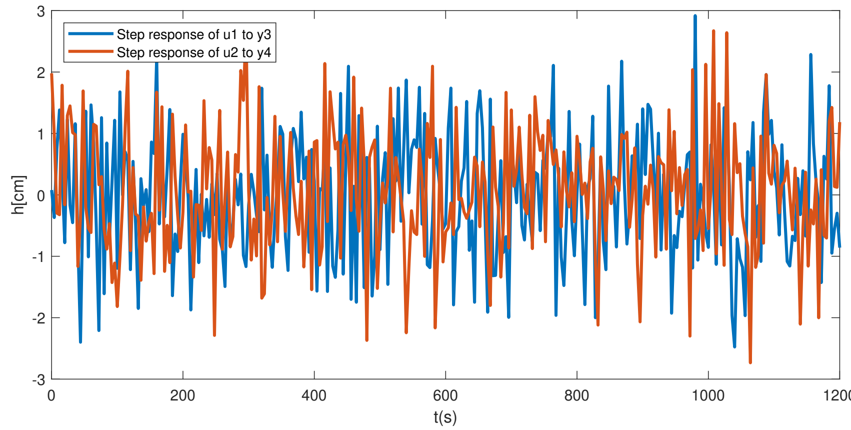

Figure 3.41: variance of low noise.

The true variance of low noise is 1, and the calculated variance is 0.9322. About the medium noise

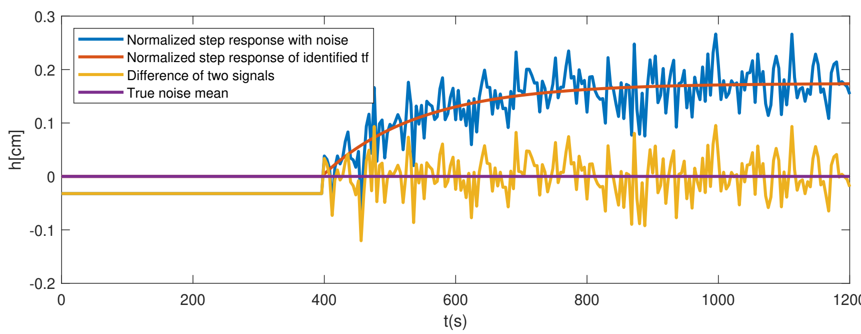

Figure 3.42: Mean of medium noise.

The true mean of medium noise is 0, and the calculated mean is 0.0033.

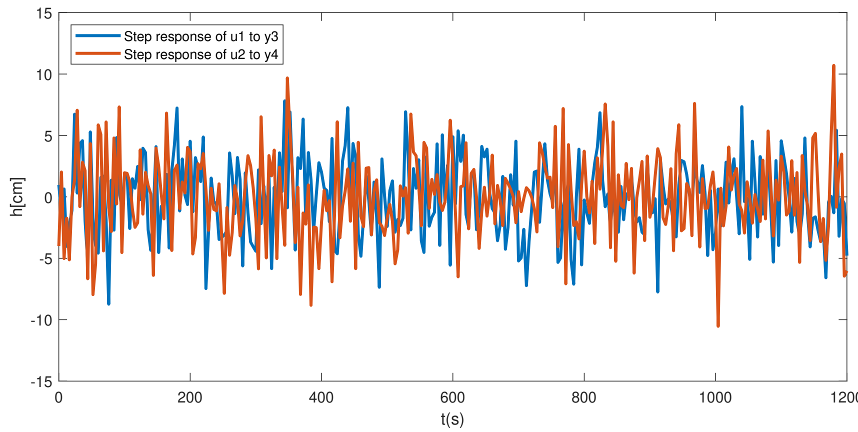

Figure 3.43: variance of medium noise.

The true variance of medium noise is 10, and the calculated variance is 10.5659. About the high noise

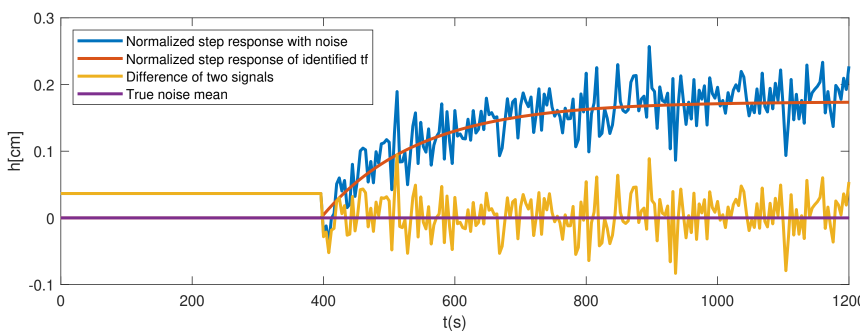

Figure 3.44: Mean of high noise.

The true mean of high noise is 0, and the calculated mean is 0.0068.

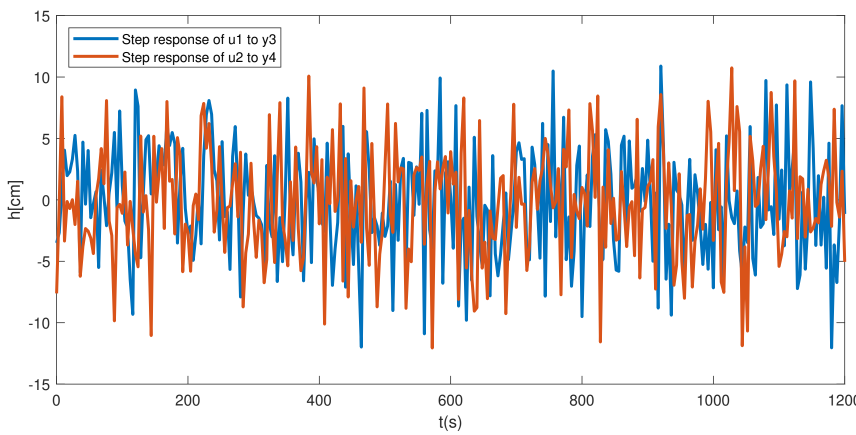

Figure 3.45: variance of high noise.

The true variance of high noise is 20, and the calculated variance is 17.4317.

3.6 Problem 3.7

Question: Report the identified linear model estimate from the step responses. Discuss the accuracy of the model and the requirements of a step experiment.

Regarding the identification result, the accuracy of the identification has been ensured by comparing with the original signal in section 3.5. It can be found from the results that the identified transfer function of u1 to y2, u2 to y1 has a time delay, and the selected system is a non-minimum phase system, which is consistent with the identification result.

3.7 Problem 3.8

Question: Compute the corresponding impulse response coefficients (Markov parameters, discrete-time, you choose a sampling time) and plot them in appropriate plots.

In order to obtain Markov parameters, the impulse response of the nonlinear model needs to be obtained. The impulse signal is structured as

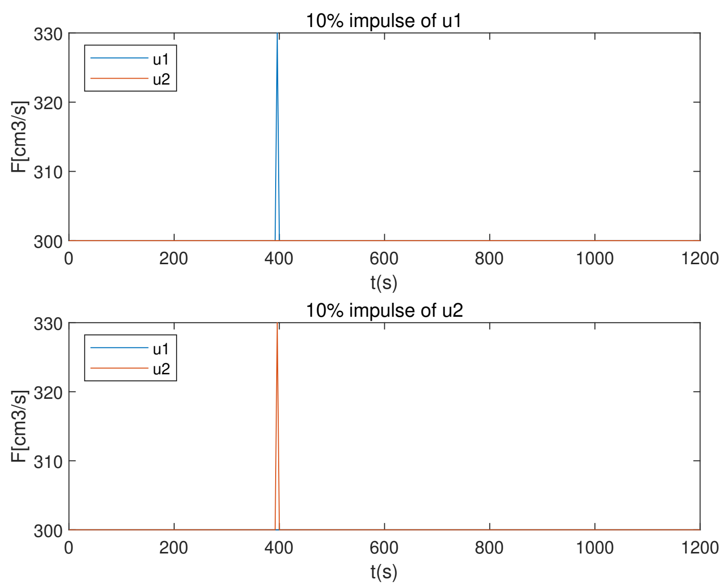

Figure 3.46: Impulse signal.

The impulse response of u1 for the nonlinear model is

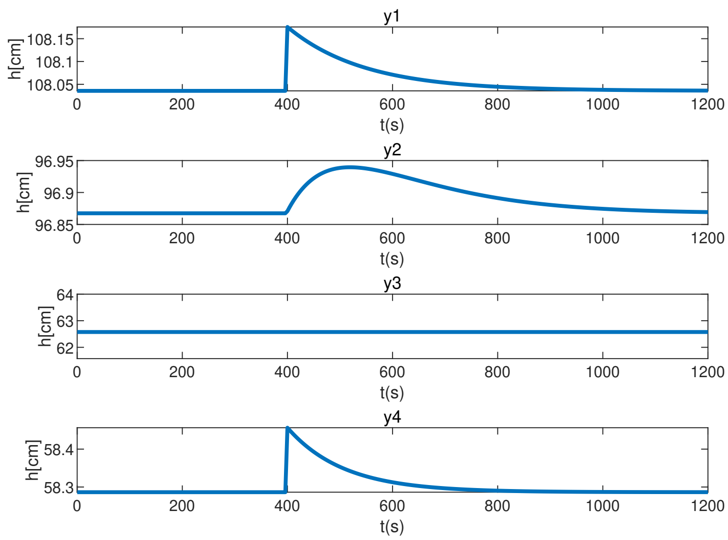

Figure 3.47: Impulse response of u1 for the nonlinear model.

The normalized impulse response of u1 for the nonlinear model is

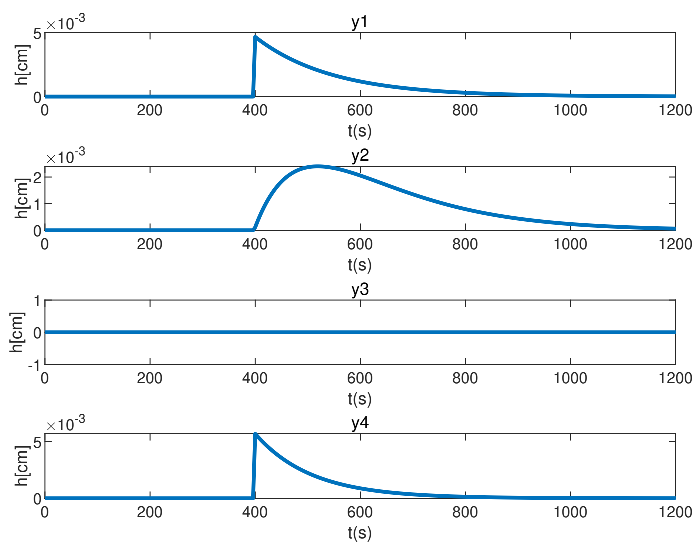

Figure 3.48: Normalized impulse response of u1 for the nonlinear model.

The impulse response of u2 for the nonlinear model is

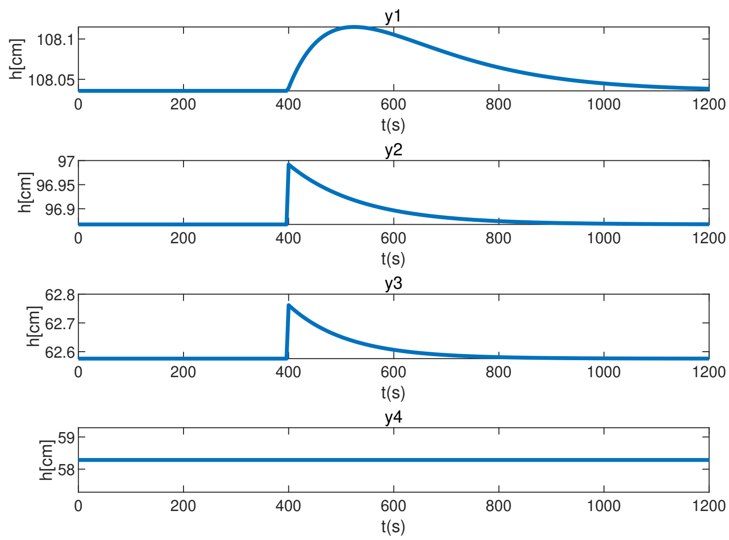

Figure 3.49: Impulse response of u2 for the nonlinear model.

The normalized impulse response of u2 for the nonlinear model is

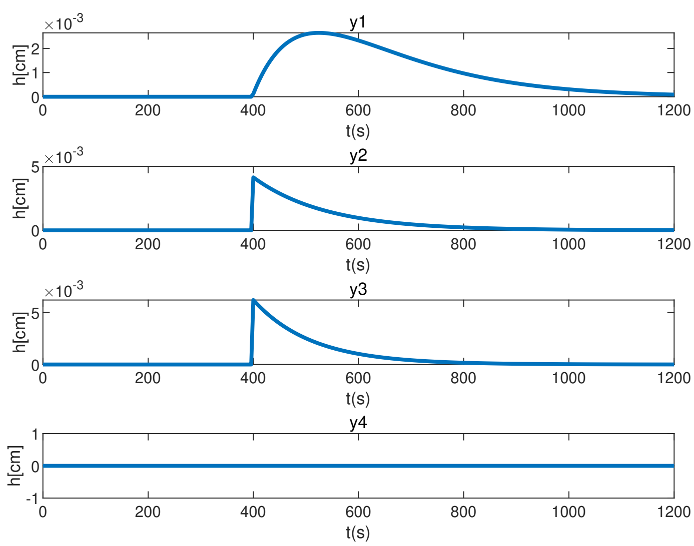

Figure 3.50: Normalized impulse response of u2 for the nonlinear model.

Markov parameters of u1 with sampling time of 4s is

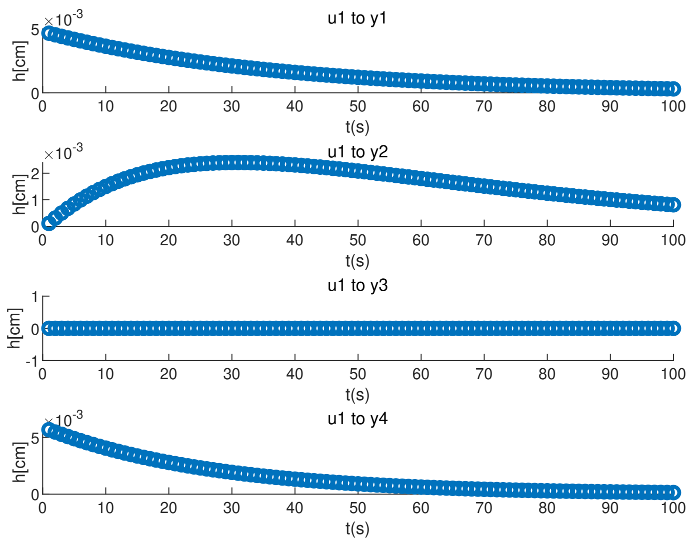

Figure 3.51: Markov parameters of u1.

Markov parameters of u2 with sampling time of 4s is

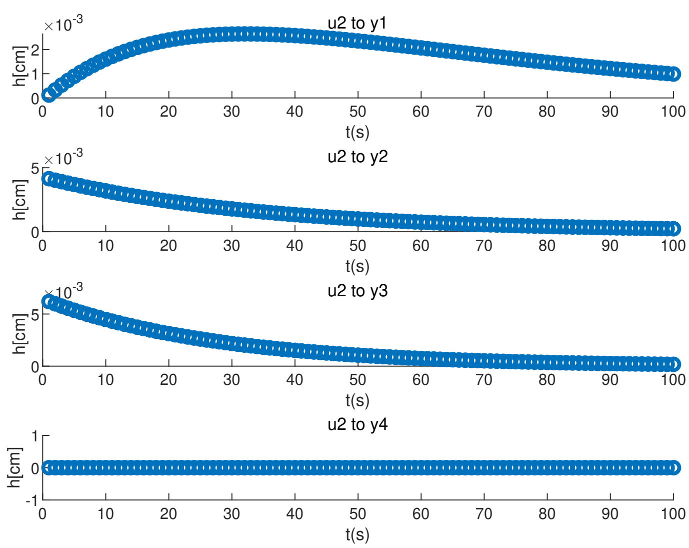

Figure 3.52: Markov parameters of u2.

CHAPTER 4

Linearization and Discretization

4.1 Problem 4.1

Question: Compute continuous-time linearized models for the 3 models developed in Problem 2.

First the steady state value of the state x and measurement y according to the operating point of the manipulated variables u

$$u_{ss} = \begin{bmatrix} 300 \\ 300 \end{bmatrix} \quad d_{ss} = \begin{bmatrix} 250 \\ 250 \end{bmatrix} \quad x_{ss} = \begin{bmatrix} 4.1068 \\ 3.6822 \\ 2.3787 \\ 2.2157 \end{bmatrix} \times 10^4$$

$$y_{ss} = \begin{bmatrix} 108.0357 \\ 96.8675 \\ 62.5759 \\ 58.2863 \end{bmatrix} \quad z_{ss} = \begin{bmatrix} 108.0357 \\ 96.8675 \end{bmatrix}$$

About the continuous-time linearization for the deterministic mathematical model

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), d(t), p) \qquad x(t_0) = x_0 \\ y(t) &= g(x(t)) \\ g(t) &= h(x(t)) \end{aligned}$$

The linear system is defined as

$$\begin{aligned} \dot{X}(t) &= AX(t) + BU(t) + ED(t) \qquad X(t_0) = X_0 \\ Y(t) &= CX(t) \\ Z(t) &= C_z X(t) \end{aligned}$$

Where

$$\begin{aligned} X(t) &= x(t) - x_{ss} \\ U(t) &= u(t) - u_{ss} \\ D(t) &= d(t) - d_{ss} \\ Y(t) &= Y(t) - y_{ss} \\ Z(t) &= Z(t) - z_{ss} \end{aligned}$$

The system matrices is computed as

$$A = \frac{\partial f}{\partial x}(x_s, u_s, d_s) = \begin{bmatrix} -0.0069 & 0 & 0.0090 & 0 \\ 0 & -0.0073 & 0 & 0.0094 \\ 0 & 0 & -0.0090 & 0 \\ 0 & 0 & 0 & -0.0094 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial u}(x_s, u_s, d_s) = \begin{bmatrix} 0.4500 & 0 \\ 0 & 0.4000 \\ 0 & 0.6000 \\ 0.5500 & 0 \end{bmatrix}$$

$$E = \frac{\partial f}{\partial d}(x_s, u_s, d_s) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \frac{\partial g}{\partial x}(x_s) = \begin{bmatrix} 0.0026 & 0 & 0 & 0 \\ 0 & 0.0026 & 0 & 0 \\ 0 & 0 & 0.0026 & 0 \\ 0 & 0 & 0 & 0.0026 \end{bmatrix}$$

$$C_z = \frac{\partial h}{\partial x}(x_s) = \begin{bmatrix} 0.0026 & 0 & 0 & 0 \\ 0 & 0.0026 & 0 & 0 \end{bmatrix}$$

About the continuous-time linearization for the deterministic mathematical model, and the disturbances F3 and F4 are stochastic variables but piecewise constant.

$$\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), d_k, p) \qquad t_k \leq t \leq t_k + 1 \qquad x(t_0) = x_0 \\
y(t) &= g(x(t)) + v(t) \\
g(t) &= h(x(t))
\end{aligned}$$

The linearization is same as the first one.

About the continuous-time linearization for the deterministic mathematical model, and the disturbances F3 and F4 can be modeled as Brownian motion with some variance.

$$\begin{aligned}
d\boldsymbol{x}(t) &= f(\boldsymbol{x}(t), u(t), p)dt + \sigma d\boldsymbol{w}(t) \\
y(t) &= g(x(t)) + v(t) \\
g(t) &= h(x(t))
\end{aligned}$$

The system matrices is computed as

$$A = \frac{\partial f}{\partial x}(x_s, u_s) = \begin{bmatrix} -0.0069 & 0 & 0.0090 & 0 \\ 0 & -0.0073 & 0 & 0.0094 \\ 0 & 0 & -0.0090 & 0 \\ 0 & 0 & 0 & -0.0094 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial u}(x_s, u_s) = \begin{bmatrix} 0.4500 & 0 \\ 0 & 0.4000 \\ 0 & 0.6000 \\ 0.5500 & 0 \end{bmatrix}$$

$$E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \sigma & 0 \\ 0 & \sigma \end{bmatrix}$$

$$C = \frac{\partial g}{\partial x}(x_s) = \begin{bmatrix} 0.0026 & 0 & 0 & 0 \\ 0 & 0.0026 & 0 & 0 \\ 0 & 0 & 0.0026 & 0 \\ 0 & 0 & 0 & 0.0026 \end{bmatrix}$$

$$C_z = \frac{\partial h}{\partial x}(x_s) = \begin{bmatrix} 0.0026 & 0 & 0 & 0 \\ 0 & 0.0026 & 0 & 0 \end{bmatrix}$$

4.2 Problem 4.2

Question: Compute the gains, poles and zeros of these models.

The gains, poles and zeros of these models are

$$\text{gains} = \begin{bmatrix} 0.0012 & 0 \\ 0 & 0.0011 \\ 0 & 0.0016 \\ 0.0014 & 0 \end{bmatrix} \quad \text{zeros} = \begin{bmatrix} [] & [] \\ [] & [] \\ [] & [] \\ [] & [] \end{bmatrix}$$

$$\text{poles} = \begin{bmatrix} [-0.0069] & [-0.0069, -0.0090] \\ [-0.0073, -0.0094] & [-0.0073] \\ [] & [-0.0090] \\ [-0.0094] & [] \end{bmatrix}$$

4.3 Problem 4.3

Question: Compute the continuous-time transfer functions for the continuous-time linearized models.

The continuous-time transfer functions for the continuous-time linearized models are

$$
\begin{aligned}
g_{u_1,y_1}(s) &= \frac{0.001184}{s+0.006879}\\
g_{u_1,y_2}(s) &= \frac{1.3552\times 10^{-5}}{(s+0.009365)(s+0.007265)}\\
g_{u_1,y_3}(s) &= 0\\
g_{u_1,y_4}(s) &= \frac{0.0014471}{s+0.009365}\\
g_{u_2,y_1}(s) &= \frac{1.4267\times 10^{-5}}{(s+0.009038)(s+0.006879)}\\
g_{u_2,y_2}(s) &= \frac{0.0010524}{s+0.007265}\\
g_{u_2,y_3}(s) &= \frac{0.0015786}{s+0.009038}\\
g_{u_2,y_4}(s) &= 0
\end{aligned}
$$

4.4 Problem 4.4

Question: Compare the gains and time constants to the gains and time constants obtained from the step response experiments in Problem 3.

The following table shows the comparison between the identified transfer function and the linearized transfer function.

	Identified tf			Linearized tf		
	gains	time constants	1/zero	gains	time constants	1/zero
g_{u_1,y_1}	0.17415	[148.96,73.054]	[73.908]	0.1721	[145.37]	[]
g_{u_1,y_2}	0.20236	[148.56,102.81]	[4.0165]	0.1992	[137.65,106.78]	[]
g_{u_1,y_3}	0	[]	[]	0	[]	[]
g_{u_1,y_4}	0.15756	[111.13,53.974]	[55.011]	0.1545	[106.78]	[]
g_{u_2,y_1}	0.23328	[157.1,106.48]	[4.1492]	0.2295	[145.37,110.64]	[]
g_{u_2,y_2}	0.14647	[140.86,69.184]	[69.944]	0.1449	[137.65]	[]
g_{u_2,y_3}	0.17829	[115.38,55.938]	[57.068]	0.1747	[110.64]	[]
g_{u_2,y_4}	0	[]	[]	0	[]	[]

The gain value is basically the same. Because the identified transfer function model is set to two poles and one zero, the transfer function obtained by identification has two time constants. The reciprocal of the zero point is also displayed in the table. It can be seen that when one of the poles and zero is cancelled , The time constant is basically the same.

4.5 Problem 4.5

Question: Compute discrete-time state space models using a sampling time of your choice.

The discrete-time state space model is defined as

$$
\begin{aligned}
X_{k+1} &= FX_k + GU_k + G_d D_k \\
Y_k &= CX_k \\
Z_k &= C_z X_k
\end{aligned}
$$

Select the sampling time as 4s for discretization

$$
F = \begin{bmatrix} 0.9729 & 0 & 0.0350 & 0 \\ 0 & 0.9714 & 0 & 0.0362 \\ 0 & 0 & 0.9645 & 0 \\ 0 & 0 & 0 & 0.9632 \end{bmatrix}
$$

$$
G = \begin{bmatrix} 1.7755 & 0.0425 \\ 0.0403 & 1.5770 \\ 0 & 2.3571 \\ 2.1593 & 0 \end{bmatrix}
$$

$$
G_d = \begin{bmatrix} 0.0708 & 0 \\ 0 & 0.0733 \\ 3.9286 & 0 \\ 0 & 3.9260 \end{bmatrix}
$$

$$
C = \begin{bmatrix} 0.0026 & 0 & 0 & 0 \\ 0 & 0.0026 & 0 & 0 \\ 0 & 0 & 0.0026 & 0 \\ 0 & 0 & 0 & 0.0026 \end{bmatrix}
$$

$$
C_z = \begin{bmatrix} 0.0026 & 0 & 0 & 0 \\ 0 & 0.0026 & 0 & 0 \end{bmatrix}
$$

4.6 Problem 4.6

Question: Compute the Markov parameters for these discrete-time state space models and compare them to the the Markov parameters you obtained from the step response experiments

The Markov parameters for these discrete-time state space models are computed and compared to the the Markov parameters obtained from the step response.

Markov parameters of u1 with sampling time of 4s is

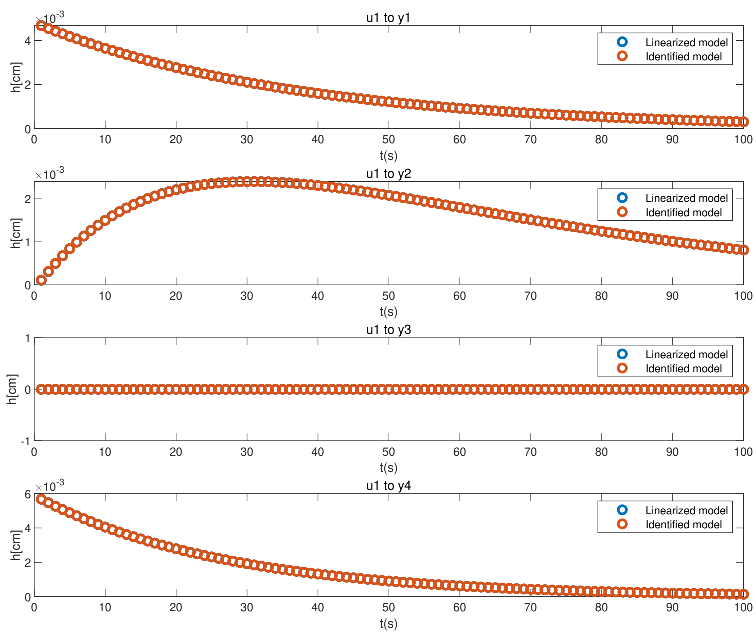

Figure 4.1: Markov parameters of u1 for linearized model and identified model.

Markov parameters of u2 with sampling time of 4s is

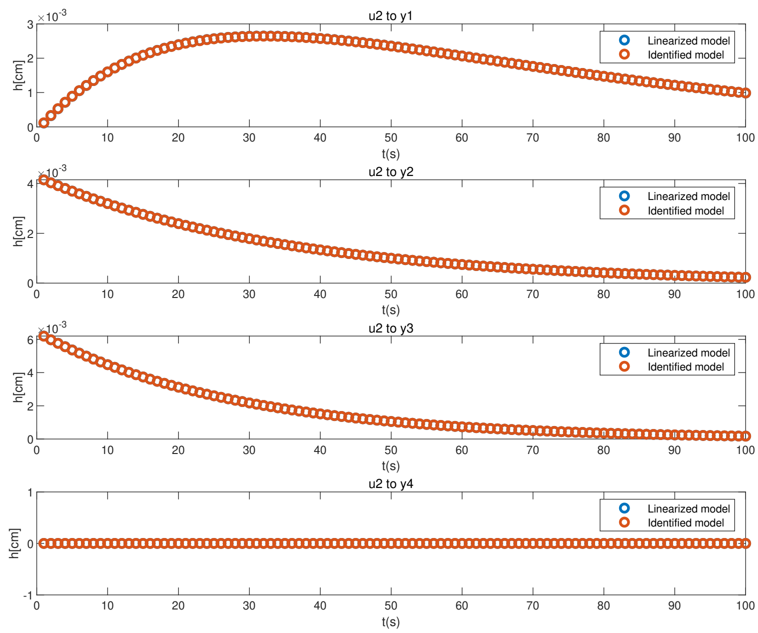

Figure 4.2: Markov parameters of u2 for linearized model and identified model.

It can be seen that the markov parameters obtained by the linearized model and identified model are exactly the same.

4.7 Problem 4.7

Question: Discuss and comment on the linearization approach for obtaining discrete-time linear state space models.

It can be seen from the comparison of Markov parameters between the model obtained by direct linearization and the model identified near the steady-state point that although the system is a non-minimum phase system with time delay. However, the linearization approximation near the steady-state point can almost restore the time delay characteristics of the original system.

It is very fast and convenient to obtain the discrete state equation of the system by directly calculating the steady-state point near the operating point and linearizing it. And if the control of the system is performed near the steady-state point, although linearization loses part of the properties of the nonlinear system, the controller based on direct linearization can show good performance. But if the position of the control work is far from the steady-state point, the error of the linearization approximation will be amplified, and the performance of the controller will be greatly reduced.

Although linearization can be performed near multiple operating points, and then different linearization models can be selected by judging which steady-state point is near to control, in reality this will undoubtedly increase the burden of memory.

CHAPTER 5

State Estimation

5.1 Problem 5.1

Question: Show how the models in Problem 3 and Problem 4 can be represented as linear state space models in discrete time.

The linearized model in problem 4has been discretized in section 4.5. This is mainly to describe how to discretize the state space of the identified model in problem3.

