# EE208 EXPERIMENT 7

STATE FEEDBACK DESIGN IN THE DIGITAL DOMAIN

GROUP NO. 9
ARYAN BANSAL – 2020EEB1162
HIYA KWATRA – 2020EEB1173
ISHAAN SETHI – 2020EEB1174

## Objective

To design state feedback for a given digital state-space system, so as to realize performance specifications for different applications.

## System

The discretized state open-loop space representation for an armature-controlled DC motor is described by:

$$\mathbf{F} = \begin{bmatrix} 1.0 & 0.1 & 0.0 \\ 0.0 & 0.9995 & 0.0095 \\ 0.0 & -0.0947 & 0.8954 \end{bmatrix} \quad ; \quad \mathbf{g} = \begin{bmatrix} 1.622 \times 10^{-6} \\ 4.821 \times 10^{-4} \\ 9.468 \times 10^{-2} \end{bmatrix}$$

Corresponding to a sampling time of 0.01s.

## Tasks

1. Design state feedback gain matrices for three applications, for which the closed loop discrete time eigenvalue specifications are respectively as follows:
   a. $0.1,\ 0.4 \pm 0.4j$
   b. $0.4,\ 0.6 \pm 0.33j$
   c. All three eigenvalues at the origin.
2. For each case, discuss the response of the closed loop system for zero input with initial state vector as: $\boldsymbol{x_0} = [1\ 1\ 1]^T$.
3. Also examine the responses for step and ramp inputs.

## MATLAB Functions Used

```
eye, ss, eig, place, acker, lsim, lsiminfo, initial, step,stepinfo,
pzmap, stem
```

# Pole placement

## Open loop system

We have assumed the C and D matrices as follows:

```
% C and D matrices not given in the handout, setting D = 0 and
% C = identity matrix to view all states' time response.
C = eye(3);
D = 0;
```

The eigenvalues of the open-loop system are $1,\ 0.99,\ 0.9049$.

## Designing gain matrices

Designing the gain matrices in MATLAB is simple, and can be done with a single command – **place** or **acker** (for poles with multiplicity > 1).

Using this command, we obtain the following gain matrices corresponding to each application –

$$K_1 = 1e3 * [4.9268\ \ 1.4324\ \ 0.0137]$$

$$K_2 = 1e3 * [1.6985\ \ 0.7009\ \ 0.0101]$$

$$K_3 = 1e4 * [1.0527\ \ 0.2621\ \ 0.0017]$$

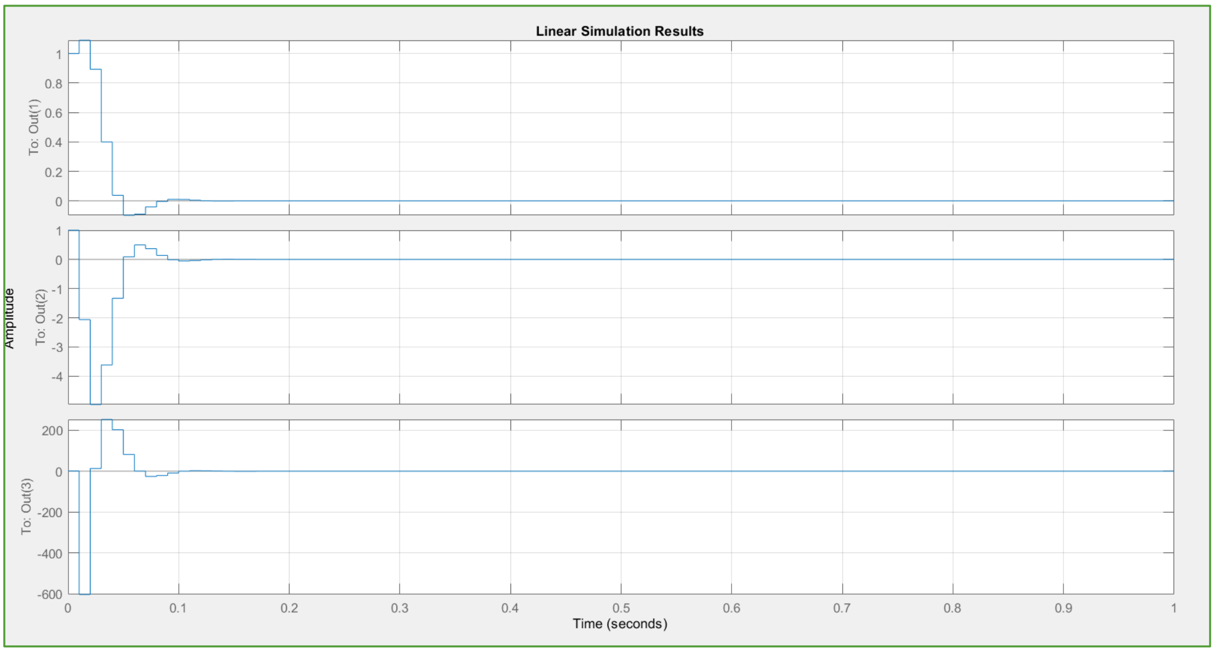
We can also see the actual poles placed –

```
>> pole(system_CL_1)
ans =
   0.4000 + 0.4000i
   0.4000 - 0.4000i
   0.1000 + 0.0000i

>> pole(system_CL_2)
ans =
   0.6000 + 0.3300i
   0.6000 - 0.3300i
   0.4000 + 0.0000i

>> pole(system_CL_3)
ans =
   1.0e-05 *
   0.2661 + 0.4609i
   0.2661 - 0.4609i
  -0.5322 + 0.0000i
```

