**Computer based laboratory on State Feedback Controllers and Observers**

State Feedback Controller and State Observer Design

MATLAB Codes

G = tf([1],[1,0,0]) h = 0.02

[Phi Gama Cd Dd] = ssdata(c2d(G,h,'zoh'))

Wn = 15 Zeta = 0.8 roots([1, 2\*Zeta\*Wn, Wn\*Wn])

poled = exp(h\*roots([1, 2\*Zeta\*Wn, Wn\*Wn]))

L = place(Phi, Gama, poled)

Q = eig(Phi-Gama\*L) Gd = c2d(G,h,'zoh') poleobs = exp(h\*[-10 10]')

K = acker(Phi', Cd', poleobs)'

R = eig(Phi-K\*Cd)

F = Cd\*inv((eye(2)-Phi+Gama\*L))\*Gama

N = 1/F

>>

G =

1

---

s^2

Continuous-time transfer function.

h =

0.0200

Phi =

2 -1

1 0

Gama =

0.0156

0

Cd =

0.0128 0.0128

Dd =

0

Wn =

15

Zeta =

0.8000

ans =

-12.0000 + 9.0000i

-12.0000 - 9.0000i

poled =

0.7739 + 0.1408i

0.7739 - 0.1408i

L =

28.9384 -24.3979

Q =

0.7739 + 0.1408i

0.7739 - 0.1408i

Gd =

0.0002 z + 0.0002

-----------------

z^2 - 2 z + 1

Sample time: 0.02 seconds

Discrete-time transfer function.

poleobs =

0.8187

1.2214

K =

-2.3516

-0.7839

R =

1.2214

0.8187

F =

0.0056

N =

177.3642

>>

OBSERVATIONS

Implementation of the State Feedback Controller and the Full State Observer in Simulation

