Take Home Final

Nathan Rose

December 6, 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

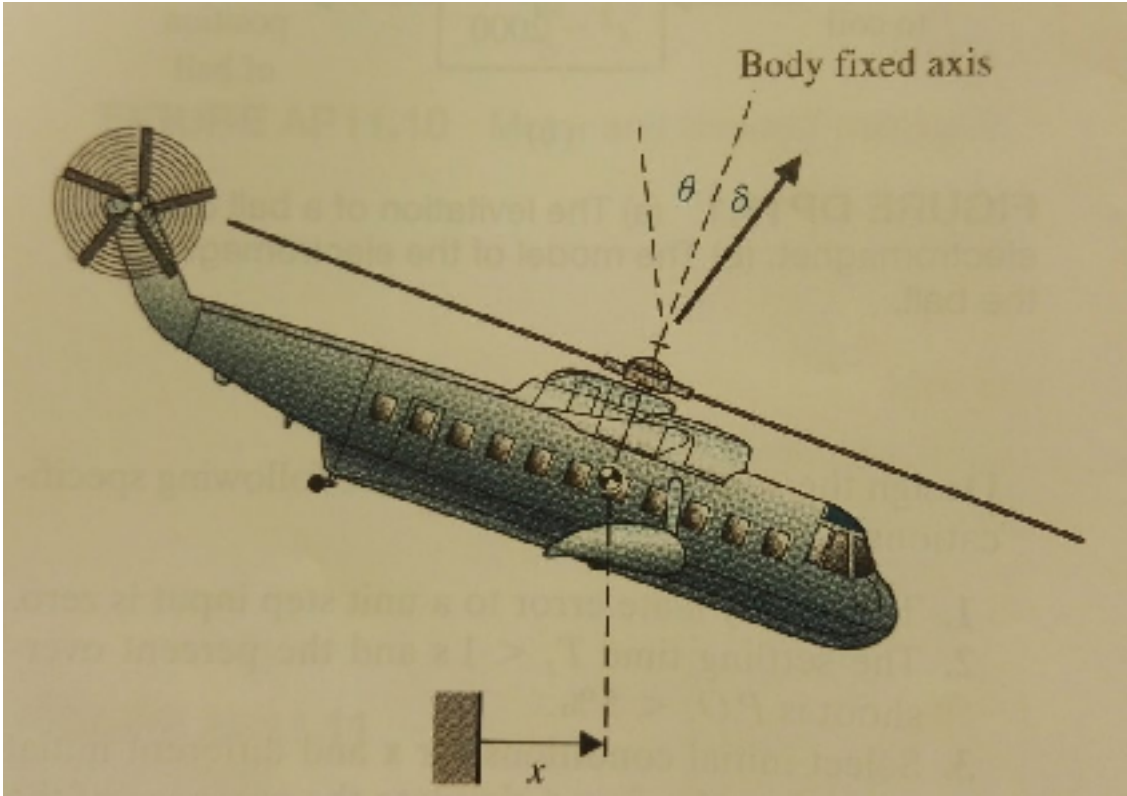

Figure 1:

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

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

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 system with 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:

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

2

(c) part c

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

2 (30pts)
The open loop system

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 4 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} u \tag{9}$$

$$y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \tag{10}$$

$$x(0) = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \tag{11}$$

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: $|y_1(t)| \le 0.05$, $|y_2(t)| \le 0.05$, 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) 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: $|y_1(t)| \le 0.05$, $|y_2(t)| \le 0.05$, for all $t > 5$. Plot $y_1(t)$ and $y_2(t)$ in the same figure for $t \in [0, 20]$
   For the system described by:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 4 & 5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} u \tag{12}$$