In problem3, the continuous transfer function from input u1, u2 to output y1, y2, y3, y4 is obtained. The continuous transfer function needs to be minimal realized to obtain the discrete state space equation. In addition, in the following sections, a Kalman filter based on the model obtained by problem3 needs to be designed to observe the disturbances, so the transfer function of the disturbances d1, d2 to the output y1, y2, y3, y4 also needs to be obtained, and the transfer functions of u1 u2 to y1, y2, y3, and y4 are implemented together for minimum realization.

First, perform step test of disturbances d1, d2 and obtain the transfer function of disturbances d1, d2 to output y1, y2, y3, y4 through identification.

The identified discrete-time transfer function of d1 to y1 is

$$g_{d_1,y_1}(s) = \frac{K_{d_1,y_1}\left(\beta_{d_1,y_1}s+1\right)}{\left(\tau_{1,d_1y_1}s+1\right)\left(\tau_{2,d_1y_1}s+1\right)}e^{-\tau_{d,d_1y_1}s}$$

$$K_{d_1,y_1} = 0.39128 \qquad \tau_{1,d_1y_1} = 161.16 \qquad \tau_{2,d_1y_1} = 105.27$$
$$\tau_{d,d_1y_1} = 3.948 \qquad \beta_{d_1,y_1} = 4.871$$

The identified discrete-time transfer function of d1 to y2 is 0.

The identified discrete-time transfer function of d1 to y3 is

$$g_{d_1,y_3}(s) = \frac{K_{d_1,y_3}\left(\beta_{d_1,y_3}s+1\right)}{\left(\tau_{1,d_1y_3}s+1\right)\left(\tau_{2,d_1y_3}s+1\right)}e^{-\tau_{d,d_1y_3}s}$$

$$K_{d_1,y_3} = 0.29951 \qquad \tau_{1,d_1y_3} = 117.16 \qquad \tau_{2,d_1y_3} = 56.082$$
$$\tau_{d,d_1y_3} = 0 \qquad \beta_{d_1,y_3} = 57.635$$

The identified discrete-time transfer function of d1 to y4 is 0.

The identified discrete-time transfer function of d2 to y1 is 0.

The identified discrete-time transfer function of d2 to y2 is

$$g_{d_2,y_2}(s) = \frac{K_{d_2,y_1}\left(\beta_{d_2,y_2}s+1\right)}{\left(\tau_{1,d_2y_2}s+1\right)\left(\tau_{2,d_2y_2}s+1\right)}e^{-\tau_{d,d_2y_2}s}$$

$$K_{d_2,y_2} = 0.37093 \qquad \tau_{1,d_2y_2} = 153.5 \qquad \tau_{2,d_2y_2} = 101.33$$
$$\tau_{d,d_2y_2} = 3.964 \qquad \beta_{d_2,y_2} = 4.9225$$

The identified discrete-time transfer function of d2 to y3 is 0.

The identified discrete-time transfer function of d2 to y4 is

$$g_{d_2,y_4}(s) = \frac{K_{d_2,y_4}\left(\beta_{d_2,y_4}s+1\right)}{\left(\tau_{1,d_2y_4}s+1\right)\left(\tau_{2,d_2y_4}s+1\right)}e^{-\tau_{d,d_2y_4}s}$$

$$K_{d_2,y_4} = 0.28936 \qquad \tau_{1,d_2y_4} = 113.3 \qquad \tau_{2,d_2y_4} = 54.17$$
$$\tau_{d,d_2y_4} = 0 \qquad \beta_{d_2,y_4} = 55.721$$

The discrete-time state space model is defined as

$$\begin{aligned} X_{k+1} &= FX_k + [G, G_d]\begin{bmatrix} U_k \\ D_k \end{bmatrix} \\ Y_k &= CX_k \\ Z_k &= C_z X_k \end{aligned}$$

Use a function for minimal realization to get

$$F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix}$$

where

$$F_{11} = \begin{bmatrix} 0.9841 & 0.0005 & -0.0039 & 0.0077 & -0.0015 & -0.0002 & 0.0017 & 0.0000 \\ 0.0003 & 0.9841 & -0.0067 & -0.0038 & 0.0003 & -0.0016 & -0.0004 & 0.0019 \\ 0.0109 & 0.0249 & 0.9533 & -0.0011 & 0.0014 & -0.0050 & 0.0027 & 0.0122 \\ -0.0214 & 0.0130 & -0.0008 & 0.9527 & -0.0052 & -0.0016 & -0.0099 & 0.0055 \\ -0.0016 & 0.0002 & 0.0006 & -0.0017 & 0.9785 & 0.0006 & -0.0122 & 0.0020 \\ 0.0002 & 0.0010 & -0.0061 & -0.0019 & 0.0003 & 0.9515 & -0.0006 & 0.1823 \\ -0.0030 & 0.0006 & 0.0030 & -0.0106 & -0.0346 & 0.0011 & 0.8037 & 0.0447 \\ -0.0004 & -0.0052 & 0.0119 & 0.0061 & -0.0007 & 0.0919 & 0.0362 & 0.1003 \end{bmatrix}$$

$$F_{12} = \begin{bmatrix} -0.0030 & -0.0002 & 0.0002 & 0.0007 & -0.0002 & 0.0000 & 0.0000 & -0.0000 \\ 0.0004 & -0.0003 & 0.0021 & -0.0001 & 0.0000 & -0.0001 & -0.0000 & -0.0000 \\ -0.0038 & 0.0000 & 0.0002 & -0.0004 & 0.0003 & 0.0002 & -0.0000 & 0.0000 \\ 0.0092 & 0.0002 & 0.0000 & 0.0011 & -0.0007 & 0.0002 & 0.0001 & -0.0000 \\ 0.0038 & -0.0008 & -0.0014 & 0.0177 & -0.0068 & 0.0012 & 0.0006 & -0.0003 \\ -0.0188 & -0.0072 & 0.0538 & -0.0002 & -0.0006 & -0.0020 & 0.0005 & 0.0009 \\ 0.3553 & 0.0119 & 0.0030 & 0.0167 & 0.0127 & -0.0017 & 0.0056 & -0.0028 \\ 0.0039 & 0.0637 & -0.1631 & -0.0067 & 0.0062 & 0.0517 & -0.0143 & -0.0177 \end{bmatrix}$$

$$F_{21} = \begin{bmatrix} 0.0034 & -0.0002 & -0.0049 & 0.0124 & 0.0324 & -0.0147 & 0.3720 & 0.0094 \\ 0.0001 & 0.0002 & -0.0010 & 0.0000 & 0.0006 & -0.0105 & 0.0128 & 0.0163 \\ -0.0001 & -0.0005 & 0.0024 & 0.0009 & 0.0000 & 0.0549 & 0.0068 & -0.3860 \\ 0.0000 & -0.0000 & -0.0001 & 0.0003 & -0.0028 & -0.0004 & -0.0401 & 0.0130 \\ 0.0003 & -0.0000 & 0.0000 & -0.0002 & 0.0104 & -0.0005 & 0.0148 & -0.0036 \\ -0.0000 & 0.0000 & -0.0002 & -0.0000 & -0.0010 & 0.0017 & -0.0004 & -0.0527 \\ 0.0000 & -0.0000 & -0.0001 & -0.0001 & 0.0017 & -0.0016 & 0.0016 & -0.0040 \\ -0.0000 & -0.0000 & -0.0002 & -0.0000 & -0.0008 & -0.0027 & -0.0009 & -0.0051 \end{bmatrix}$$

$$F_{22} = \begin{bmatrix} 0.1564 & -0.0341 & 0.0125 & -0.0497 & -0.0892 & 0.0058 & -0.0179 & 0.0115 \\ -0.0401 & 0.9949 & 0.0566 & 0.0009 & -0.0066 & -0.0045 & 0.0007 & 0.0024 \\ 0.0309 & -0.0102 & 0.8135 & -0.0023 & 0.0077 & 0.0404 & -0.0042 & -0.0066 \\ 0.1825 & 0.0088 & 0.0032 & 0.9337 & 0.0361 & -0.0046 & -0.0090 & 0.0046 \\ -0.0051 & 0.0014 & -0.0036 & -0.0055 & 0.9604 & 0.0024 & -0.0055 & 0.0053 \\ 0.0048 & 0.0027 & -0.0498 & 0.0006 & 0.0041 & 0.9774 & 0.0104 & 0.0130 \\ -0.0012 & -0.0006 & 0.0003 & 0.0037 & -0.0145 & -0.0021 & 0.9468 & -0.0057 \\ 0.0009 & -0.0011 & 0.0015 & -0.0018 & 0.0059 & -0.0068 & -0.0057 & 0.9425 \end{bmatrix}$$

$$G = \begin{bmatrix} -0.0473 & -0.0529 \\ 0.0239 & -0.0111 \\ -0.0466 & 0.0528 \\ -0.0222 & 0.0040 \\ -0.0051 & -0.0050 \\ -0.0047 & 0.0048 \\ -0.0035 & -0.0036 \\ 0.0029 & -0.0029 \\ 0.0021 & 0.0032 \\ -0.0005 & 0.0006 \\ 0.0017 & -0.0018 \\ -0.0002 & -0.0001 \\ 0.0012 & 0.0013 \\ -0.0003 & 0.0001 \\ 0.0001 & 0.0003 \\ -0.0003 & 0.0001 \end{bmatrix} \qquad G_d = \begin{bmatrix} -0.0640 & -0.0430 \\ -0.0534 & 0.0731 \\ 0.0570 & -0.0207 \\ -0.0447 & -0.0699 \\ -0.0005 & -0.0002 \\ 0.0010 & -0.0010 \\ -0.0065 & -0.0064 \\ -0.0094 & 0.0115 \\ 0.0091 & 0.0073 \\ 0.0005 & -0.0001 \\ -0.0005 & 0.0007 \\ 0.0002 & 0.0002 \\ -0.0003 & -0.0003 \\ 0.0000 & 0.0000 \\ -0.0001 & -0.0000 \\ -0.0000 & 0.0001 \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} \end{bmatrix}$$

Where

$$C_{11} = \begin{bmatrix} -0.0608 & -0.0262 & -0.0658 & 0.0293 & 0.0002 & -0.0051 & 0.0072 & 0.0090 \\ -0.0457 & 0.0444 & 0.0375 & 0.0600 & 0.0003 & 0.0051 & 0.0073 & -0.0111 \\ -0.0592 & -0.0463 & 0.0499 & -0.0268 & -0.0051 & -0.0007 & -0.0017 & -0.0016 \\ -0.0406 & 0.0635 & -0.0242 & -0.0488 & -0.0051 & 0.0009 & -0.0015 & 0.0015 \end{bmatrix}$$

$$C_{12} = \begin{bmatrix} -0.0100 & -0.0007 & 0.0026 & 0.0006 & -0.0000 & 0.0001 & 0.0000 & 0.0000 \\ -0.0079 & 0.0000 & -0.0028 & 0.0005 & -0.0000 & -0.0001 & -0.0000 & -0.0000 \\ -0.0006 & -0.0007 & 0.0034 & 0.0019 & -0.0008 & -0.0001 & 0.0001 & -0.0000 \\ -0.0008 & 0.0004 & -0.0037 & 0.0018 & -0.0007 & 0.0003 & 0.0001 & -0.0000 \end{bmatrix}$$

$$C_z = \begin{bmatrix} C_{z,11} & C_{z,12} \end{bmatrix}$$

Where

$$C_{z,11} = \begin{bmatrix} -0.0100 & -0.0007 & 0.0026 & 0.0006 & -0.0000 & 0.0001 & 0.0000 & 0.0000 \\ -0.0079 & 0.0000 & -0.0028 & 0.0005 & -0.0000 & -0.0001 & -0.0000 & -0.0000 \end{bmatrix}$$

$$C_{z,12} = \begin{bmatrix} -0.0006 & -0.0007 & 0.0034 & 0.0019 & -0.0008 & -0.0001 & 0.0001 & -0.0000 \\ -0.0008 & 0.0004 & -0.0037 & 0.0018 & -0.0007 & 0.0003 & 0.0001 & -0.0000 \end{bmatrix}$$

5.2 Problem 5.2

Question: Design and evaluate static and dynamic Kalman filters for the linear models identified in Problem 3 and Problem 4. You should simulate the case where the unknown disturbances are stochastic variables but do not contains step changes..

First, based on the discrete state space equations identified in problem3 and linearized in problem4, static Kalman observers are designed respectively. The Riccati equation can be used to calculate the gain of the static Kalman observer in infinite time offline. The Kalman observer in this chapter is to observe the disturbances term, and the discrete state space model needs to be constructed as

$$\begin{bmatrix} X_{k+1} \\ D_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A & G_d \\ 0 & I \end{bmatrix}}_{A_{hat}} \begin{bmatrix} X_k \\ D_k \end{bmatrix} + \underbrace{\begin{bmatrix} G \\ 0 \end{bmatrix}}_{B_{hat}} U_k + \underbrace{\begin{bmatrix} I & G_d \\ 0 & I \end{bmatrix}}_{G_{w,hat}} \begin{bmatrix} W_{x,k} \\ W_{d,k} \end{bmatrix}$$

$$Y_k = \underbrace{\begin{bmatrix} C \\ 0 \end{bmatrix}}_{C_{hat}} \begin{bmatrix} X_k \\ D_k \end{bmatrix} + V_k$$

$$E\{ww'\} = Q \qquad E\{vv'\} = R$$

Where Q and R are covariances of process noise and measurements noise.

For the discrete state space equations linearized in problem4, Q and R are chosen as

$$Q = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.008 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.008 \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The *dlqe* function of matlab is used to calculate Riccati solution P and gain K

$$P = \begin{bmatrix} 1.6565 & 0 & 1.7571 & 0 & 0.0182 & 0 \\ 0 & 1.6356 & 0 & 1.7342 & 0 & 0.0185 \\ 1.7571 & 0 & 2.4394 & 0 & 0.0292 & 0 \\ 0 & 1.7342 & 0 & 2.3705 & 0 & 0.0290 \\ 0.0182 & 0 & 0.0292 & 0 & 0.0004 & 0 \\ 0 & 0.0185 & 0 & 0.0290 & 0 & 0.0004 \end{bmatrix} \times 10^3$$

$$K = \begin{bmatrix} 4.2547 & 0 & 4.4952 & 0 \\ 0 & 4.2025 & 0 & 4.4395 \\ 4.4952 & 0 & 6.2578 & 0 \\ 0 & 4.4395 & 0 & 6.0836 \\ 0.0465 & 0 & 0.0750 & 0 \\ 0 & 0.0473 & 0 & 0.0746 \end{bmatrix}$$

For the discrete state space equations identified in problem3, Q and R are chosen as

$$Q = \underbrace{\begin{bmatrix} 4 & 0 & \dots & 0 & 0 & 0 \\ 0 & \ddots & & \vdots & \vdots & \vdots \\ \vdots & & \ddots & 0 & & \\ 0 & \dots & & 4 & 0 & 0 \\ 0 & \dots & & 0 & 40 & 0 \\ 0 & \dots & & 0 & 0 & 40 \end{bmatrix}}_{18\times 18} \qquad R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For the discrete state space equations identified in problem3, The *dlqe* function of matlab is used to calculate Riccati solution P and gain K

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

where

$$P_{11}=\begin{bmatrix} 36.9074 & 1.1161 & -7.2365 & 16.2445 & -6.7123 & -0.5448 & 5.0169 & 0.1627 & 1.4135 \\ 1.1161 & 45.4144 & -9.7861 & -6.5960 & 1.3189 & -2.9017 & -0.7917 & 0.1778 & -0.1067 \\ -7.2365 & -9.7861 & 27.3462 & -3.4879 & 1.0411 & -4.0473 & 0.2521 & -0.2659 & 0.2193 \\ 16.2445 & -6.5960 & -3.4879 & 37.8198 & -3.6934 & -1.2611 & -1.2451 & -0.1376 & -0.9372 \\ -6.7123 & 1.3189 & 1.0411 & -3.6934 & 108.5737 & 0.7405 & -44.2411 & -1.6513 & -15.5628 \\ -0.5448 & -2.9017 & -4.0473 & -1.2611 & 0.7405 & 78.3969 & -1.8228 & 6.8906 & -2.2287 \\ 5.0169 & -0.7917 & 0.2521 & -1.2451 & -44.2411 & -1.8228 & 83.6822 & 2.7608 & 33.7891 \\ 0.1627 & 0.1778 & -0.2659 & -0.1376 & -1.6513 & 6.8906 & 2.7608 & 7.8483 & 0.2005 \\ 1.4135 & -0.1067 & 0.2193 & -0.9372 & -15.5628 & -2.2287 & 33.7891 & 0.2005 & 19.4789 \end{bmatrix}$$

$$P_{12}=\begin{bmatrix} -0.5788 & 0.1598 & 1.8741 & -0.1603 & -0.0079 & 0.0941 & 0.0049 & -20.5773 & -13.5893 \\ -2.7847 & 0.5960 & -0.2892 & 0.0127 & 0.0319 & 0.1399 & 0.2356 & -17.6991 & 23.9605 \\ 2.0459 & -1.1809 & 0.4299 & 0.3803 & 1.2635 & 0.0596 & -0.0420 & 16.4697 & -5.2302 \\ 0.6243 & -0.4916 & -1.1988 & -0.5481 & 0.4375 & -0.1877 & 0.1081 & -14.1402 & -21.1859 \\ 0.2101 & -0.2549 & -7.7246 & 7.4920 & 0.1896 & -0.2565 & 0.0678 & 0.4566 & 0.5866 \\ 5.7833 & 7.8902 & -1.6053 & 1.5335 & 0.5781 & -3.0052 & -3.6308 & 0.2546 & -0.2481 \\ -0.7674 & 1.4530 & 27.3717 & -10.0735 & 1.1837 & 4.8931 & -2.6129 & -1.5619 & -1.3181 \\ 16.7328 & -6.8368 & 0.9078 & 0.7324 & 6.4505 & -0.9555 & -1.7312 & -0.8564 & 0.8576 \\ -6.8549 & 2.1984 & 8.9462 & -9.3159 & -0.2545 & 2.6653 & -0.9647 & 0.0089 & -0.0046 \end{bmatrix}$$

$$P_{21}=\begin{bmatrix} -0.5685 & -2.7986 & 2.0470 & 0.6388 & 0.2101 & 5.7838 & -0.7669 & 16.7323 & -6.8549 \\ 0.0857 & 0.0644 & -0.8937 & -0.3817 & -0.2561 & 7.8875 & 1.4520 & -6.8612 & 2.1998 \\ 1.6511 & -0.2533 & 0.5065 & -1.4309 & -7.7255 & -1.6051 & 27.3580 & 0.9080 & 8.9493 \\ -0.0651 & -0.0020 & 0.3478 & -0.4498 & 7.4916 & 1.5333 & -10.0673 & 0.7324 & -9.3166 \\ -0.0215 & -0.0153 & 1.2946 & 0.4413 & 0.1895 & 0.5771 & 1.1833 & 6.4475 & -0.2541 \\ 0.1149 & 0.1415 & 0.0500 & -0.1678 & -0.2565 & -3.0052 & 4.8944 & -0.9552 & 2.6651 \\ -0.0017 & 0.2440 & -0.0435 & 0.0987 & 0.0678 & -3.6308 & -2.6134 & -1.7307 & -0.9646 \\ -48.0793 & -40.9767 & 39.8717 & -33.1423 & 0.3232 & 0.1825 & -3.2349 & -2.5351 & 0.8601 \\ -31.9110 & 55.6376 & -13.3184 & -50.4059 & 0.5679 & -0.1417 & -2.8322 & 2.7219 & 0.5839 \end{bmatrix}\times 10^{3}$$

$$P_{22} = \begin{bmatrix} 144.4173 & -29.8149 & 2.5684 & 3.2499 & 17.1778 & -1.9280 & -4.0233 & 0.5917 & -0.6695 \\ -29.8149 & 31.5224 & 0.0532 & -0.3951 & -1.6189 & 0.8297 & 1.2199 & 2.0025 & -2.1509 \\ 2.5684 & 0.0532 & 46.2932 & 7.2216 & -0.2902 & -3.5690 & 1.9490 & 0.6952 & 0.6800 \\ 3.2499 & -0.3951 & 7.2216 & 52.5389 & 5.4007 & -10.0207 & 4.7486 & -0.8224 & -0.6683 \\ 17.1778 & -1.6189 & -0.2902 & 5.4007 & 82.1085 & 0.7064 & -1.6768 & -0.1888 & 0.2941 \\ -1.9280 & 0.8297 & -3.5690 & -10.0207 & 0.7064 & 41.8660 & -4.7655 & -0.0743 & -0.0544 \\ -4.0233 & 1.2199 & 1.9490 & 4.7486 & -1.6768 & -4.7655 & 37.4016 & 0.0164 & 0.0312 \\ 0.5917 & 2.0025 & 0.6952 & -0.8224 & -0.1888 & -0.0743 & 0.0164 & 250.7799 & -1.5086 \\ -0.6695 & -2.1509 & 0.6800 & -0.6683 & 0.2941 & -0.0544 & 0.0312 & -1.5086 & 250.2398 \end{bmatrix}$$

$$K = \begin{bmatrix} -1.0042 & -0.7051 & -2.0330 & -1.3670 \\ -0.6276 & 0.9263 & -1.6811 & 2.3004 \\ -1.0542 & 0.6113 & 1.6245 & -0.5767 \\ 0.4821 & 0.9929 & -1.2728 & -1.9704 \\ 0.0268 & 0.0275 & 0.0014 & 0.0094 \\ 0.0254 & -0.0262 & -0.0343 & 0.0356 \\ -0.0201 & -0.0205 & -0.0612 & -0.0507 \\ 0.0221 & -0.0289 & -0.0526 & 0.0525 \\ -0.0422 & -0.0362 & 0.0052 & 0.0001 \\ 0.0029 & -0.0017 & 0.0103 & -0.0143 \\ 0.0235 & -0.0290 & 0.0609 & -0.0632 \\ -0.0195 & -0.0183 & 0.0312 & 0.0298 \\ -0.0083 & -0.0046 & -0.0131 & -0.0121 \\ -0.0123 & 0.0142 & 0.0073 & -0.0059 \\ -0.0017 & -0.0009 & -0.0022 & -0.0007 \\ -0.0001 & 0.0017 & -0.0006 & 0.0015 \\ 0.1221 & -0.1247 & 5.2316 & 0.0110 \\ -0.1289 & 0.1163 & 0.0111 & 5.2374 \end{bmatrix}$$

Then the static Kalman observer works in each circle

$$\widehat{X}_{k+1|k} = A_{hat}\widehat{X}_{k|k} + B_{hat}U_k + A_{hat}K\left(Y_k - C_{hat}\widehat{X}_{k|k}\right)$$

The dynamic Kalman observer does not calculate the infinite time gain offline like the static Kalman observer, but updates the gain of the next step in each cycle

Algorithm 1 Dynamic Kalman observer

1: Given the last observed $\widehat{X}_{k|k-1}$,last Riccati solution $P_{k|k-1}(P_0 = G_{w,hat}QG'_{w,hat})$ and current measurements Y_k
2: Compute the innovation $\widehat{Y}_{k|k-1} = C_{hat}\widehat{X}_{k|k-1}$, $e_k = Y_k - \widehat{Y}_{k|k-1}$ and $R_{e,k} = C_{hat}P_{k|k-1}C'_{hat} + R_{k|k-1}$
3: Compute the filtered state $K = P_{k|k-1}C'_{HAT}R^{-1}_{e,k}$, $\widehat{X}_{k|k} = \widehat{X}_{k|k-1} + Ke_k$ and $P_{k|k} = P_{k|k-1} - KR_{e,k}K'$
4: State Predictions $\widehat{X}_{k+1|k} = A_{hat}\widehat{X}_{k|k} + B_{hat}U_k$ and $P_{k+1|k} = A_{hat}P_{k|k}A'_{hat} + G_{w,hat}QG'_{w,hat}$
5: If multiple steps need to be predicted j=1,2,...,N-1 $\widehat{X}_{k+1+j|k} = A_{hat}\widehat{X}_{k+j|k} + B_{hat}U_k$ and $P_{k+1+j|k} = A_{hat}P_{k+j|k}A'_{hat} + G_{w,hat}QG'_{w,hat}$
6: Output Predictions $\widehat{Y}_{k+1|k} = C_{hat}\widehat{X}_{k+1|k}$ and $R_{k+1|k} = C_{hat}P_{k+1|k}C_{hat}$ =0

The designed observer was tested under eight conditions

- Linear system using static Kalman observer of the linearized model
- Linear system using dynamic Kalman observer of the linearized model
- Nonlinear system using static Kalman observer of the linearized model
- Nonlinear system using dynamic Kalman observer of the linearized model
- Linear system using static Kalman observer of the identified model
- Linear system using dynamic Kalman observer of the identified model
- Nonlinear system using static Kalman observer of the identified model
- Nonlinear system using dynamic Kalman observer of the identified model

Since the unknown disturbances are stochastic variables but do not contains step changes, the signal of disturbance of test for linear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 30 \\ 30 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

the signal of disturbance of test for nonlinear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 280 \\ 280 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

In the case of Linear system using static Kalman observer of the linearized model, the test result is

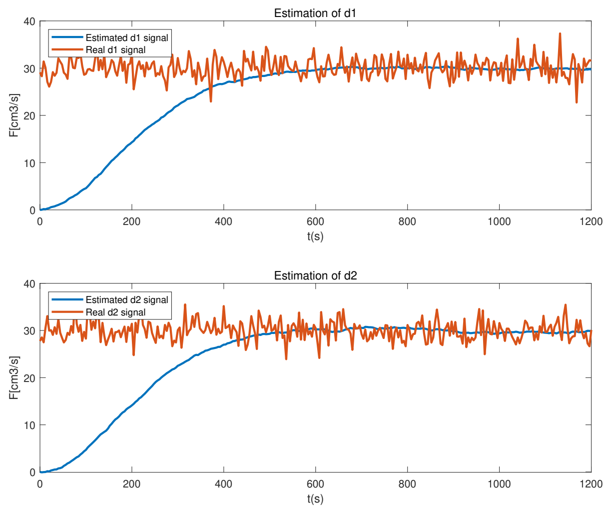

Figure 5.1: Estimated d of Linear system using static Kalman observer of the linearized model.

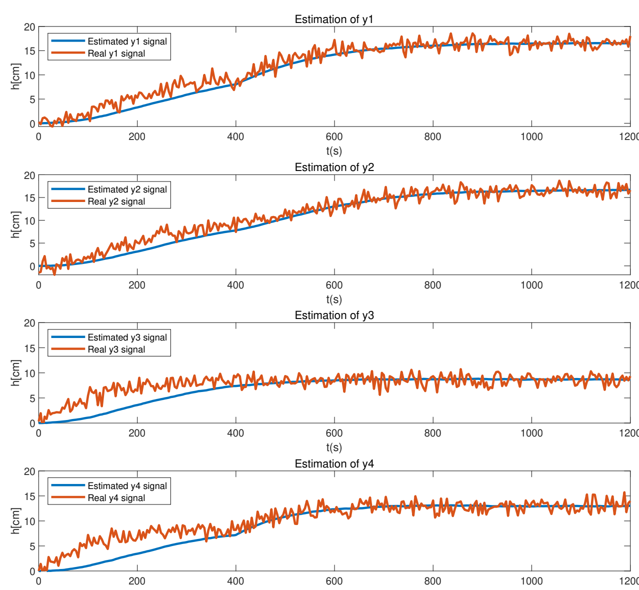

Figure 5.2: Estimated y of Linear system using static Kalman observer of the linearized model.

In the case of Linear system using dynamic Kalman observer of the linearized model, the test result is

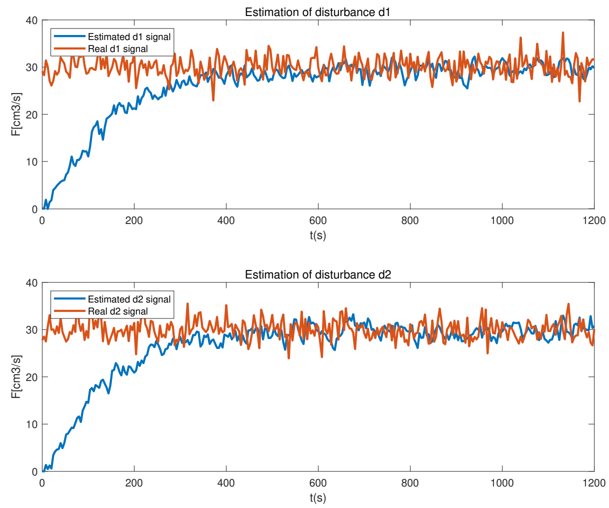

Figure 5.3: Estimated d of Linear system using dynamic Kalman observer of the linearized model.

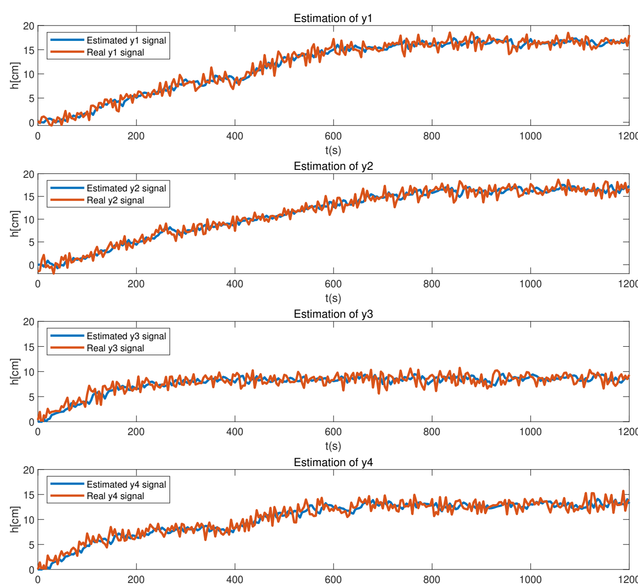

Figure 5.4: Estimated y of Linear system using dynamic Kalman observer of the linearized model.

