Name: Sahil T Chaudhary
Andrew ID: stchaudh

Modern Control Theory

HW-1

Exercise 1 :-

1. $y(t) = 0$

The system is linear & time invariant as it does not depend on any input $u(t)$.

2. $y(t) = u^3(t)$

Linearity — $y_1 = u_1^3$
$y_2 = u_2^3$

$\therefore u_3 = \alpha u_1 + \beta u_2$

To prove: $y_3 = \alpha y_1 + \beta y_2$

$$(\alpha u_1 + \beta u_2)^3 \neq \alpha u_1^3 + \beta u_2^3$$

$\therefore$ $\boxed{\text{Not linear}}$

Time invariance — Delay in input
$u_d(t) = u(t-\tau)$

$$y_1 = y(t) = u_d^3(t)$$
$$= u^3(t-\tau)$$

Delay in output

$$y_2 = y(t-\tau) = u^3(t-\tau)$$

$\therefore y_1 = y_2 \Rightarrow$ $\boxed{\text{time invariant}}$

3. $y(t) = u(3t)$

Linearity — $y_1 = u_1(3t)$
$y_2 = u_2(3t)$

$u_3 = \alpha u_1 + \beta u_2$

T.P. : $y_3 = \alpha y_1 + \alpha y_2$

$\alpha u_1 + \beta u_2 = \alpha u_1 + \beta u_2$

$\therefore$ $\boxed{\text{Linear}}$

Time invariance — I/p delay,

$y_1 = y(t) = u(3t - \tau)$

O/P delay,

$y_2 = y(t-\tau) = u(3t - 3\tau)$

$\therefore$ $\boxed{\text{time varying}}$

4. $y(t) = e^{-t} u(t-T)$

Linearity — $y_1 = e^{-t} u_1(t-T)$
$y_2 = e^{-t} u_2(t-T)$

$u_3 = \alpha u_1 + \beta u_2$

T.P. : $y_3 = \alpha y_1 + \beta y_2$

$$e^{-t}(\alpha u_1 + \beta u_2)(t-T) \neq \alpha e^{-t} u_1(t-T) + \beta e^{-t} u_2(t-T)$$

$\therefore$ $\boxed{\text{not linear}}$

Time invariance – I/P delay,

$$y_1 = y(t) = e^{-t} u(t-T-\tau)$$

O/P delay,

$$y_2 = y(t-\tau) = e^{-(t-\tau)} u(t-T-\tau)$$

$\therefore y_1 \neq y_2$

$\therefore$ $\boxed{\text{time varying}}$

5. $$y(t) = \begin{cases} 0 & , t \le 1 \\ u(t) & , t > 0 \end{cases}$$

linearity –

<u>Case I</u> :- Both inputs are at $t \le 0$

y is linear as it doesn't depend on $u(t)$

Case II :-

Both inputs are at $t > 0$

$$y_3 = \alpha y_1 + \alpha y_2$$

$$\alpha u_1 + \beta u_2 = \alpha u_1 + \beta u_2$$

$\therefore$ linear

Case III :-

One input at $t \leq 0$ & one at $t > 0$

$$y_1 = 0$$

$$y_2 = u(t)$$

$$u_3 = \alpha u_1(t) + \beta u_2(t)$$

T.P. - $y_3 = \alpha y_1 + \beta y_2$

$$\alpha u_1 + \beta u_2 \neq \beta y_2$$

$\therefore$ $\boxed{\text{not linear}}$

Since at $t \leq 0$ the input $u(t)$ may still exist.

Time invariance -

For $t \leq 0$, it is time invariant as doesn't depend on input.

For $t > 0$, I/p delay,

$$y_1 = y(t) = u(t-\tau)$$

o/p delay,

$$y_2 = y(t-\tau) = u(t-\tau)$$

$$\therefore y_1 = y_2$$

$\therefore$ $\boxed{\cancel{\text{time invariant}}}$

Since the ~~pn~~ o/p itself changes with t, it is time varying.