For Applications 1 and 2 the poles are placed at the exact locations, but pole placement for Application 3 is inaccurate because of numerical limitations in MATLAB – placing multiple poles at the z-domain origin is bound to induce some numerical error.

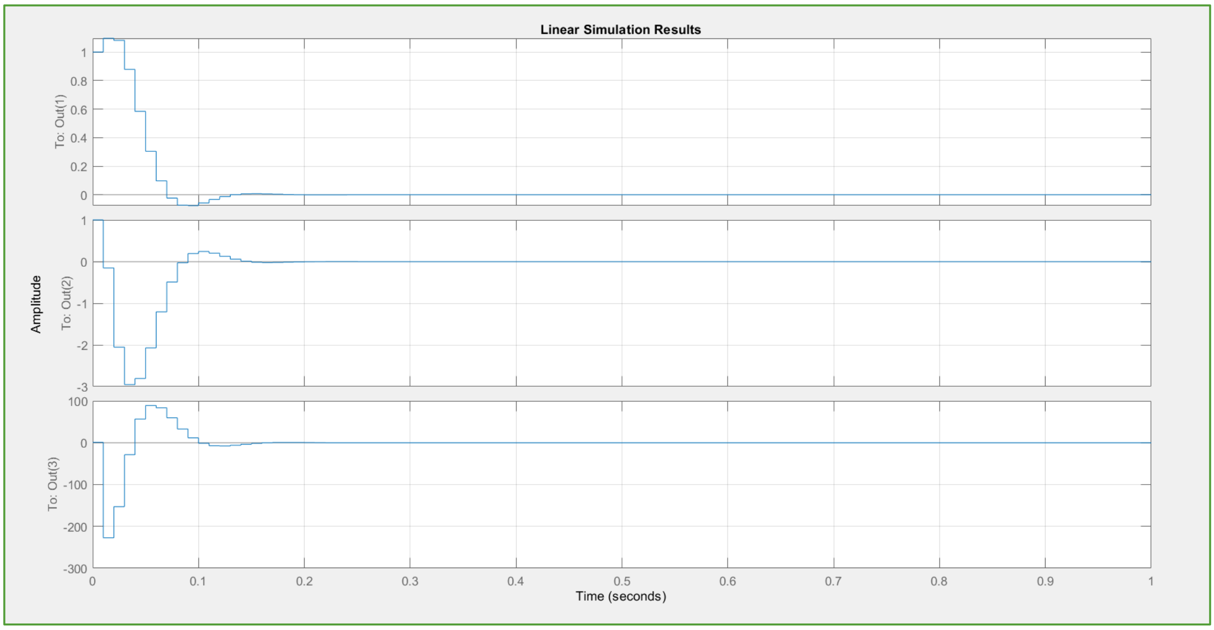
# Zero input responses with initial state $\boldsymbol{x_0} = [1\ 1\ 1]^T$

## Application 1




- Settling time for all three states is around 0.08s.
- Steady-state values are 0, as is expected for natural response.
- General behavior is oscillatory.