In the case of nonlinear system using static Kalman observer of the linearized model, the test result is

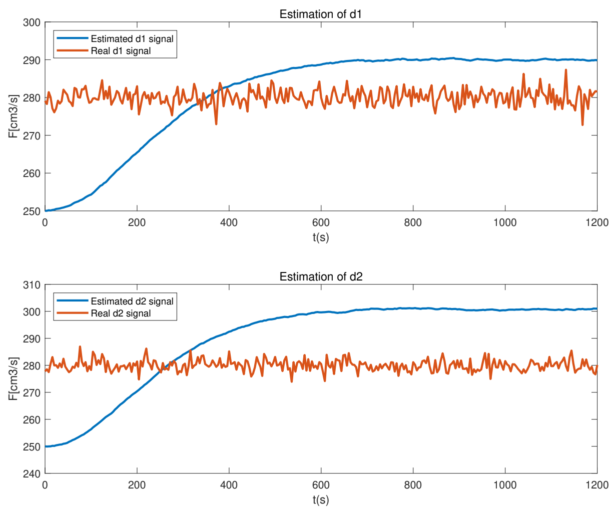

Figure 5.5: Estimated d of nonlinear system using static Kalman observer of the linearized model.

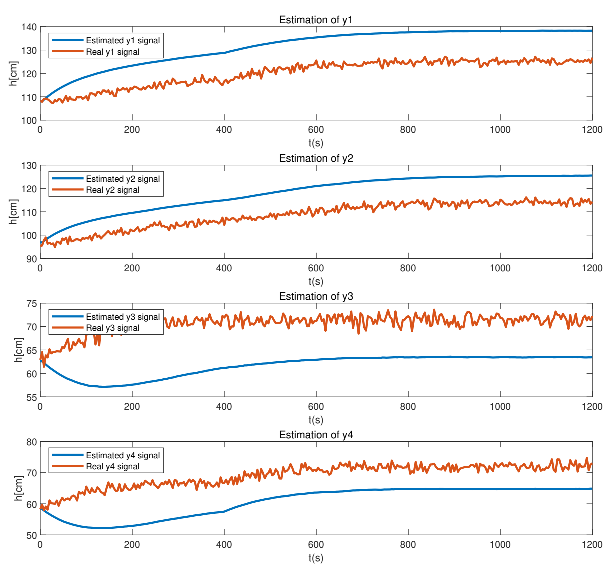

Figure 5.6: Estimated y of nonlinear system using static Kalman observer of the linearized model.

In the case of Nonlinear system using dynamic Kalman observer of the linearized model, the test result is

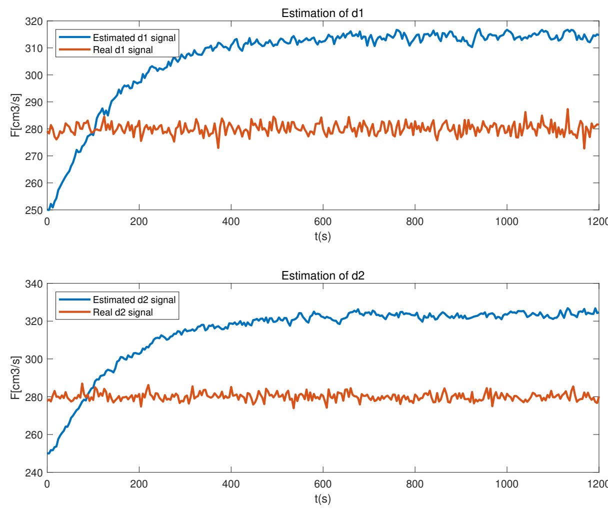

Figure 5.7: Estimated d of Nonlinear system using dynamic Kalman observer of the linearized model.

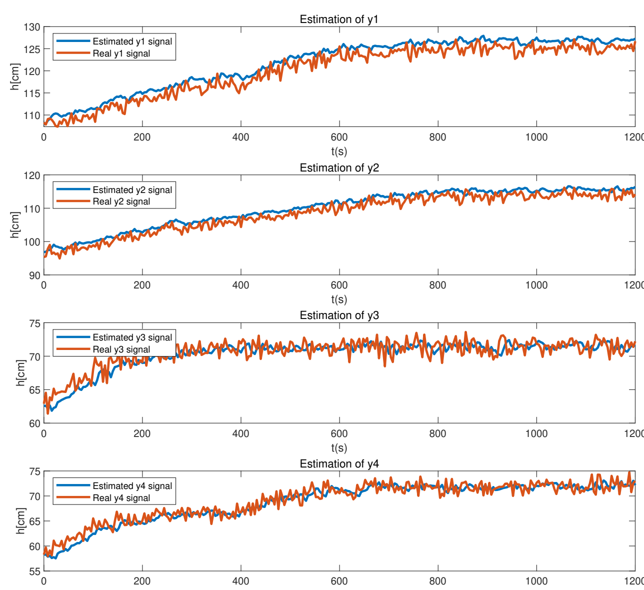

Figure 5.8: Estimated y of Nonlinear system using dynamic Kalman observer of the linearized model.

In the case of Linear system using static Kalman observer of the identified model, the test result is

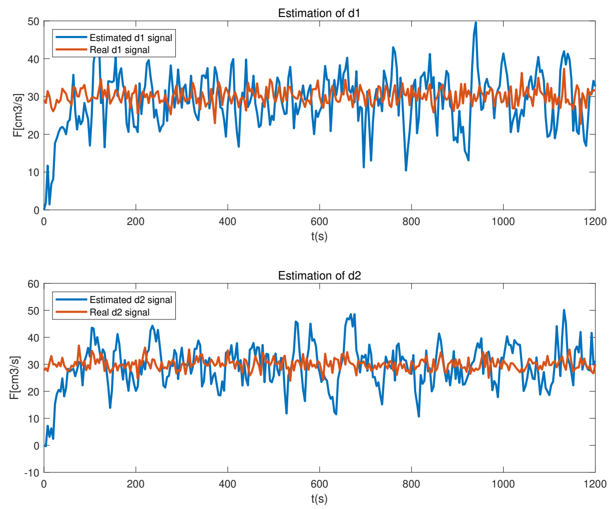

Figure 5.9: Estimated d of Linear system using static Kalman observer of the identified model.

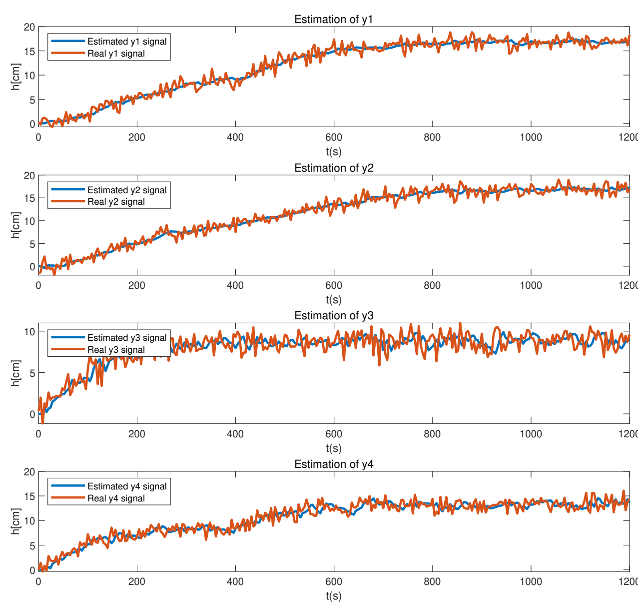

Figure 5.10: Estimated y of Linear system using static Kalman observer of the identified model.

In the case of Linear system using dynamic Kalman observer of the identified model, the test result is

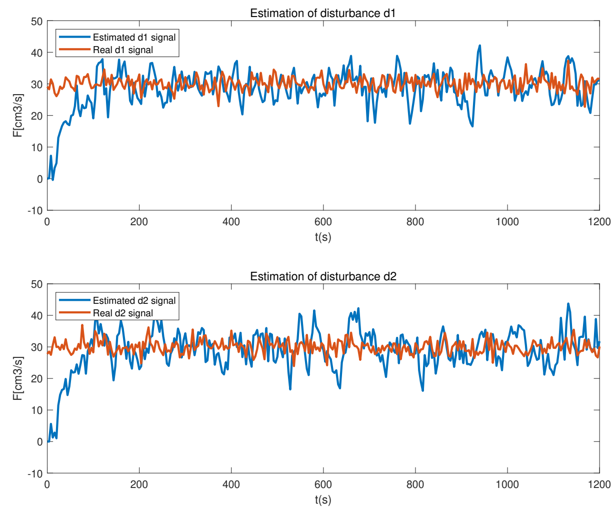

Figure 5.11: Estimated d of Linear system using dynamic Kalman observer of the identified model.

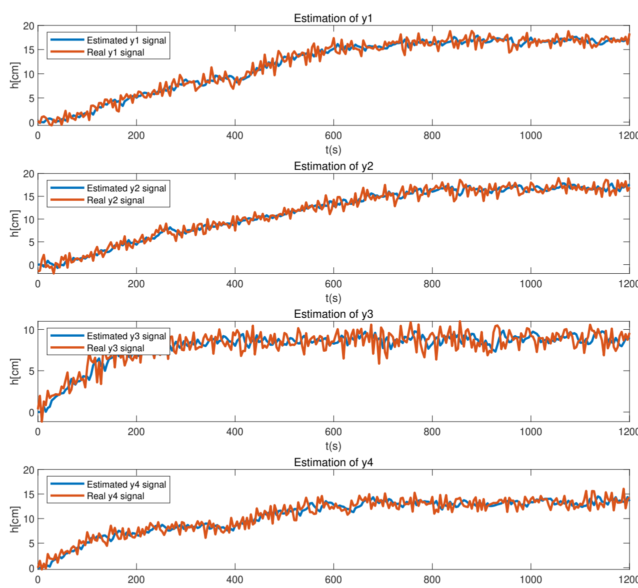

Figure 5.12: Estimated y of Linear system using dynamic Kalman observer of the identified model.

In the case of Nonlinear system using static Kalman observer of the identified model, the test result is

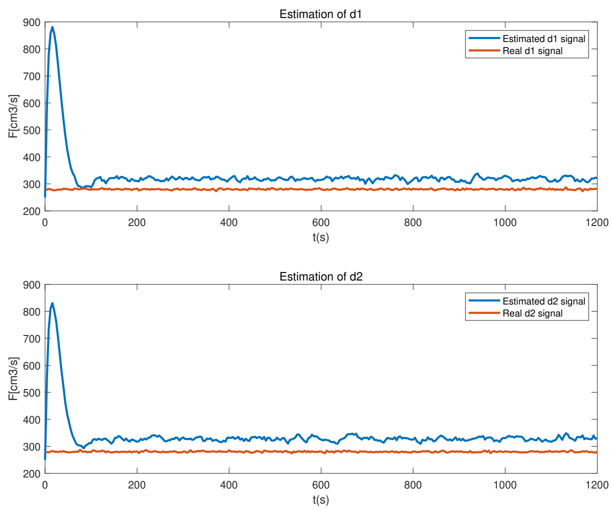

Figure 5.13: Estimated d of Nonlinear system using static Kalman observer of the identified model.

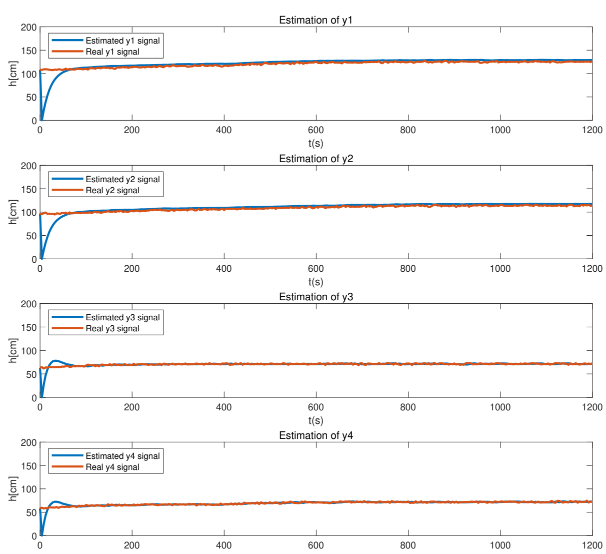

Figure 5.14: Estimated y of nonlinear system using static Kalman observer of the identified model.

In the case of Nonlinear system using dynamic Kalman observer of the identified model, the test result is

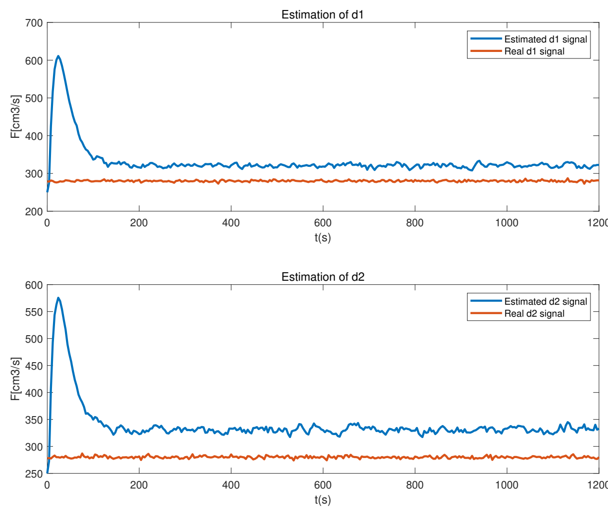

Figure 5.15: Estimated d of Nonlinear system using dynamic Kalman observer of the identified model.

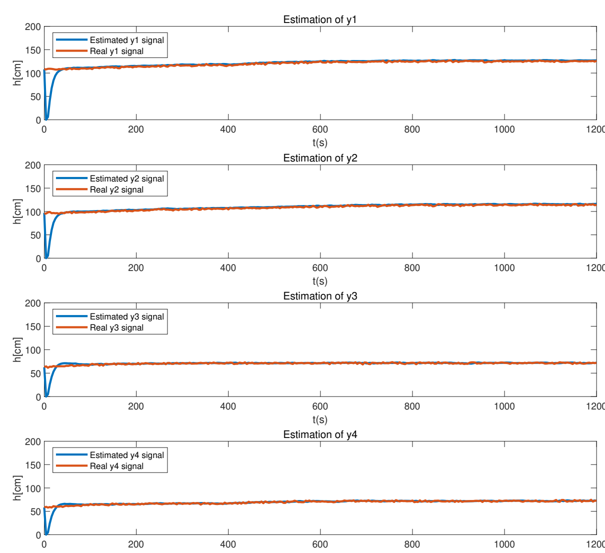

Figure 5.16: Estimated y of Nonlinear system using dynamic Kalman observer of the identified model.

5.3 Problem 5.3

Question: Design and evaluate static and dynamic Kalman filters for the linear models identified in Problem 3 and Problem 4. You should simulate the case where the unknown disturbances are stochastic variables but DO CONTAIN step changes. Design Kalman Filters that do not have steady state offsets.

The designed observer was tested under eight conditions

- Linear system using static Kalman observer of the linearized model
- Linear system using dynamic Kalman observer of the linearized model
- Nonlinear system using static Kalman observer of the linearized model
- Nonlinear system using dynamic Kalman observer of the linearized model
- Linear system using static Kalman observer of the identified model
- Linear system using dynamic Kalman observer of the identified model
- Nonlinear system using static Kalman observer of the identified model
- Nonlinear system using dynamic Kalman observer of the identified model

Since the unknown disturbances are stochastic variables but DO CONTAIN step changes, the signal of disturbance of test for linear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 30 \\ 30 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right) \Longrightarrow N_{iid}\left(\begin{bmatrix} 100 \\ 100 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

the signal of disturbance of test for nonlinear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 280 \\ 280 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right) \Longrightarrow N_{iid}\left(\begin{bmatrix} 350 \\ 350 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

In the case of Linear system using static Kalman observer of the linearized model, the test result is

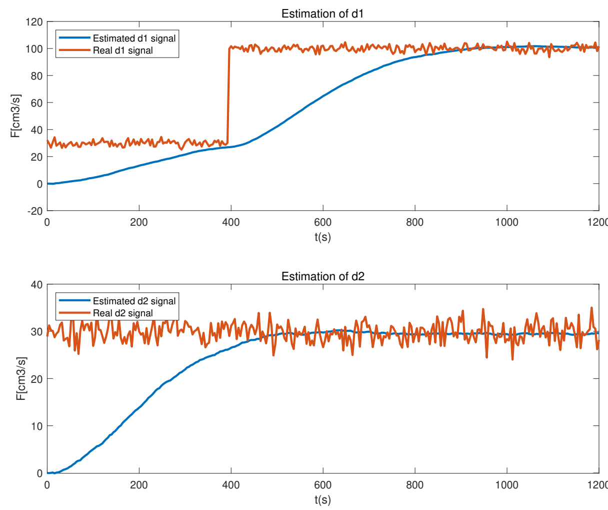

Figure 5.17: Estimated d of Linear system using static Kalman observer of the linearized model.

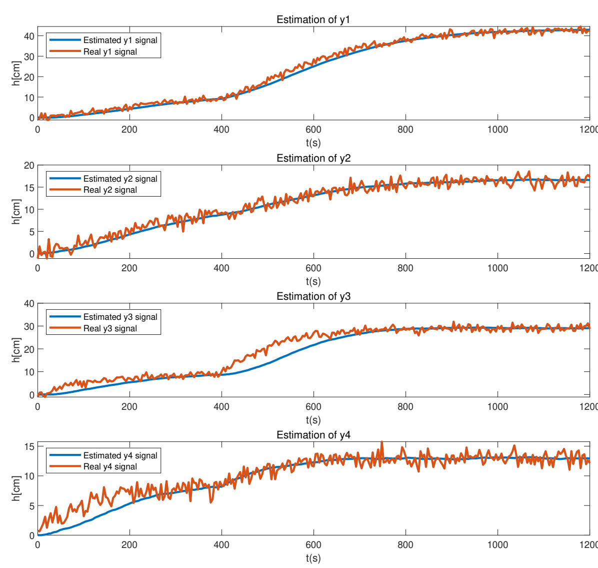

Figure 5.18: Estimated y of Linear system using static Kalman observer of the linearized model.

In the case of Linear system using dynamic Kalman observer of the linearized model, the test result is

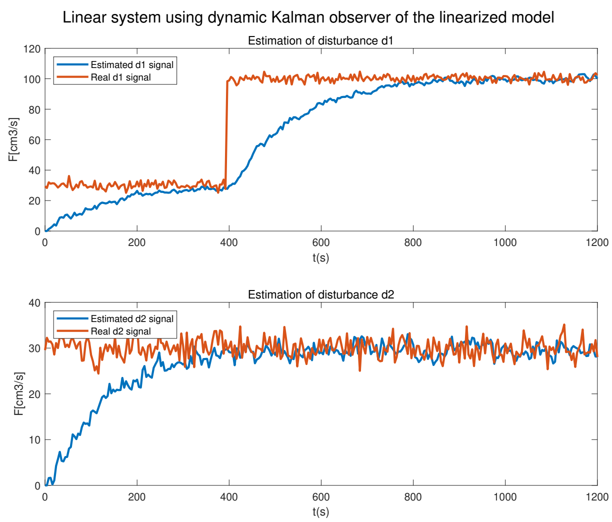

Figure 5.19: Estimated d of Linear system using dynamic Kalman observer of the linearized model.

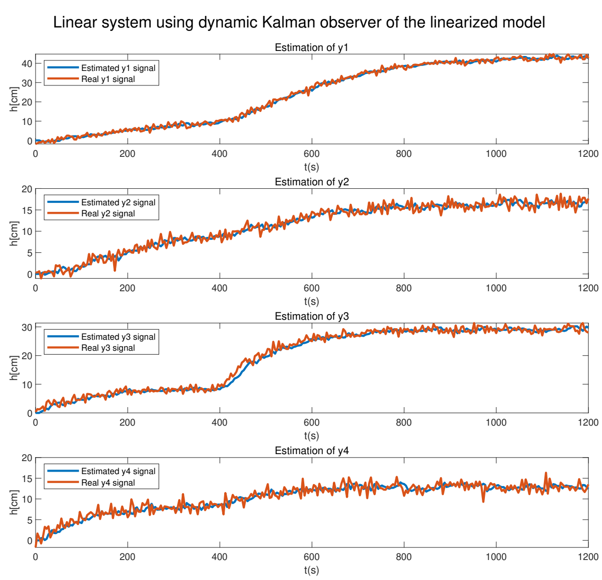

Figure 5.20: Estimated y of Linear system using dynamic Kalman observer of the linearized model.

In the case of Linear system using dynamic Kalman observer of the linearized model, the test result is

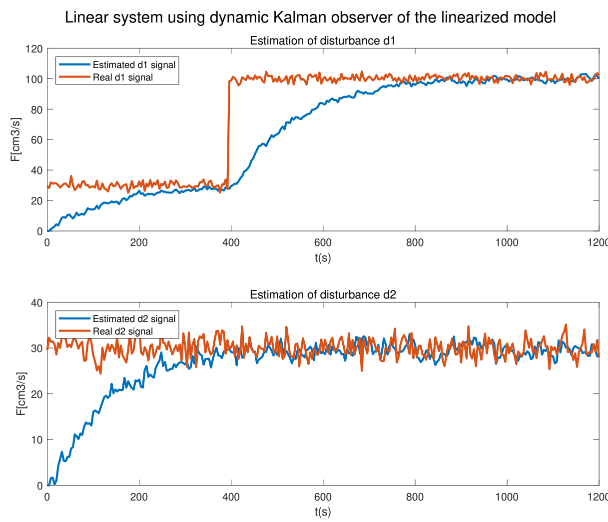

Figure 5.21: Estimated d of Linear system using dynamic Kalman observer of the linearized model.

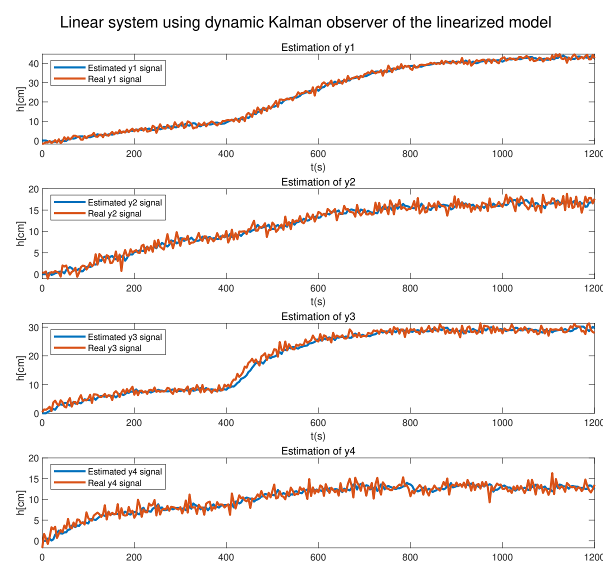

Figure 5.22: Estimated y of Linear system using dynamic Kalman observer of the linearized model.

In the case of Nonlinear system using dynamic Kalman observer of the linearized model, the test result is

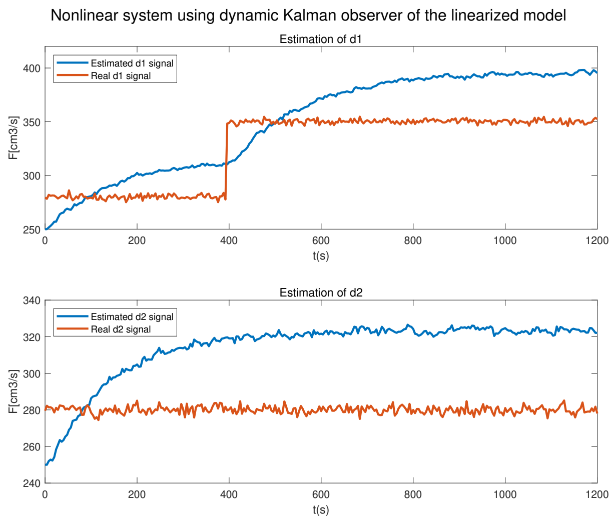

Figure 5.23: Estimated d of Nonlinear system using dynamic Kalman observer of the linearized model.

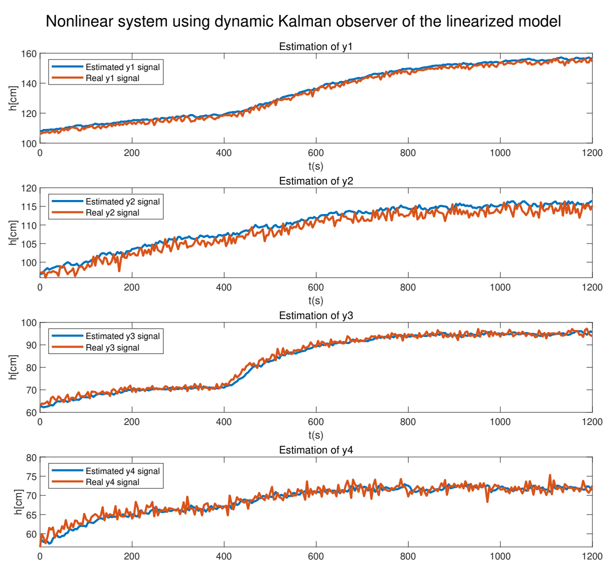

Figure 5.24: Estimated y of Nonlinear system using dynamic Kalman observer of the linearized model.

In the case of Linear system using static Kalman observer of the identified model, the test result is

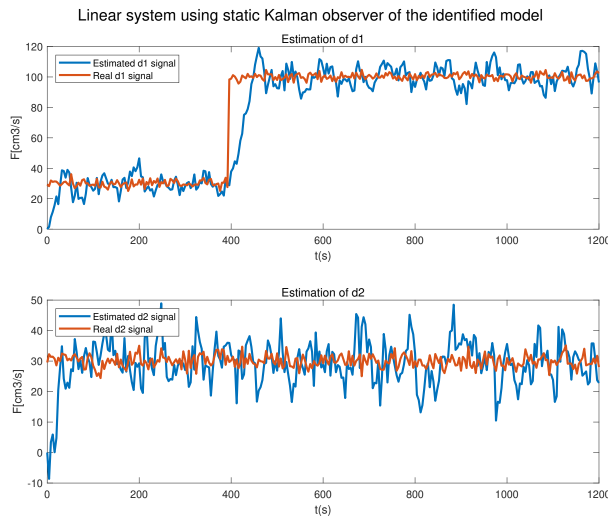

Figure 5.25: Estimated d of Linear system using static Kalman observer of the identified model.

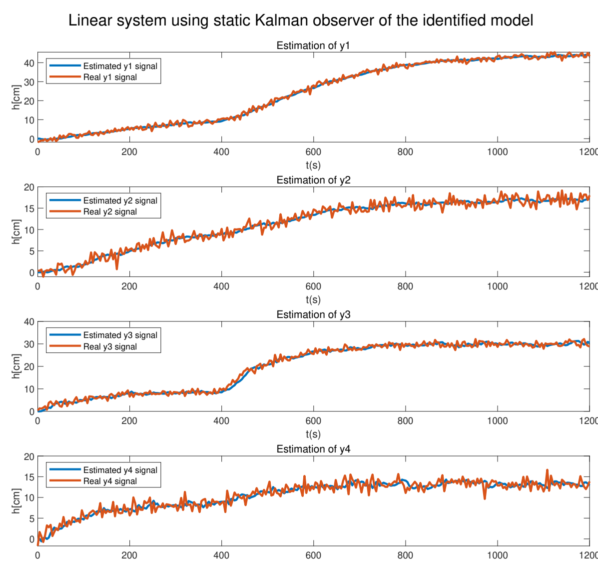

Figure 5.26: Estimated y of Linear system using static Kalman observer of the identified model.

In the case of Linear system using dynamic Kalman observer of the identified model, the test result is

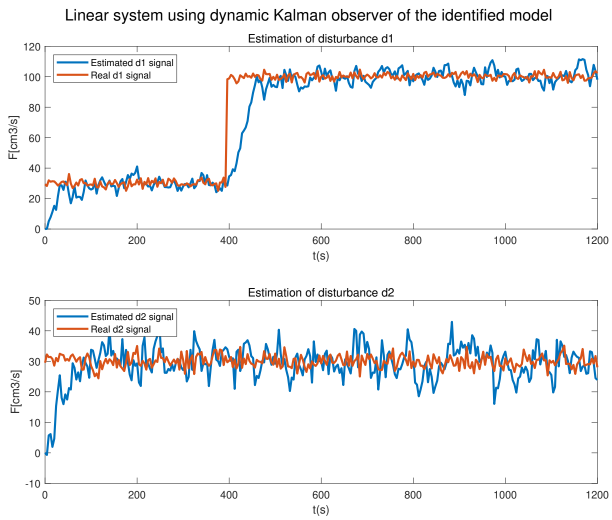

Figure 5.27: Estimated d of Linear system using dynamic Kalman observer of the identified model.

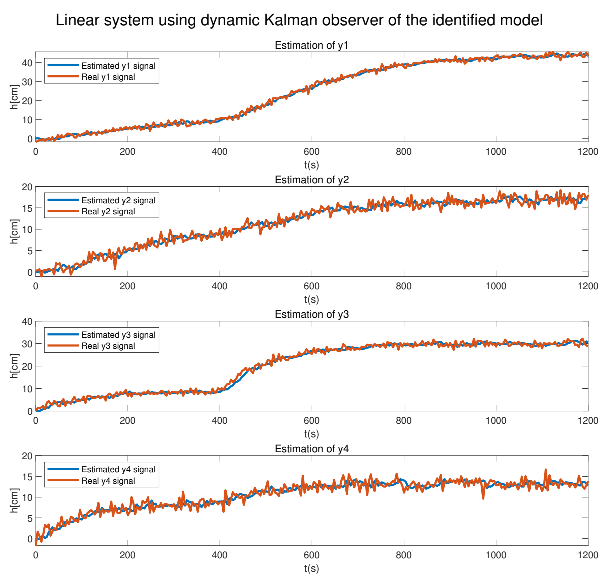

Figure 5.28: Estimated y of Linear system using dynamic Kalman observer of the identified model.

