Take Home Final

Nathan Rose

December 10, 2021

This final was made with python and latex. There are scripts that produce latex fragments and images, these are linked into the main document with the subimport package. if you wish to change some of the inputs and mess with the output you can do so in the various json files, located in scripts/resources. To see instructions for how to build easily see the README.md file

1 (30pts)

A High performance helicopter has a model shown in Figure 1. The goal is to control the pitch angle θ of the helicopter by adjusting the rotor thrust δ. The equations of motion of the helicopter are

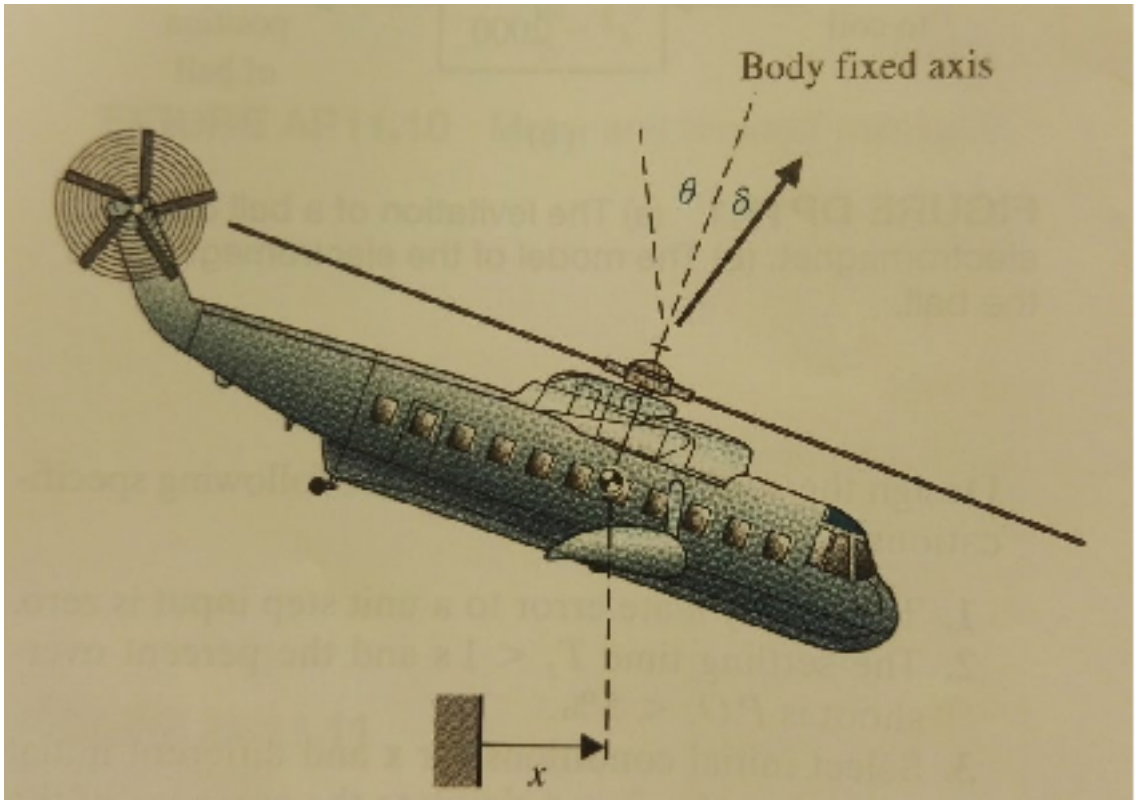

Figure 1:

$$\frac{d^2\theta}{dt^2} = -\sigma_1 \frac{d\theta}{dt} - \alpha_1 \frac{dx}{dt} + n\delta \tag{1}$$

$$\frac{d^2x}{dt^2} = g\theta - \alpha_2 \frac{d\theta}{dt} - \sigma_2 \frac{dx}{dt} + g\delta \tag{2}$$