1. Implementation of the plant model using discrete state space model in SimulinkTM.

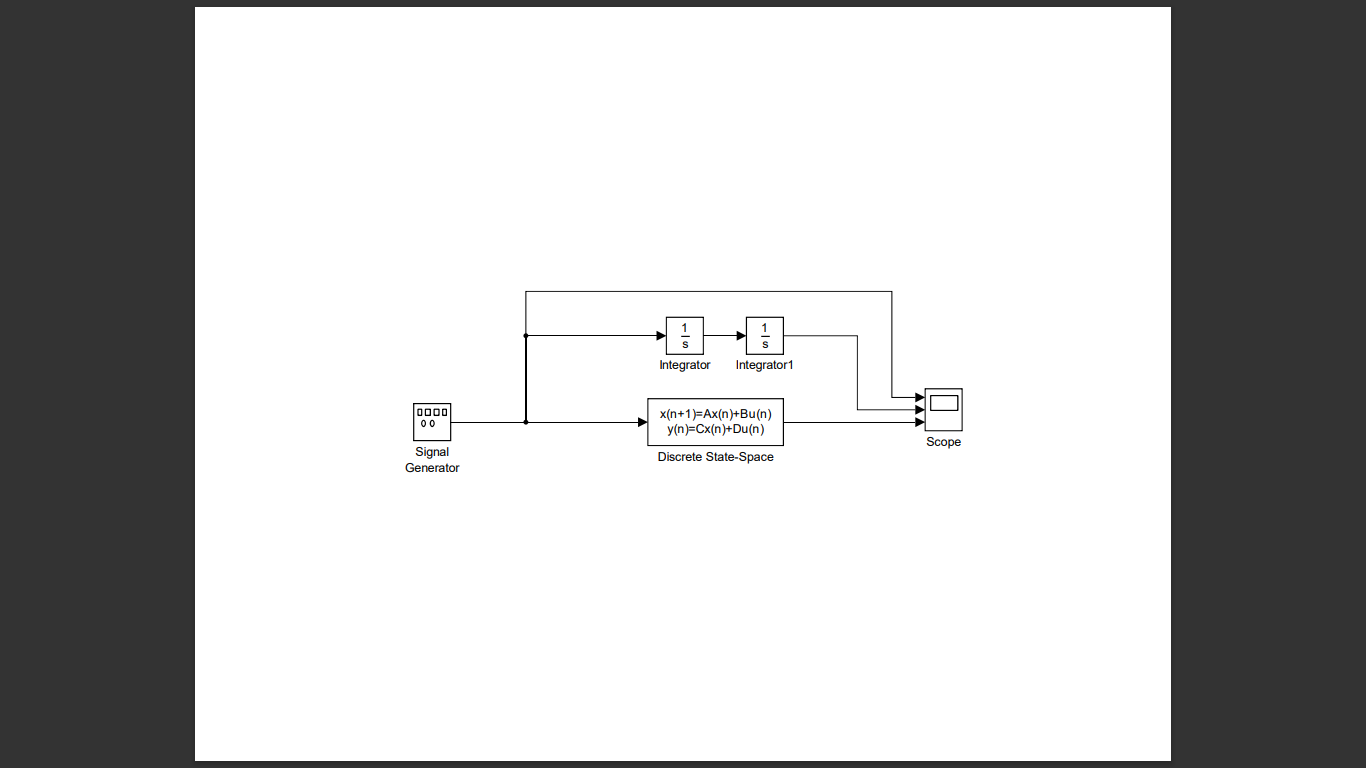


Figure 01: Block diagram of the plant model in state space model

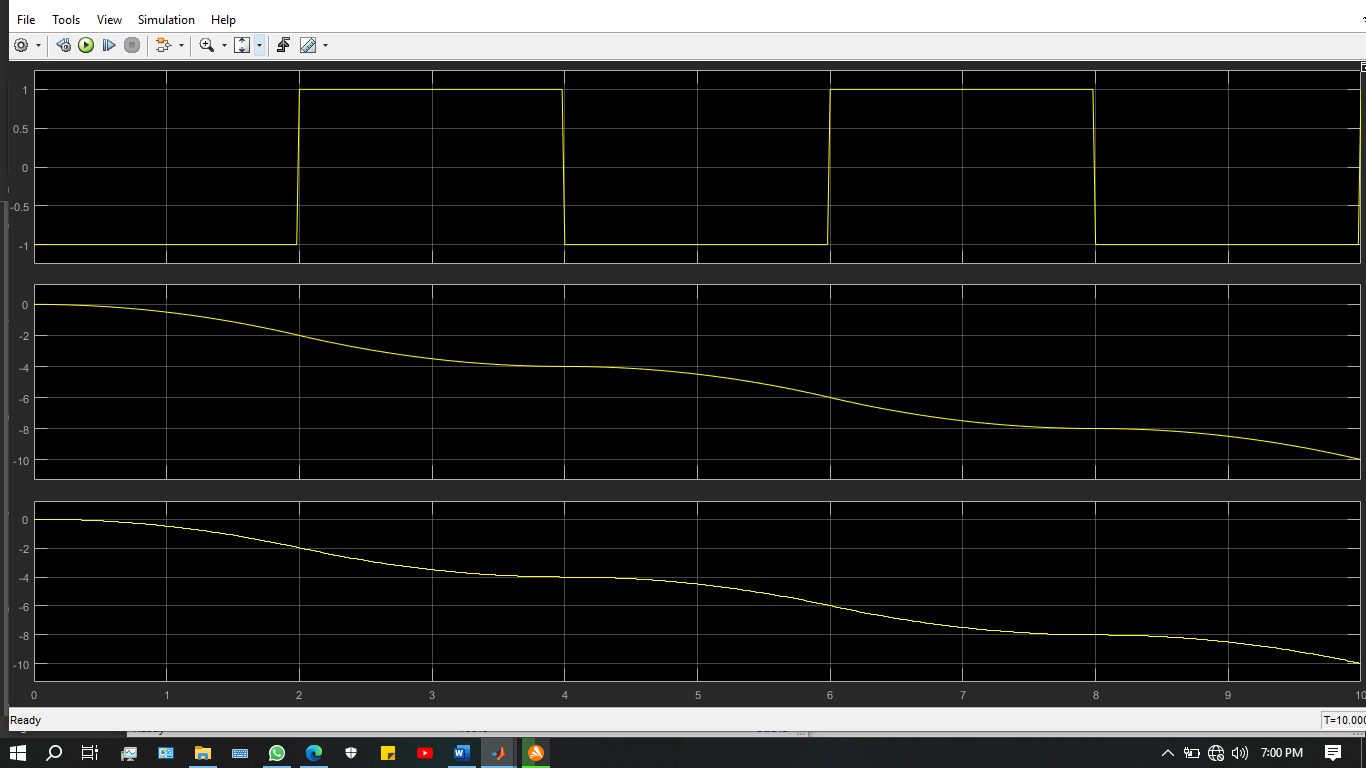


Figure 02: Output response of the plant models

1. Implementation of another plant model using gain matrix blocks, delay blocks, etc., in SimulinkTM so that both plant models are identical.