In the case of Linear system using dynamic Kalman observer of the identified model, the test result is

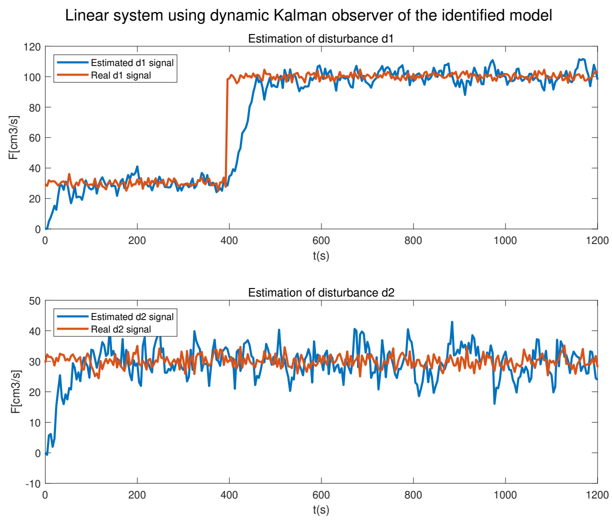

Figure 5.29: Estimated d of Linear system using dynamic Kalman observer of the identified model.

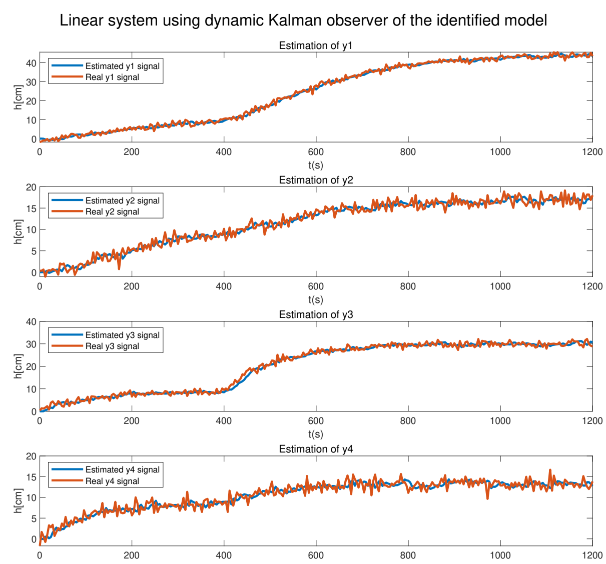

Figure 5.30: Estimated y of Linear system using dynamic Kalman observer of the identified model.

In the case of Nonlinear system using dynamic Kalman observer of the identified model, the test result is

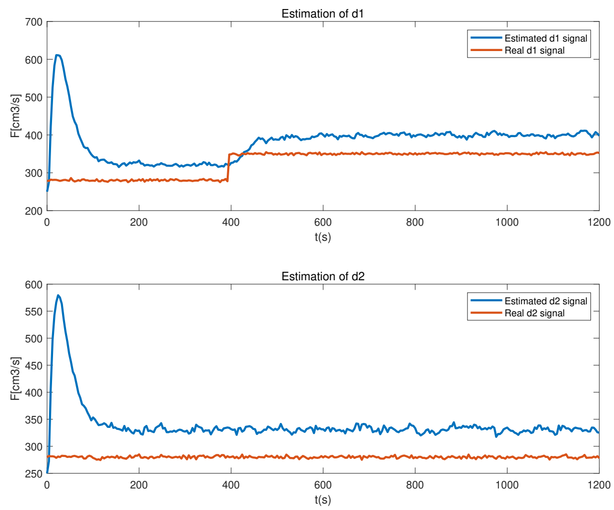

Figure 5.31: Estimated d of Nonlinear system using dynamic Kalman observer of the identified model.

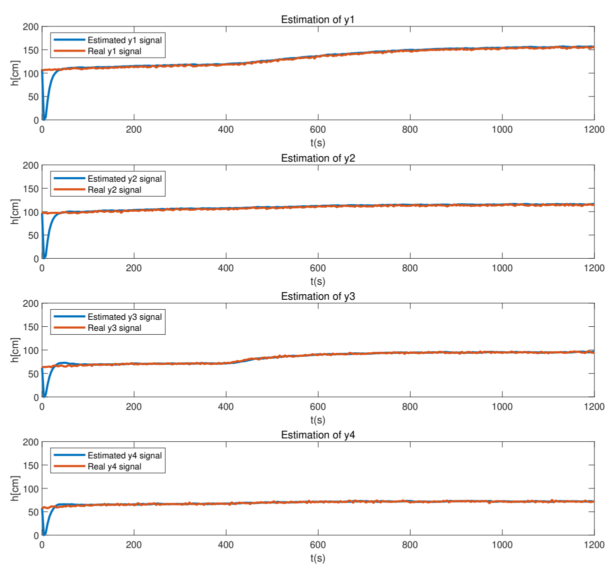

Figure 5.32: Estimated y of Nonlinear system using dynamic Kalman observer of the identified model.

5.4 Problem 5.4

Question: Discuss and evaluate the Kalman filters by simulation on the linear and nonlinear models.

First of all, whether it is based on the model obtained by direct linearization (Problem 4) or the model obtained by identification (Problem 3), it is an observer based on the linearized model design. Both static and dynamic Kalman observers can quickly and accurately observe the output and disturbances in a linear system, and can quickly follow the trend of output and interference in a nonlinear system, but there will be a constant deviation.

The comparison of the performance of the static and dynamic observers in the linear system based on the problem 3 model is shown

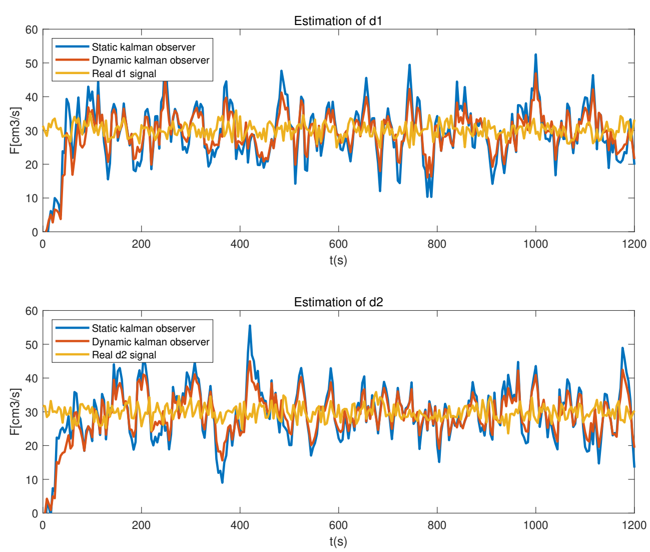

Figure 5.33: Estimated d of linear system using static and dynamic Kalman observer of the problem 3 model.

The comparison of the performance of the static and dynamic observers in the linear system based on the problem 3 model is shown

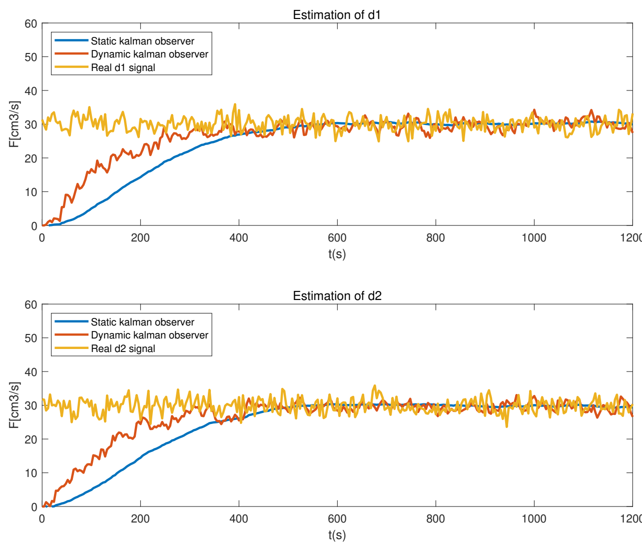

Figure 5.34: Estimated d of linear system using static and dynamic Kalman observer of the problem 4 model.

At the same time, the observer designed based on the model obtained by direct linearization (Problem 4) performs better in the observation process of the nonlinear system than the observer designed by the model obtained by identification (Problem 3). There is no overshoot and the steady-state error is smaller.

The comparison of the performance of the static observers in the nonlinear system based on the problem 3 and 4 model is shown

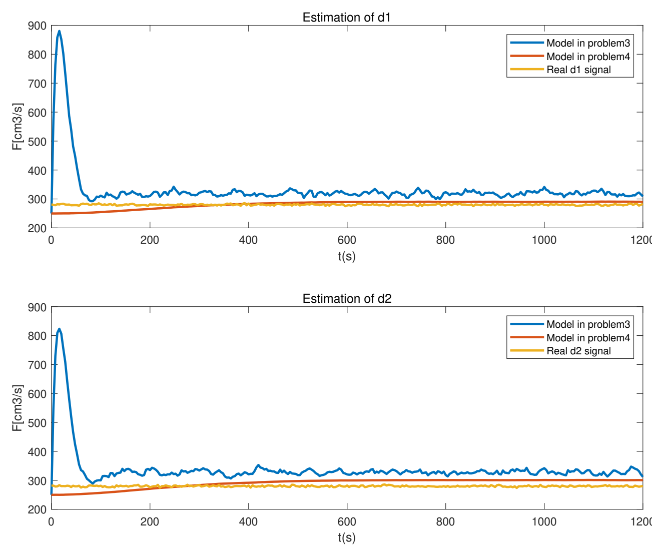

Figure 5.35: Estimated d of nonlinear system using static Kalman observer of the problem 3 and 4 model.

The comparison of the performance of the dynamic observers in the nonlinear system based on the problem 3 and 4 model is shown

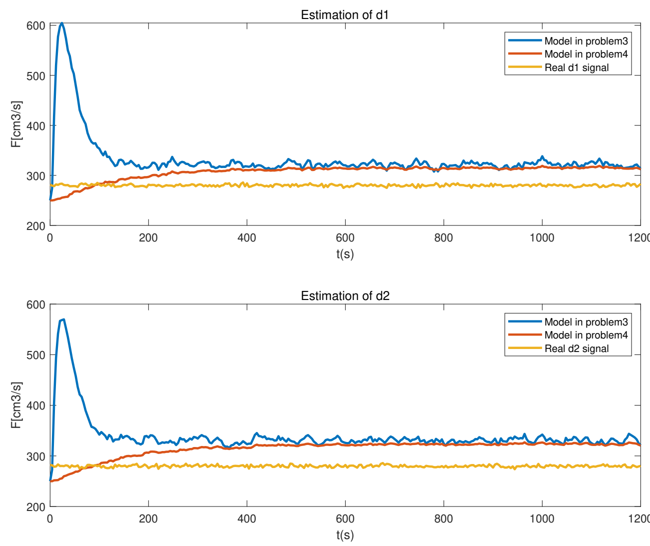

Figure 5.36: Estimated d of nonlinear system using dynamic Kalman observer of the problem 3 and 4 model.

This is supposed to be related to the model. The directly linearized model has only 4 states, and has a physical meaning (the quality of the water in the tank), which can be observed based on the accurate initial value of the state. However, the identified model has 16 states through the minimum realization, and the corresponding initial state values cannot be known.

CHAPTER 6

QP solver interface

Question: Implement a QP solver interface
function [x,info] = qpsolver(H,g,l,u,A,bl,bu,xinit)
for solution of the convex quadratic program

$$\begin{array}{ll} \min_{x \in R^n} & \phi = \frac{1}{2}x'Hx + g'x \\ \text{s.t.} & l \leq x \leq u \\ & b_l \leq Ax \leq b_u \end{array}$$

using the QP solver in Matlab, i.e. quadprog

```
function [x,info] = qpsolver(H,g,l,u,A,bl,bu,xinit)
%A QP solver interface
% min   J=0.5*x'*H*x+g'*x
% s.t.  l<=x<=u
% s.t.  bl<=A*x<=bu
% Syntax: [x,info] = qpsolver(H,g,l,u,A,bl,bu,xinit)
%          info.fval: minimum value
%          info.exitflag: Reason quadprog stopped
%          info.output: optimization process
%          info.lambda: Lagrange multipliers at the solution
Ai=[A;-A];
bi=[bu;-bl];
options = optimoptions('quadprog','Algorithm',"active-set");
```

```
[x,fval,exitflag,output,lambda] = quadprog(H,g,Ai,bi,[],[],l,u,
    xinit,options);
info.fval=fval;
info.exitflag=exitflag;
info.output=output;
info.lambda=lambda;
end
```

.

CHAPTER 7

Unconstrained MPC

7.1 Problem 7.1

Question: Implement a function for design of an Unconstrained MPC based on discrete-time state space models. You should explain in the report how your Matlab function work and its theoretical background.

In unconstrained linear MPC, an optimization problem is constructed to solve the optimal input(u) in the prediction period to obtain the best performance. First of all, the performance is represented by a cost function, which includes the difference between the controlled value(z) and the given reference value(r) during the prediction period and the speed of the input change during the prediction period. The relationship between the controlled value(z) and the input(u) is expressed by a discrete state space equation and used as an equation constraint for the optimization problem

$$\begin{aligned} &\min \phi = \tfrac{1}{2}\textstyle\sum_{k=0}^{N} \|z_k - r_k\|_{Qz}^2 + \tfrac{1}{2}\textstyle\sum_{k=0}^{N-1} \|\Delta u_k\|_S^2 \\ &\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k + Ed_k \quad k = 0, 1, \ldots, N-1 \\ &\qquad\quad z_k = C_z x_k \quad k = 0, 1, \ldots, N \end{aligned}$$

For this optimization problem, both analytical and numerical solutions were implemented. First, the relationship between the controlled value and the input can be expressed in discrete time as

$$\begin{aligned} z_k &= C_z A^k x_0 + \sum_{j=0}^{k-1} C_z A^{k-1-j} B u_j + \sum_{j=0}^{k-1} C_z A^{k-1-j} E d_j \\ &= C_z A^k x_0 + \sum_{j=0}^{k-1} H_{k-j} u_j + \sum_{j=0}^{k-1} H_{k-j}^E d_j \end{aligned}$$

Where H is the Impulse Response Coefficients of the input u, H^E is the Impulse Response Coefficients of the disturbance d

$$\begin{aligned} H_i &= 0 && i=0 \\ H_i &= C_z A^{i-1} B && i \geq 1 \\ H_i^E &= 0 && i=0 \\ H_i^E &= C_z A^{i-1} E && i \geq 1 \end{aligned}$$

It can be expressed as a matrix in the prediction period as

$$\phi_z = \frac{1}{2}\sum_{k=1}^{N} \|z_k - r_k\|_{Q_z}^2 = \frac{1}{2}\left\| \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} - \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \right\|_{Q_z}^2 \qquad Q_z = \begin{bmatrix} Q_{z1} & & & \\ & Q_{z2} & & \\ & & \ddots & \\ & & & Q_{zN} \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_N \end{bmatrix}}_{Z} = \underbrace{\begin{bmatrix} C_z A \\ C_z A^2 \\ C_z A^3 \\ \vdots \\ C_z A^N \end{bmatrix}}_{\Phi} x_0 + \underbrace{\begin{bmatrix} H_1 & 0 & 0 & \dots & 0 \\ H_2 & H_1 & 0 & \dots & 0 \\ H_3 & H_2 & H_1 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ H_N & H_{N-1} & H_{N-2} & \dots & H_1 \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{U}$$
$$+ \underbrace{\begin{bmatrix} H_1^E & 0 & 0 & \dots & 0 \\ H_2^E & H_1^E & 0 & \dots & 0 \\ H_3^E & H_2^E & H_1^E & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ H_N^E & H_{N-1}^E & H_{N-2}^E & \dots & H_1^E \end{bmatrix}}_{\Gamma^d} \underbrace{\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1} \end{bmatrix}}_{D}$$

It can be simply expressed as

$$\phi_z = \frac{1}{2}\sum_{k=1}^{N} \|z_k - r_k\|_{Q_z}^2 = \frac{1}{2}\|\Gamma U - b\|_{Q_z}^2$$

$$b = R - \Phi x_0 - \Gamma^d D$$

For the form $\phi_z = U'H_zU + g_z'U + \rho_z$, we have

$$\begin{aligned} H_z &= \Gamma' Q_z \Gamma \\ g_z &= -\Gamma' Q_z b = -\Gamma' Q_z \left(R - \Phi x_0 - \Gamma^d D\right) \\ &= M_R R + M_{x_0} x_0 + M_d D \\ \rho_z &= \frac{1}{2} b' \mathcal{Q}_z b \end{aligned}$$

where

$$M_R = -\Gamma' Q_z \qquad M_{x_0} = \Gamma' Q_z \Phi \qquad M_d = \Gamma' Q_z \Gamma^d$$

The speed of the input change during the prediction period can be expressed in the form of a matrix as

$$\phi_{\triangle u} = \frac{1}{2} \sum_{k=0}^{N-1} \|\Delta u_k\|_S^2$$

$$\phi_{\triangle u} = \frac{1}{2} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}' \begin{bmatrix} 2S & -S & & & \\ -S & 2S & -S & & \\ & -S & 2S & -S & \\ & & \ddots & \ddots & -S \\ & & & -S & S \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} + \frac{1}{2} u_0 S u_0 - u_{-1} S u_0 + \frac{1}{2} u_{-1} S u_{-1}$$

It can be further expressed as

$$\phi_{\triangle u} = \frac{1}{2} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}' \underbrace{\begin{bmatrix} 2S & -S & & & \\ -S & 2S & -S & & \\ & -S & 2S & -S & \\ & & \ddots & \ddots & -S \\ & & & -S & S \end{bmatrix}}_{H_s} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} + \underbrace{\left(- \begin{bmatrix} S \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_{-1} \right)'}_{M_{u_{-1}}} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} + \frac{1}{2} u_{-1} S u_{-1}$$

Then the complete value function can be expressed as

$$\begin{aligned} H &= \Gamma' \mathcal{Q}_z \Gamma + H_s \\ g &= M_R R + M_{x_0} x_0 + M_d D + M_{u_{-1}} u_{-1} \end{aligned}$$

Regarding the numerical solution, H and g can be directly put into the solver implemented in the previous section for solution.

Regarding the analytical solution, deriving the value function

$$\begin{aligned} U^* =& - H^{-1} g \\ =& - H^{-1} \left(M_{x_0} x_0 + M_R R + M_{u_{-1}} u_{-1} + M_d D \right) \\ =& L_{x_0} x_0 + L_R R + L_{u_{-1}} u_{-1} + L_d D \end{aligned}$$

Where

$$L_{x_0} = -H^{-1} M_{x_0} \qquad L_R = -H^{-1} M_R \qquad L_{u_{-1}} = -H^{-1} M_{u_{-1}} \qquad L_d = -H^{-1} M_d$$

The input at the first time point can be expressed as

$$u^* = K_{x_0} x_0 + K_R R + K_{u_{-1}} u_{-1} + K_d D$$

Where

$$K_{x_0} = L_{x_0,0} \qquad K_R = L_{R,0} \qquad K_{u_{-1}} = L_{u_{-1},0} \qquad K_d = L_{d,0}$$

However, if the observer is not effective in observing disturbances, then using the disturbance of the entire prediction time period will cause bigger error, so only using the disturbances observed at the current time point can appropriately reduce the error, so Γ^d will be removed from and expressed by Φ^d in g, then the optimization problem can be expressed as

$$\begin{aligned} H &= \Gamma' Q_z \Gamma + H_s \\ g &= M_R R + M_{x_0} x_0 + M_d D + M_{u_{-1}} u_{-1} \end{aligned}$$

Where

$$M_R = -\Gamma' Q_z \qquad M_{x_0} = \Gamma' Q_z \Phi \qquad M_d = \Gamma' Q_z \Phi^d \qquad \Phi^d = \begin{bmatrix} C_z A E \\ C_z A^2 E \\ C_z A^3 E \\ \vdots \\ C_z A^N E \end{bmatrix}$$

In the unconstrained MPC function, these necessary matrices are constructed and the optimal input is calculated

Algorithm 2 Unconstrained MPC for analytical solution

1: Given the last input u_{-1},current state x and disturbance d, and the reference R_z throughout the prediction period
2: Constructing matrix Q_z, Γ and Compute $H_z = \Gamma' Q_z \Gamma$
3: Constructing matrix $\Phi, \Phi^d, M_{u_{-1}}$ and Compute $M_R = -\Gamma' Q_z, M_{x_0} = \Gamma' Q_z \Phi, M_d = \Gamma' Q_z \Phi^d$
4: Constructing matrix H_s and Compute $H = H_z + H_s Q_z \Gamma, g = M_R R + M_{x_0} x_0 + M_d D + M_{u_{-1}} u_{-1}$
5: Cholesky factorize H: $H = R'_H R_H$
6: Solve by backsubstitution: $U = \backslash(R'_H \backslash - g)$
=0

Algorithm 3 Unconstrained MPC for Numerical solution

1: Given the last input u_{-1},current state x and disturbance d, and the reference R_z throughout the prediction period
2: Constructing matrix Q_z, Γ and Compute $H_z = \Gamma' Q_z \Gamma$
3: Constructing matrix $\Phi, \Phi^d, M_{u_{-1}}$ and Compute $M_R = -\Gamma' Q_z, M_{x_0} = \Gamma' Q_z \Phi, M_d = \Gamma' Q_z \Phi^d$
4: Constructing matrix H_s and Compute $H = H_z + H_s Q_z \Gamma, g = M_R R + M_{x_0} x_0 + M_d D + M_{u_{-1}} u_{-1}$
5: Use qpsolver to compute the optimal U
6: Choose U's first input u_0
=0

7.2 Problem 7.2

Question: Design unconstrained MPC for the models identified in Problem 3 and Problem 4.

The model in Problem 3 is constructed by obtaining the transfer function (u, d to y) through system identification of the step response of the nonlinear model, and further transforms it into a discrete state space equation.

The model in Problem 4 is a discrete state space equation obtained by linearization near the steady state point of the nonlinear model.

When the regulator function of MPC is designed, MPC regulator needs

- Input(u_{-1}) at the previous time point
- Current status(x)
- Current disturbance(d) or disturbance(D) throughout the prediction period
- The reference of the controlled variables(R_z) throughout prediction period

Where 1 can be obtained directly, and 4 needs to be manually selected. 2 and 3 need the previously designed observer to observe.

As we said when designing the MPC regulator in the previous section, if the observer observes the disturbance well, the disturbance in the entire prediction period can make the input obtained by the MPC better. But if the observer's effect on the disturbance is poor, it is better to only use the disturbance value at the current time point.

The weight selection of the optimization problem in MPC is

$$Q_z = 0.03 \qquad S = 10$$

The prediction length of MPC is

$$N_{pre} = 50$$

7.3 Problem 7.3

Question: Implement and discuss a compute and prediction function for this MPC.

The Matlab code of unconstrained MPC for analytical solution is showed here

```
function u_new=LMPCcompute_uncons_analytical(R,X0,D,U,Ad,Bd,
    Bd_d,Czss,N,Q_cof,u_delta_cof)
% Syntax: u_new=LMPCcompute_analytical(R,Y,D,U,Ad,Bd,Bd_d,Czss,
    N,Q_cof,u_delta_cof)
%         R: Reference throughout the prediction period
%         X0: Observed state x0
%         D: Observed disturbance d0
%         U: Last input U-1
%         Ad,Bd,Bd_d,Czss: Linear model state space matrix
%         N: prediction length
%         Q_cof: Weight matrix for tracking the reference
%         u_delta_cof: Weight matrix for input change rate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         u_new: The first input to be executed
num_x=size(Ad,1);
num_u=size(Bd,2);
num_d=size(Bd_d,2);
num_y=4;
num_z=2;
Phi=zeros(N*num_z,num_x);
Phi_d=zeros(N*num_z,num_d);
Gam=zeros(N*num_z,N*num_u);
%Weight matrix for tracking the reference Qz
```

```
Qz=diag(Q_cof*ones(N*num_z,1)) ;
%Weight matrix for input rate S
Sz=diag(u_delta_cof*ones(num_u,1));
%Quadratic objective term Hs
Hs=zeros(N*num_u,N*num_u);
%Linear objective term Mu_delta
Mu_delta=zeros(N*num_u,num_u);
Mu_delta(1:num_u,:)=-Sz;
U0=zeros(N*num_u,1);
for i=1:N
    %fill the Phi and Phi_d
    Phi((i-1)*num_z+1:i*num_z,:)=Czss*Ad^i;
    Phi_d((i-1)*num_z+1:i*num_z,:)=Czss*Ad^i*Bd_d;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Gam
    Gam_i=zeros(num_z,N*num_u);%[num_x * num_u*N]
    for j=1:i
        Gam_i(:,(j-1)*num_u+1:j*num_u)=Czss*(Ad^(i-j))*Bd;
    end
    Gam((i-1)*num_z+1:i*num_z,:)=Gam_i;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Hs
    if i==1
        Hs(1:num_u,1:2*num_u)=[2*Sz,-Sz];
    elseif i==N
        Hs(end-num_u+1:end,end-2*num_u+1:end)=[-Sz,Sz];
    else
        Hs((i-1)*num_u+1:(i)*num_u,(i-2)*num_u+1:(i+1)*num_u)
           =[-Sz,2*Sz,-Sz];
    end
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the U0
    U0((i-1)*num_u+1:i*num_u,:)=U;
end
Mx0=Gam'*Qz*Phi;
Md=Gam'*Qz*Phi_d;
%Linear objective term Mr
Mr=-Gam'*Qz;
%Quadratic objective term Hr
Hr=Gam'*Qz*Gam;
Hu=Hr+Hs;
g=Mx0*X0+Mr*R+Mu_delta*U+Md*D;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RH=chol(Hu);
u_new_list = RH\(RH'\-g);
u_new=u_new_list(1:num_u);
end
```

.

The Matlab code of unconstrained MPC for numerical solution is showed here

```
function u_new=LMPCcompute_uncons_numerical(R,X0,D,U,Ad,Bd,Bd_d
    ,Czss,N,Q_cof,u_delta_cof)
% Syntax: u_new=LMPCcompute_uncons_numerical(R,X0,D,U,Ad,Bd,
    Bd_d,Czss,N,Q_cof,u_delta_cof)
%         R: Reference throughout the prediction period
%         X0: Observed state x0
%         D: Observed disturbance d0
%         U: Last input U-1
%         Ad,Bd,Bd_d,Czss: Linear model state space matrix
%         N: prediction length
```

```
%         Q_cof: Weight matrix for tracking the reference
%         u_delta_cof: Weight matrix for input change rate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         u_new: The first input to be executed
num_x=size(Ad,1);
num_u=size(Bd,1);
num_d=size(Bd_d,1);
num_y=4;
num_z=2;
Phi=zeros(N*num_z,num_x);
Phi_d=zeros(N*num_z,num_d);
Gam=zeros(N*num_z,N*num_u);
%Weight matrix for tracking the reference Qz
Qz=diag(Q_cof*ones(N*num_z,1)) ;
%Weight matrix for input rate S
Sz=diag(u_delta_cof*ones(num_u,1));
%Quadratic objective term Hs
Hs=zeros(N*num_u,N*num_u);
%Linear objective term Mu_delta
Mu_delta=zeros(N*num_u,num_u);
Mu_delta(1:num_u,:)=-Sz;
U0=zeros(N*num_u,1);
for i=1:N
    %fill the Phi
    Phi((i-1)*num_z+1:i*num_z,:)=Czss*Ad^i;
    Phi_d((i-1)*num_z+1:i*num_z,:)=Czss*Ad^i*Bd_d;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Gam
    Gam_i=zeros(num_z,N*num_u);
    for j=1:i
```

```
        Gam_i(:,(j-1)*num_u+1:j*num_u)=Czss*(Ad^(i-j))*Bd;
    end
    Gam((i-1)*num_z+1:i*num_z,:)=Gam_i;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Hs
    if i==1
        Hs(1:num_u,1:2*num_u)=[2*Sz,-Sz];
    elseif i==N
        Hs(end-num_u+1:end,end-2*num_u+1:end)=[-Sz,Sz];
    else
        Hs((i-1)*num_u+1:(i)*num_u,(i-2)*num_u+1:(i+1)*num_u)
            =[-Sz,2*Sz,-Sz];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the U0
    U0((i-1)*num_u+1:i*num_u,:)=U;
end
Mx0=Gam'*Qz*Phi;
Md=Gam'*Qz*Phi_d;
%Linear objective term Mr
Mr=-Gam'*Qz;
%Quadratic objective term Hr
Hr=Gam'*Qz*Gam;
Hu=Hr+Hs;
g=Mx0*X0+Mr*R+Mu_delta*U+Md*D;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[u_opt,~] = qpsolver(0.5*(Hu+Hu'),g,[],[],[],[],[],U0);
u_new=u_opt(1:num_u);
end
```

.

CHAPTER 8

Input Constrained MPC

8.1 Problem 8.1

Question: Implement a function for design of an input constrained MPC based on discrete-time state space models. You should explain in the report how your Matlab function work and its theoretical background.