Exercise 2 :-

1. $$s_i = G_{ii} p_i$$

$$q_i = \sigma^2 + \sum_{j \neq i} G_{ij} p_j$$

$$S_i = \frac{s_i}{q_i} = \frac{G_{ii} p_i}{\sigma^2 + \sum_{j \neq i} G_{ij} p_j} \quad \alpha \gamma$$

Controller,

$$p_i(k+1) = p_i(k) \frac{\alpha \gamma}{S_i(k)}$$

Now, $$p_i(k+1) = \frac{s_i}{G_{ii}} \frac{\alpha \gamma}{s_i} \times q_i$$

$$= \frac{\alpha \gamma \sigma^2}{G_{ii}} + \frac{\alpha \gamma}{G_{ii}} \sum G_{ij} p_j$$

$$\therefore \mu_i(k+1) = A\mu(k) + B\sigma^2$$

$$\begin{bmatrix} \mu_1(k+1) \\ \mu_2(k+1) \\ \mu_3(k+1) \end{bmatrix} = \alpha\gamma \begin{bmatrix} 0 & G_{12}/G_{11} & G_{13}/G_{11} \\ G_{21}/G_{22} & 0 & G_{23}/G_{22} \\ G_{31}/G_{33} & \frac{G_{32}}{G_{33}} & 0 \end{bmatrix} \begin{bmatrix} \mu_1(k) \\ \mu_2(k) \\ \mu_3(k) \end{bmatrix}$$

$$+ \alpha\gamma \begin{bmatrix} 1/G_{11} \\ 1/G_{22} \\ 1/G_{33} \end{bmatrix} \sigma^2$$

$$\therefore A = \begin{bmatrix} 0 & \alpha\gamma\, G_{12}/G_{11} & \alpha\gamma\, G_{13}/G_{11} \\ G_{21}/G_{22} \times \alpha\gamma & 0 & \alpha\gamma\, G_{23}/G_{22} \\ G_{31}/G_{33} \times \alpha\gamma & \alpha\gamma\, G_{32}/G_{33} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \alpha\gamma/G_{11} \\ \alpha\gamma/G_{22} \\ \alpha\gamma/G_{33} \end{bmatrix}$$

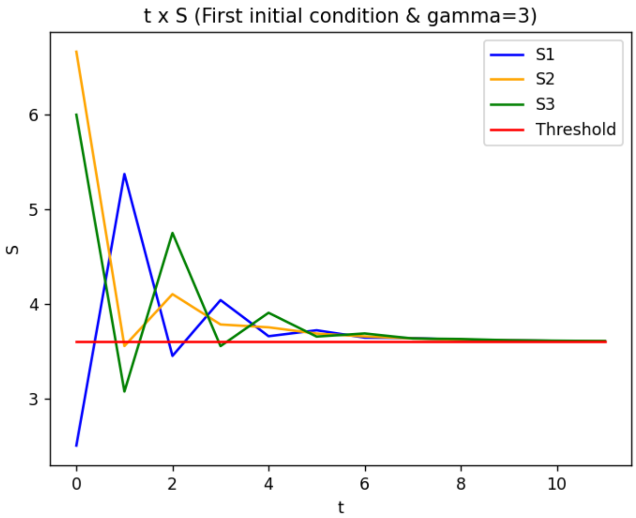
EXERCISE 2:

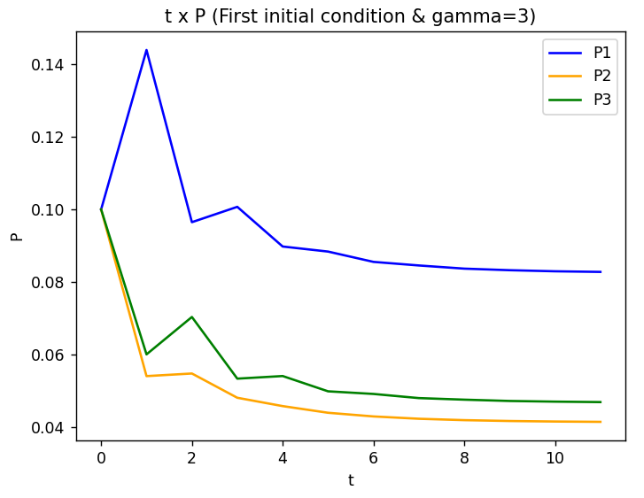
2.