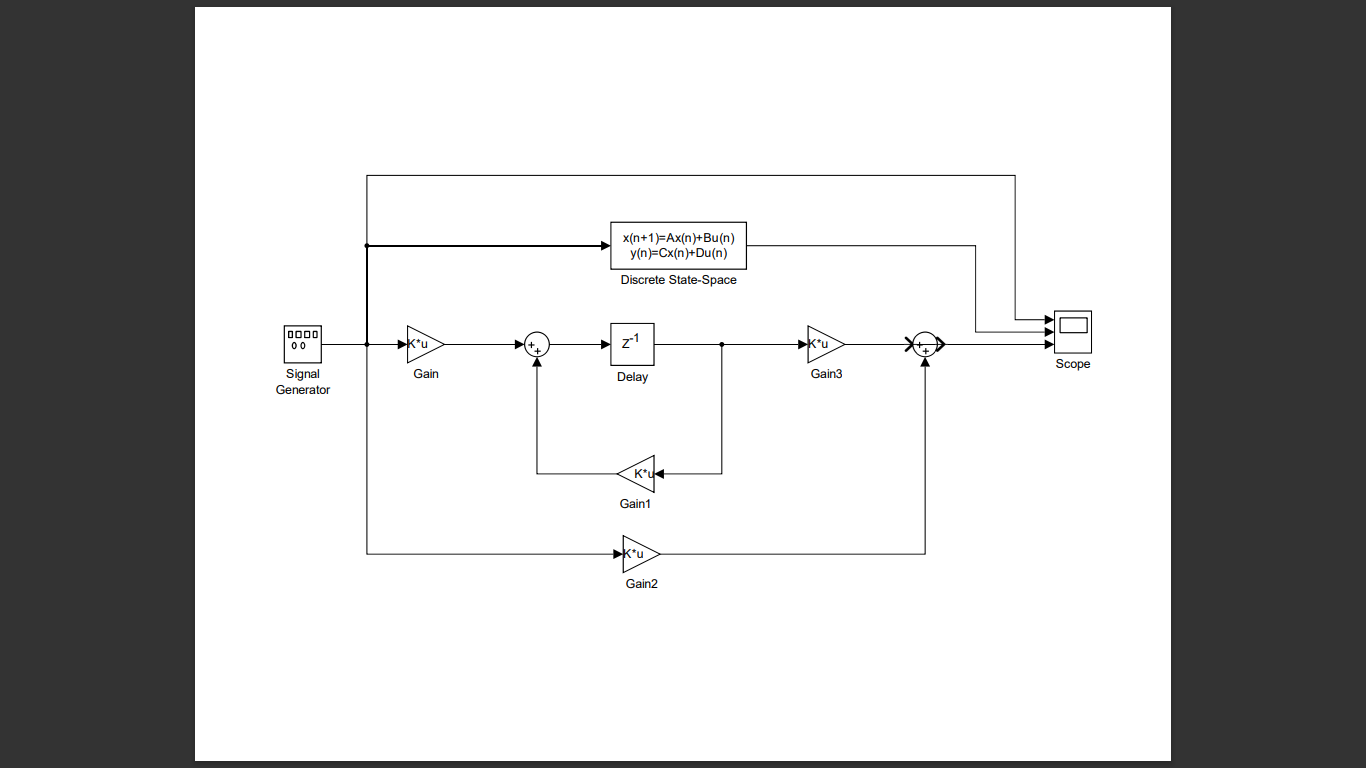


Figure 03: Plant model using blocks.

