Homework 1

24-677 Special Topics: Linear Control Systems

Prof. D. Zhao

Due: Sept 10, 2020, 9:50 am. Submit within deadline.

- All assignments will be submitted through Gradescope. Your online version and its timestamp will be used for assessment. Gradescope is a tool licensed by CMU and integrated with Canvas for easy access by students and instructors. When you need to complete a Gradescope assignment, here are a few easy steps you will take to prepare and upload your assignment, as well as to see your assignment status and grades. Take a look at Q&A about Gradescope to understand how to submit and monitor HW grades. https://www.cmu.edu/teaching//gradescope/index.html

- You will need to upload your solution in .pdf to Gradescope (either scanned handwritten version or LaTeX or other tools). If you are required to write Python code, upload the code to Gradescope as well.

- Grading: The score for each question or sub-question is discrete with three outcomes: fully correct (full score), partially correct/unclear (half the score), and totally wrong (zero score).

- Regrading: please review comments from TAs when the grade is posted and make sure no error in grading. If you find a grading error, you need to inform the TA as soon as possible but no later than a week from when your grade is posted. The grade may NOT be corrected after 1 week.

- At the start of every exercise you will see topic(s) on what the given question is about and what will you be learning.

- We advise you to start with the assignment early. All the submissions are to be done before the respective deadlines of each assignment. For information about the late days and scale of your Final Grade, refer to the Syllabus in Canvas.

Exercise 1. *Types of Systems* ***(20 points)***

A system has an input $u(t)$ and an output $y(t)$, which are related by the information provided below. Classify each system as linear or non-linear and time invariant or time-varying, and explain why.

1. $y(t) = 0$ for all t **(4 points)**
2. $y(t) = u^3(t)$ **(4 points)**
3. $y(t) = u(3t)$ **(4 points)**
4. $y(t) = e^{-t}u(t-T)$ **(4 points)**
5. $y(t) = \begin{cases} 0 & t \leq 0 \\ u(t) & t > 0 \end{cases}$ **(4 points)**

Solution:

1. $y(t) = 0$ for all t
   $y_1(t) = 0$ for $u_1(t)$ input
   $y_2(t) = 0$ for $u_2(t)$ input
   $y_3(t) = \alpha \cdot 0 + \beta \cdot 0$ for input:
   $\alpha u_1(t) + \beta u_2(t)$
   Hence it is **linear**

   $y(t+\tau) = 0$
   output for $u(t+\tau) = 0$
   Hence **time invariant**

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

   (a) Non-linear:
   Consider
   $$u(t) = \alpha u_1(t) + \beta u_2(t)$$
   with
   $$y_1(t) = u_1^3(t)$$
   $$y_2(t) = u_1^3(t)$$
   then the output $y(t)$ corresponding to the input $u(t)$ is
   $$\begin{aligned} y(t) &= u^3(t) \\ &= (\alpha u_1(t) + \beta u_2(t))^3 \\ &= \alpha^3 u_1^3(t) + \beta^3 u_2^3(t) + 3\alpha^2\beta u_1^2(t)u_2(t) + 3\alpha\beta^2 u_1(t)u_2^2(t) \\ &\neq \alpha y_1(t) + \beta y_2(t) \end{aligned}$$

(b) Time-Invariant:

$$D_\tau\{u(t)\} = u^3(t-\tau) \tag{1}$$
$$D_\tau\{y(t)\} = y(t-\tau) \tag{2}$$
$$= u^3(t-\tau) \tag{3}$$
$$= D_\tau\{u(t)\} \tag{4}$$

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

(a) Linear:
Consider

$$u(3t) = \alpha u_1(3t) + \beta u_2(3t)$$

with

$$y_1(t) = u_1(3t)$$
$$y_2(t) = u_1(3t)$$

then the output $y(t)$ corresponding to the input $u(3t)$ is

$$\begin{aligned} y(t) &= u(3t) \\ &= \alpha u_1(3t) + \beta u_2(3t) \\ &= \alpha y_1(t) + \beta y_2(t) \end{aligned}$$