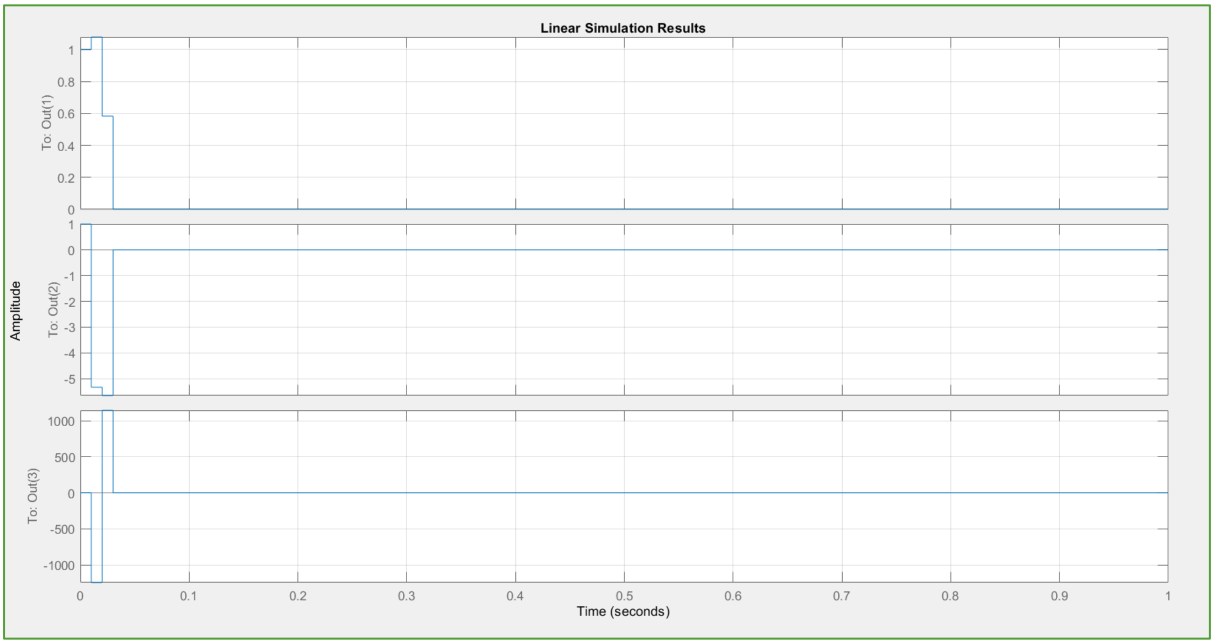
## Application 2




- Settling times are respectively: 0.1153 s, 0.1299 s, and 0.1362 s.
- Comparing with Application 1, damping of complex poles is increased (0.685 from 0.566), but frequency is decreased (63 rad/s from 97 rad/s). For the real pole, frequency is decreased from 230 rad/s to 91.6 rad/s.

- Thus, even though damping is increased, the frequencies of all three poles have been reduced, hence settling time increases slightly.