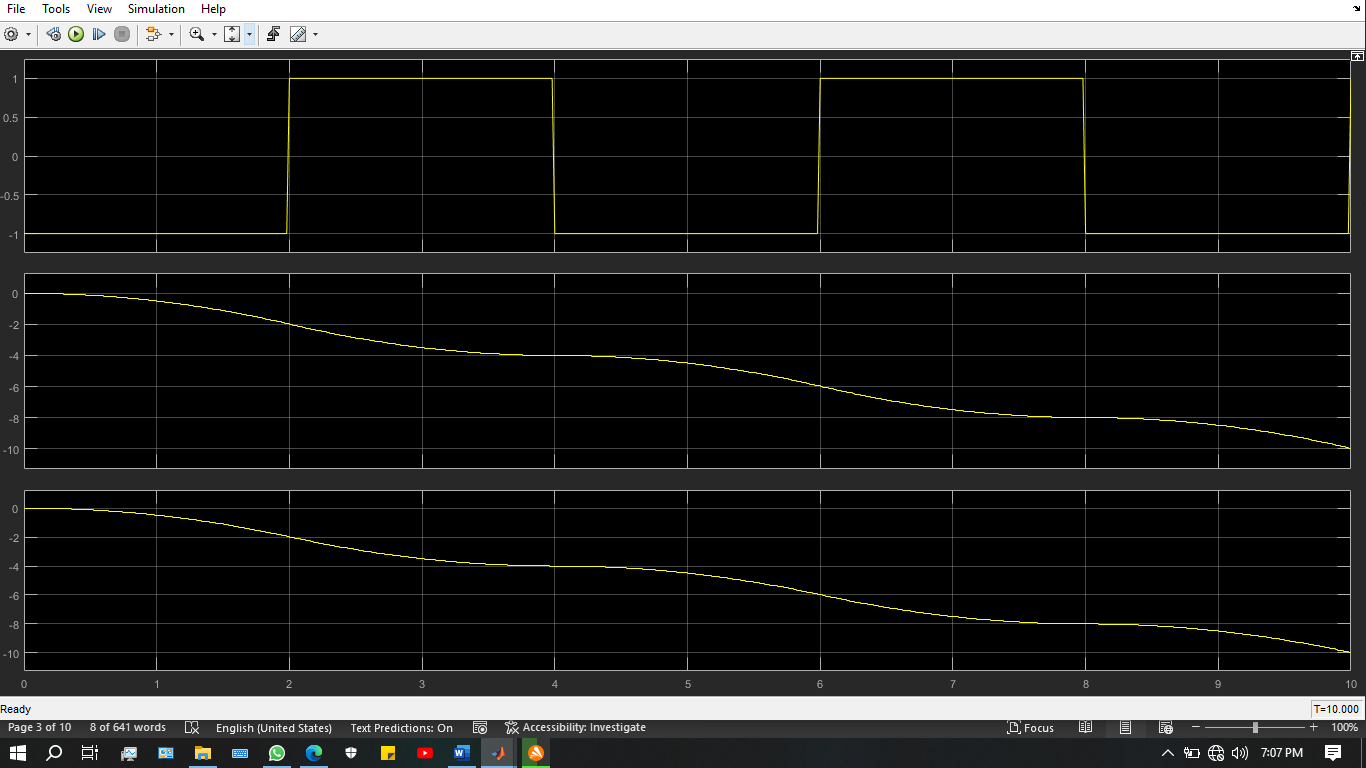


Figure 04: Output responses of the pulse train and plant model in state space and blocks