(b) Time-Varying:

$$D_\tau\{u(3t)\} = u(3t-\tau) \tag{1}$$
$$D_\tau\{y(t)\} = u(3(t-\tau)) \tag{2}$$
$$= u(3t-3\tau)) \tag{3}$$
$$\neq D_\tau\{u(t)\} \tag{4}$$

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

(a) Linear:
Consider

$$u(t-T) = \alpha u_1(t-T) + \beta u_2(t-T)$$

with

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

then the output $y(t)$ corresponding to the input is

$$\begin{aligned} y(t) &= e^{-t}u(t-T) \\ &= e^{-t}\left(\alpha u_1(t-T) + \beta u_2(t-T)\right) \\ &= \alpha y_1(t) + \beta y_2(t) \end{aligned}$$

(b) Time Varying:

$$D_\tau\{e^{-t}u(t)\} = e^{-t}u(t-\tau-T) \tag{1}$$
$$D_\tau\{y(t)\} = e^{-(t-\tau)}u(t-\tau-T) \tag{2}$$
$$\neq D_\tau\{u(t)\} \tag{3}$$

5. $y(t) = \begin{cases} 0, & t < 0 \\ u(t) & t \geq 0 \end{cases}$

(a) Linear:
Consider

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

with

$$y_1(t) = \begin{cases} 0, & t < 0 \\ u_1(t) & t \geq 0 \end{cases}$$
$$y_2(t) = \begin{cases} 0, & t < 0 \\ u_2(t) & t \geq 0 \end{cases}$$

then the output $y(t)$ corresponding to the input is

$$\begin{aligned} y(t) &= \begin{cases} 0, & t < 0 \\ u(t) & t \geq 0 \end{cases} \\ &= \begin{cases} 0, & t < 0 \\ \alpha u_1(t) + \beta u_2(t) & t \geq 0 \end{cases} \\ &= \alpha y_1(t) + \beta y_2(t) \end{aligned}$$

(b) Time-Varying: $D_\tau\{u(t)\} = \begin{cases} 0, & t < 0 \\ u(t-\tau) & t \geq 0 \end{cases}$

$D_\tau\{y(t)\} = \begin{cases} 0, & t < \tau \\ u(t-\tau) & t \geq \tau \end{cases}$

$\neq D_T\{u(t)\}$

Exercise 2. *State space representations* ***(30 points)***

A company deployed 3 teams of drones in a region. Each team consists of a pair of drones. One drone in the team carries a transmitter and the other one carries a receiver. Transmitter i transmits at power level p_i ($p_i > 0$). The path gain from transmitter j to receiver i is G_{ij} ($G_{ij} \geqslant 0$ for $j \neq i$, and $G_{ii} > 0$). The signal power at receiver i is given by $s_i = G_{ii}p_i$. The noise plus interference power (caused by other transmitters $j \neq i$) at receiver i is given by

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

where $\sigma^2 > 0$ is the self-noise power of the receivers.

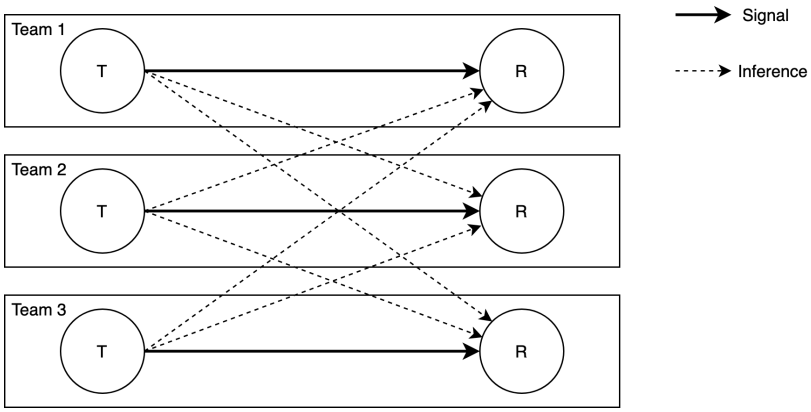

Figure 1: The wireless network

The signal to interference plus noise ratio (SINR) at receiver i is defined as $S_i = s_i/q_i$. For signal reception to occur, the SINR must exceed some threshold value γ (*i.e.*, $S_i \geq \gamma$). We assume p, q and S are discrete-time signals. For example, $p_i(k)$ represents the transmit power of transmitter i at time k ($k = 0, 1, 2, ...$). We want to have a certain SINR, e.g.