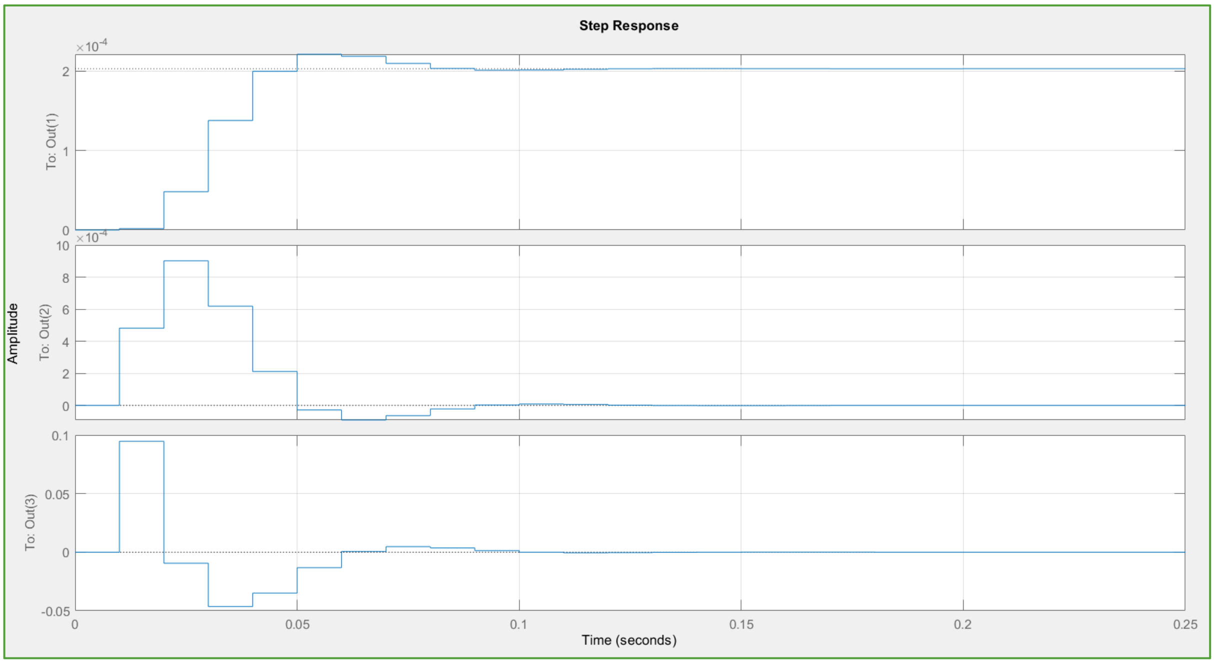
## Application 3




- Settling time for all three states is approximately 0.03 s.
- All three poles are at origin, having a higher frequency, hence the settling times are lesser.

# Step responses

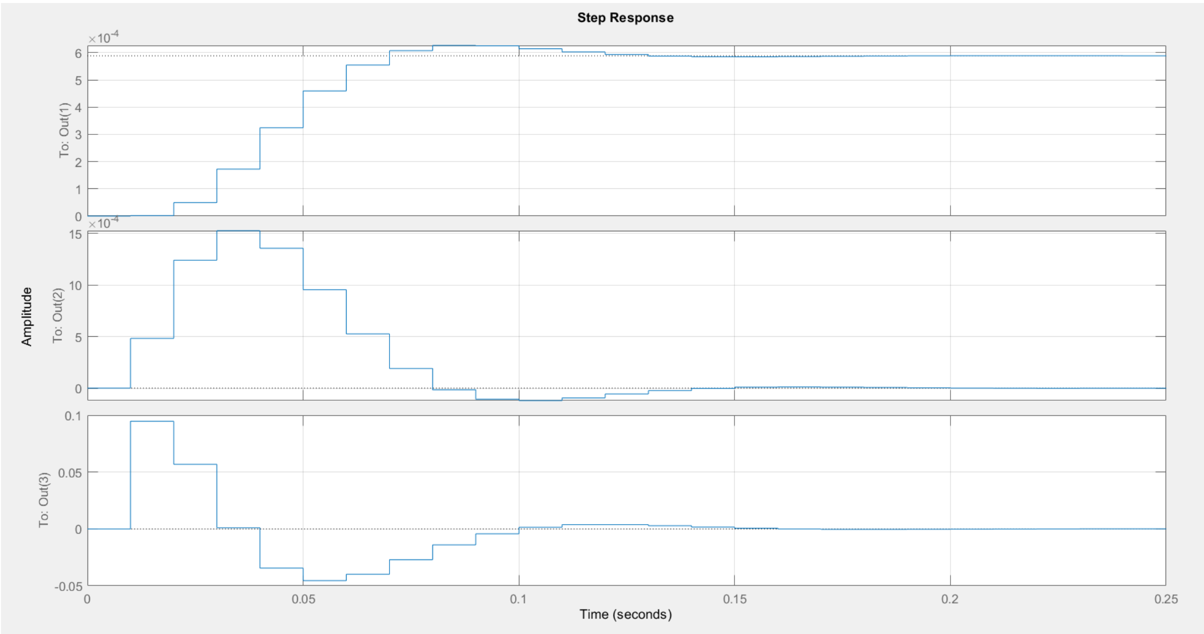
## Application 1




| State | y_ss | Transient Time |
|---|---|---|
| *x1* | 0.000203 | 0.0741 |
| *x2* | 0 | 0.0815 |
| *x3* | 0 | 0.0876 |

- Steady-state value for x2 and x3 is zero.
- Overall, the response is of very small magnitude.