1. Implementation of the state feedback control law on the latter plant model

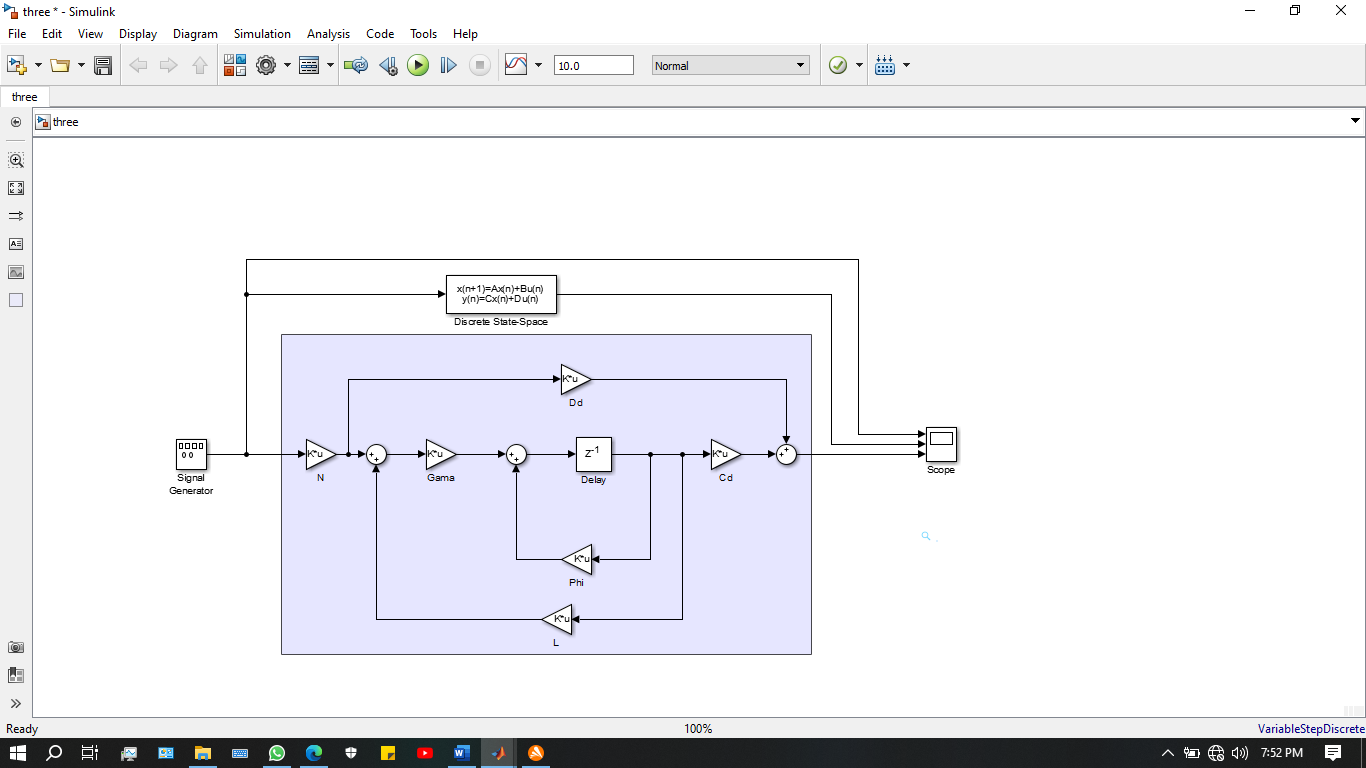


Figure 05: Implementation of the state feedback control on the latter plant model with a unity gain

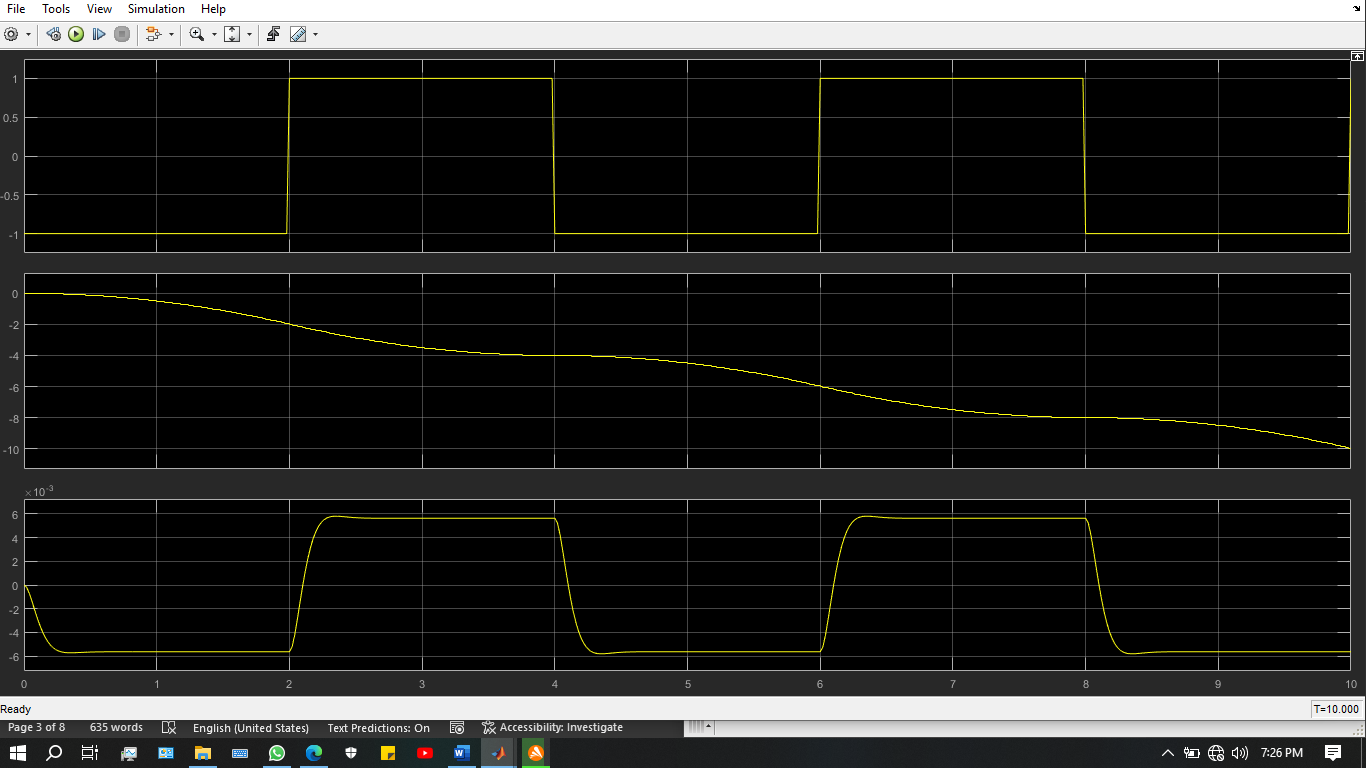


Figure 06: Output responses of the first plant model and the second plant model with the state feedback control