In MPC with input constraints, the structure of the optimization problem needs to be changed. Generally speaking, the input constraints for the actual problem must be observed, so the input constraints should be in the form of hard constraints of inequality constraints, rather than soft constraints in the value function. In addition, because of the inequality constraints, optimization problems no longer have analytical solutions, only numerical solutions. The optimization problem is restructured as

$$
\begin{aligned}
\min \quad & \phi = \frac{1}{2}\sum_{k=1}^{N} \|z_k - r_k\|_{Q_z}^2 + \frac{1}{2}\sum_{k=0}^{N-1} \|\Delta u_k\|_S^2 & \\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k + Ed_k & k = 0, 1, \dots, N-1 \\
& z_k = C_z x_k & k = 0, 1, \dots, N \\
& u_{\min} \leq u_k \leq u_{\max} & k = 0, 1, \dots, N-1 \\
& \Delta_{u,\min} \leq \Delta u_k \leq \Delta_{u,\max} & k = 0, 1, \dots, N-1
\end{aligned}
$$

The form of qpsolver constructed in the previous chapter is

$$
\begin{aligned}
\min_{x \in R^n} \quad & \phi = \tfrac{1}{2}x'Hx + g'x \\
\text{s.t.} \quad & l \leq x \leq u \\
& b_l \leq Ax \leq b_u
\end{aligned}
$$

Then the inequality constraints of input can be simply constructed as

$$\underbrace{\begin{bmatrix} u_{min} \\ u_{min} \\ \vdots \\ u_{min} \end{bmatrix}}_{U_{min}} \leq \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \leq \underbrace{\begin{bmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \end{bmatrix}}_{U_{max}}$$

The inequality constraint on the rate of change of input can be expressed as

$$\underbrace{\begin{bmatrix} \Delta_{u,min} + u_{-1} \\ \Delta_{u,min} \\ \Delta_{u,min} \\ \vdots \\ \Delta_{u,min} \end{bmatrix}}_{\Delta_{u,min}} \leq \underbrace{\begin{bmatrix} I & & & & \\ -I & I & & & \\ & -I & I & & \\ & & \ddots & \ddots & \\ & & & -I & I \end{bmatrix}}_{A_{\Delta u}} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} \leq \underbrace{\begin{bmatrix} \Delta_{u,max+u_{-1}} \\ \Delta_{u,max} \\ \Delta_{u,max} \\ \vdots \\ \Delta_{u,max} \end{bmatrix}}_{\Delta_{u,max}}$$

The value function of input Constrained MPC is the same as that of unconstrained mpc. In the input Constrained MPC function, these necessary matrices are constructed and the optimal input is calculated

Algorithm 4 Input Constrained MPC for Numerical solution

1: Given the last input u_{-1},current state x and disturbance d, and the reference R_z throughout the prediction period
2: Constructing matrix Q_z, Γ and Compute $H_z = \Gamma' Q_z \Gamma$
3: Constructing matrix $\Phi, \Phi^d, M_{u_{-1}}$ and Compute $M_R = -\Gamma' Q_z, M_{x_0} = \Gamma' Q_z \Phi, M_d = \Gamma' Q_z \Phi^d$
4: Constructing matrix H_s and Compute $H = H_z + H_s, g = M_R R + M_{x_0} x_0 + M_d D + M_{u_{-1}} u_{-1}$
5: Constructing matrix U_{min},U_{max},$\Delta_{u,min}$,A_Δ and $\Delta_{u,max}$
6: Use qpsolver to compute the optimal U
7: Choose U's first input u_0
=0

8.2 Problem 8.2

Question: Design input constrained MPC for the models identified in Problem 3 and Problem 4.

Same as the section on unconstrained MPC

The weight selection of the optimization problem in MPC is

$$Q_z = 0.03 \qquad S = 10$$

The prediction length of MPC is

$$N_{pre} = 50$$

The constraints of Δ_u and u are

$$-30 \leq \Delta_u \leq 30 \qquad 0 \leq u \leq 600$$

8.3 Problem 8.3

Question: Implement and discuss a compute and prediction function for this MPC.

The Matlab code of input Constrained MPC for numerical solution is showed here

```
function u_new=LMPCcompute_inputcons(R,X0,D,U,Ad,Bd,Bd_d,Czss,
    u_min,u_max,u_delta_min,u_delta_max,N,Q_cof,u_delta_cof)
% Syntax: u_new=LMPCcompute_inputcons(R,X0,D,U,Ad,Bd,Bd_d,Czss,
    u_min,u_max,u_delta_min,u_delta_max,N,Q_cof,u_delta_cof)
%          R: Reference throughout the prediction period
%          X0: Observed state x0
```

```
%           D: Observed disturbance d0
%           U: Last input U-1
%           Ad,Bd,Bd_d,Czss: Linear model state space matrix
%           u_min: Minimum of input
%           u_max: Maximum of input
%           u_delta_min: Minimum of input rate
%           u_delta_max: Maximum of input rate
%           N: prediction length
%           Q_cof: Weight matrix for tracking the reference
%           u_delta_cof: Weight matrix for input change rate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           u_new: The first input to be executed
num_x=size(Ad,1);
num_u=size(Bd,2);
num_d=size(Bd_d,2);
num_y=4;
num_z=2;
num_u_delta_cons=N*num_u;
Phi=zeros(N*num_z,num_x);
Phi_d=zeros(N*num_z,num_d);
Gam=zeros(N*num_z,N*num_u);
%Weight matrix for tracking the reference Qz
Qz=diag(Q_cof*ones(N*num_z,1)) ;
%Weight matrix for input rate S
Sz=diag(u_delta_cof*ones(num_u,1));
%Quadratic objective term Hs
Hs=zeros(N*num_u,N*num_u);
%Linear objective term Mu_delta
Mu_delta=zeros(N*num_u,num_u);
Mu_delta(1:num_u,:)=-Sz;
```

```
U0=zeros(N*num_u,1);
Iu=diag(ones(num_u,1));
Au_delta_cons=zeros(num_u_delta_cons,N*num_u);
for i=1:N
    %fill the Phi
    Phi((i-1)*num_z+1:i*num_z,:)=Czss*Ad^i;
    Phi_d((i-1)*num_z+1:i*num_z,:)=Czss*Ad^i*Bd_d;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Gam
    Gam_i=zeros(num_z,N*num_u);%[num_x * num_u*N]
    for j=1:i
        Gam_i(:,(j-1)*num_u+1:j*num_u)=Czss*(Ad^(i-j))*Bd;
    end
    Gam((i-1)*num_z+1:i*num_z,:)=Gam_i;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Hs
    if i==1
        Hs(1:num_u,1:2*num_u)=[2*Sz,-Sz];
    elseif i==N
        Hs(end-num_u+1:end,end-2*num_u+1:end)=[-Sz,Sz];
    else
        Hs((i-1)*num_u+1:(i)*num_u,(i-2)*num_u+1:(i+1)*num_u)
            =[-Sz,2*Sz,-Sz];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the U0
    U0((i-1)*num_u+1:i*num_u,:)=U;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Au_delta_cons
    if i==1
```

```
        Au_delta_cons(1:num_u,1:num_u)=Iu;
    else
        Au_delta_cons((i-1)*num_u+1:(i)*num_u,(i-2)*num_u+1:i*num_u)=[-Iu,Iu];
    end
end
Mx0=Gam'*Qz*Phi;
Md=Gam'*Qz*Phi_d;
%Linear objective term Mr
Mr=-Gam'*Qz;
%Quadratic objective term Hr
Hr=Gam'*Qz*Gam;
Hu=Hr+Hs;
g=Mx0*X0+Mr*R+Mu_delta*U+Md*D;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U_min=u_min*ones(N*num_u,1);
U_max=u_max*ones(N*num_u,1);
U_delta_min=u_delta_min*ones(num_u_delta_cons,1);
U_delta_max=u_delta_max*ones(num_u_delta_cons,1);
U_delta_min(1)=u_delta_min+U(1);
U_delta_min(2)=u_delta_min+U(2);
U_delta_max(1)=u_delta_max+U(1);
U_delta_max(2)=u_delta_max+U(2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[u_opt,~] = qpsolver(0.5*(Hu+Hu'),g,U_min,U_max,Au_delta_cons,U_delta_min,U_delta_max,U0);
u_new=u_opt(1:num_u);
end
```

.

CHAPTER 9

MPC with Input Constraints and Soft Output Constraints

9.1 Problem 9.1

Question: Implement a function for design of an MPC based on discrete-time state space models. The MPC should have input constraints and soft constraints. You should explain in the report how your Matlab function work and its theoretical background.

When MPC wants to constrain the output during optimization, it will generally be added to the value function in the form of soft constraints. Because if the output constraint becomes an inequality constraint as a hard constraint, the efficiency of solving the optimization problem will be greatly reduced, and there will even be no solution. This is obviously what we want to avoid. The soft constraint method requires the introduction of slack variables, and the solution variables of the optimization problem have changed from the original input U for the entire prediction period to the input U and slack variables for the entire prediction period.

$$\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \Longrightarrow \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \eta_{min,1} \\ \vdots \\ \eta_{min,N} \\ \eta_{max,1} \\ \vdots \\ \eta_{max,N} \end{bmatrix}$$

The optimization problem is structured as

$$\begin{aligned}
\phi = \sum_{k=1}^{N} \frac{1}{2} \left\| z_k - r_k \right\|_{Q_z}^2 + \frac{1}{2} \left\| \eta_{min,k} \right\|_{S_{\eta,min}}^2 \quad &+ \frac{1}{2} \left\| \eta_{max,k} \right\|_{S_{\eta,max}}^2 + \frac{1}{2} \sum_{k=0}^{N-1} \left\| \Delta u_k \right\|_S^2 \\
\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k + Ed_k \quad & k = 0, 1, \ldots, N-1 \\
z_k = C_z x_k \quad & k = 0, 1, \ldots, N \\
u_{\min} \leq u_k \leq u_{\max} \quad & k = 0, 1, \ldots, N-1 \\
\Delta_{u,\min} \leq \Delta u_k \leq \Delta_{u,\max} \quad & k = 0, 1, \ldots, N-1 \\
z_k - \eta_{max,k} \leq z_{\max,k} \quad & k = 1, 2, \ldots, N \\
z_k + \eta_{min,k} \geq z_{\min,k} \quad & k = 1, 2, \ldots, N \\
\eta_{max,k} \geq 0 \quad & k = 1, 2, \ldots, N \\
\eta_{min,k} \geq 0 \quad & k = 1, 2, \ldots, N
\end{aligned}$$

When constraining the output z, considering that although the heights y3, y4 of the water tanks 3 and 4 are not controlled variables, they should also be restricted in height. So z contains y3, y4, and becomes $[z1, z2, y3, y4]$. However, in the approximation and optimization of the reference value, the Q matrix can be changed to select only z1, z2

$$\underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_N \end{bmatrix}}_{Z} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^N \end{bmatrix}}_{\Phi} x_0 + \underbrace{\begin{bmatrix} CAE \\ CA^2E \\ CA^3E \\ \vdots \\ CA^NE \end{bmatrix}}_{\Phi^E} d_0 + \underbrace{\begin{bmatrix} H_1 & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ H_2 & H_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ H_3 & H_2 & H_1 & & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & 0 \\ H_N & H_{N-1} & H_{N-2} & \dots & H_1 & 0 & \dots & 0 \end{bmatrix}}_{\Gamma} \underbrace{\begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \eta_{min,1} \\ \vdots \\ \eta_{min,N} \\ \eta_{max,1} \\ \vdots \\ \eta_{max,N} \end{bmatrix}}_{U}$$

The speed of the input change during the prediction period can be expressed in the form of a matrix as

$$\phi_{\triangle u} = \frac{1}{2} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \eta_{min,1} \\ \vdots \\ \eta_{min,N} \\ \eta_{max,1} \\ \vdots \\ \eta_{max,N} \end{bmatrix}' \underbrace{\begin{bmatrix} 2S & -S & & & & 0 & \dots & 0 \\ -S & 2S & -S & & & \vdots & & \vdots \\ & -S & 2S & -S & & \vdots & & \\ & & \ddots & \ddots & -S & \vdots & & \vdots \\ & & & -S & S & 0 & \dots & 0 \\ 0 & \dots & & \dots & 0 & 0 & \dots & 0 \\ \vdots & \dots & & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & & \dots & 0 & 0 & \dots & 0 \end{bmatrix}}_{H_s} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \eta_{min,1} \\ \vdots \\ \eta_{min,N} \\ \eta_{max,1} \\ \vdots \\ \eta_{max,N} \end{bmatrix}$$

$$+ \underbrace{\left(- \begin{bmatrix} S \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} u_{-1} \right)'}_{M_{u_{-1}}} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \eta_{min,1} \\ \vdots \\ \eta_{min,N} \\ \eta_{max,1} \\ \vdots \\ \eta_{max,N} \end{bmatrix} + \frac{1}{2} u_{-1} S u_{-1}$$

The inequality constraints of input can be simply constructed as

$$\underbrace{\begin{bmatrix} u_{min} \\ \vdots \\ u_{min} \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{U_{min}} \leq \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \eta_{min,1} \\ \vdots \\ \eta_{min,N} \\ \eta_{max,1} \\ \vdots \\ \eta_{max,N} \end{bmatrix} \leq \underbrace{\begin{bmatrix} u_{max} \\ \vdots \\ u_{max} \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{U_{max}}$$

The inequality constraint on the rate of change of input can be expressed as

$$\underbrace{\begin{bmatrix} \Delta_{u,min} + u_{-1} \\ \Delta_{u,min} \\ \Delta_{u,min} \\ \vdots \\ \Delta_{u,min} \end{bmatrix}}_{\Delta_{u,min}} \leq \underbrace{\begin{bmatrix} I & & & & 0 & \dots & 0 \\ -I & I & & & \vdots & & \vdots \\ & -I & I & & & & \\ & & \ddots & \ddots & & & \\ & & & -I \quad I & 0 & \dots & 0 \end{bmatrix}}_{A_{\Delta u}} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \eta_{min,1} \\ \vdots \\ \eta_{min,N} \\ \eta_{max,1} \\ \vdots \\ \eta_{max,N} \end{bmatrix} \leq \underbrace{\begin{bmatrix} \Delta_{u,max+u_{-1}} \\ \Delta_{u,max} \\ \Delta_{u,max} \\ \vdots \\ \Delta_{u,max} \end{bmatrix}}_{\Delta_{u,max}}$$

The output inequality constraint can be expressed as

$$\begin{aligned} Z_{min} - \Phi x_0 - \Phi_d D \leq \Gamma U + \eta_{min} &\leq +\infty \\ -\infty \leq \Gamma U - \eta_{max} &\leq Z_{max} - \Phi x_0 - \Phi_d D \end{aligned}$$

The weight matrix Q for tracking the reference value is

$$Q_z = \begin{bmatrix} Q_{z1} & & & \\ & Q_{z2} & & \\ & & \ddots & \\ & & & Q_{zN} \end{bmatrix} \qquad Q_{zN} = \begin{bmatrix} Q_{zN,1} & 0 & 0 & 0 \\ 0 & Q_{zN,2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Algorithm 5 MPC with Input Constraints and Soft Output Constraints

1: Given the last input u_{-1},current state x and disturbance d, and the reference R_z throughout the prediction period
2: Constructing matrix Q_z, Γ and Compute $H_z = \Gamma' Q_z \Gamma$
3: Constructing matrix $\Phi, \Phi^d, M_{u_{-1}}$ and Compute $M_R = -\Gamma' Q_z, M_{x_0} = \Gamma' Q_z \Phi, M_d = \Gamma' Q_z \Phi^d$
4: Constructing matrix H_s,$H_{\eta,min}$, $H_{\eta,max}$and Compute $H = H_z + H_s + H_{\eta,min} + H_{\eta,max}, g = M_R R + M_{x_0} x_0 + M_d D + M_{u_{-1}} u_{-1}$
5: Constructing matrix $\Delta_{u,min}$, A_Δ and $\Delta_{u,max}$
6: Constructing matrix $A_{\eta,min}$,$A_{\eta,max}$, U_{min},U_{max}, Z_{min} and Z_{max}
7: Use qpsolver to compute the optimal U
8: Choose U's first input u_0
=0

9.2 Problem 9.2

Question: Design the input- and soft-constrained output MPC for the models identified in Problem 3 and Problem 4.

The biggest difference from the previous two MPC algorithms is that soft-constrained output MPC introduces soft constraints on the output, which gives the MPC algorithm more possibilities for individual situations.

For example, when the output target value is close to the dangerous value, the unconstrained MPC or only constraining the input cannot guarantee the safety of control, but soft-constrained output MPC with can give the upper limit of the output a large penalty weight as much as possible. The output is below the dangerous value. For another example, in some models, the time and magnitude of the disturbance are unknown, but the positive and negative are known, which means that influence trend of the disturbance on the output is predictable, then the soft constraint on the output can be restricted to the positive and negative of the disturbance.

In the design of the soft-constrained output MPC algorithm for the output of the water tank system, if the output target value is greater than the initial value, it will be determined whether the lower limit of the output needs to be changed before each MPC calculation. Because if a lower limit value greater than the initial output value is used at the beginning, overshoot will inevitably cause danger, which is obviously not what we want.

This situation is shown below. In the case of linear system using soft-constrained output MPC with disturbances are stochastic variables but do contains step changes

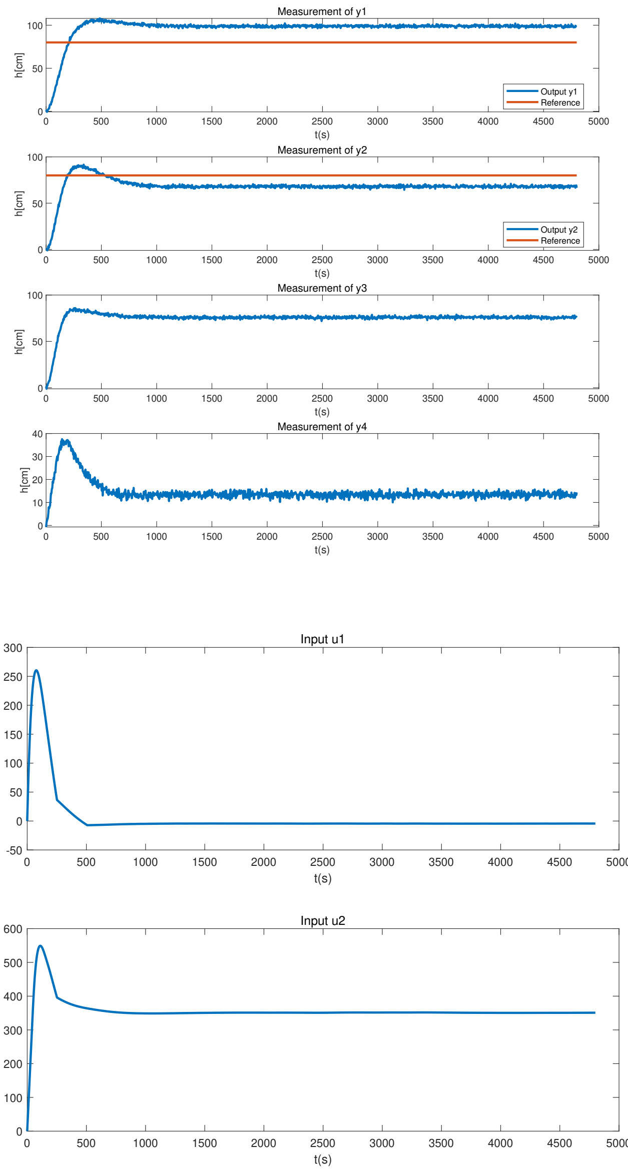

Figure 9.1: overshoot Y and U of Linear system using soft-constrained output MPC without step changing.

It can be seen from the signal of u that at the beginning, because the lower limit of the output is much larger than the initial value, the sharp change of u causes the output to overshoot. The adjusted situation shows better results in section 10.

The following shows the framework logic when the output target value is greater than the initial value

Algorithm 6 Framework of MPC with Input Constraints and Soft Output Constraints

Given input constraint U_{min},U_{max}, input rate constraint $\Delta_{u,min}$ and $\Delta_{u,max}$, the initial state x_0, initial state $y_0 = [0, 0, 0, 0]$, reference $R_z = [50, 50]$, output constraint $Y_{min1} = [0, 0, 0, 0]$, $Y_{min2} = [40, 40, 0, 0]$ and $Y_{max} = [60, 60, 60, 60]$
while 1 **do**
  Kalman observer compute the observed $\hat{x}$ and $\hat{d}$
  if $(y_k > Y_{min2} - 10)$ **then**
    $Y_{min} = Y_{min2}$
  else
    $Y_{min} = Y_{min1}$
  end if
  MPC compute the u_k
  State update x_k
  sensor and output update y_k, z_k
end while=0

The weight selection of the optimization problem in MPC is

$$Q_z = 0.03 \qquad S = 10 \qquad Q_{eta,min} = 10 \qquad Q_{eta,max} = 10$$

The constraints of Δ_u and u are

$$-30 \leq \Delta_u \leq 30 \qquad 0 \leq u \leq 600$$

9.3 Problem 9.3

Question: Implement and discuss a compute and prediction function for this MPC.

The Matlab code of MPC with input constraints and soft output constraints is showed here

```
function u_new=LMPCcompute_inputcons_outputSoftCons(R,X0,D,U,Ad
    ,Bd,Bd_d,Css,u_min,u_max,u_delta_min,u_delta_max,z_min,z_max
    ,N,Q_cof,u_delta_cof,eta_cof1,eta_cof2)
% Syntax: u_new=LMPCcompute_inputcons_outputSoftCons(R,X0,D,U,
   Ad,Bd,Bd_d,Css,u_min,u_max,u_delta_min,u_delta_max,z_min,
   z_max,N,Q_cof,u_delta_cof,eta_cof1,eta_cof2)
%          R: Reference throughout the prediction period
%          X0: Observed state x0
%          D: Observed disturbance d0
%          U: Last input U-1
%          Ad,Bd,Bd_d,Czss: Linear model state space matrix
%          u_min: Minimum of input
%          u_max: Maximum of input
%          u_delta_min: Minimum of input rate
%          u_delta_max: Maximum of input rate
%          z_min: Minimum of output
%          z_max: Maximum of input
%          N: prediction length
%          Q_cof: Weight matrix for tracking the reference
%          u_delta_cof: Weight matrix for input change rate
%          eta_cof1: Weight matrix for Minimum of output
%          eta_cof2: Weight matrix for Maximum of output
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%          u_new: The first input to be executed
num_x=size(Ad,1);
num_u=size(Bd,2);
num_d=size(Bd_d,2);
num_y=4;
num_z=2;
num_eta=num_y;
num_u_delta_cons=N*num_u;
num_z_cons=N*num_y;
Phi=zeros(N*num_y,num_x);
Phi_d=zeros(N*num_y,num_d);
Gam=zeros(N*num_y,N*num_u+2*N*num_eta);
%Weight matrix for tracking the reference Qz
Qz=diag(zeros(N*num_y,1));
%Weight matrix for input rate S
Sz=diag(u_delta_cof*ones(num_u,1));
%Quadratic objective term Hs
Hs=zeros(N*num_u+2*N*num_eta,N*num_u+2*N*num_eta);
%Linear objective term Mu_delta
Mu_delta=zeros(N*num_u+2*N*num_eta,num_u);
Mu_delta(1:num_u,:)=-Sz;
%Quadratic objective term H_eta
H_eta1=diag([zeros(N*num_u,1);eta_cof1*ones(N*num_eta,1);zeros(N*num_eta,1)]);
H_eta2=diag([zeros(N*num_u,1);zeros(N*num_eta,1);eta_cof2*ones(N*num_eta,1)]);
U0=zeros(N*num_u+2*N*num_eta,1);
Iu=diag(ones(num_u,1));
Au_delta_cons=zeros(num_u_delta_cons,N*num_u+2*N*num_eta);
for i=1:N
```

```
    %fill the Phi
    Phi((i-1)*num_y+1:i*num_y,:)=Css*Ad^i;
    Phi_d((i-1)*num_y+1:i*num_y,:)=Css*Ad^i*Bd_d;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Gam
    Gam_i=zeros(num_y,N*num_u+2*N*num_eta);
    for j=1:i
        Gam_i(:,(j-1)*num_u+1:j*num_u)=Css*(Ad^(i-j))*Bd;
    end
    Gam((i-1)*num_y+1:i*num_y,:)=Gam_i;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Qz
    qz=diag([Q_cof*ones(num_z,1);zeros(num_y-num_z,1)]);
    Qz((i-1)*num_y+1:i*num_y,(i-1)*num_y+1:i*num_y)=qz;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Hs
    if i==1
        Hs(1:num_u,1:2*num_u)=[2*Sz,-Sz];
    elseif i==N
        Hs((i-1)*num_u+1:(i)*num_u,(i-2)*num_u+1:i*num_u)=[-Sz,
            Sz];
    else
        Hs((i-1)*num_u+1:(i)*num_u,(i-2)*num_u+1:(i+1)*num_u)
            =[-Sz,2*Sz,-Sz];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the U0
    U0((i-1)*num_u+1:i*num_u,:)=U;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %fill the Au_delta_cons
```

```
    if i==1
        Au_delta_cons(1:num_u,1:num_u)=Iu;
    else
        Au_delta_cons((i-1)*num_u+1:(i)*num_u,(i-2)*num_u+1:i*num_u)=[-Iu,Iu];
    end
end
Mx0=Gam'*Qz*Phi;
Md=Gam'*Qz*Phi_d;
%Linear objective term Mr
Mr=-Gam'*Qz;
%Quadratic objective term Hr
Hr=Gam'*Qz*Gam;
Hu=Hr+Hs+H_eta1+H_eta2;
g=Mx0*X0+Mr*R+Mu_delta*U+Md*D;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Az_cons_min=Gam;
Az_cons_min(:,N*num_u+1:N*num_u+N*num_eta)=diag(1*ones(N*num_eta,1));
Az_cons_min(:,N*num_u+N*num_eta+1:end)=diag(zeros(N*num_eta,1));
Az_cons_max=Gam;
Az_cons_max(:,N*num_u+1:N*num_u+N*num_eta)=diag(zeros(N*num_eta,1));
Az_cons_max(:,N*num_u+N*num_eta+1:end)=diag(-1*ones(N*num_eta,1));
Z_min=repmat(z_min,N,1)-Phi*X0-Phi_d*D;
Z_max_fake=5000*ones(num_z_cons,1);
Z_max=repmat(z_max,N,1)-Phi*X0-Phi_d*D;
Z_min_fake=zeros(num_z_cons,1);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
U_min=[u_min*ones(N*num_u,1);zeros(2*N*num_eta,1)];
U_max=[u_max*ones(N*num_u,1);5000*ones(2*N*num_eta,1)];
U_delta_min=u_delta_min*ones(num_u_delta_cons,1);
U_delta_max=u_delta_max*ones(num_u_delta_cons,1);
U_delta_min(1)=u_delta_min+U(1);
U_delta_min(2)=u_delta_min+U(2);
U_delta_max(1)=u_delta_max+U(1);
U_delta_max(2)=u_delta_max+U(2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_ieq=[Au_delta_cons;Az_cons_min;Az_cons_max];
b_ieq_min=[U_delta_min;Z_min;Z_min_fake];
b_ieq_max=[U_delta_max;Z_max_fake;Z_max];
[u_opt,~] = qpsolver(0.5*(Hu+Hu'),g,U_min,U_max,A_ieq,b_ieq_min
    ,b_ieq_max,U0);
u_new=u_opt(1:num_u);
end
```

CHAPTER 10

Closed-Loop Simulations

Question: Do closed-simulations of your MPCs. You should do the simulation for both linear and nonlinear models. Discuss the results. Present movies and plots that illustrate the performance of your MPCs.

The three MPC algorithms designed unconstrained MPC, input constrained MPC and MPC with input constraints and soft output constraints are tested in the following situations

- Linear system with disturbances are stochastic variables but do not contains step changes
- Linear system with disturbances are stochastic variables but do contains step changes
- Nonlinear system with disturbances are stochastic variables but do not contains step changes
- Nonlinear system with disturbances are stochastic variables but do contains step changes

Compared with the unconstrained mpc, input constrained MPC only limits the input, so the performance of the control is the same as that of the unconstrained mpc, and only the results of the iteration time are shown.

Since the unknown disturbances are stochastic variables but do not contains step changes, the reference of output and signal of disturbance of test for linear system is constructed as

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 80 \\ 80 \end{bmatrix} \qquad \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 50 \\ 50 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

The signal of disturbance of test for nonlinear system is constructed as

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 200 \\ 200 \end{bmatrix} \qquad \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 300 \\ 300 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

In the case of linear system using all MPC with disturbances are stochastic variables but do not contains step changes

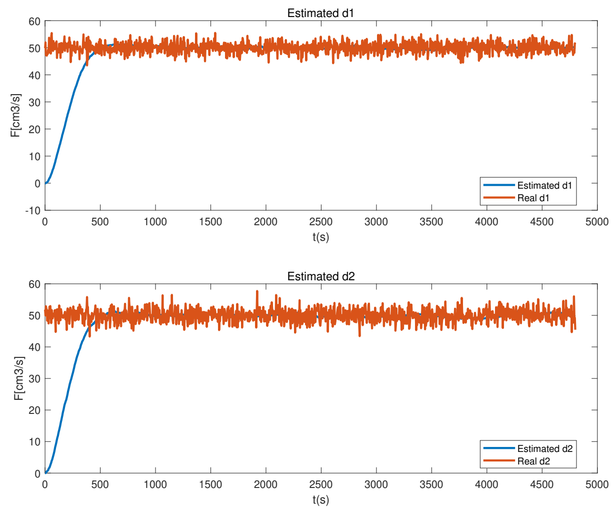

Figure 10.1: D of Linear system using Unconstrained MPC without step changing.

In the case of nonlinear system using all MPC with disturbances are stochastic variables but do not contains step changes

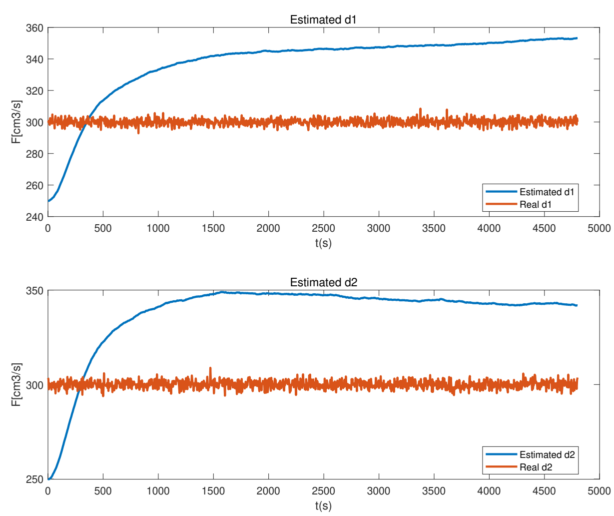

Figure 10.2: D of nonlinear system using Unconstrained MPC without step changing.