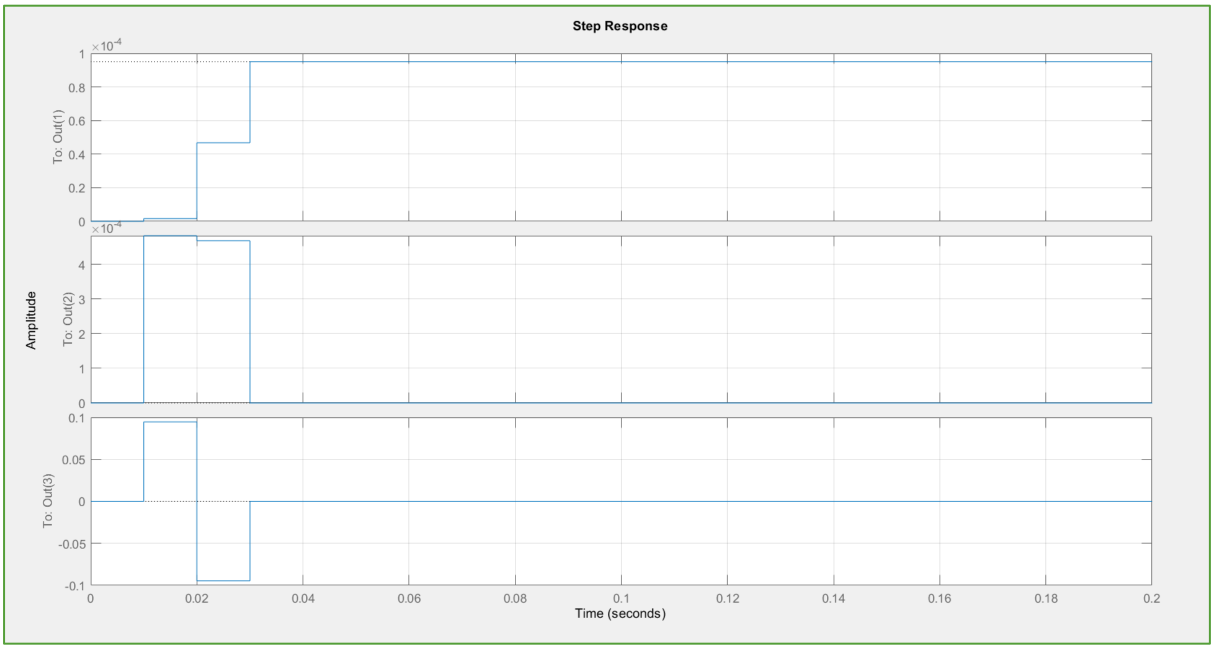
## Application 2



| State | y_ss | Transient Time |
|---|---|---|
| x1 | 0.000589 | 0.112 |
| x2 | 0 | 0.128 |
| x3 | 0 | 0.137 |

- Very similar response to that of Application 1.
- Transient Times have been increased, due to lesser frequency of poles, as discussed for the natural response case.