$$y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{13}$$

and $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = [1]$

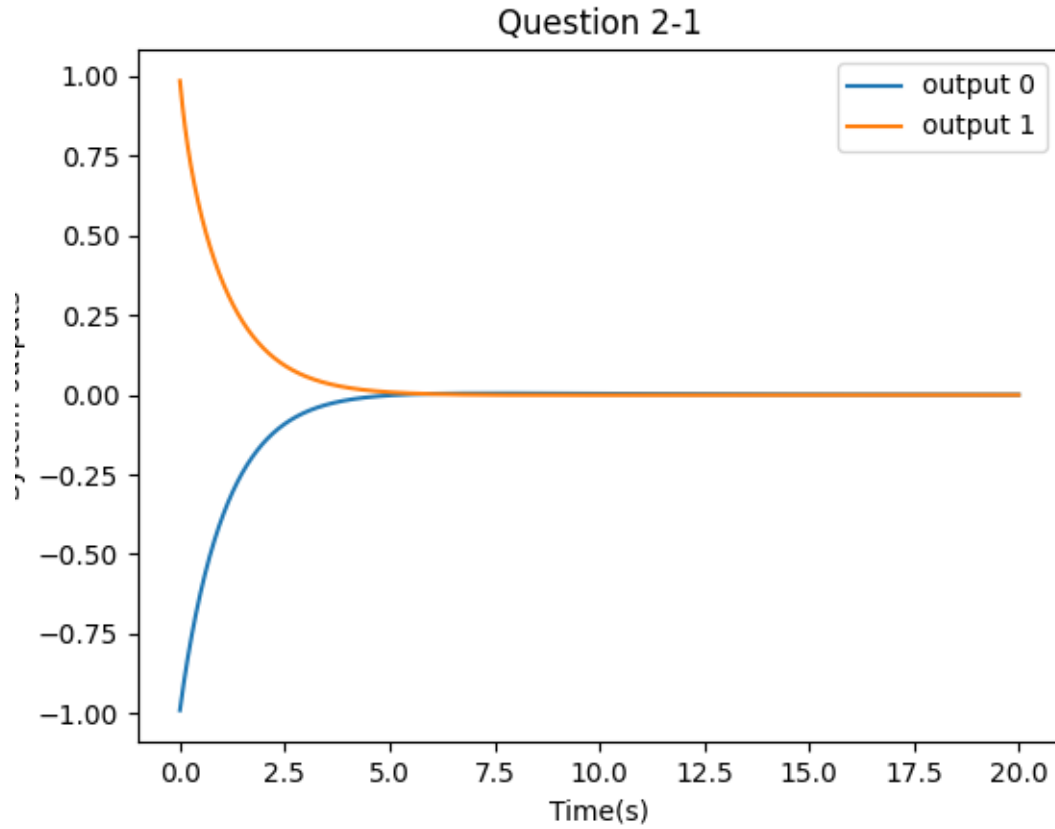The following is the outputs of the LQR system assuming the inputs are 0

Figure 2: LQR system

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 $t \in [0, 20]$ 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 chosen for the observer:

$$L_0 \begin{vmatrix} \begin{bmatrix} 0.0 & 1.0 \\ 0.1 & 0.0 \\ 0.1 & 0.25 \end{bmatrix} \end{vmatrix}$$

$$F \begin{vmatrix} \begin{bmatrix} -5.0 & 5.0 & 0.0 \\ -5.0 & -5.0 & 0.0 \\ 0.0 & 0.0 & -10.0 \end{bmatrix} \end{vmatrix}$$

These were used to calculate:

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

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

The following is the outputs of the LQR system assuming the inputs are 0 and an initial estimate of state of $\begin{bmatrix} 1.0 \\ -1.0 \\ 1.0 \end{bmatrix}$
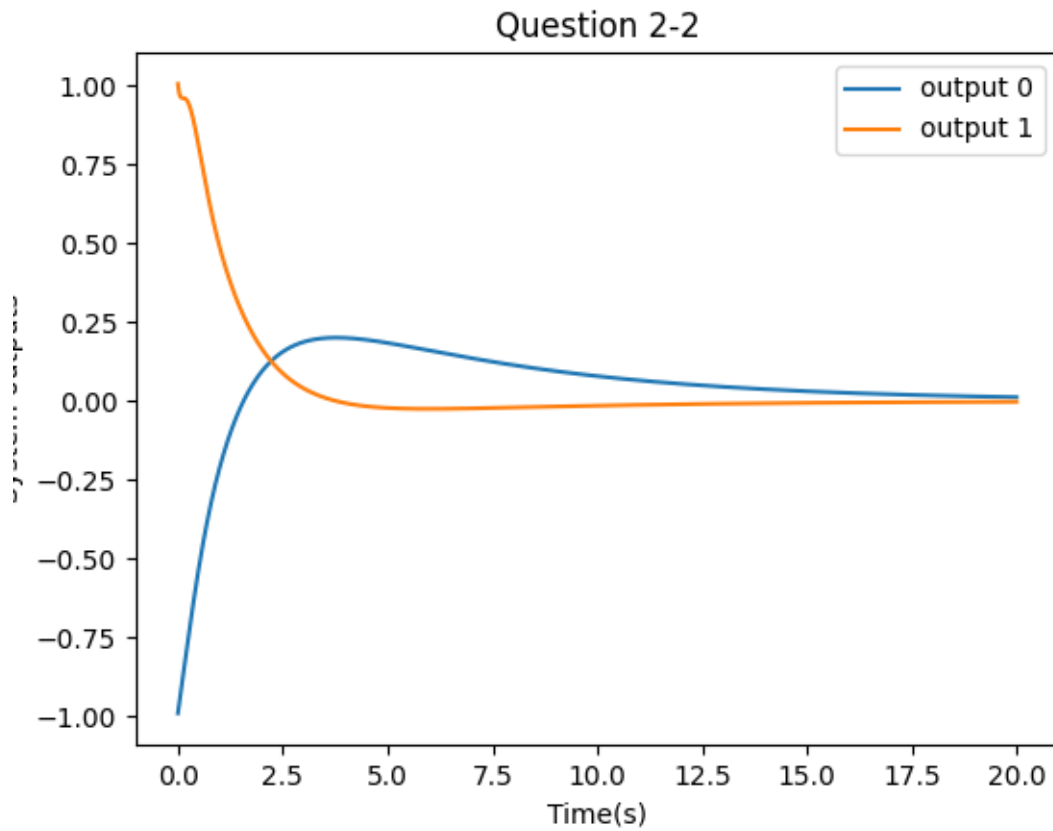

Figure 3: LQR system

3 (40pts)

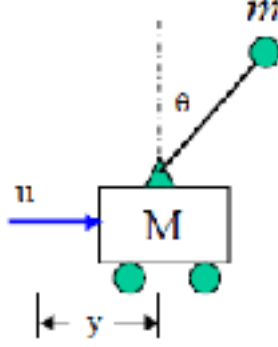A cart with an inverted pendulum as seen in Figure 4

Figure 4:

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

$$x = \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{16}$$

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.

$$\begin{bmatrix} M + m & ml\cos(\theta) \\ \cos(\theta) & l \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} - \begin{bmatrix} u + ml\dot{\theta}^2 \sin\theta \\ g\sin\theta \end{bmatrix} \tag{17}$$

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 $x = 0$

$$\begin{bmatrix} M + m & ml \\ 1 & l \end{bmatrix} \begin{bmatrix} \ddot{y} \\ \ddot{\theta} \end{bmatrix} - \begin{bmatrix} u \\ g\theta \end{bmatrix} \tag{18}$$

the state space description for the linearlized system.

$$\dot{x} = Ax + Bu \tag{19}$$

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 $x(0) = (-1.5, 0, 1, 3)$.