$$S_i(k) = s_i(k)/q_i(k) = \alpha\gamma,$$

where $\alpha > 1$ is an SINR safety margin. To achieve this goal, someone designed the following control rule

$$p_i(k+1) = p_i(k)\left(\alpha\gamma/S_i(k)\right).$$

1. Show that the power control update algorithm can be expressed as a linear dynamical system with constant input, *i.e.*, in the form

   $$p(k+1) = Ap(k) + B\sigma^2,$$

   where $A \in \mathbb{R}^{3\times3}$ and $B \in \mathbb{R}^{3\times1}$ are constant and $p(k) = [p_1, p_2, p_3]^T$. Describe A and b explicitly in terms of σ, γ, α and the components of G. **(10 points)**

2. Use Python to simulate the power control algorithm. Use the problem data

$$G = \begin{bmatrix} 1 & .2 & .1 \\ .1 & 2 & .1 \\ .3 & .1 & 3 \end{bmatrix}, \quad \gamma = 3, \quad \alpha = 1.2, \quad \sigma = 0.1$$

Experiment with two different initial conditions: $p_1 = p_2 = p_3 = 0.1$ and $p_1 = 0.1, p_2 = 0.01, p_3 = 0.02$. Plot S_i and p as a function of t, and compare it to the target value $\alpha\gamma$. Repeat for $\gamma = 5$. Can the controller achieve the goal to make $S_i(t) \to \alpha\gamma$? Plot all the $p_i(k)$ as well. Submit your code to Gradescope. **(20 points)**

Solution:

1. For the transmitter i, the power update rule can be represented by

$$p_i(k+1) = \frac{\alpha\gamma p_i(k)}{S_i(k)} = \frac{\alpha\gamma p_i(k) q_i(k)}{s_i(k)} = \frac{\alpha\gamma p_i(k)\left[\sigma^2 + \sum_{j\neq i} G_{ij} p_j(k)\right]}{G_{ii} p_i(k)}$$
$$= \frac{\alpha\gamma\left[\sigma^2 + \sum_{j\neq i} G_{ij} p_j(k)\right]}{G_{ii}}$$

If we write it in matrix form for n transmitter:

$$\underbrace{\begin{bmatrix} p_1(k+1) \\ p_2(k+1) \\ p_3(k+1) \end{bmatrix}}_{p(k+1)} = \underbrace{\begin{bmatrix} 0 & \frac{\alpha\gamma G_{12}}{G_{11}} & \frac{\alpha\gamma G_{13}}{G_{11}} \\ \frac{\alpha\gamma G_{21}}{G_{22}} & 0 & \frac{\alpha\gamma G_{23}}{G_{22}} \\ \frac{\alpha\gamma G_{31}}{G_{33}} & \frac{\alpha\gamma G_{32}}{G_{33}} & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} p_1(k) \\ p_2(k) \\ p_3(k) \end{bmatrix}}_{p(k)} + \underbrace{\begin{bmatrix} \frac{\alpha\gamma}{G_{11}} \\ \frac{\alpha\gamma}{G_{22}} \\ \frac{\alpha\gamma}{G_{33}} \end{bmatrix}}_{B} \sigma^2$$

2. Here is the python code for $\gamma = 3$ and initial power of 0.1 for all transmitters. You can access Colab via https://colab.research.google.com/drive/1DV13F4vYFEw0ZjA3Rx1r6-Ai7TWB6yEg?usp=sharing

