



# State Space Model Implementation with Python

---

Graduate School of Energy Convergence

Prof. Yun-Su Kim

# Exercise – Problem

## 1. First order differential equation

$$3 \frac{dx}{dt} + 12x = 6u$$

$$y = x$$

## 2. Second order differential equations

$$2 \frac{dx_1}{dt} + 6x_1 = 8u$$

$$3 \frac{dx_2}{dt} + 6x_1 + 9x_2 = 0$$

$$y = \frac{x_1 + x_2}{2}$$

## 3. Second order differential equation

$$4 \frac{d^2y}{dt^2} + 2 \frac{dy}{dt} + y = 3u$$

## 4. Nonlinear differential equations

$$2 \frac{dx_1}{dt} + x_1^2 + 3x_2^2 = 16u$$

$$3 \frac{dx_2}{dt} + 6x_1^2 + 6x_2^2 = 0$$

$$y_1 = x_2$$

$$y_2 = x_1$$

$$\bar{u} = 1, \quad \bar{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

# Exercise – Code

```
import numpy as np
from scipy import signal

# problem 1
A = [-4.0]
B = [2.0]
C = [1.0]
D = [0.0]
sys1 = signal.StateSpace(A,B,C,D)
t1,y1 = signal.step(sys1)

# problem 2
A = [[-3.0,0.0],[-2.0,-3.0]]
print(np.linalg.eig(A)[0])
B = [[4.0],[0.0]]
C = [0.5,0.5]
D = [0.0]
sys2 = signal.StateSpace(A,B,C,D)
t2,y2 = signal.step(sys2)

# problem 3
A = [[0.0,1.0],[-0.25,-0.5]]
print(np.linalg.eig(A)[0])
B = [[0.0],[0.75]]
C = [1.0,0.0]
D = [0.0]
sys3 = signal.StateSpace(A,B,C,D)
t = np.linspace(0,30,100)
u = np.zeros(len(t))
u[5:50] = 1.0 # first step input
u[50:] = 2.0 # second step input
t3,y3,x3 = signal.lsim(sys3,u,t)

# problem 4
A = [[-2.0,-6.0],[-8.0,-8.0]]
print(np.linalg.eig(A)[0])
B = [[8.0],[0.0]]
C = [[0.0,1.0],[1.0,0.0]]
D = [[0.0],[0.0]]
sys4 = signal.StateSpace(A,B,C,D)
t4,y4 = signal.step(sys4)

import matplotlib.pyplot as plt
plt.figure(1)
plt.subplot(4,1,1)
plt.plot(t1,y1,'r-', linewidth=3)
plt.ylabel('Problem 1')
plt.legend(['y'], loc='best')
plt.subplot(4,1,2)
plt.plot(t2,y2,'b--', linewidth=3)
plt.ylabel('Problem 2')
plt.legend(['y'], loc='best')
plt.subplot(4,1,3)
plt.plot(t3,y3,'k-', linewidth=3)
plt.plot(t,u,'r-')
plt.ylabel('Problem 3')
plt.legend(['y', 'u'], loc='best')
plt.subplot(4,1,4)
plt.plot(t4,y4[:,0]+2.0,'r--', linewidth=3)
plt.plot(t4,y4[:,1]+2.0,'b-', linewidth=3)
plt.legend(['y1', 'y2'], loc='best')
plt.ylabel('Problem 4')
plt.xlabel('Time')
plt.show()
```

# Exercise – Result