- p1 = p2 = p3 = 0.1 and gamma = 3

```
import numpy as np
import matplotlib.pyplot as plt

# With gamma = 3
G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
gamma = 3
alpha = 1.2
sigma = 0.1**2

t = np.arange(0,12,1)

# First initial condition
p_init = np.array([[0.1],[0.1],[0.1]])

A = alpha*gamma*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
                           [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
                           [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])

B = alpha*gamma*np.array([[1/G[0][0]],
                          [1/G[1][1]],
                          [1/G[2][2]]])

P1arr = []
P2arr = []
P3arr = []
S1arr = []
S2arr = []
S3arr = []

p=p_init

for x in range(0,12):
    q1 = sigma + (0.2*p[1]) + (0.1*p[2])
    q2 = sigma + (0.1*p[0]) + (0.1*p[2])
    q3 = sigma + (0.3*p[0]) + (0.1*p[1])
    S = np.array([G[0][0]*p[0]/q1, G[1][1]*p[1]/q2, G[2][2]*p[2]/q3])
    P1arr.append(p[0])
    P2arr.append(p[1])
    P3arr.append(p[2])
    S1arr.append(S[0])
    S2arr.append(S[1])
    S3arr.append(S[2])
    p = np.dot(A,p) + np.dot(B,sigma)

target = [alpha*gamma]*12

plt.plot(t,S1arr, color='b', label='S1')
plt.plot(t,S2arr, color='orange', label='S2')
plt.plot(t,S3arr, color='g', label='S3')
```

```
plt.plot(t,target, color='r', label='Threshold')
plt.xlabel("t")
plt.ylabel("S")
plt.title("t x S (First initial condition & gamma=3)")
plt.legend()
plt.show()

plt.plot(t,P1arr, color='b', label='P1')
plt.plot(t,P2arr, color='orange', label='P2')
plt.plot(t,P3arr, color='g', label='P3')

plt.xlabel("t")
plt.ylabel("P")
plt.title("t x P (First initial condition & gamma=3)")
plt.legend()
plt.show()
```