Where x is the translation in the horizontal direction. For a miltary high-preformance helicopter we find: $\sigma_1 = 0.415$, $\sigma_2 = 0.0198$, $\alpha_1 = 0.0111$,$\alpha_2 = 1.43$, $n = 6.27$, $g = 9.8$ all in appropriate SI units. Find:

(a) A state variable representation of this system

(b) The transfer function representation for $\frac{\theta(s)}{\delta(s)}$

(c) Use state variable feedback to achieve adaquate performances for the controlled system. Desired specifications include:

  (1) A steady-state for an input step command for $\theta_d(s)$, the desired pitch angle, less than 20% of the input step magnitude

  (2) An overshoot for a step input command is less than 20%

  (3) a settling (with a 2% criterion) time for a step command of less than 1.5 seconds

(d) If the state variable is not available, design the observer and control law to meet the design specifications included in part

work:

(a) For the sytem with the following state variables:

$$x = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{3}$$

the state space representation is:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\sigma_2 & g & -\alpha_1 \\ 0 & 0 & 0 & 1 \\ 0 & -\sigma_1 & 0 & -\alpha_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ g \\ 0 \\ n \end{bmatrix} u \tag{4}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{5}$$

(b) For the system described by:

$$\dot{x} = \begin{bmatrix} 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & -0.02 & 9.8 & -1.43 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ 0.0 & -0.415 & 0.0 & -0.011 \end{bmatrix} x + \begin{bmatrix} 0.0 \\ 9.8 \\ 0.0 \\ 6.27 \end{bmatrix} u \tag{6}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{7}$$

A realization was determined by taking the realization of the state space representation. The result of this is:

$$g(s) = \begin{bmatrix} \frac{9.8s^2 - 8.857s + 61.446}{1.0s^4 + 0.031s^3 - 0.593s^2 + 4.067s} \\ \frac{6.27s^2 - 3.943s}{1.0s^4 + 0.031s^3 - 0.593s^2 + 4.067s} \end{bmatrix} \tag{8}$$

(c) To meet requirements the following eigenvalues were chose:

(a) $-2.667 \pm 5.205j$
(b) $-13.333 \pm 1j$

These eigenvalues were then use to create a feedback matrix of:

$$k = \begin{bmatrix} 99.523 & 39.6 & -78.583 & -56.796 \end{bmatrix} \tag{9}$$

This produced a reeponse which can be seen below in Figure 2

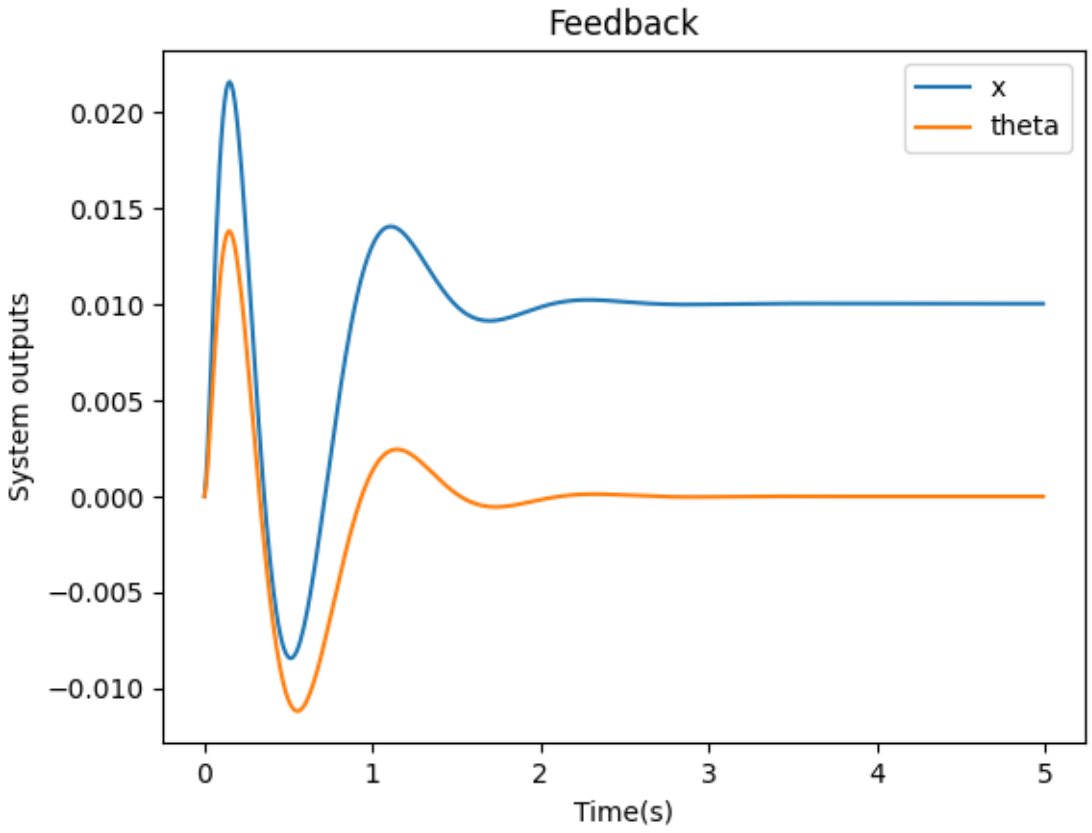

Figure 2: Feedback

(d) The observer was chosen to have eigenvalues that are observer 5 further away than the controller eigenvalues listed above in the previous step. This resulted in eigenvalues of:

(a) $-13.333 \pm 5.205j$