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

# Constants
G = np.array([[1, .2, .1], [.1, 2, .1], [.3, .1, 3]])
GAMMA = 3
ALPHA = 1.2
SIGMA = 0.1


def calculate_SINR(p):
    n = p.shape[0]
```

```
    q = np.zeros((3,1))
    S = np.zeros((3,1)) # SINR
    for i in range(n):
        S[i] = G[i,i]*p[i]
        q[i] = SIGMA**2
        for j in range(n):
            if j != i:
                q[i] += G[i,j]*p[j]
        S[i] = S[i]/q[i]

    return S


def main():
    A = np.zeros((3,3))
    b = np.zeros((3,1))
    for i in range(3):
        b[i] = ALPHA*GAMMA*SIGMA**2/G[i,i]
        for j in range(3):
            if j != i:
                A[i,j] = ALPHA*GAMMA*G[i,j]/G[i,i]

    num_iterations = 200
    p_t = np.array([.1, .1, .1]).reshape(3,1)  # initial condition
    p = p_t  # array to store the history of power
    S = calculate_SINR(p_t)

    for _ in range(num_iterations):
        p_t = A.dot(p_t) + b
        S_t = calculate_SINR(p_t)
        p = np.hstack((p, p_t))
        S = np.hstack((S, S_t))

    fig = plt.figure(figsize=(10,8), dpi=200)
    ax1 = plt.subplot(2, 1, 1)
    ax1.title.set_text(r"$\gamma=5$, initial powers = [.1, .1, .1]")
    ax1.plot(range(num_iterations+1), p[0,:], '-', label="Transmitter 1")
    ax1.plot(range(num_iterations+1), p[1,:], '--', label="Transmitter 2")
    ax1.plot(range(num_iterations+1), p[2,:], '-.', label="Transmitter 3")
    ax1.legend(loc="upper right")
    ax1.set_ylabel('Transmitter Power', fontsize=12)
    ax1.set_xlabel('Time Step k', fontsize=12)
    # ax1.yaxis.set_major_locator(ticker.MultipleLocator(0.02))
    ax1.grid()
```

```
    ax2 = plt.subplot(2, 1, 2)
    ax2.title.set_text(r"$\gamma=5$, initial powers = [.1, .1, .1]")
    ax2.plot(range(num_iterations+1), S[0,:], '-', label="Transmitter 1")
    ax2.plot(range(num_iterations+1), S[1,:], '--', label="Transmitter 2")
    ax2.plot(range(num_iterations+1), S[2,:], '-.', label="Transmitter 3")
    ax2.plot(range(num_iterations+1), np.ones((num_iterations+1))*ALPHA*GAMMA,
        'k--', label="Target Value")
    ax2.legend(loc="upper right")
    ax2.set_ylabel('SINR', fontsize=12)
    ax2.set_xlabel('Time Step k', fontsize=12)
    ax2.yaxis.set_major_locator(ticker.MultipleLocator(1))
    ax2.grid()

    plt.tight_layout()
    plt.show()