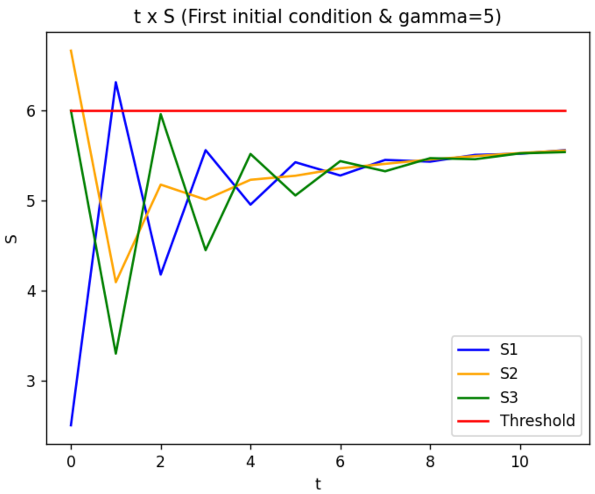


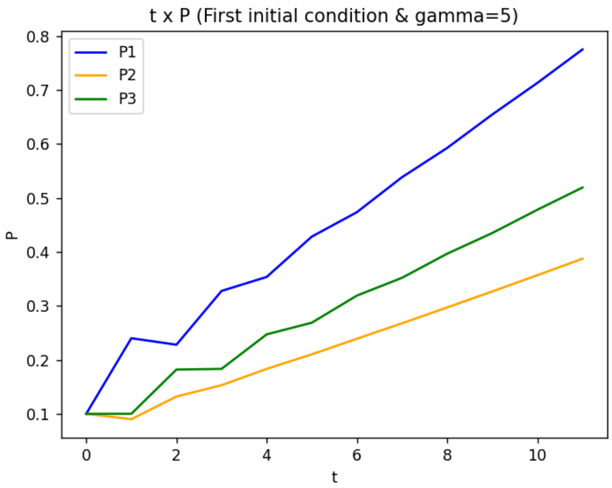


- p1 = p2 = p3 = 0.1 and gamma = 5

```
import numpy as np
import matplotlib.pyplot as plt

# With gamma = 5
G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
gamma = 5
alpha = 1.2
sigma = 0.1**2

t = np.arange(0,12,1)

# First initial condition
p_init = np.array([[0.1],[0.1],[0.1]])

A = alpha*gamma*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
                          [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
                          [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])

B = alpha*gamma*np.array([[1/G[0][0]],
                         [1/G[1][1]],
```

```
                            [1/G[2][2]]])

P1arr = []
P2arr = []
P3arr = []
S1arr = []
S2arr = []
S3arr = []

p=p_init

for x in range(0,12):
    q1 = sigma + (0.2*p[1]) + (0.1*p[2])
    q2 = sigma + (0.1*p[0]) + (0.1*p[2])
    q3 = sigma + (0.3*p[0]) + (0.1*p[1])
    S = np.array([G[0][0]*p[0]/q1, G[1][1]*p[1]/q2, G[2][2]*p[2]/q3])
    P1arr.append(p[0])
    P2arr.append(p[1])
    P3arr.append(p[2])
    S1arr.append(S[0])
    S2arr.append(S[1])
    S3arr.append(S[2])
    p = np.dot(A,p) + np.dot(B,sigma)

target = [alpha*gamma]*12

plt.plot(t,S1arr, color='b', label='S1')
plt.plot(t,S2arr, color='orange', label='S2')
plt.plot(t,S3arr, color='g', label='S3')
plt.plot(t,target, color='r', label='Threshold')
plt.xlabel("t")
plt.ylabel("S")
plt.title("t x S (First initial condition & gamma=5)")
plt.legend()
plt.show()

plt.plot(t,P1arr, color='b', label='P1')
plt.plot(t,P2arr, color='orange', label='P2')
plt.plot(t,P3arr, color='g', label='P3')

plt.xlabel("t")
plt.ylabel("P")
plt.title("t x P (First initial condition & gamma=5)")
plt.legend()
plt.show()
```