(b) $-66.667 \pm 1j$

The L matrix was determined from $L_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$ which resulted in:

$$L = \begin{bmatrix} 79.104 & -7.305 \\ 815.808 & -444.737 \\ 6.291 & 80.865 \\ 360.09 & 943.008 \end{bmatrix} \tag{10}$$

This produces a response which can be seen below in Figure 3 below.

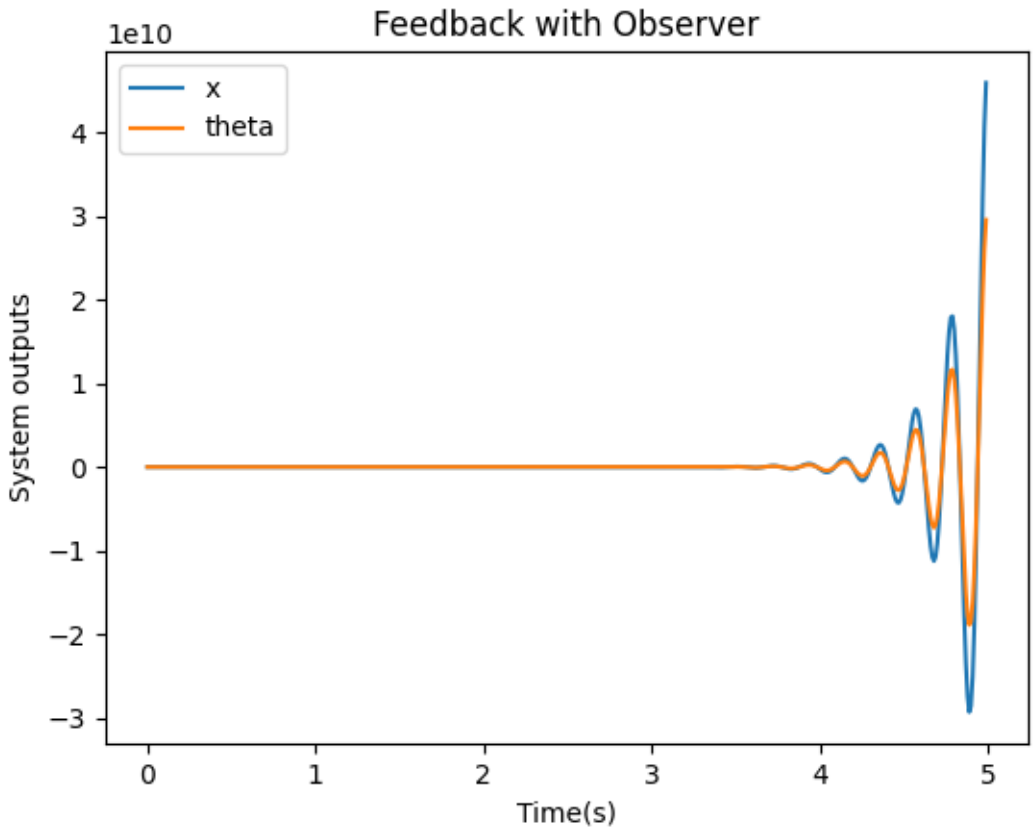

Figure 3: Observer

2 (30pts)