if __name__ == "__main__":
    main()
```

We can observe that the update algorithm seems to work for $\gamma = 3$. The SINR approaches $\alpha\gamma = 3.6$ for both initial powers tried.

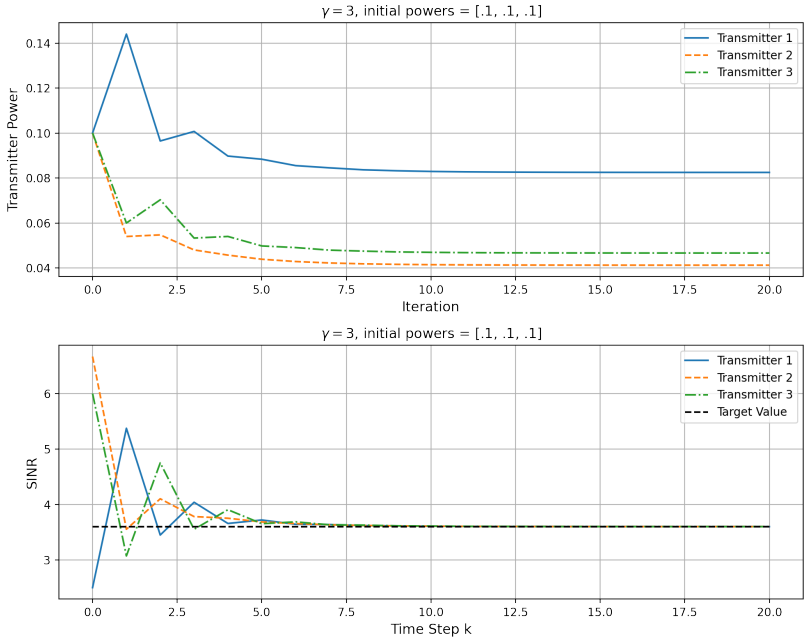

Figure 2: $\gamma = 3$, initial power = [.1; .1; .1]

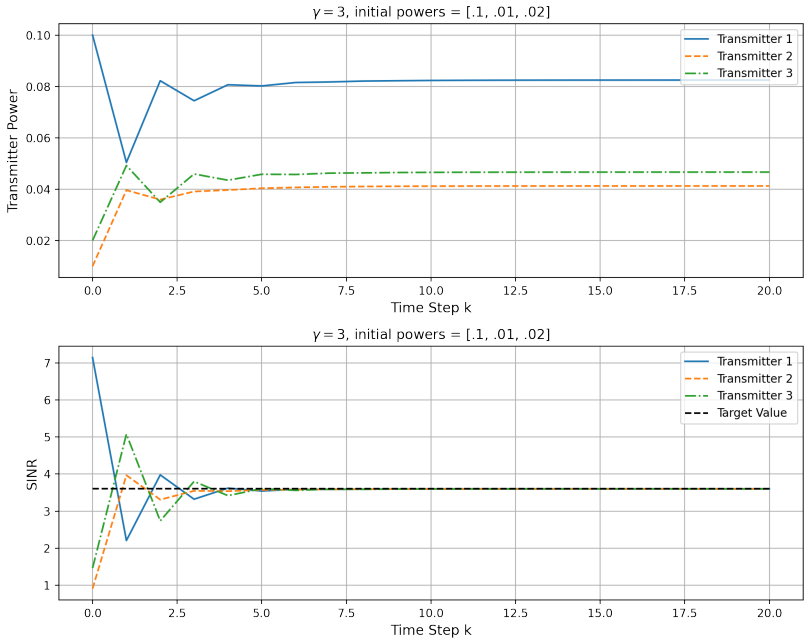

Figure 3: $\gamma = 3$, initial power = [.1; .01; .02]

However, for $\gamma = 5$, the algorithm does not always work. The transmit powers grow exponentially. SINR is always below $\alpha\gamma = 6$.

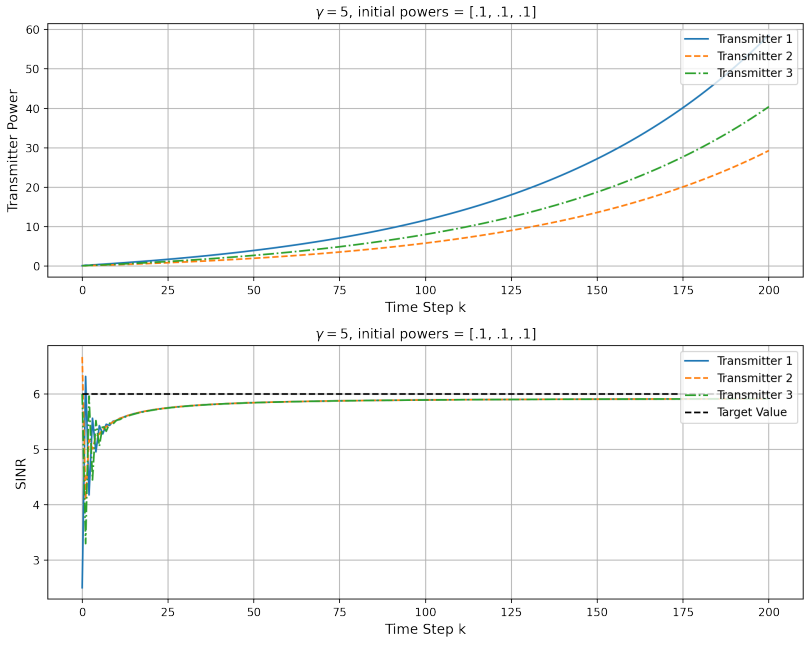

Figure 4: $\gamma = 5$, initial power = [.1; .1; .1]

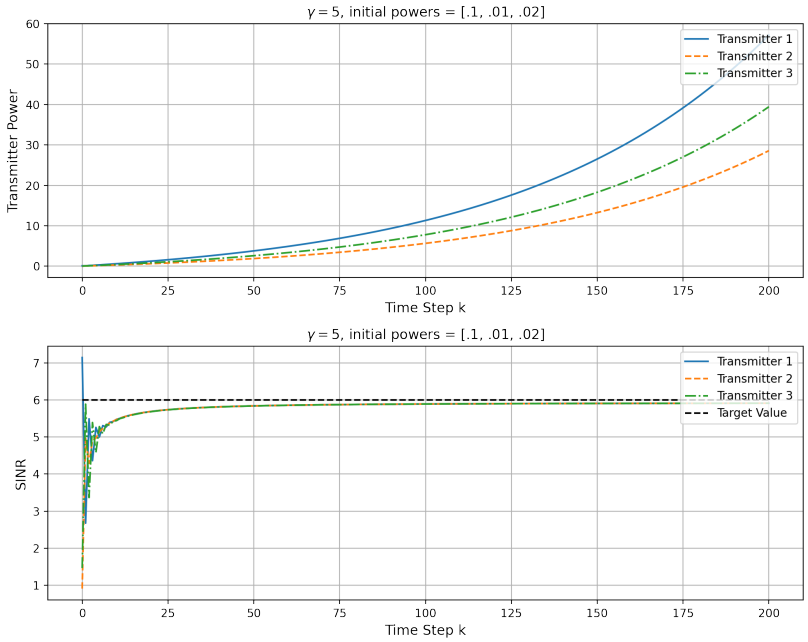

Figure 5: $\gamma = 5$, initial power = [.1; .01; .02]

Exercise 3. *Linearization* ***(15 points)***

Perform linearization on the given differential equation

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

Solution: Let $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$, also $x_1 = y$ and $x_2 = \dot{y}$ gives the state variable model

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} x_2 \\ 2x_1 - 0.5{x_1}^3 - (1+x_1)x_2 \end{bmatrix} = f(x)$$

Equilibrium points are solutions of $f(x) = 0$, so each must have $x_2 = 0$ and $2x_1 - 0.5{x_1}^3 = 0$, three solutions are

$$x_{e1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad x_{e2} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad x_{e3} = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$$

and we have the Jacobian matrix

$$\begin{bmatrix} 0 & 1 \\ 2 - \frac{3{x_1}^2}{2} - x_2 & -(1+x_1) \end{bmatrix}$$

such that at (0,0), (2,0) and (-2,0)

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

Exercise 4. *Equilibrium* ***(10 points)***

The simplified dynamics of the vertical ascent of a Space X rocket can be modeled as

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -g(\frac{D}{x_1(t)+D})^2 + \frac{\ln(u)}{m} \end{bmatrix}$$

where D is the distance from earth to the surface of the rocket, m is the actual mass of the rocket, g is the gravity constant, and u is the thrust. During a short period of time, we can assume D, m, g, u are all constant. $ln(*)$ is the natural logarithmic,