In the case of linear system using unconstrained MPC with disturbances are stochastic variables but do not contains step changes

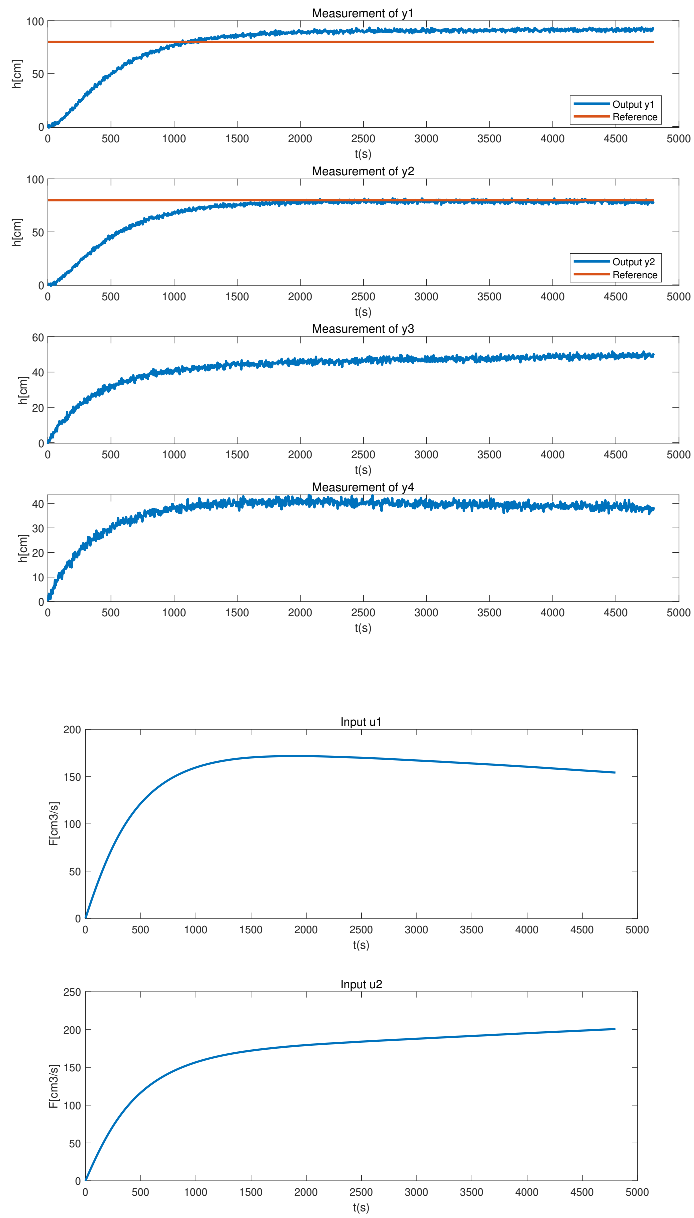

Figure 10.3: Y and U of Linear system using Unconstrained MPC without step changing.

In the case of linear system using MPC with input constraints and soft output constraints with disturbances are stochastic variables but do not contains step changes

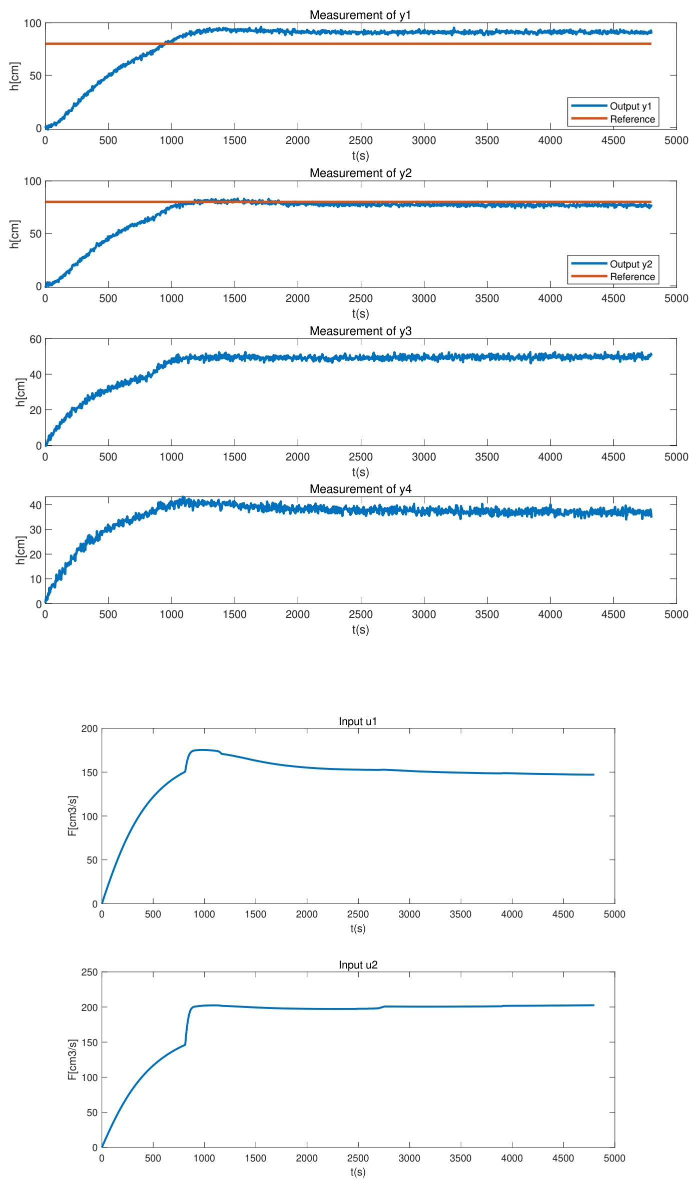

Figure 10.4: Y and U of Linear system using input constrained MPC without step changing.

In the case of nonlinear system using unconstrained MPC with disturbances are stochastic variables but do not contains step changes

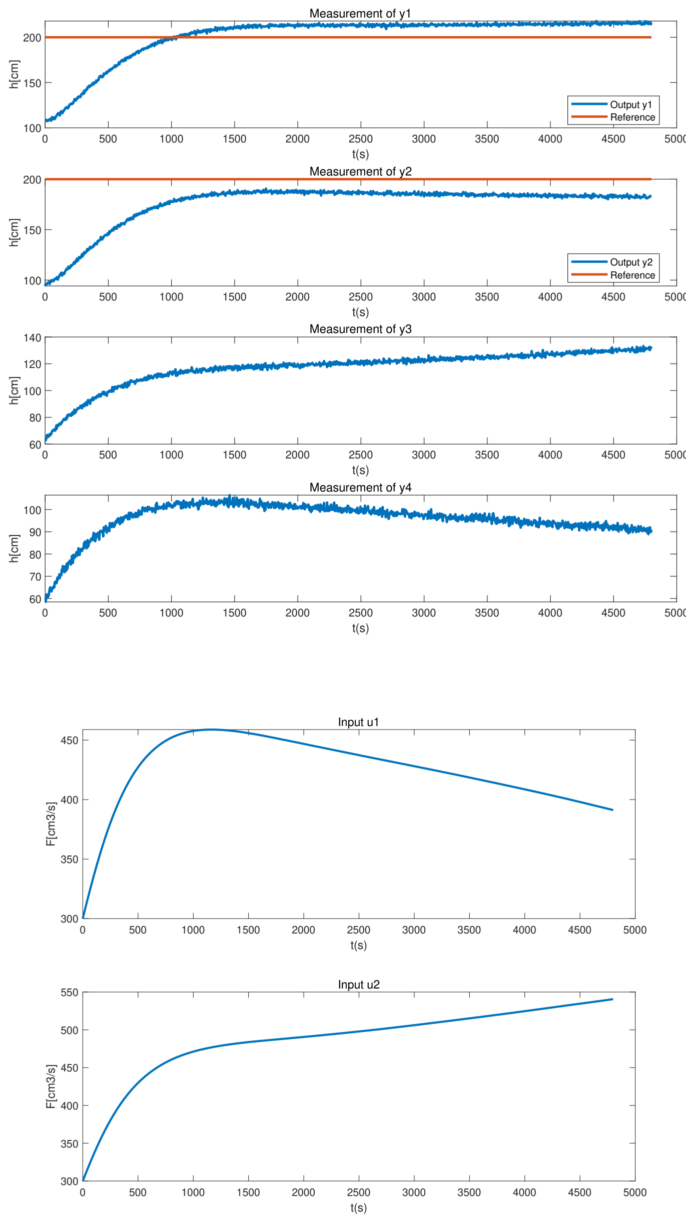

Figure 10.5: Y and U of nonlinear system using unconstrained MPC without step changing.

In the case of nonlinear system using MPC with input constraints and soft output constraints with disturbances are stochastic variables but do not contains step changes

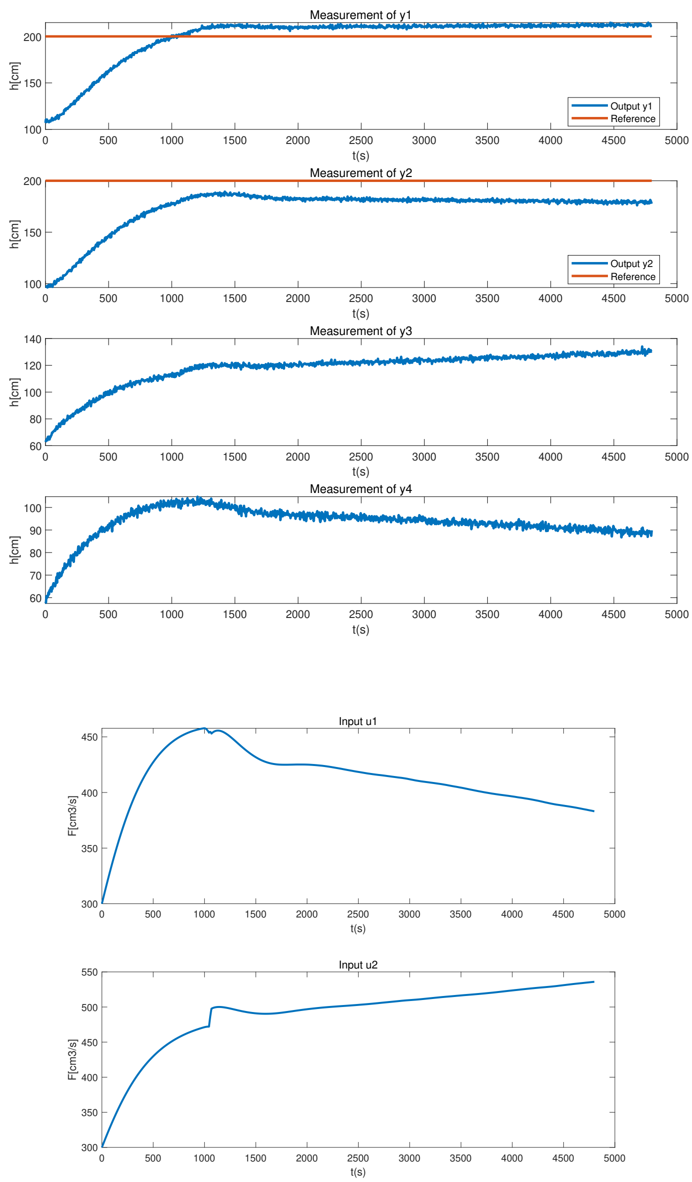

Figure 10.6: Y and U of nonlinear system using MPC with input constraints and soft output constraints without step changing.

The iteration times of the three MPC algorithms are also compared in the linear system and nonlinear system tests.

When the linear system is tested

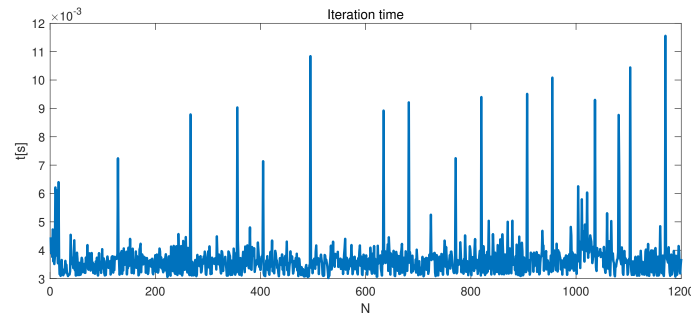

Figure 10.7: iteration times of linear system using unconstrained MPC without step changing.

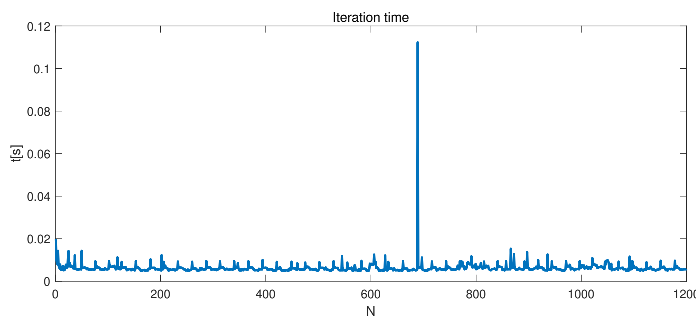

Figure 10.8: iteration times of linear system using input constrained MPC without step changing.

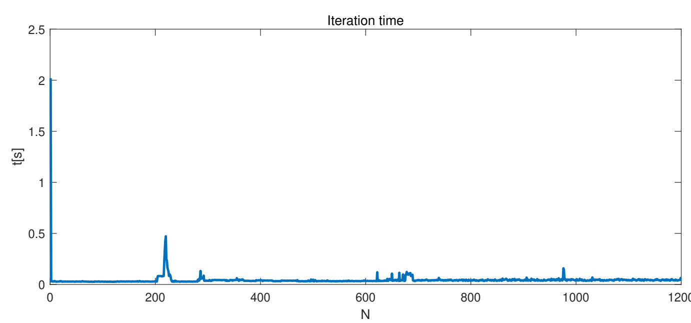

Figure 10.9: iteration times of linear system using MPC with input constraints and soft output constraints without step changing.

When the nonlinear system is tested

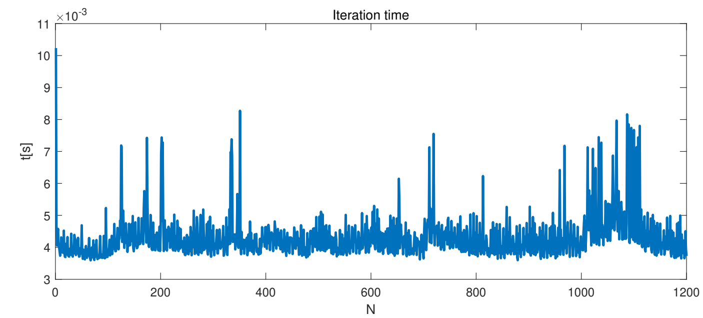

Figure 10.10: iteration times of nonlinear system using unconstrained MPC without step changing.

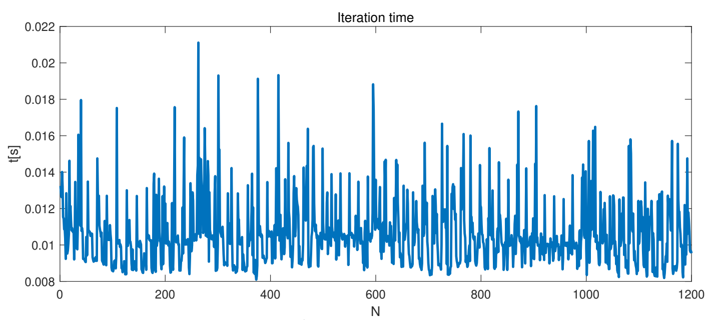

Figure 10.11: iteration times of nonlinear system using input constrained MPC without step changing.

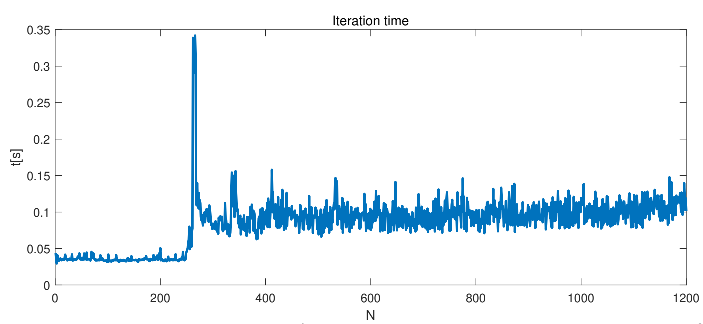

Figure 10.12: iteration times of nonlinear system using MPC with input constraints and soft output constraints without step changing.

Since the unknown disturbances are stochastic variables but DO CONTAIN step changes, the signal of disturbance of test for linear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 50 \\ 50 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right) \Longrightarrow N_{iid}\left(\begin{bmatrix} 100 \\ 100 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

the signal of disturbance of test for nonlinear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 300 \\ 300 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right) \Longrightarrow N_{iid}\left(\begin{bmatrix} 350 \\ 350 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

In the case of linear system using all MPC with disturbances are stochastic variables but do not contains step changes

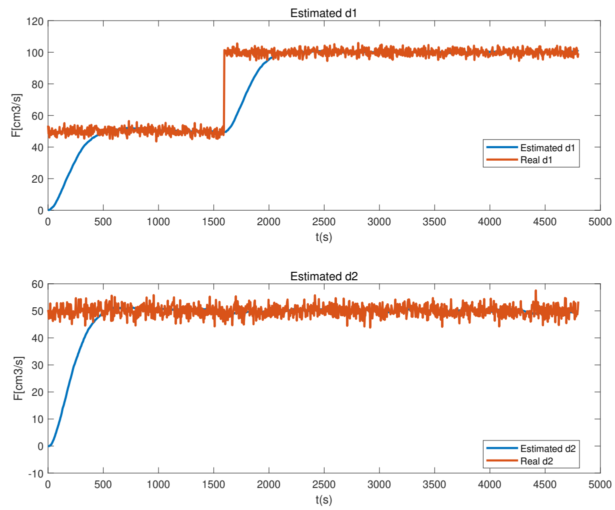

Figure 10.13: D of linear system using Unconstrained MPC with step changing.

In the case of nonlinear system using all MPC with disturbances are stochastic variables but do not contains step changes

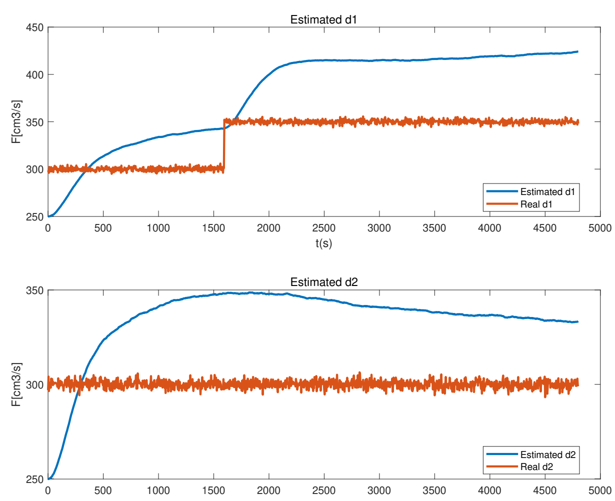

Figure 10.14: D of nonlinear system using Unconstrained MPC with step changing.

In the case of linear system using unconstrained MPC with disturbances are stochastic variables but do contains step changes

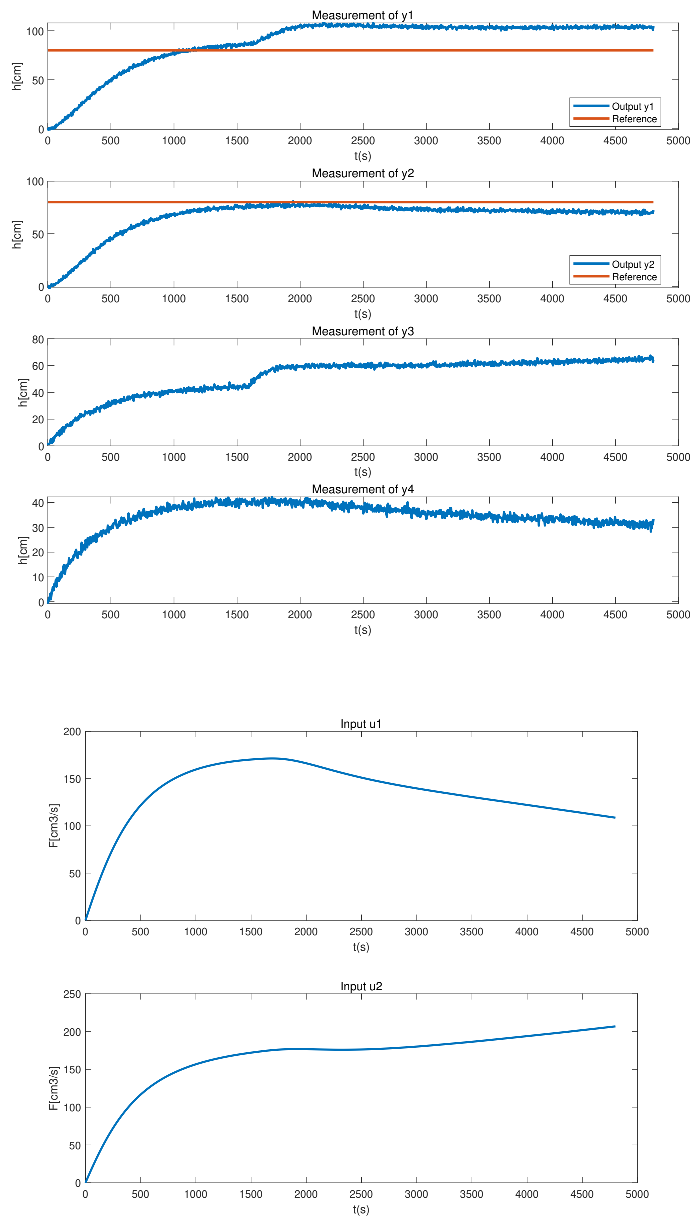

Figure 10.15: Y and U of Linear system using Unconstrained MPC with step changing.

In the case of linear system using MPC with input constraints and soft output constraints with disturbances are stochastic variables but do contains step changes

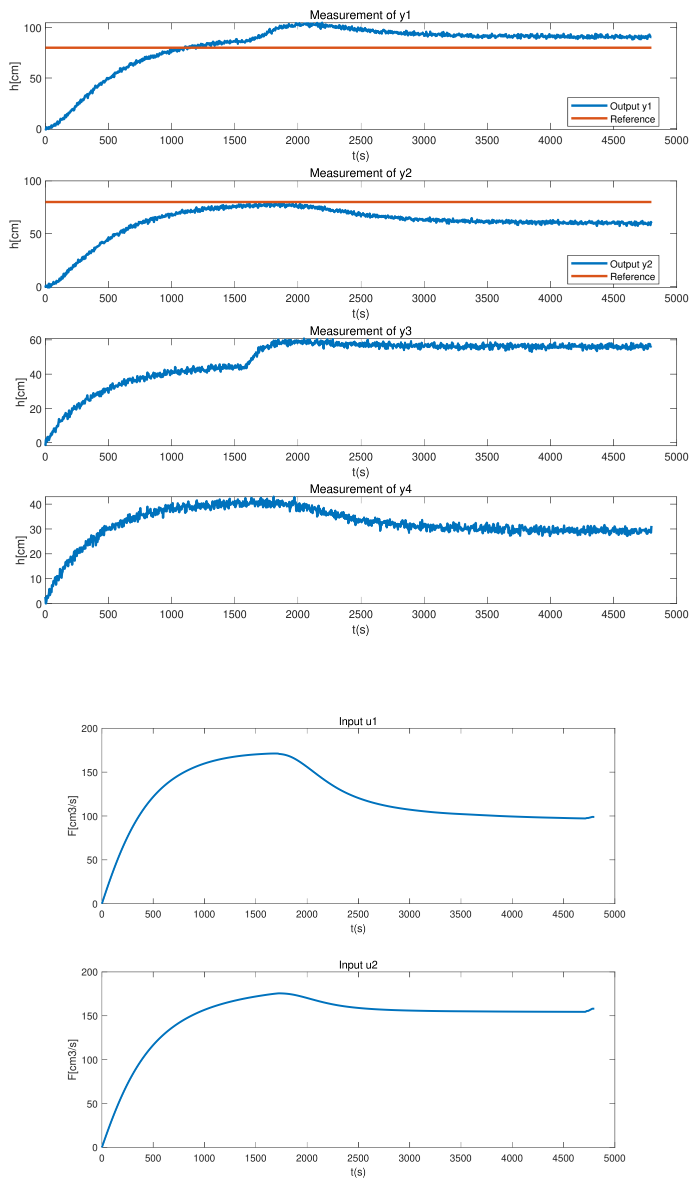

Figure 10.16: Y and U of Linear system using input constrained MPC with step changing.

In the case of nonlinear system using unconstrained MPC with disturbances are stochastic variables but do contains step changes

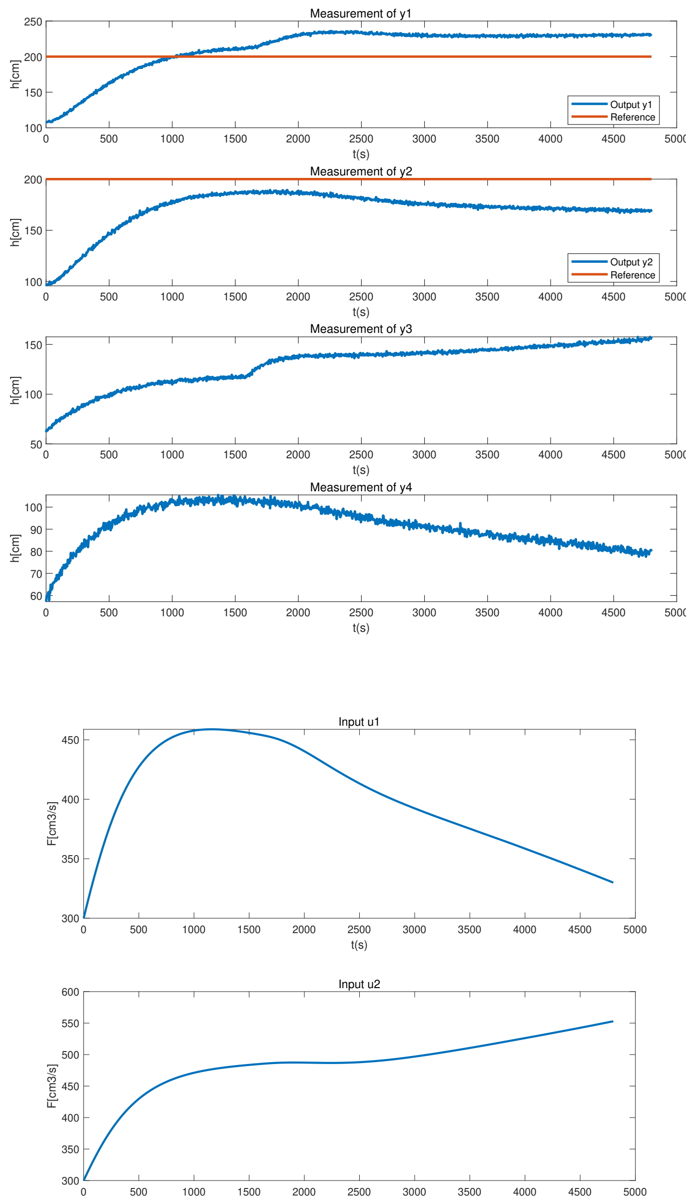

Figure 10.17: Y and U of nonlinear system using unconstrained MPC with step changing.

In the case of nonlinear system using MPC with input constraints and soft output constraints with disturbances are stochastic variables but do contains step changes

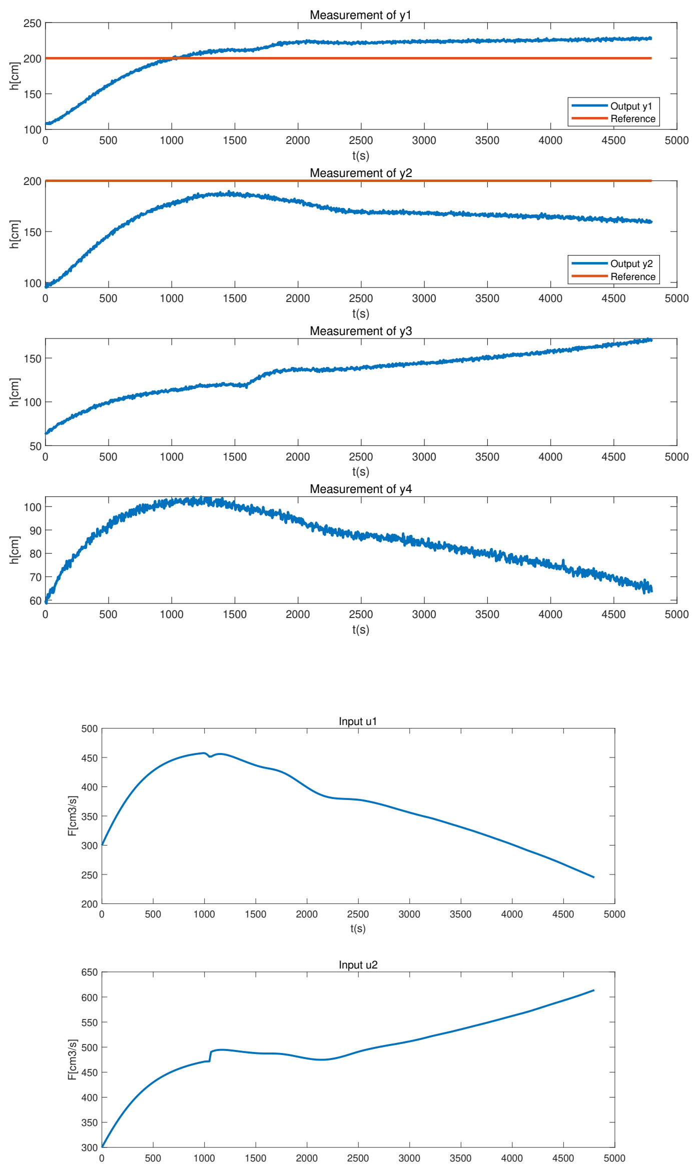

Figure 10.18: Y and U of nonlinear system using MPC with input constraints and soft output constraints with step changing.

The iteration times of the three MPC algorithms are also compared in the linear system and nonlinear system tests.