The open loop system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 4 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} u \tag{11}$$

$$y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \tag{12}$$

$$x(0) = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \tag{13}$$

1) Assume that x is availbale for state feedback. Design and LQR control law by letting $R = 1$ and choosing Q so that all the elements of the feedback gain K have absolute value less than 50.
Requirement: $|y_1(t)| \leq 0.05$,$|y_2(t)| \leq 0.05$, for all $t > 5$. Plot $y_1(t)$ and $y_2(t)$ in the same figure for $t \in [0, 20]$

2) Assume that only the output y is available. Design an observer so that the poles of the observer are $-5 \pm j5, -10$. Choose the observer gain so that all the elments have absolute value less than 80. Form a closed loop system along with the LQR controller in step 1). Plot $y_1(t)$ and $y_2(t)$ in the same figure for $t \in [0, 20]$

work:

1) For the system described by:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 4 & 5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} u \tag{14}$$

$$y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{15}$$

The following variables were used to generate the observer:

along with $R = \begin{bmatrix} 1 \end{bmatrix}$ and $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ set to

This produced $k = \begin{bmatrix} 0.0 & 5.324 & 6.046 \end{bmatrix}$

The following is the output of the LQR of the system, This also assumes the input is the unit step function.

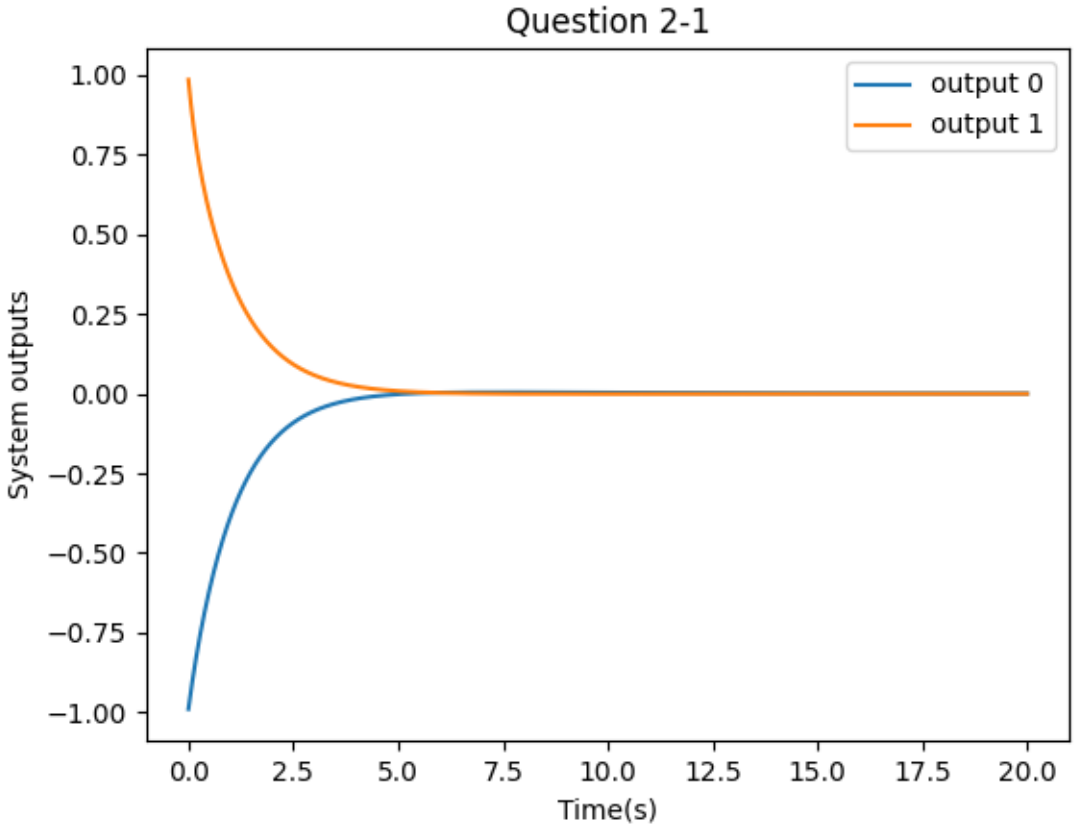

Figure 4: LQR System

2) For the system described by:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 4 & 5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} u \tag{16}$$

$$y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{17}$$

The following variables were used to generate the observer:

$$L_0 \left| \begin{bmatrix} 0.0 & 1.0 \\ 0.1 & 0.0 \\ 0.1 & 0.25 \end{bmatrix} \right.$$

$$F \left| \begin{bmatrix} -5.0 & 5.0 & 0.0 \\ -5.0 & -5.0 & 0.0 \\ 0.0 & 0.0 & -10.0 \end{bmatrix} \right.$$

With these values, the following outputs were calculated:

$$T \left| \begin{bmatrix} 0.003 & -0.002 & 0.078 \\ -0.012 & 0.061 & -0.045 \\ 0.002 & 0.003 & 0.016 \end{bmatrix} \right.$$

$$L \left| \begin{bmatrix} 58.497 & 1.337 \\ 11.91 & 9.973 \\ -2.128 & 13.09 \end{bmatrix} \right.$$

The following is the output of the LQR of the system, This also assumes the inputs are the unit step function with an initial estimate of the state of: $x_{e0} = \begin{bmatrix} 1.0 \\ -1.0 \\ 1.0 \end{bmatrix}$