## Application 3

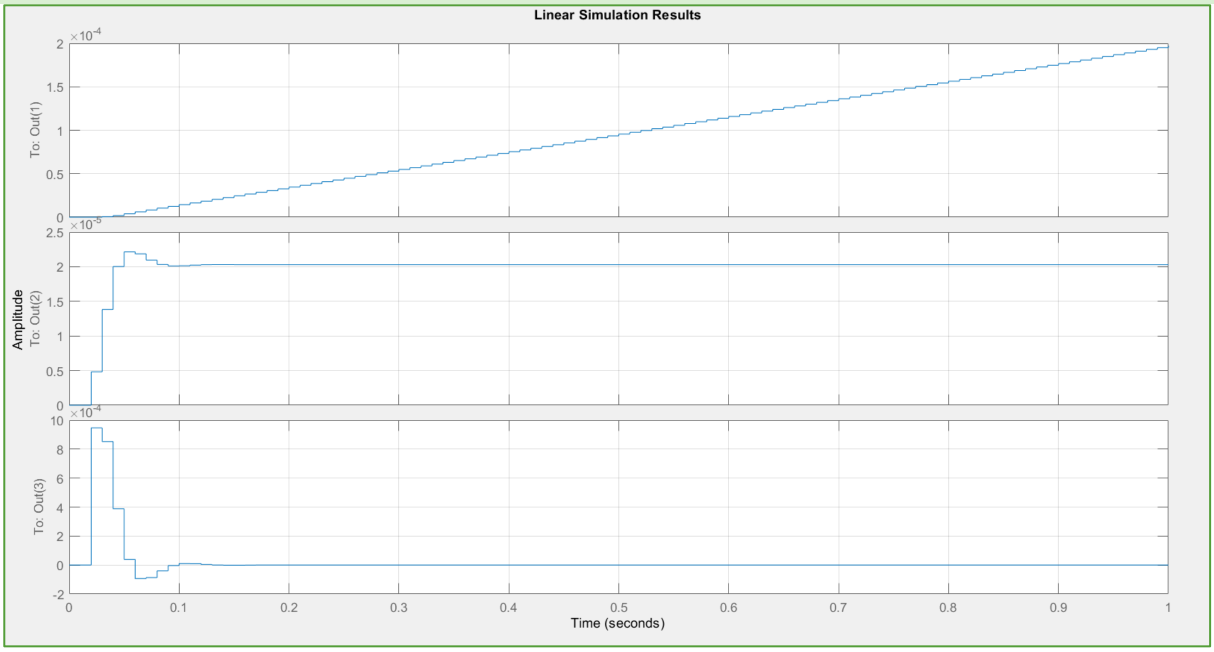



| State | y_ss | Transient Time |
|---|---|---|
| x1 | 9.5*e-5 | 0.0296 |
| x2 | 0 | 0.0298 |
| x3 | 0 | 0.0298 |

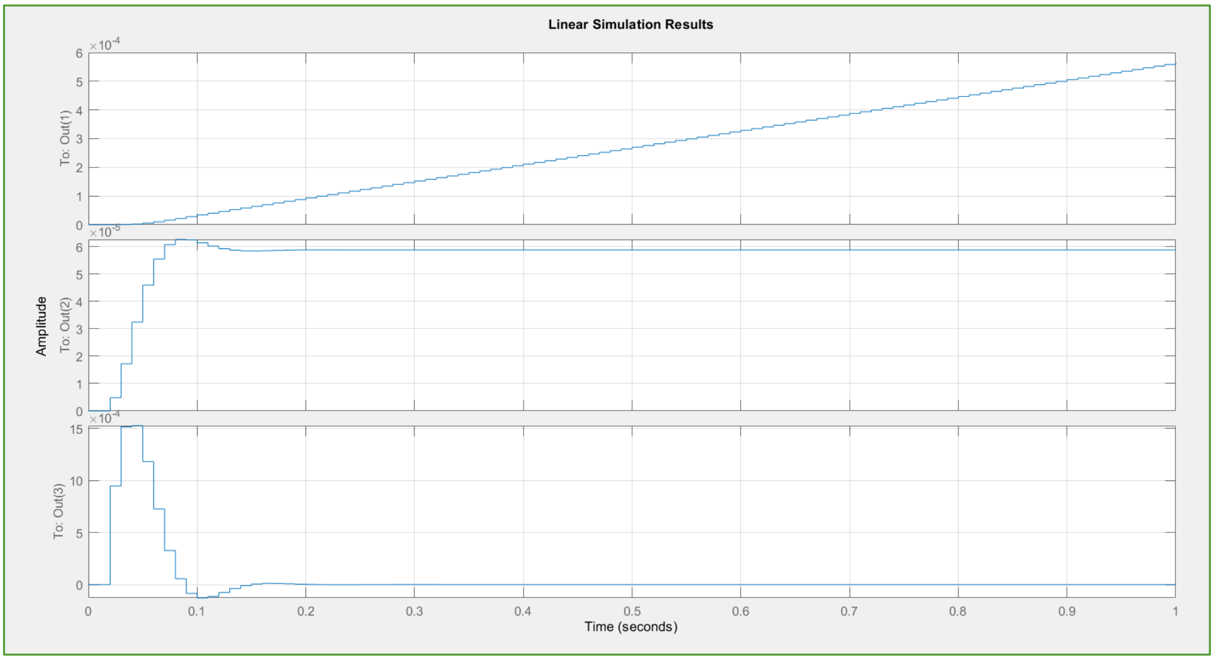
- The response is very short-lived, due to high-frequency poles (in digital domain, poles near the origin are high-frequency).
- Similar to above cases, steady state-value is zero for x2 and x3.