1. Implement the full state observer. (plant states and the observer states are identical.)

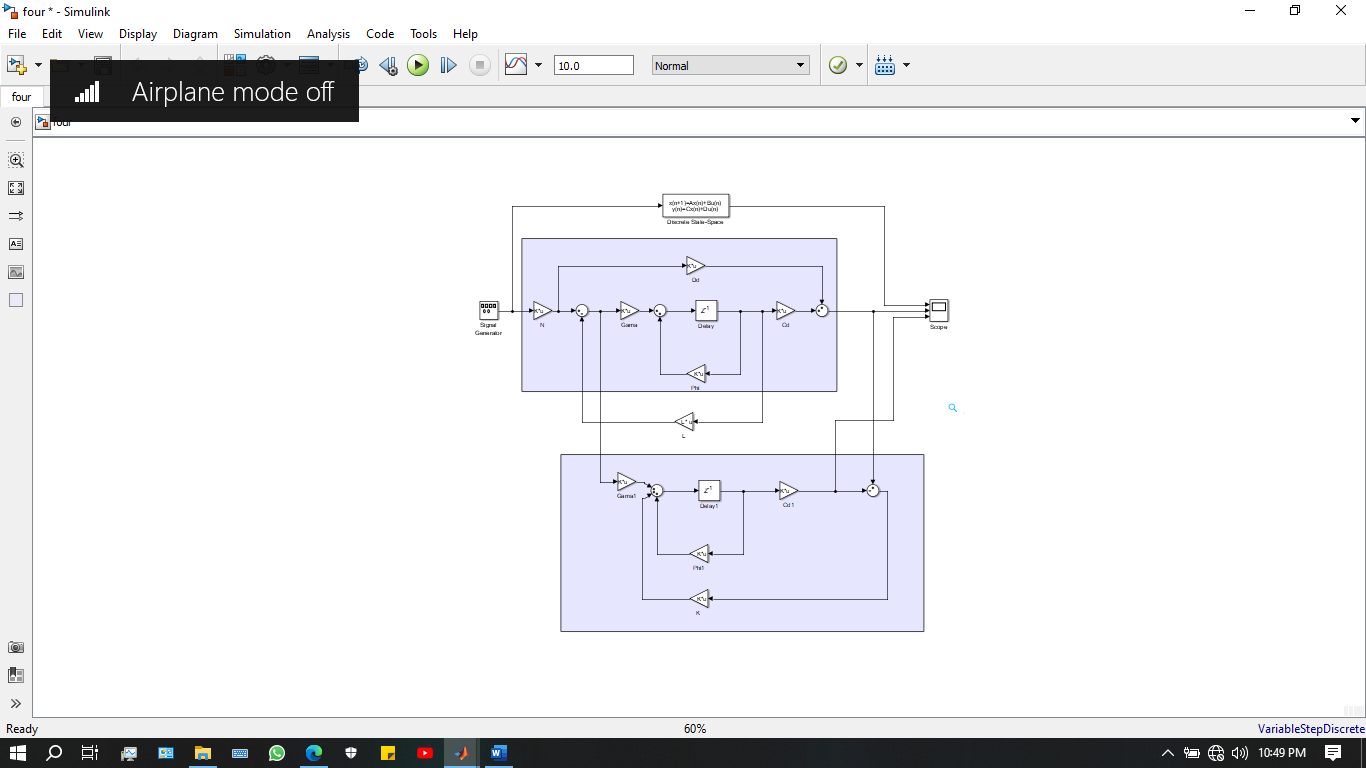


Figure 07: Block diagram setup for the state comparison of the plant model and the full state observer