When the linear system is tested

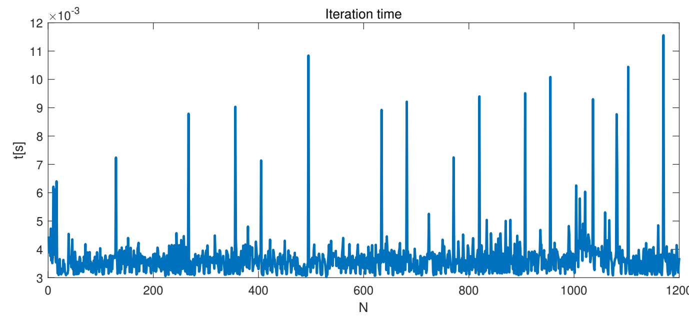

Figure 10.19: iteration times of linear system using unconstrained MPC with step changing.

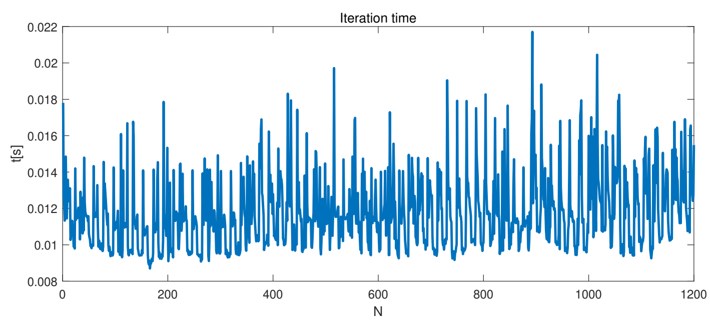

Figure 10.20: iteration times of linear system using input constrained MPC with step changing.

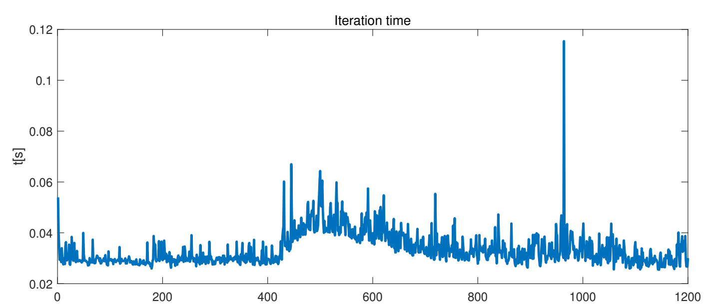

Figure 10.21: iteration times of linear system using MPC with input constraints and soft output constraints with step changing.

When the nonlinear system is tested

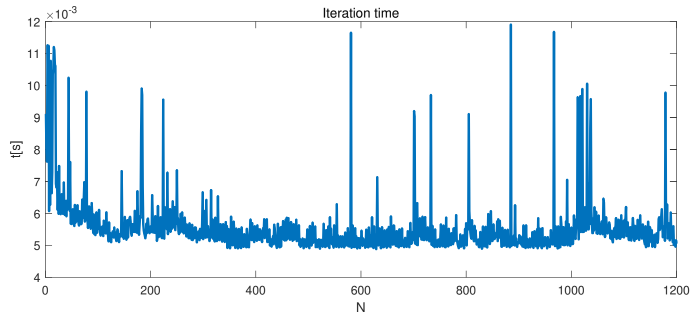

Figure 10.22: iteration times of nonlinear system using unconstrained MPC with step changing.

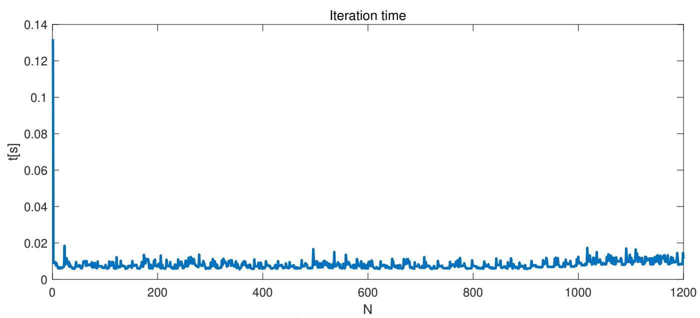

Figure 10.23: iteration times of nonlinear system using input constrained MPC with step changing.

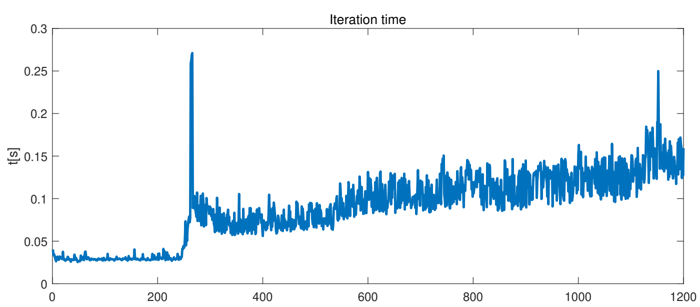

Figure 10.24: iteration times of nonlinear system using MPC with input constraints and soft output constraints with step changing.

In the case that the input calculated by the MPC algorithm does not exceed a set limit, the control performance of the three MPC algorithms is similar.

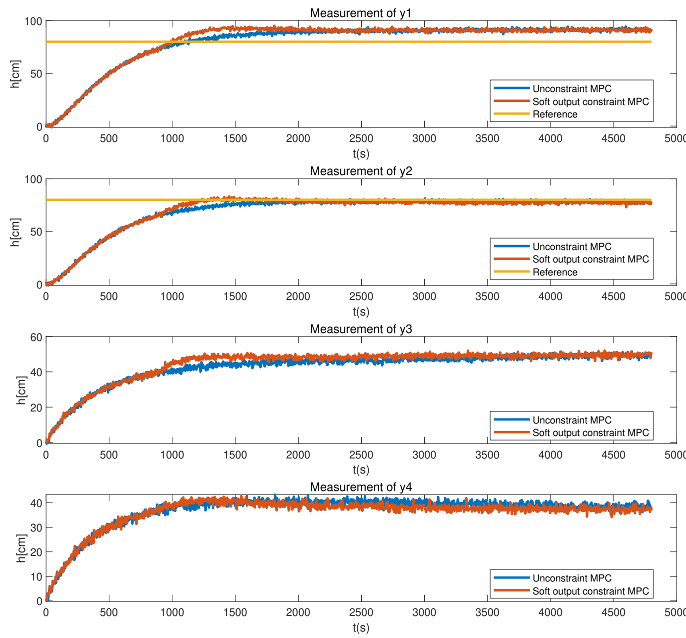

Figure 10.25: Comparison of linear system using unconstrained MPC and MPC with input constraints and soft output constraints with step changing.

However, in the case of linear system using unconstrained MPC with disturbancesare stochastic variables but do not contains step changes. From figure 10.3, It can be found that in order to achieve the control effect, , the input u calculated by unconstrained MPC exceeds $150[cm_3/s]$ in the process. But in practice, the input usually has a limit. If the input is required to not exceed 150, then the unconstrained MPC algorithm cannot be used. However, MPC with input constraints can still continue to work safely while maintaining control performance.

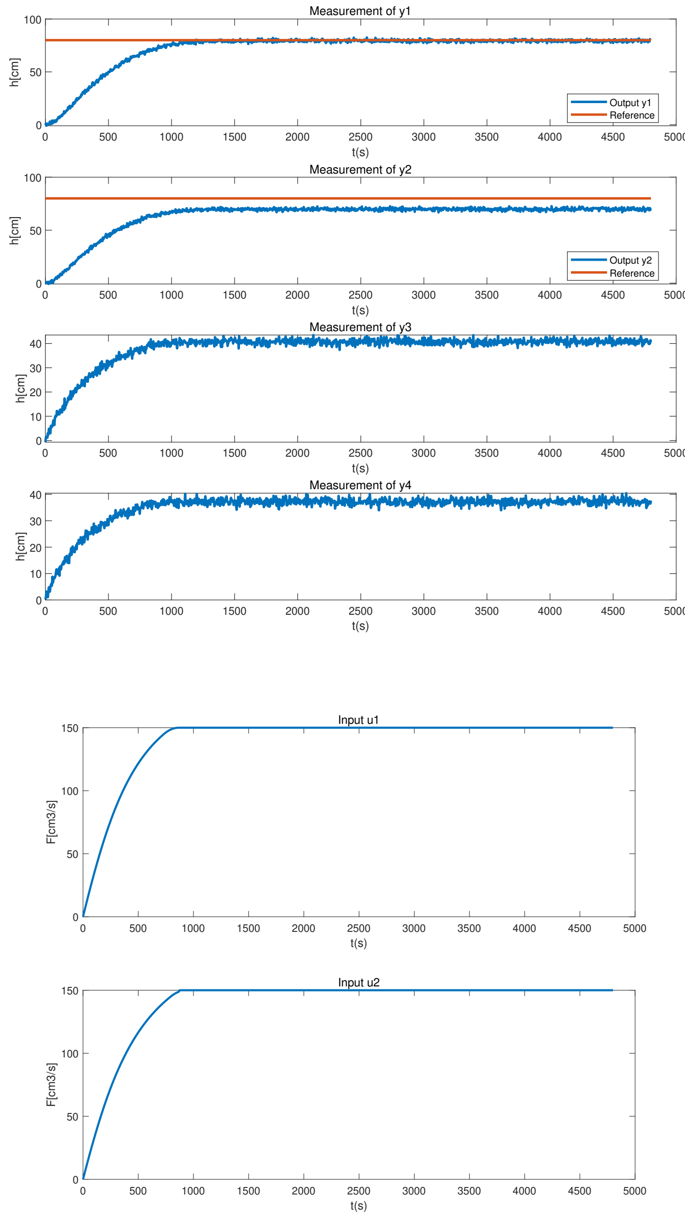

Figure 10.26: Y and U of linear system using input constrained MPC without step changing.

It can be found from the test results that the performance of MPC in a linear system is better than that in a nonlinear system. It is speculated that this is related to the error of the established system model and the error of the Kalman observer.

During the control process, the output and input signals are well obeyed the constraints designed in the MPC, and it can be seen that the input u has been changing very smoothly without sudden changes and fluctuations. The output y did not overshoot, and finally stabilized around the target value. However, it can be seen that there is a steady-state error in the output y, which is speculated to be related to the error of the established system model and the error of the observer.

It can be seen from the iteration time that as the constraints in the MPC increase, the calculation time will become longer, but they are all within the allowable range.

CHAPTER 11

PID control

11.1 Problem 13.1

Question: Discuss the pairing of inputs and outputs for the four tank system.

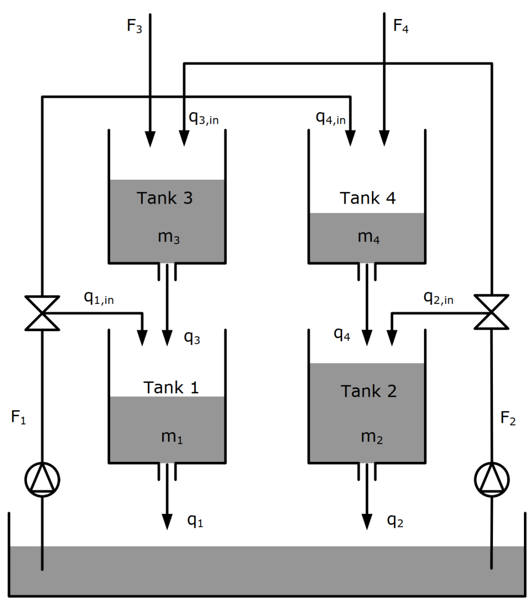

Figure 11.1: The Modified Four Tank System.

Figure 1.1 is a modified model of four water tanks. It can be seen from the figure that input u1 directly acts on y1, and directly acts on y4 and indirectly acts on y2. At the same time, input u2 directly acts on y2, and directly acts on y3 and indirectly acts on y1. It can be seen that the relationship between manipulated variables u1, u2 and controlled variables z1, z2 is coupled.

In order to understand the nature and degree of coupling in the system, the RGA(Relative Gain Array) needs to be calculated. The steady-state gain of the transfer function is

$$K = \begin{bmatrix} 0.1721 & 0.1992 \\ 0.2295 & 0.1449 \end{bmatrix} \qquad F = \frac{K_{12}K_{21}}{K_{11}K_{22}} = 1.8333$$

The RGA(Relative Gain Array) is

$$T = \begin{bmatrix} \frac{1}{1-F} & -\frac{F}{1-F} \\ -\frac{F}{1-F} & \frac{1}{1-F} \end{bmatrix} = \begin{bmatrix} -1.2 & 2.2 \\ 2.2 & -1.2 \end{bmatrix}$$

Because T_{11} and T_{22} are less than 0, u1 to z1 and u2 to z2 are not chosen to pair. In contrast, u1 to z2 and u2 to z1 are chosen to pair.

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{11} \end{bmatrix} = \begin{bmatrix} 0 & G_{12} \\ G_{21} & 0 \end{bmatrix}$$

Decoupled matrix is

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{11} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}^{-1} \begin{bmatrix} 0 & G_{12} \\ G_{21} & 0 \end{bmatrix}$$

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{11} \end{bmatrix} = \frac{1}{K_{11}K_{22} - K_{12}K_{21}} \begin{bmatrix} -K_{12}K_{21} & K_{22}K_{12} \\ K_{11}K_{21} & -K_{21}K_{12} \end{bmatrix} = \begin{bmatrix} 2.2 & -1.3887 \\ -1.9010 & 2.2 \end{bmatrix}$$

Decoupled steady-state gain of the transfer function is

$$K = \begin{bmatrix} 0.0597 & 0.2359 \\ 0.1768 & -0.0596 \end{bmatrix} \qquad F = \frac{K_{12}K_{21}}{K_{11}K_{22}} = -11.7107$$

Decoupled RGA(Relative Gain Array) is

$$T = \begin{bmatrix} 0.0787 & 0.9213 \\ 0.9213 & 0.0787 \end{bmatrix}$$

It can be seen that through the decoupling matrix, the coupling of the system is reduced to a controllable level.

11.2 Problem 13.2

Question: Implement a P-, a PI- and a PID-controller for the four tank system.

Algorithm 7 P-controller

Given the reference output $\hat{Y}$, current measured output Y ,last measured output Y_{old} and decoupling matrix M_{dec}
Compute the error $e = \hat{Y} - Y$
Compute the decoupled error $e = M_{dec}e$
Compute the Proportional term $P_k = K_P e$
Compute the input $u_k = P_k$
if $(u_{min} > u_k || u_k > u_{max})$ **then**
    $u_k = max\{u_{min}, min\{u_{max}, u_k\}\}$
end if
=0

Proportional coefficient	$K_P = 4.5811$

Algorithm 8 PI-controller

Given the reference output $\hat{Y}$, current measured output Y, last measured output Y_{old}, updated integral term I_k and decoupling matrix M_{dec}
Compute the error $e = \hat{Y} - Y$
Compute the decoupled error $e = M_{dec}e$
Compute the Proportional term $P_k = K_P e$
Compute the input $u_k = P_k + I_k$
if $(u_{min} < u_k < u_{max})$ **then**
    Compute the integral term $I_{k+1} = I_k + K_I e \Delta t$
else
    $u_k = max\{u_{min}, min\{u_{max}, u_k\}\}$
end if
=0

Proportional coefficient	$K_P = 1.9144$
Integral coefficient	$K_I = 0.010683$

Algorithm 9 PID-controller

Given the reference output $\hat{Y}$, current measured output Y, last measured output Y_{old}, updated integral term I_k and decoupling matrix M_{dec}
Compute the error $e = \hat{Y} - Y$
Compute the decoupled error $e = M_{dec}e$
Compute the Proportional term $P_k = K_P e$
Compute the Derivative term$D_k = -K_D \frac{Y - Y_{old}}{\Delta t}$
Compute the input $u_k = P_k + D_k + I_k$
if $(u_{min} < u_k < u_{max})$ **then**
  Compute the integral term $I_{k+1} = I_k + K_I e \Delta t$
else
  $u_k = max\{u_{min}, min\{u_{max}, u_k\}\}$
end if
=0

Proportional coefficient	$K_P = 2.2715$
Integral coefficient	$K_I = 0.012046$
Derivative coefficient	$K_I = 4.4948$

The Matlab code of PID-controller is showed here

```
function [u,I] = MIMOPID(ubar,ybar,y,yold,I,KP,KI,KD,dt,umin,umax,decoup)
% Syntax: [u,I] = MIMOPID(ubar,ybar,y,yold,I,KP,KI,KD,dt,umin,umax,decop)
%         ubar: Reference input
%         ybar: Reference output
%         y: measured output
%         yold: Last measured output
%         I: Integral term
%         KP: Proportional coefficient
```

```
%           KI: Integral coefficient
%           KD: Derivative coefficient
%           dt: Unit time
%           u_min: Minimum of input
%           u_max: Maximum of input
%           decop: Decopling matrix
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           u: The input to be executed
%           I: Updated Integral term
e = ybar-y;
e_decop=decoup*e;
P = KP*e_decop;
D = -KD*(y-yold)/dt;
u = ubar + P + I + D;
flag=1;
for n=1:2
if (u(n) >= umax)
u(n) = umax;
flag=0;
elseif (u(n) <= umin)
u(n) = umin;
flag=0;
end
end
if(flag==1)
I = I + KI*e_decop*dt;
end
```

.

11.3 Problem 13.3

Question: Test the controllers by closed-loop simulation.

The designed P- PI- and PID- controller were tested under 4 conditions

- Linear system with disturbances are stochastic variables but do not contains step changes
- Linear system with disturbances are stochastic variables but do contains step changes
- Nonlinear system with disturbances are stochastic variables but do not contains step changes
- Nonlinear system with disturbances are stochastic variables but do contains step changes

Since the unknown disturbances are stochastic variables but do not contains step changes, the reference of output and signal of disturbance of test for linear system is constructed as

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 80 \\ 80 \end{bmatrix} \qquad \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 50 \\ 50 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \right)$$

The signal of disturbance of test for nonlinear system is constructed as

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 200 \\ 200 \end{bmatrix} \qquad \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 300 \\ 300 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \right)$$

In the case of Linear system using P- controller with disturbances are stochastic variables but do not contains step changes

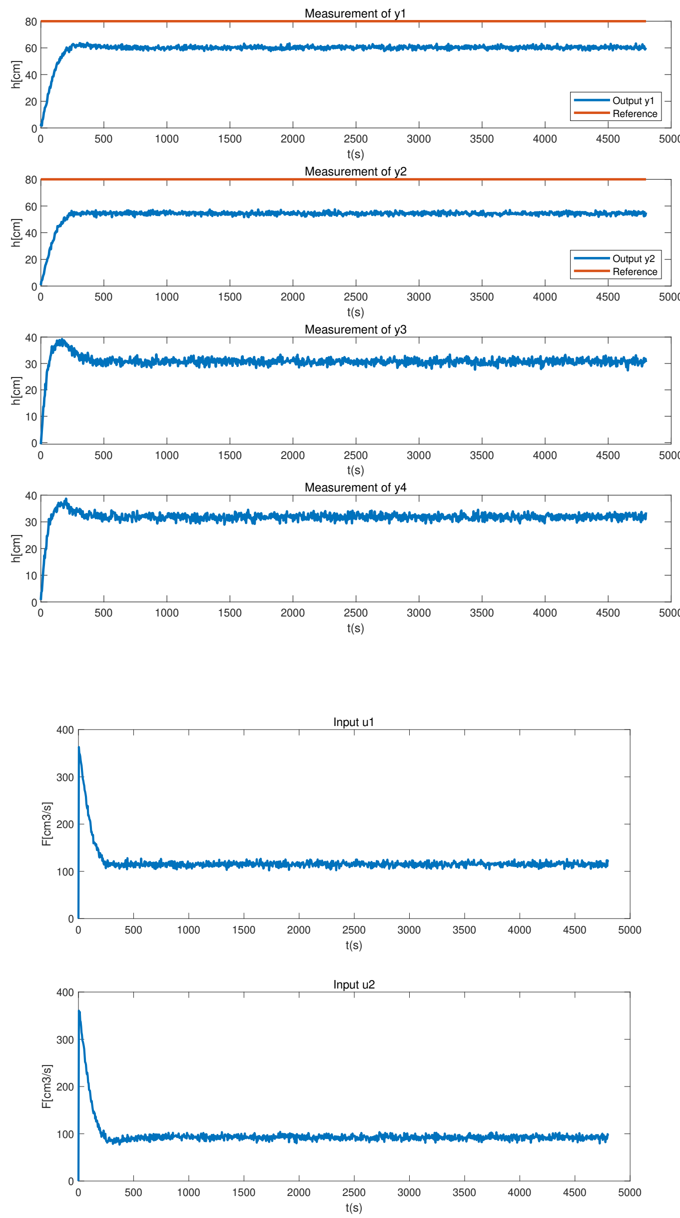

Figure 11.2: Y and U of Linear system using P- controller without step changing.

In the case of Linear system using PI- controller with disturbances are stochastic variables but do not contains step changes

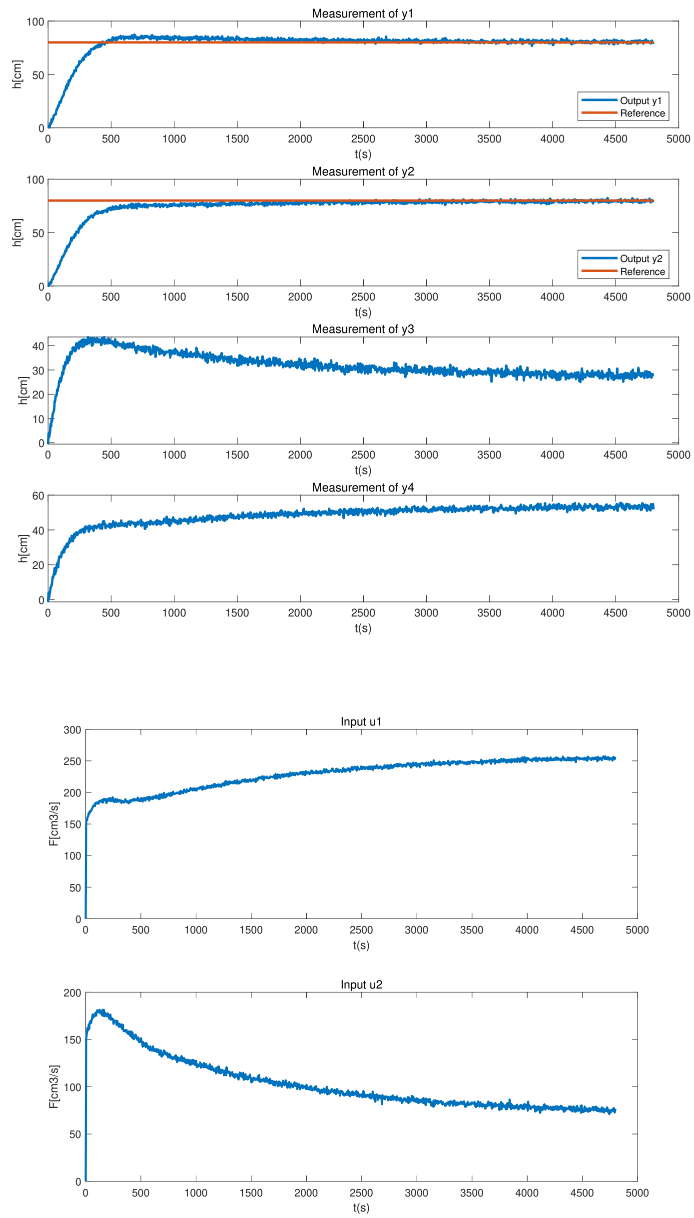

Figure 11.3: Y and U of Linear system using PI- controller without step changing.

In the case of Linear system using PID- controller with disturbances are stochastic variables but do not contains step changes

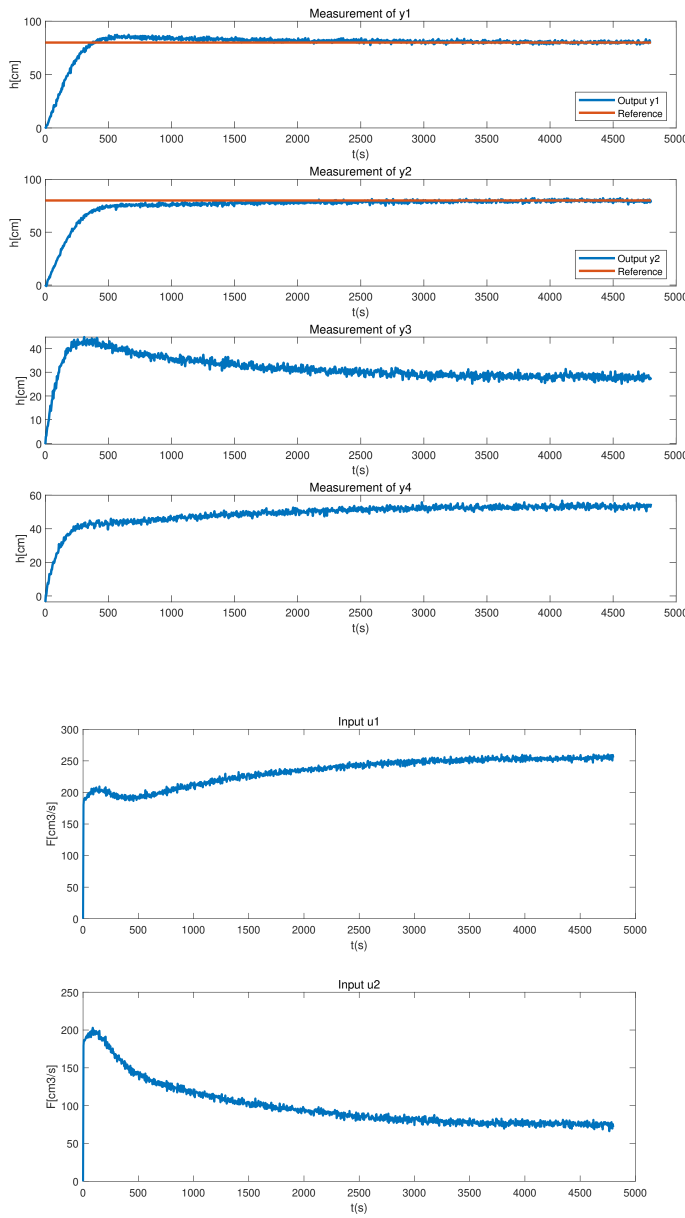

Figure 11.4: Y and U of Linear system using PID- controller without step changing.

In the case of nonlinear system using P- controller with disturbances are stochastic variables but do not contains step changes

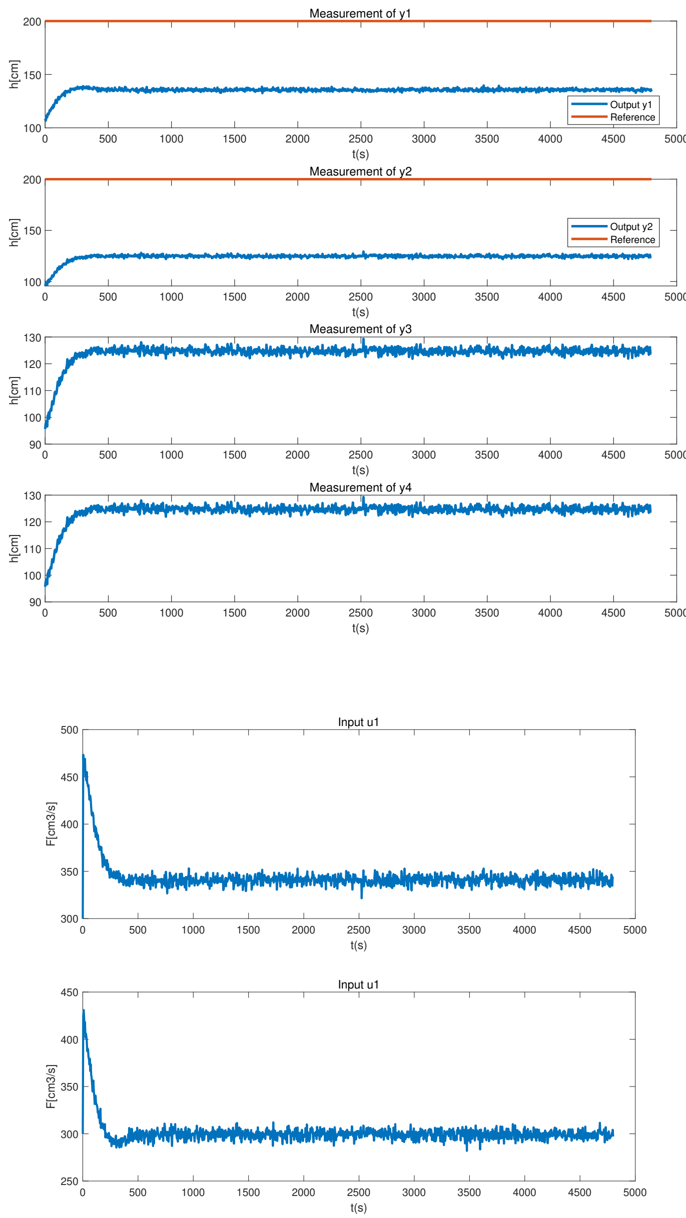

Figure 11.5: Y and U of nonlinear system using P- controller without step changing.