# Ramp responses

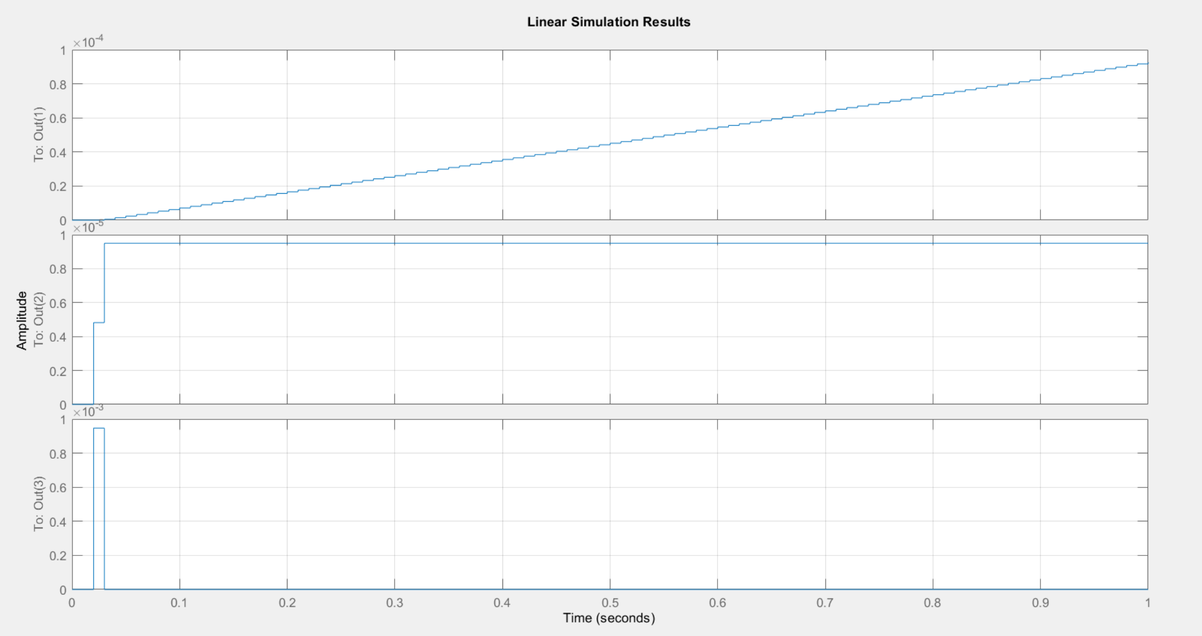
## Application 1




## Application 2



## Application 3




For all three cases, the responses are very similar so we shall discuss them together:

- State x1 follows the ramp input with some delay.
- State x2 settles down to a particular value. The settling times and peak overshoots change due to the difference in eigenvalues, as has been discussed before.
- State x3 settles down to 0, with the same reason for difference in settling times.

The various responses to initial-state, step-input and ramp-input differ because of:

1. Change in poles (i.e., CL eigenvalues), across different applications.
2. Change in zeros, across the states x1, x2, x3.

# Conclusions

In this experiment we designed state feedback gain matrices for a given digital system to achieve the required specifications. We studied the CL system’s natural response to a given initial state. We also studied the step and ramp responses for various CL eigenvalue placements.

# MATLAB Script

https://drive.google.com/file/d/1yk6o-RC_tjPzFxO1kS5BJ6eMuyK6ig-F/view?usp=sharing