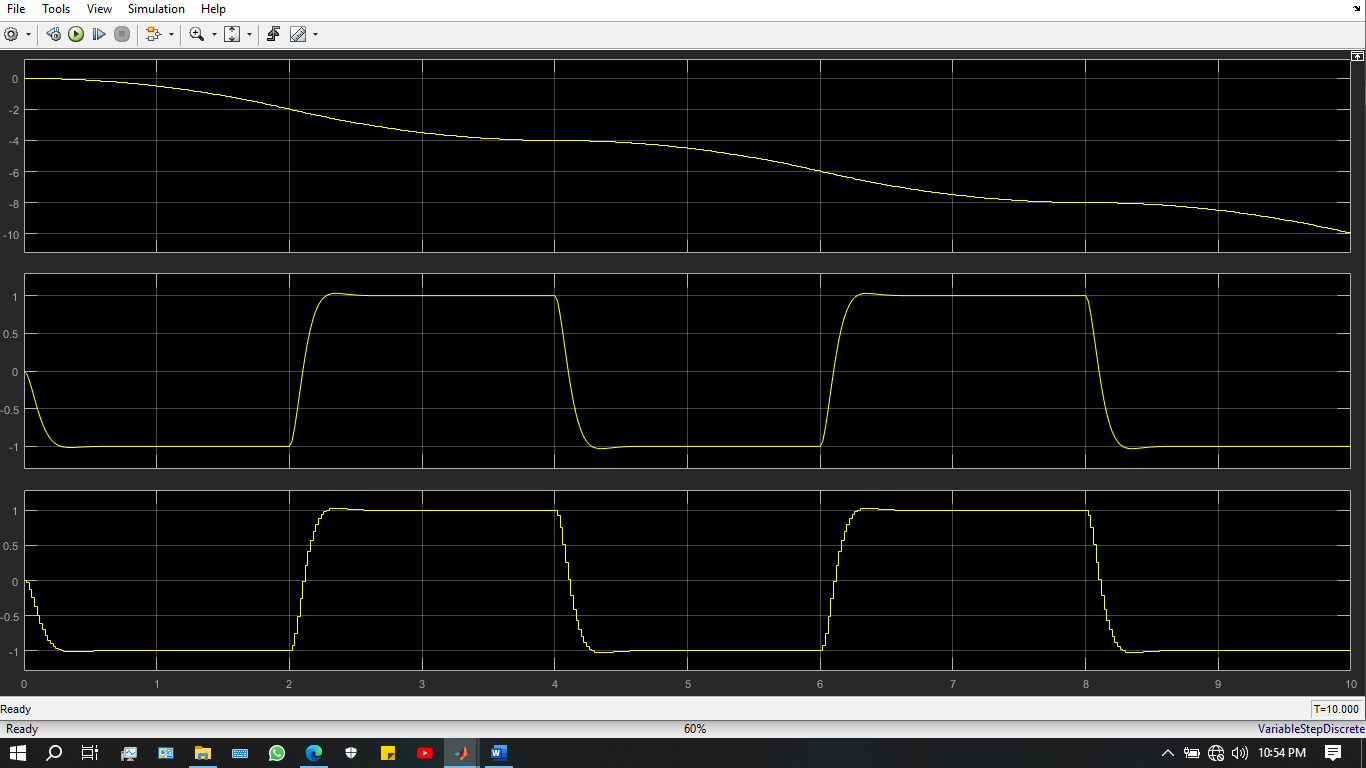


Figure 08: Output responses of the first plant model, the second plant model with the state feedback control and, full state observer