Find the equilibrium states (x_1^*, x_2^*) of the above dynamic system. Perform linearization on the system.

Solution:
Given the state variable model

$$\begin{bmatrix} \dot{x_1}(t) \\ \dot{x_2}(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -g(\frac{D}{x_1(t)+D})^2 + \frac{\ln(u)}{m} \end{bmatrix} = f(x)$$

Equilibrium points are solutions of $f(x) = 0$, so we must have $x_2(t) = 0$ and $-g(\frac{D}{x_1(t)+D})^2 + \frac{\ln(u)}{m} = 0$. The solution is

$$\begin{gathered} x_1(t) = \frac{\sqrt{mg}D}{\sqrt{\ln(u)}} - D \\ x_2(t) = 0 \end{gathered}$$

The Jacobian of the system is

$$J = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ 2gD^2(x_1(t)+D)^{-3} & 0 \end{bmatrix}$$

$$\left.\frac{\partial f}{\partial x}\right|_{eq} = \begin{bmatrix} 0 & 1 \\ 2\frac{ln(u)\sqrt{ln(u)}}{m\sqrt{mg}D} & 0 \end{bmatrix}$$

Exercise 5. *Linearization* ***(25 points)***

Model the earth and a satellite as particles. The normalized equations of motion, in an earth-fixed inertial frame, simplified to 2 dimensions (from Lagrange's equations of motion, the Lagrangian $L = T - V = \frac{1}{2}\dot{r}^2 + \frac{1}{2}r^2\dot{\theta}^2 - \frac{k}{r}$, where r is the radius of the trajectory of the satellite, θ is the angle, k is the Newtonian constant:

$$\ddot{r} = r\dot{\theta}^2 - \frac{k}{r^2} + u_1$$
$$\ddot{\theta} = -2\frac{\dot{\theta}}{r}\dot{r} + \frac{u_2}{r}$$

with $u1$, $u2$ are control input, representing the radial and tangential forces due to the thrusters. The reference orbit with $u1 = u2 = 0$ is circular with $r(t) \equiv p$ and $\theta(t) = \omega t$, where p is a representing the constant cruise radius, ω is the constant angular velocity of the satellite.

1. What's the value of k expressed in terms of p and w, when the satellite is on the reference orbit? **(10 points)**

2. Obtain the linearized equation about this orbit. (Hint: we linearize on a trajectory, not a equilibrium point, so $\dot{x} \neq 0$) **(15 points)**

Solution:

1. When the satellite is on the reference orbit, $u1 = u2 = 0$ and $\ddot{r} = 0$. From the first equation, we have:

$$0 = r\dot{\theta}^2 - \frac{k}{r^2}$$
$$k = r^3\dot{\theta}^2$$

   Since $\theta(t) = \omega t$, $\dot{\theta} = \omega$, and

$$k = p^3\omega^2$$

2. Choose r, θ, $\dot{r}$ and $\dot{\theta}$ as the states of the system. The state is given by

$$x = \begin{bmatrix} r \\ \theta \\ \dot{r} \\ \dot{\theta} \end{bmatrix}$$

   The system is given by

$$\dot{x} = f(r, \theta, \dot{r}, \dot{\theta}, u) = \begin{bmatrix} \dot{r} \\ \dot{\theta} \\ r\dot{\theta}^2 - \frac{k}{r^2} + u_1 \\ -2r\frac{\dot{\theta}^2}{\dot{r}} + \frac{u_2}{r} \end{bmatrix}$$

where $u = [u_1, u_2]^T$.
The Jacobian of the system is

$$J_1 = \frac{\partial f(r,\theta,\dot{r},\dot{\theta},u)}{\partial x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \dot{\theta}^2 + 2\frac{k}{r^3} & 0 & 0 & 2r\dot{\theta} \\ -2\frac{\dot{\theta}}{r^2}\dot{r} - \frac{u_2}{r^2} & 0 & -2\frac{\dot{\theta}}{r} & -2\frac{\dot{r}}{r} \end{bmatrix}$$

On the reference trajectory we have $u(t) \equiv 0$, $r(t) \equiv 0$, $\dot{r}(t) \equiv 0$, $\theta(t) = \omega t$, and $\dot{\theta} = \omega$ and so on the reference trajectory the Jacobian evaluates to

$$J_1|_{ref} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 2p\omega \\ 0 & 0 & -2\frac{\omega}{p} & 0 \end{bmatrix}$$

Similarly, the Jacobian of f with respect to u is

$$J_2 = \frac{\partial f(\dot{r},\dot{\theta},u)}{\partial u} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \frac{1}{r} \end{bmatrix}$$

and so

$$J_2|_{ref} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \frac{1}{p} \end{bmatrix}$$

Therefore, the linearized equation about the reference orbit is

$$\begin{bmatrix} \frac{d}{dt}\delta r \\ \frac{d}{dt}\delta\theta \\ \frac{d}{dt}\delta\dot{r} \\ \frac{d}{dt}\delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 2p\omega \\ 0 & 0 & -2\frac{\omega}{p} & 0 \end{bmatrix} \begin{bmatrix} \delta r \\ \delta\theta \\ \delta\dot{r} \\ \delta\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \frac{1}{p} \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix}$$