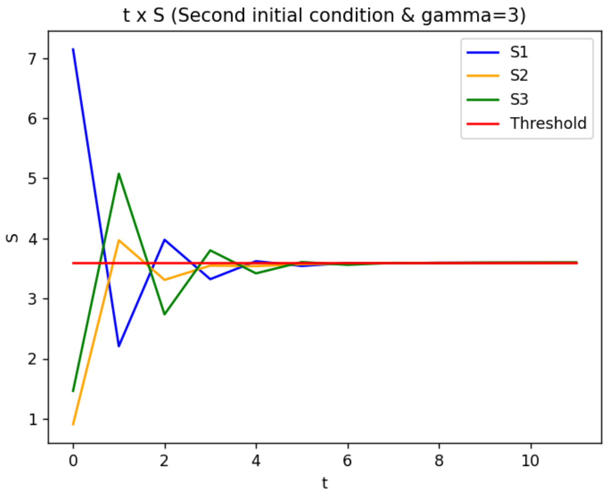


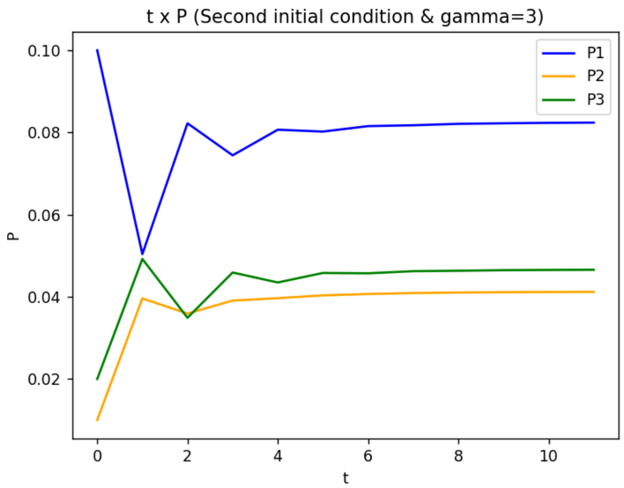


- p1 = 0.1, p2 = 0.01, p3 = 0.02 and gamma = 3

```
import numpy as np
import matplotlib.pyplot as plt

# With gamma = 3
G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
gamma = 3
alpha = 1.2
sigma = 0.1**2

t = np.arange(0,12,1)

# Second initial condition
p_init = np.array([[0.1],[0.01],[0.02]])

A = alpha*gamma*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
                          [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
                          [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])

B = alpha*gamma*np.array([[1/G[0][0]],
                         [1/G[1][1]],
                         [1/G[2][2]]])

P1arr = []
P2arr = []
P3arr = []
S1arr = []
S2arr = []
S3arr = []

p=p_init

for x in range(0,12):
    q1 = sigma + (0.2*p[1]) + (0.1*p[2])
    q2 = sigma + (0.1*p[0]) + (0.1*p[2])
    q3 = sigma + (0.3*p[0]) + (0.1*p[1])
    S = np.array([G[0][0]*p[0]/q1, G[1][1]*p[1]/q2, G[2][2]*p[2]/q3])
    P1arr.append(p[0])
    P2arr.append(p[1])
```

```
    P3arr.append(p[2])
    S1arr.append(S[0])
    S2arr.append(S[1])
    S3arr.append(S[2])
    p = np.dot(A,p) + np.dot(B,sigma)

target = [alpha*gamma]*12

plt.plot(t,S1arr, color='b', label='S1')
plt.plot(t,S2arr, color='orange', label='S2')
plt.plot(t,S3arr, color='g', label='S3')
plt.plot(t,target, color='r', label='Threshold')
plt.xlabel("t")
plt.ylabel("S")
plt.title("t x S (Second initial condition & gamma=3)")
plt.legend()
plt.show()

plt.plot(t,P1arr, color='b', label='P1')
plt.plot(t,P2arr, color='orange', label='P2')
plt.plot(t,P3arr, color='g', label='P3')

plt.xlabel("t")
plt.ylabel("P")
plt.title("t x P (Second initial condition & gamma=3)")
plt.legend()
plt.show()
```