In the case of nonlinear system using PI- controller with disturbances are stochastic variables but do not contains step changes

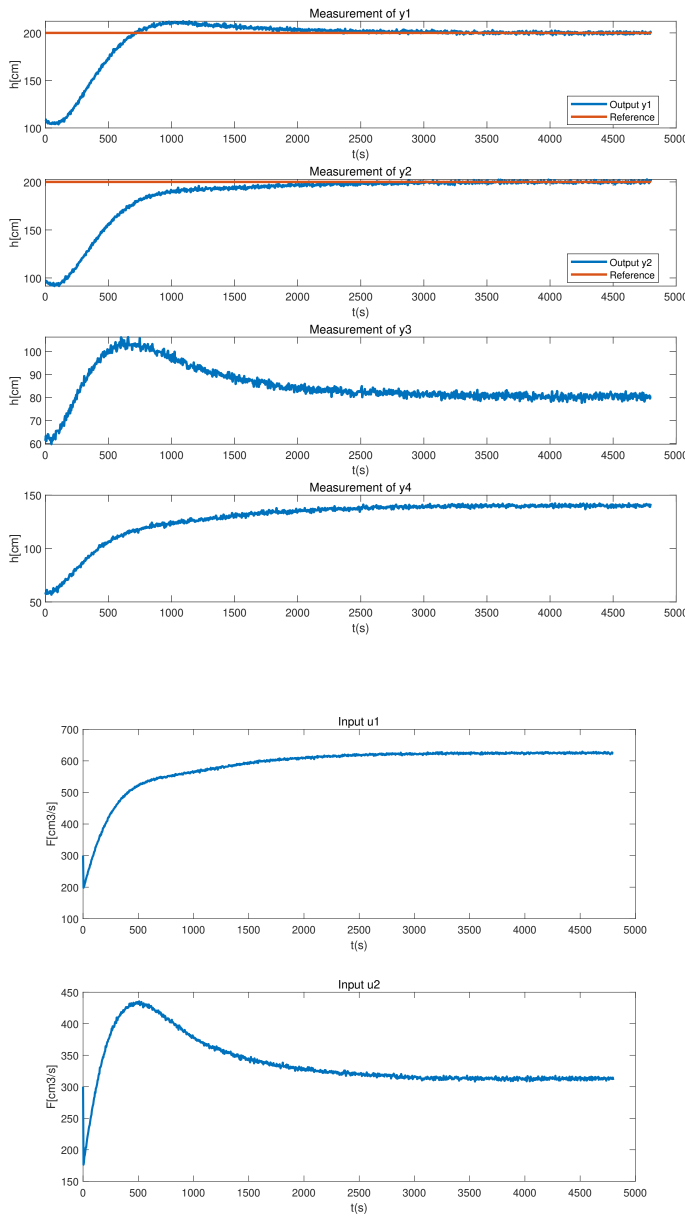

Figure 11.6: Y and U of nonlinear system using PI- controller without step changing.

In the case of nonlinear system using PID- controller with disturbances are stochastic variables but do not contains step changes

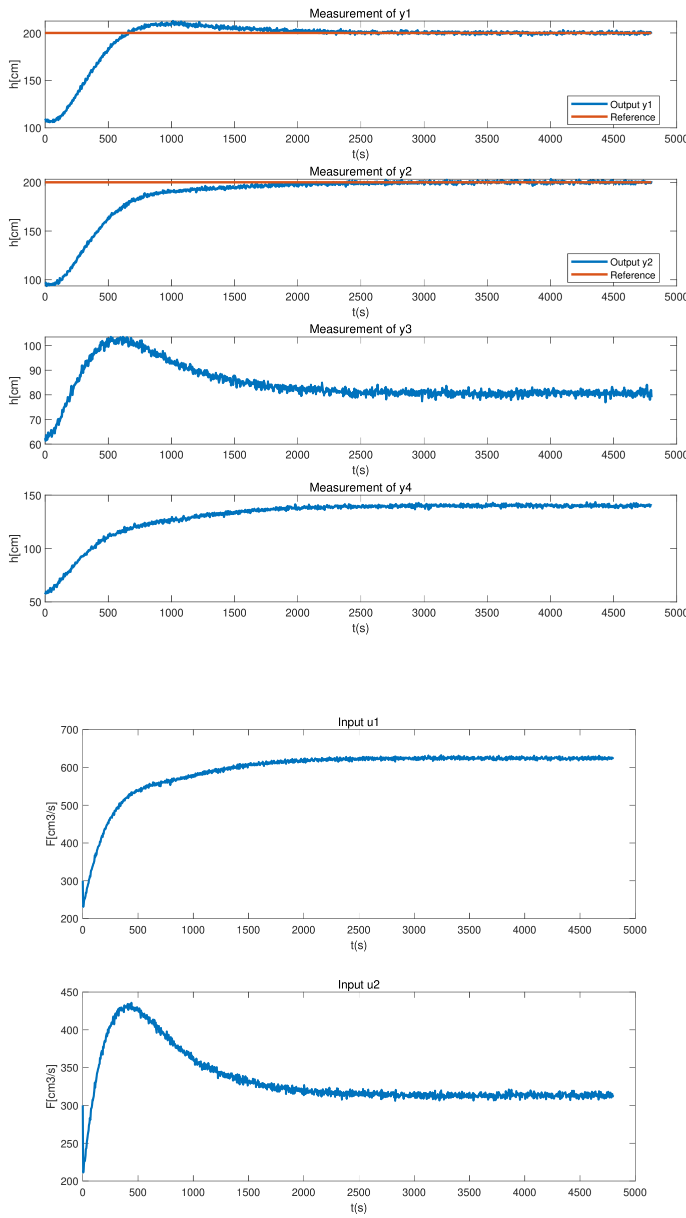

Figure 11.7: Y and U of nonlinear system using PID- controller without step changing.

Since the unknown disturbances are stochastic variables but DO CONTAIN step changes, the signal of disturbance of test for linear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 50 \\ 50 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right) \Longrightarrow N_{iid}\left(\begin{bmatrix} 100 \\ 100 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

the signal of disturbance of test for nonlinear system is constructed as

$$\begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 300 \\ 300 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right) \Longrightarrow N_{iid}\left(\begin{bmatrix} 350 \\ 350 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}\right)$$

In the case of Linear system using P- controller with disturbances are stochastic variables but do contains step changes

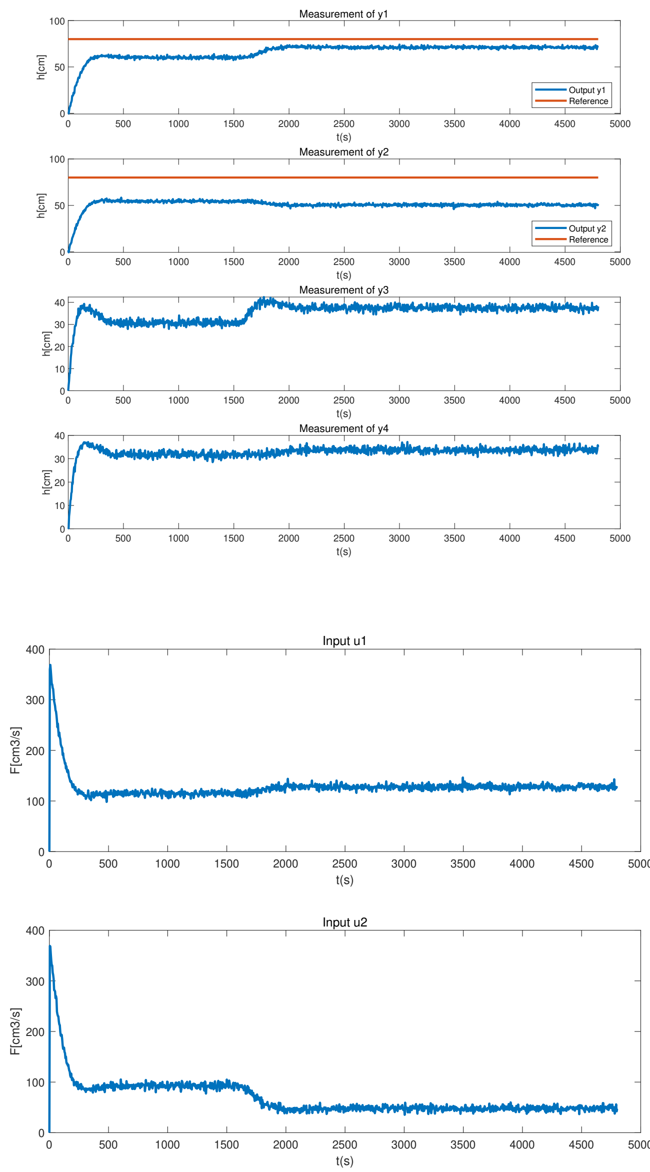

Figure 11.8: Y and U of Linear system using P- controller with step changing.

In the case of Linear system using PI- controller with disturbances are stochastic variables but do contains step changes

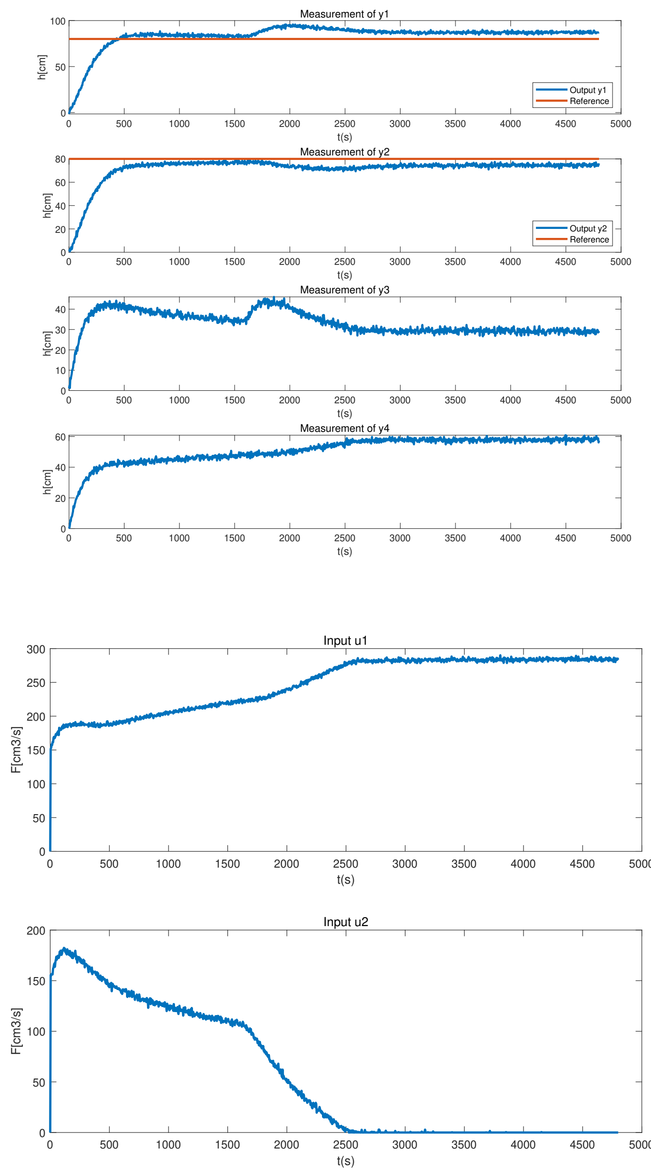

Figure 11.9: Y and U of Linear system using PI- controller with step changing.

In the case of Linear system using PID- controller with disturbances are stochastic variables but do contains step changes

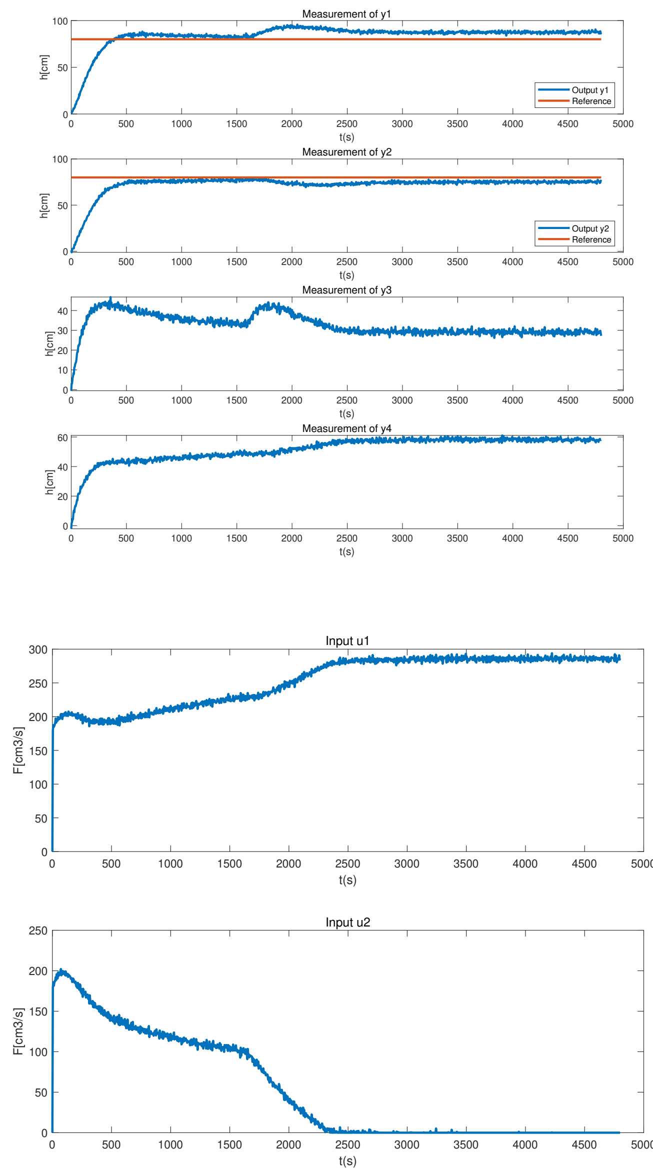

Figure 11.10: Y and U of Linear system using PID- controller with step changing.

In the case Of nonlinear system using P- controller with disturbances are stochastic variables but do contains step changes

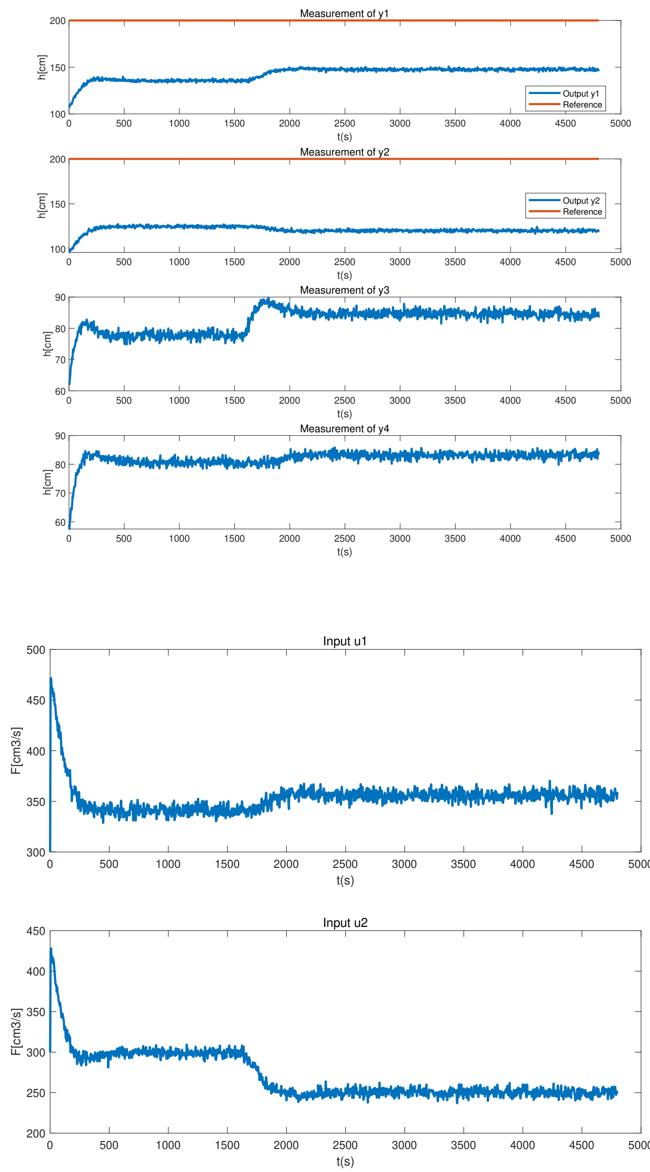

Figure 11.11: Y and U of nonlinear system using P- controller with step changing.

In the case of nonlinear system using PI- controller with disturbances are stochastic variables but do contains step changes

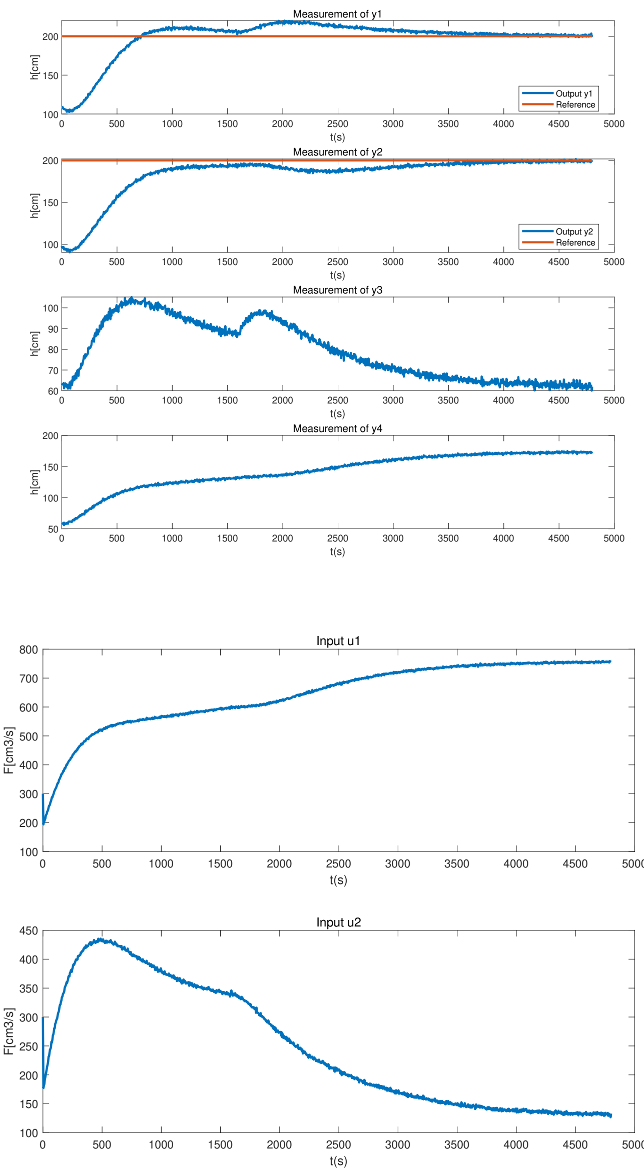

Figure 11.12: Y and U of nonlinear system using PI- controller with step changing.

In the case of nonlinear system using PID- controller with disturbances are stochastic variables but do contains step changes

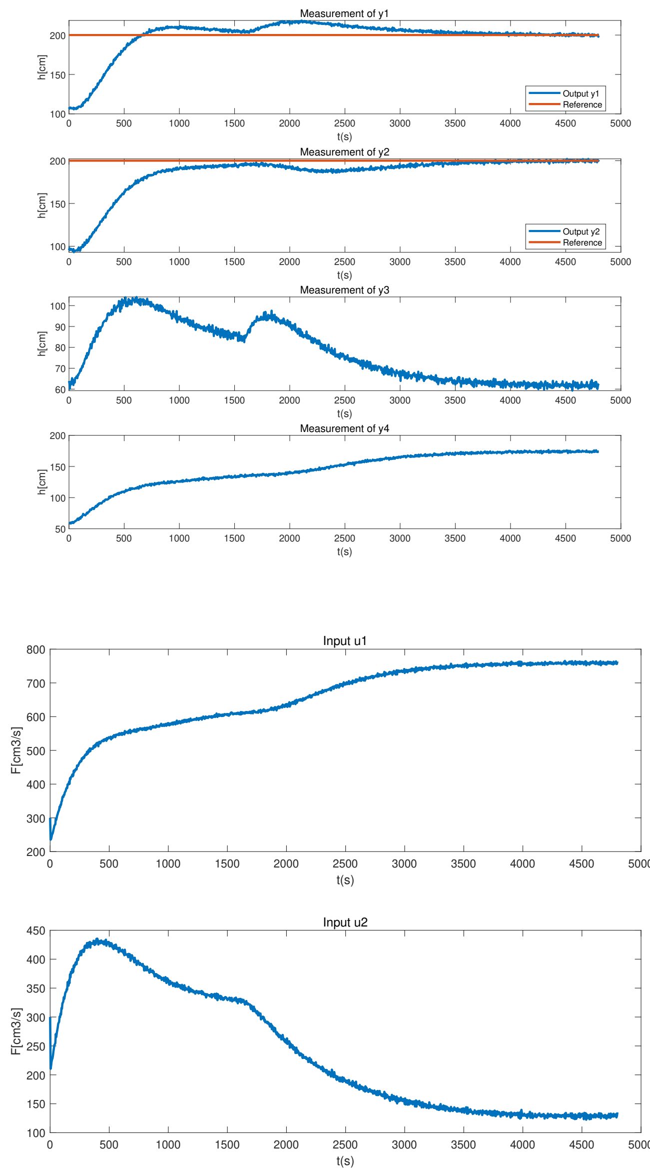

Figure 11.13: Y and U of nonlinear system using PID- controller with step changing.

The iteration times of the P-, PI- and PID controllers are also compared in the linear system and nonlinear system tests.

When the linear system is tested

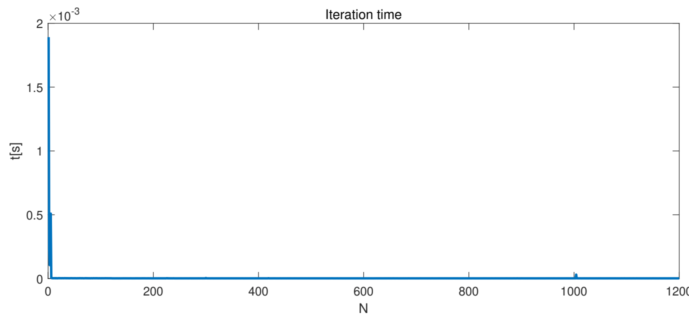

Figure 11.14: iteration times of linear system using P controller with step changing.

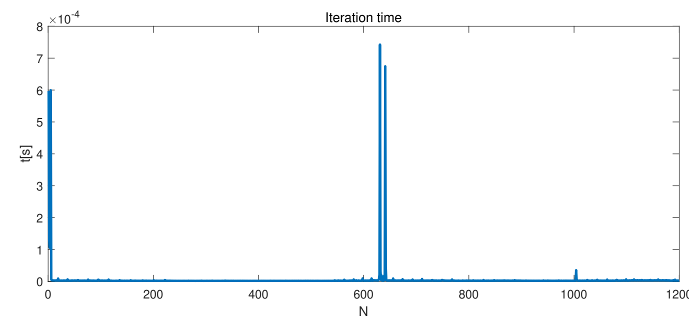

Figure 11.15: iteration times of linear system using PI controller with step changing.

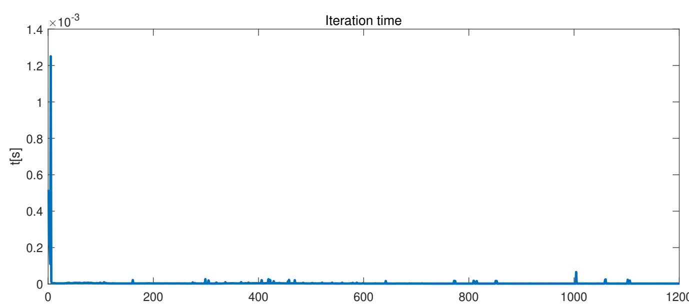

Figure 11.16: iteration times of linear system using PID controller with step changing.

When the nonlinear system is tested

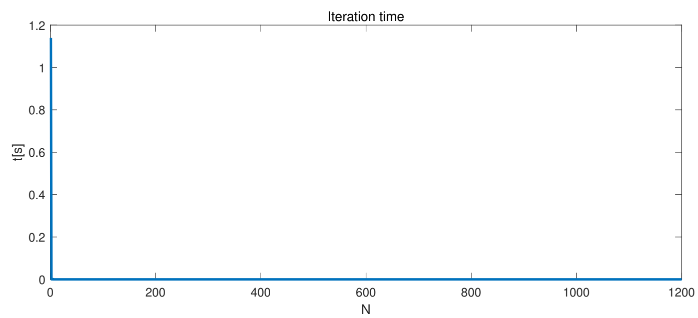

Figure 11.17: iteration times of nonlinear system using P controller with step changing.

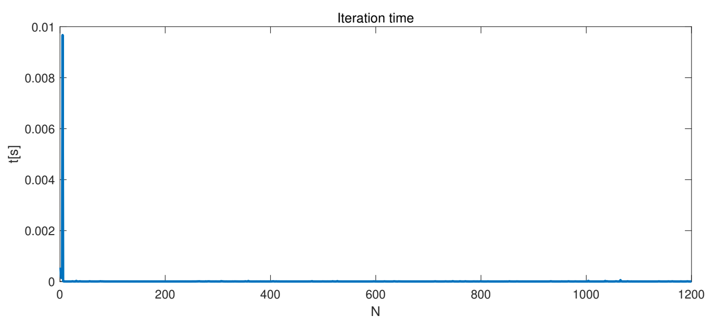

Figure 11.18: iteration times of nonlinear system using PI controller with step changing.

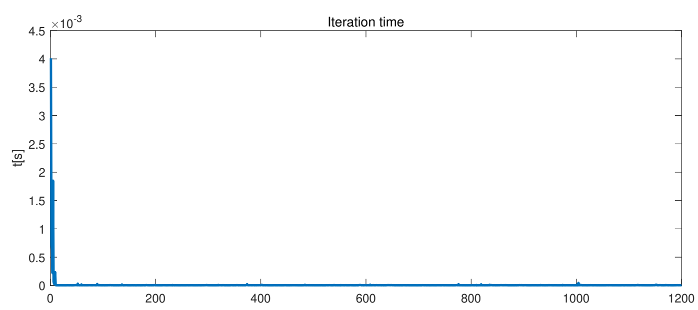

Figure 11.19: iteration times of nonlinear system using PID controller with step changing.

11.4 Problem 13.4

Question: Compare the PID type controllers to other controllers e.g. MPC.

It can be seen from the test results that a simple P-controller cannot eliminate steady-state errors and cannot resist disturbances. The PI controller and PID controller can basically eliminate the steady-state error after introducing the integral term, and can resist disturbances regardless of whether there is a step in the disturbances. The calculation cost is not high, and the calculation speed is very fast.

Although the PID control effect looks very good, it must be noted that from the characteristics of the PID controller, the PID controller can only constrain the upper and lower limits of the input, and cannot constrain the rate of change of the input. For example, as shown in Figure 11.13, when the initial value of the input is at the operating point 300, the input calculated by the PID controller in the next step is 200. It is difficult or even impossible for the actuator to complete the action within 4s of the sampling time in some cases. At the same time, in the subsequent process, the input has obviously been changing up and down, which will increase the burden on the actuator or even destroy the actuator.

In addition, PID has a good control effect on the SISO system. The better control effect of PID on the MIMO system has a great relationship with decoupling. As a MIMO system, the water tank system has only two inputs and two controlled variables, so The coupling is relatively less complicated. But if the system is more complex, decoupling will be more difficult, and it will be difficult to achieve a good control effect.

Compared with MPC controller, PID controller has obvious characteristics

- PID is aimed at model-free systems and can be used without modeling.
- PID can show good control performance for SISO systems and uncomplicated MIMO systems, but is not good at dealing with complex MIMO systems.

- PID can only make hard constraints on the upper and lower limits of the input. But PID cannot constrain the input rate and the actuator may not be able to execute the calculated input.
- PID calculation cost is low and have good real-time performance

CHAPTER 12

Discussion and Conclusion

Question: Discuss and comment on your results. Provide a discussion on the pros and conss of the different controllers.

This report mainly includes three parts, model, observer and controller.

In the model part, the chapter 1 discusses the influence of model parameters on whether the model is a minimum phase system, and selects a non-minimum phase system for research. Chapter 2 establishes deterministic and stochastic models based on the differential equations of the model. Chapter 3 performs step response and impulse response of the model near the selected operating point, and performs system identification to obtain the transfer function of the system. The chapter 4 directly linearizes around the selected operating point to obtain the continuous state space equation, and selects the appropriate sampling time to obtain the discrete space equation. The transfer functions obtained in Chapter 3 and Chapter 4 are compared, and it is found that the gain, time constant, and delay obtained by the two methods are basically the same, and the Markov parameters are basically the same. It shows that the model characteristics obtained by the linearization method near the operating point basically restore the characteristics of the system near the operating point.

In the observer design part, firstly, the minimum realization of the transfer function identified in Chapter 3 is performed to obtain the discrete state space, but it can be found that there are 16 states, and the specific physical meaning of each state is not known. In contrast, the discrete state space obtained by linearization in Chapter 4 has 4 state, and the specific physical meaning of each state quantity is known. After that, static and dynamic Kalman observers were designed based on the models in Chapter 3 and Chapter 4 respectively, and tested in linear and nonlinear systems, with and without step disturbance. It can be seen that when the observer based on the linear system design observes the nonlinear system, it can track the trend of disturbance but has a certain constant error. In addition, the observer based on the model design in Chapter 3 cannot know the initial value of the state when observing nonlinearity, which leads to a large overshoot of the observer at the beginning.

In the controller design part, unconstrained MPC, input constraint MPC, input constraint and output soft constraint MPC, P controller, PI controller and PID controller are respectively designed.

In the MPC design part, it is mainly the matrix construction of the optimization problem and the weight selection. When designing the MPC framework with input constraints and output soft constraints, the decision maker made judgments and decisions on the weights and maximum and minimum values of each MPC calculation to deal with special situations.

In the PID design part, the decoupling of the MIMO system is analyzed to obtain the decoupling gain matrix, and P, I, D parameters is selected for best performance. The tests were carried out in linear systems and nonlinear systems, with and without step disturbance.

According to the test results, it can be seen that although PID can get better control performance, the price is that the input rate is very large and cannot be restricted. And as the system becomes more complex, decoupling operations become more difficult. Compared with PID, MPC has a larger steady-state error, but MPC's constraints on input ensure that the calculated input and input rate are within a safe range.

Because the MPC algorithm and Kalman observer are designed based on linearized models, the control effect of nonlinear systems is worse than linear systems. Although not implemented in this report, the extended Kalman observer based on the nonlinear model and the nonlinear MPC algorithm can have a better control effect on the nonlinear system. Compared with unconstrained MPC, MPC with input constraints and output soft constraints can deal with more situations, and judge the weight and constraint value in the part of the decision maker. But as the constraints increase, the calculation time of MPC will increase.