1. Implementation of the full state observer with the estimated state feedback in the control law.

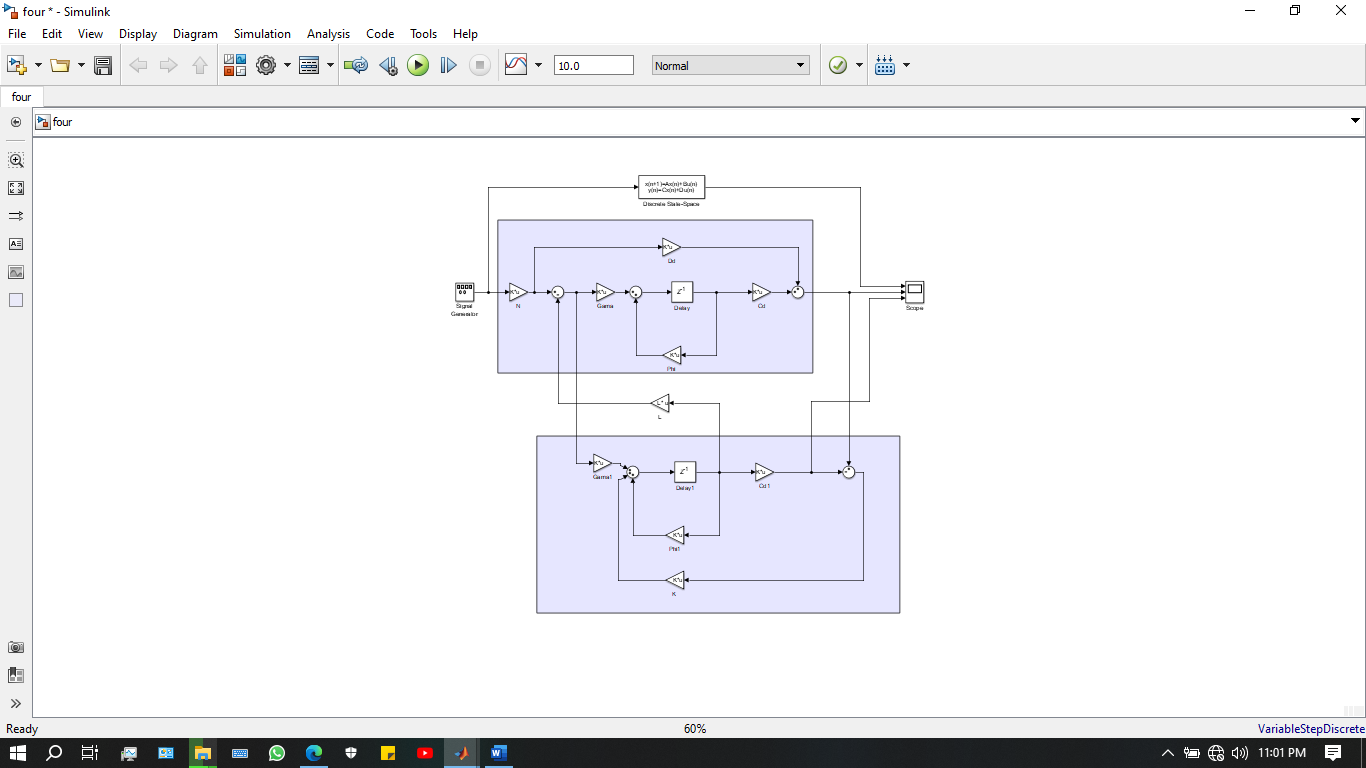


Figure 09: Implementation of the plant model and the full state observer with the estimated state feedback control

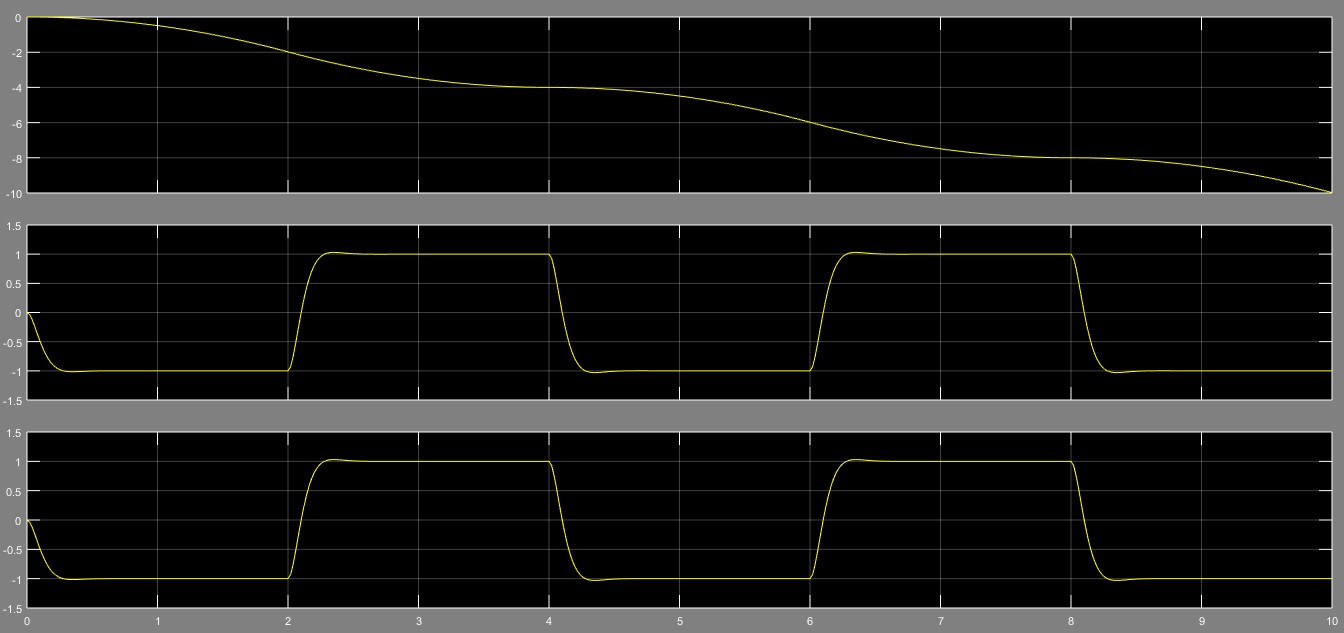


Figure 10: Output responses of the first plant model, the second plant model and, full state observer with the estimated state feedback control for a pulse train input signal

DISCUSSION

1. Explain why the transposes are used in obtaining Observer Gain Matrix using Ackermann’s formulae.

The observer gain matrix, often referred to as the observer gain vector or observer pole placement, is a design tool used in control systems theory to create an observer that calculates a system's state variables based on measurements of its input and output. The Ackermann formulas, which offer a way to assign the eigenvalues of the observer dynamics, are commonly used to compute the observer gain matrix.

Transposes are needed to make sure that the dimensions of the matrices used in the computations match up properly when using Ackermann's formulas to construct the observer gain matrix. Transposes enable the correct multiplication or addition of matrices by correctly aligning the dimensions.

REFERENCES