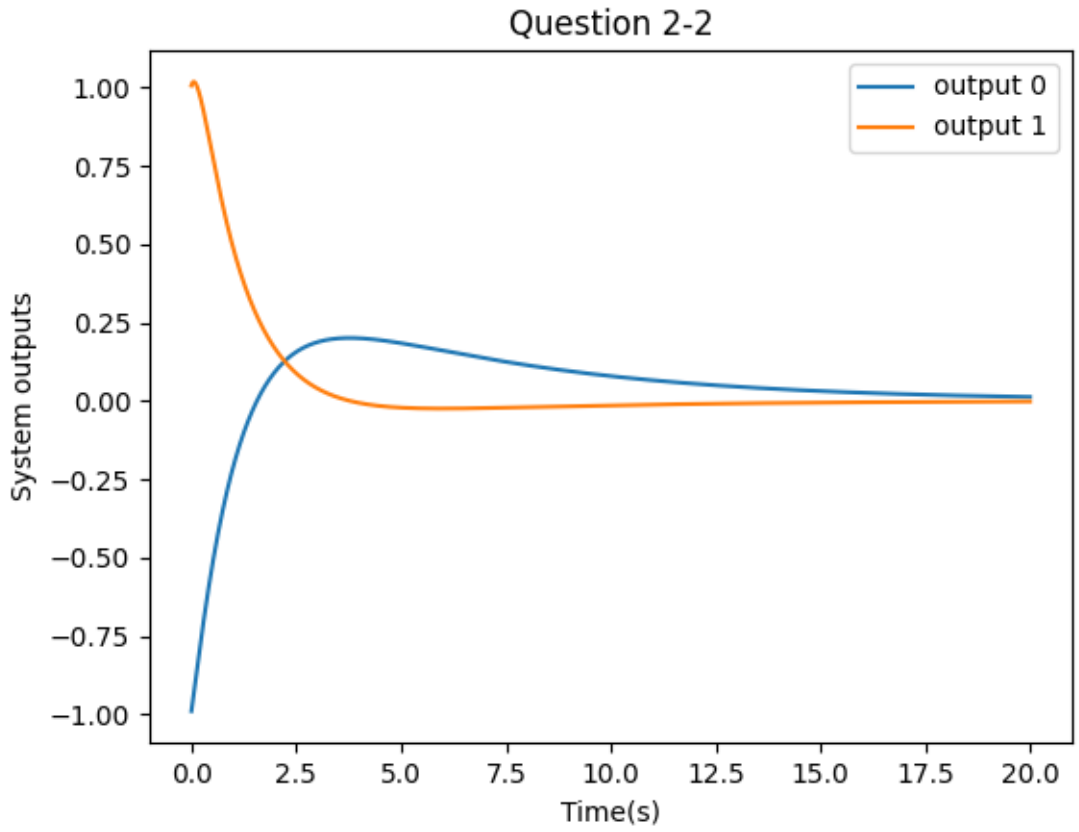

Figure 5: LQR System

3 (40pts)

A cart with an inverted pendulum as seen in Figure 6

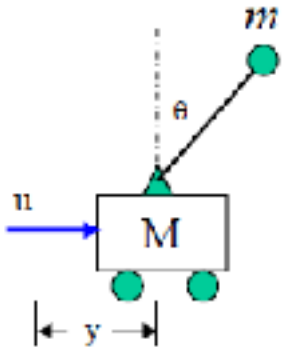

Figure 6:

n	control input(Newtons)
y	displacement of the cart(meters)
θ	angle of the pendulum(radians)

$$x = \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{18}$$

The control problems are

1: Stabalization: Design a feedback law $u\ Fx$ such that $x(t) > 0$ for $x(0)$ close to the diagram

2: For $x(0) = (0, 0, -\pi, 0)$, apply an impulse force $u(t) = u_{max}$ for $t \in [0, 0.1]$ to bring 0 to a certain range and then switch to the linear controller so that $x(t) \to 0$.

Assume that there is no friction or damping. The nonlinear model is as follows.

$$\begin{bmatrix} M+m & ml\cos(\theta) \\ \cos(\theta) & l \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} u + ml\dot{\theta}^2 \sin\theta \\ g\sin\theta \end{bmatrix} \tag{19}$$

with

$m = 1kg$	mass of the pendulum
$l = 0.2m$	length of the pendulum
$M = 5kG$	mass of the cart
$g = 9.8\frac{m}{s^2}$	mass of the cart

Linearize the system at $x = 0$

$$\begin{bmatrix} M+m & ml \\ 1 & l \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} u \\ g\theta \end{bmatrix} \tag{20}$$

the state space description for the linearlized system.

$$\dot{x} = Ax + Bu \tag{21}$$

Problems:

1. Find matrices A, B for the state space equation.

2. Design a feedback law $u - F_1 x$ so that $A + BF_1$ has eignevalues as $-3 \pm j3, -6, -8$. Build a simulink model for the closed loop linear system. Plot the response under initial condition $x(0) = (-1.5, 0, 1, 3)$.

3. Build a simulink model for the original nonlinear system, verify that stabilization is achieved by $u = F_1 x$ when $x(0)$ is close to the origin. Find the maximal θ_0 so that nonlinear system can be stabalized from $x_0 = (0, 0, \theta_0, 0)$

4. For $x(0) = (0, 0, \frac{\pi}{5}, 0)$, compare the response $y(t)$ and $\theta(t)$ for the linearized system and the nonlinear system under the same feedback $u - F_1 x$

work:

1. The system is being modeled with a state matrix of:

$$\begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{22}$$

and update and output equations of

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-mg}{M} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{l} + \frac{gm}{Ml} & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{m*l} \end{bmatrix} u \tag{23}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{24}$$

2.