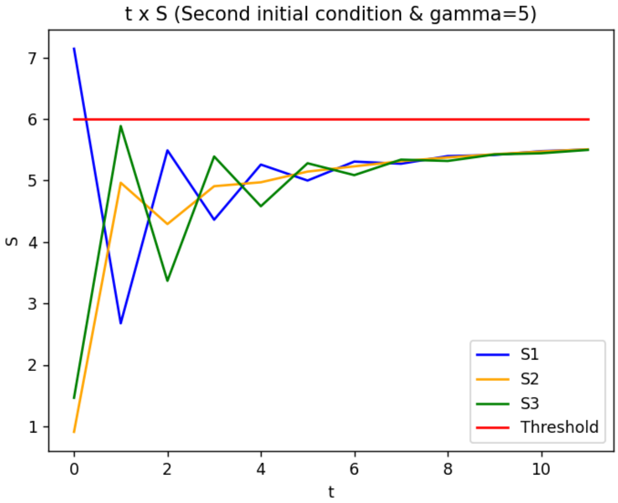


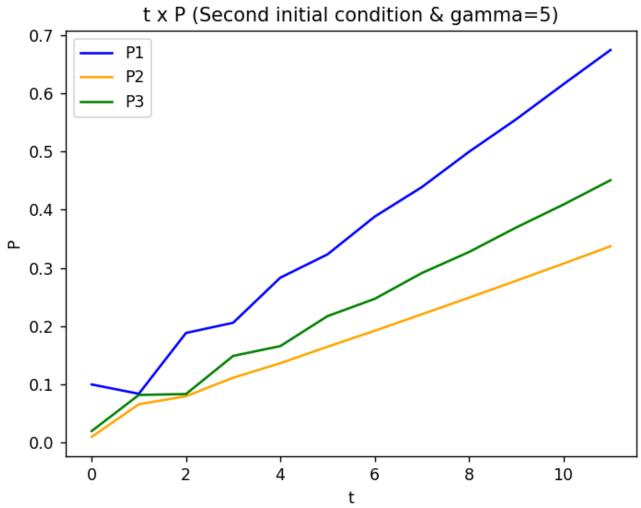


- p1 = 0.1, p2 = 0.01, p3 = 0.02 and gamma = 5

```
import numpy as np
import matplotlib.pyplot as plt

# With gamma = 5
G = np.array([[1, 0.2, 0.1],[0.1, 2, 0.1],[0.3, 0.1, 3]])
gamma = 5
alpha = 1.2
sigma = 0.1**2

t = np.arange(0,12,1)

```

```python
# Second initial condition
p_init = np.array([[0.1],[0.01],[0.02]])

A = alpha*gamma*np.array([[0, G[0][1]/G[0][0], G[0][2]/G[0][0]],
                          [G[1][0]/G[1][1], 0, G[1][2]/G[1][1]],
                          [G[2][0]/G[2][2], G[2][1]/G[2][2], 0]])

B = alpha*gamma*np.array([[1/G[0][0]],
                         [1/G[1][1]],
                         [1/G[2][2]]])

P1arr = []
P2arr = []
P3arr = []
S1arr = []
S2arr = []
S3arr = []

p=p_init

for x in range(0,12):
    q1 = sigma + (0.2*p[1]) + (0.1*p[2])
    q2 = sigma + (0.1*p[0]) + (0.1*p[2])
    q3 = sigma + (0.3*p[0]) + (0.1*p[1])
    S = np.array([G[0][0]*p[0]/q1, G[1][1]*p[1]/q2, G[2][2]*p[2]/q3])
    P1arr.append(p[0])
    P2arr.append(p[1])
    P3arr.append(p[2])
    S1arr.append(S[0])
    S2arr.append(S[1])
    S3arr.append(S[2])
    p = np.dot(A,p) + np.dot(B,sigma)

target = [alpha*gamma]*12

plt.plot(t,S1arr, color='b', label='S1')
plt.plot(t,S2arr, color='orange', label='S2')
plt.plot(t,S3arr, color='g', label='S3')
plt.plot(t,target, color='r', label='Threshold')
plt.xlabel("t")
plt.ylabel("S")
plt.title("t x S (Second initial condition & gamma=5)")
plt.legend()
plt.show()

plt.plot(t,P1arr, color='b', label='P1')
plt.plot(t,P2arr, color='orange', label='P2')
plt.plot(t,P3arr, color='g', label='P3')

plt.xlabel("t")
plt.ylabel("P")
plt.title("t x P (Second initial condition & gamma=5)")
plt.legend()
plt.show()
```





HENCE, THE CONTROLLER CAN ACHIEVE THE GOAL OF REACHING ALPHA=GAMMA FOR GAMMA = 3, BUT NOT FOR GAMMA = 5.

Exercise 3:-

$$\ddot{y} + (1+y)\dot{y} - 2y + 0.5y^3 = 0$$

Considering state variables as,

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$$

$$\therefore \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} x_2 \\ -(1+x_1)x_2 + 2x_1 - 0.5x_1^3 \end{bmatrix}$$

For equilibrium pts.

$$x_2 = 0$$

$$-(1+x_1)x_2 + 2x_1 - 0.5x_1^3 = 0$$

$$x_1(2 - 0.5x_1^2) = 0$$

$$\therefore x_1 = 0 \quad \& \quad x_1 = \pm 2$$

Now, linearising

$$\delta\dot{x} = \begin{bmatrix} 0 & 1 \\ -x_2 + 2 - 1.5x_1^2 & -1 - x_1 \end{bmatrix} \delta x$$

at $(0, 0)$

$$\dot{\delta x} = \begin{bmatrix} 0 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

at $(2, 0)$

$$\dot{\delta x} = \begin{bmatrix} 0 & 1 \\ -4 & -3 \end{bmatrix} \begin{bmatrix} x_1 - 2 \\ x_2 \end{bmatrix}$$

at $(-2, 0)$

$$\dot{\delta x} = \begin{bmatrix} 0 & 1 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} x_1 + 2 \\ x_2 \end{bmatrix}$$