3. The non-linear system was implimented with the following code

```
import numpy as np
from numpy.linalg import inv
from math import cos, sin, pi
import json


def nonLinearUpdate(x: np.matrix,
                    r: np.matrix,
                    k: np.matrix,
                    config: json,
                    dt: float) -> np.matrix:
    """
    This function is used to model the non-linear system described in
    question 3. If the pendulum falls over, will throw a validation
    error, to indicate the system fails
    """
    # a list of variables that make my life easier
    dot_y = x.item((1, 0))
    theta = x.item((2, 0))
    dot_theta = x.item((3, 0))
    u = r - (k*x)
```

```
    n = u.item((0, 0))
    M = float(config["M"])
    m = float(config["m"])
    L = float(config["l"])
    g = float(config["g"])

    # non linear update code
    accels = inv(np.matrix([
        [M+m, m*L*cos(theta)],
        [cos(theta), L]
    ])) * np.matrix([
        [n + m*L*theta*theta*sin(theta)],
        [g*sin(theta)]
    ])
    ddot_y = accels.item((0, 0))
    ddot_theta = accels.item((1, 0))
    dx = np.matrix([
        [dot_y],
        [ddot_y],
        [dot_theta],
        [ddot_theta]
    ])
    next_x = x + (dx*dt)

    # rage quit if it falls over lol
    next_theta = next_x.item((2, 0))
    while (next_theta) > pi:
        raise AssertionError("collapsed")
        next_theta = next_theta - (2*pi)
    while (next_theta) < -pi:
        raise AssertionError("collapsed")
        next_theta = next_theta + (2*pi)
    next_x.itemset((2, 0), next_theta)

    return next_x


def nonLinearOutput(x: np.matrix, r: np.matrix, k: np.matrix, config: json, dt: float)
    """
    A simple output function, just grabs the theta stat variable
    """
    theta = x.item((2, 0))
    return np.matrix([
        [theta]
    ])
```

A sample of the non-linear system with a starting state of $x_0 = \begin{bmatrix} \end{bmatrix}$ can be seen below in Figure 7