Exercise 4:—

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -gD^2\left(\frac{1}{x_1+D}\right)^2 + \frac{\ln(u)}{m}$$

For equilibrium pts;

$$\dot{x}_1 = 0 \Rightarrow x_2 = 0$$

$$\dot{x}_2 = 0 \Rightarrow -gD^2\left(\frac{1}{x_1+D}\right)^2 + \frac{\ln(u)}{m} = 0$$

$$\therefore x_1 = \pm\sqrt{\frac{mgD^2}{\ln(u)}} - D$$

Now linearising,

$$\delta\dot{x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \delta x$$

$$\therefore \begin{bmatrix} \delta\dot{x}_1 \\ \delta\dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-gD^2}{(x_1+D)^3} & 0 \end{bmatrix} \begin{bmatrix} x \pm\sqrt{\frac{mgD^2}{\ln(u)}} + D \\ x - 0 \end{bmatrix}$$

$$= \begin{bmatrix} x \\ \frac{-gD^2}{(x_1+D)^3}\left(x \pm\sqrt{\frac{mgD^2}{\ln(u)}} + D\right) \end{bmatrix}$$

Exercise 5 :-

1. $\ddot{r} = r\dot{\theta}^2 - \frac{k}{r^2} + u_1$

$\ddot{\theta} = -2\frac{\dot{\theta}}{r}\dot{r} + \frac{u_2}{r}$

on reference orbit,

$\ddot{r} = p(\omega)^2 - \frac{k}{p^2}$

$\ddot{\theta} = -2\frac{\omega}{p} \times 0$

$\therefore p\omega^2 - \frac{k}{p^2} = 0$

$\boxed{\therefore k = p^3\omega^2}$

Let

2. $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} r \\ \dot{r} \\ \theta \\ \dot{\theta} \end{bmatrix}$

$$\therefore \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \ddot{r} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 x_4^2 - k/x_1^2 + u_1 \\ x_4 \\ -2\frac{x_4}{x_1} x_2 + \frac{u_2}{x_1} \end{bmatrix}$$

Now, taylor Series expansion

$$\delta \dot{x} = f(x,u) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial u}\delta u$$

$$= \begin{bmatrix} x_2 \\ x_1 x_4^2 - k/x_1^2 + u_1 \\ x_4 \\ -2\frac{x_4}{x_1} x_2 + \frac{u_2}{x_1} \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 1 & 0 & 0 \\ x_4^2 + \frac{k}{x_1^3} & 0 & 0 & 2x_1x_4 \\ 0 & 0 & 0 & 1 \\ \left(2\frac{x_4 x_2}{x_1^2} - \frac{u_2}{x_1^2}\right) & -\frac{2x_4}{x_1} & 0 & -2\frac{x_2}{x_1} \end{bmatrix} \delta x$$

$$+ \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1/x_1 \end{bmatrix} \delta u$$

$\because u_1 = u_2 = 0$

$$\delta \dot{x} = \begin{bmatrix} & x_2 \\ x_1 x_4^2 & - k/x_1^2 \\ & x_4 \\ -\frac{2x_4}{x_1} x_2 & \end{bmatrix}$$

$$+ \begin{bmatrix} 0 & 1 & 0 & 0 \\ x_4^2 + \frac{k}{x_1^3} & 0 & 0 & 2x_1x_4 \\ 0 & 0 & 0 & 1 \\ \frac{2x_4x_2}{x_1^2} & -\frac{2x_4}{x_1} & 0 & -\frac{2x_2}{x_1} \end{bmatrix} \delta x$$

$$+ \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1/x_1 \end{bmatrix} \delta u$$