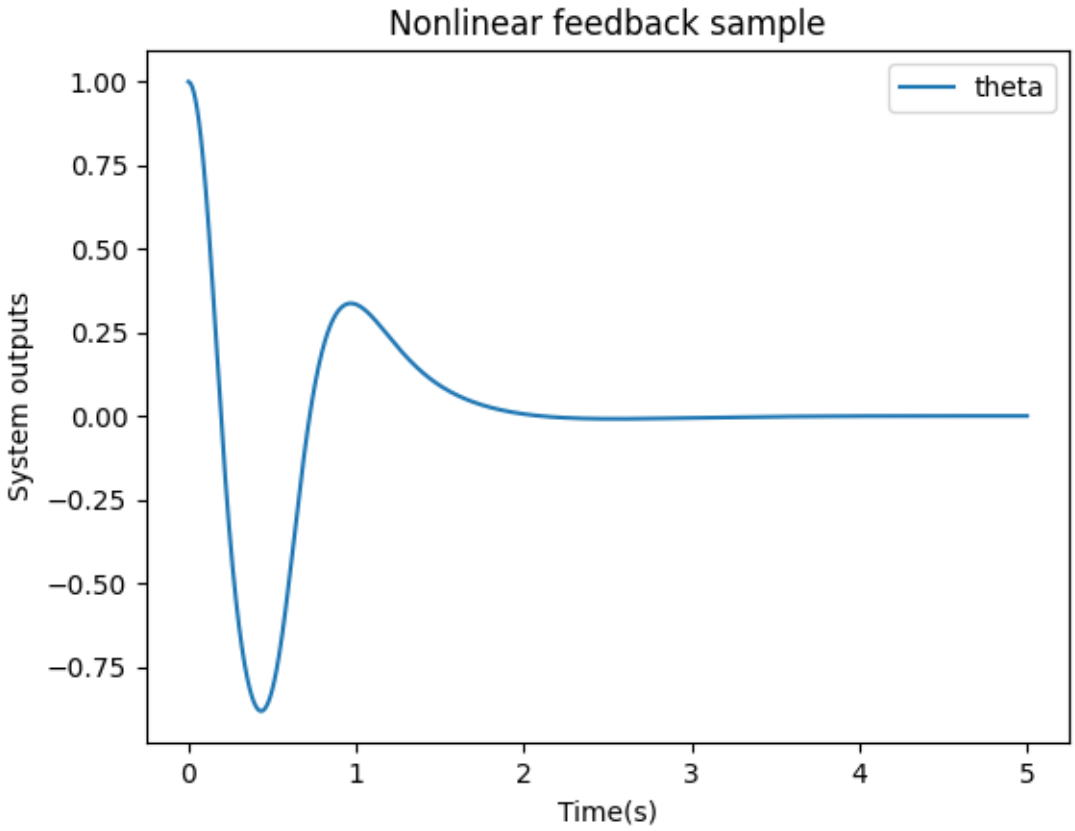

Figure 7: 3-3 System

For the limit, the system was condidered stabalized if after 5 seconds and if θ was within 0.01 radians of 0, and had not fallen over yet.

This value is calculated by code that can be seen in scripts/questionThree.py and is done by dividing up the region into n segments and determining the boundrary values for where it is stabalized and where it is not this repeats until the the the region under question is smaller than 0.001.

The result of this is a θ limit of 1.058 radians or 60.645 degrees.

4. The comparison between the linear and non-linear system can be seen below in Figure 8

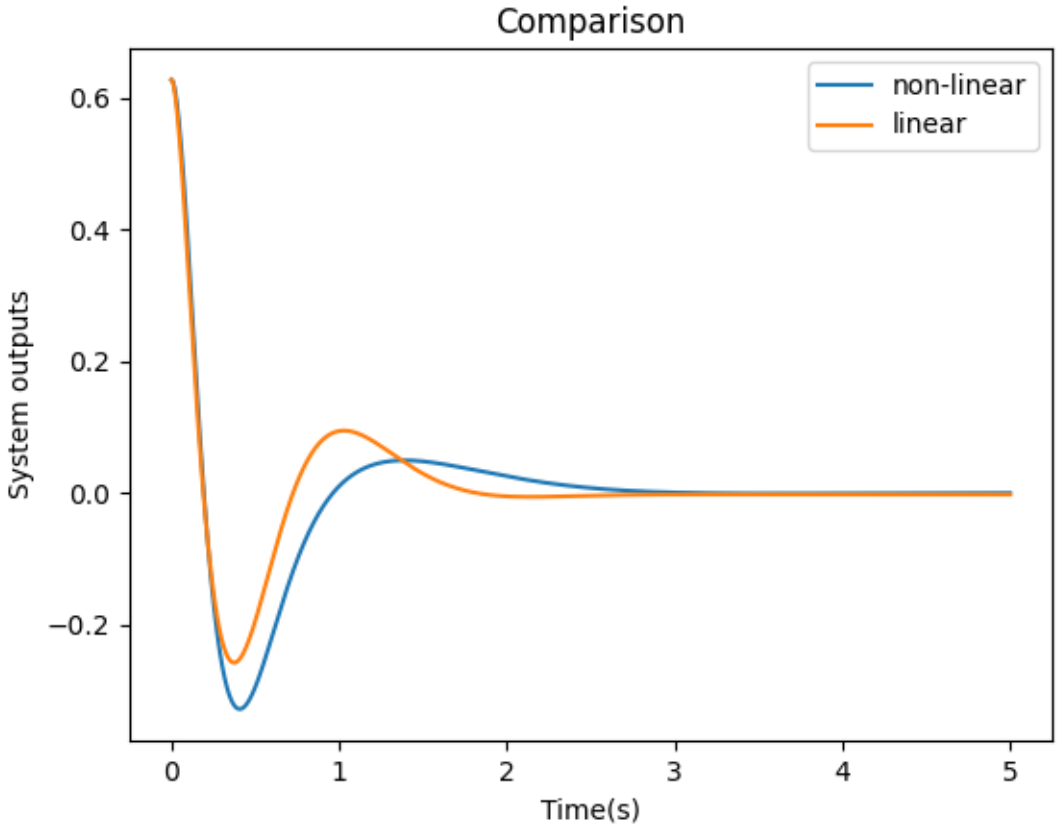

Figure 8: Comparison