6

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. For the system with state matrix:

$$x = \begin{bmatrix} y \\ \dot{y} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{20}$$

The following system is described:

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-mg}{M} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{g}{l} + \frac{gm}{Ml} & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{m*l} \end{bmatrix} u \tag{21}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \tag{22}$$

2. The input values were: $\begin{array}{c} K_0 \\ \\ F \end{array} \left| \begin{array}{c} \begin{bmatrix} 10 & 20 & 30 & 40 \end{bmatrix} \\ \begin{bmatrix} -3.0 & -3.0 & 0.0 & 0.0 \\ 3.0 & -3.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & -6.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -8.0 \end{bmatrix} \end{array} \right.$ this produced a k of:

$$k = \begin{bmatrix} -47.222 & -41.211 & -216.284 & -28.242 \end{bmatrix} \tag{23}$$

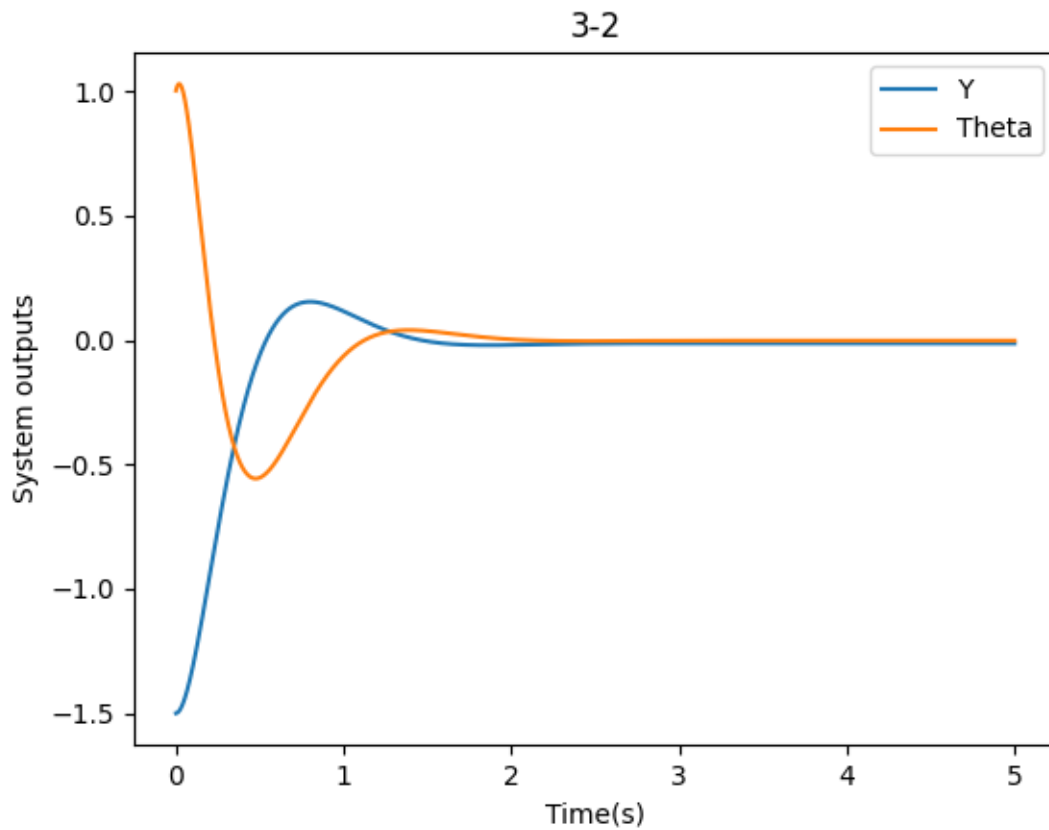With this feedback value the following outputs were made:

Figure 5: 3-2 system

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

```python
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)
    """
    This function is used to model the non-linear system, this will throw an exceptio
    there is a rollover
    """
    # 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"])
```

```python
        L = float(config["l"])
        g = float(config["g"])

        # non-linear functions
        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 with starting inputs of: $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ produces the followung outputs
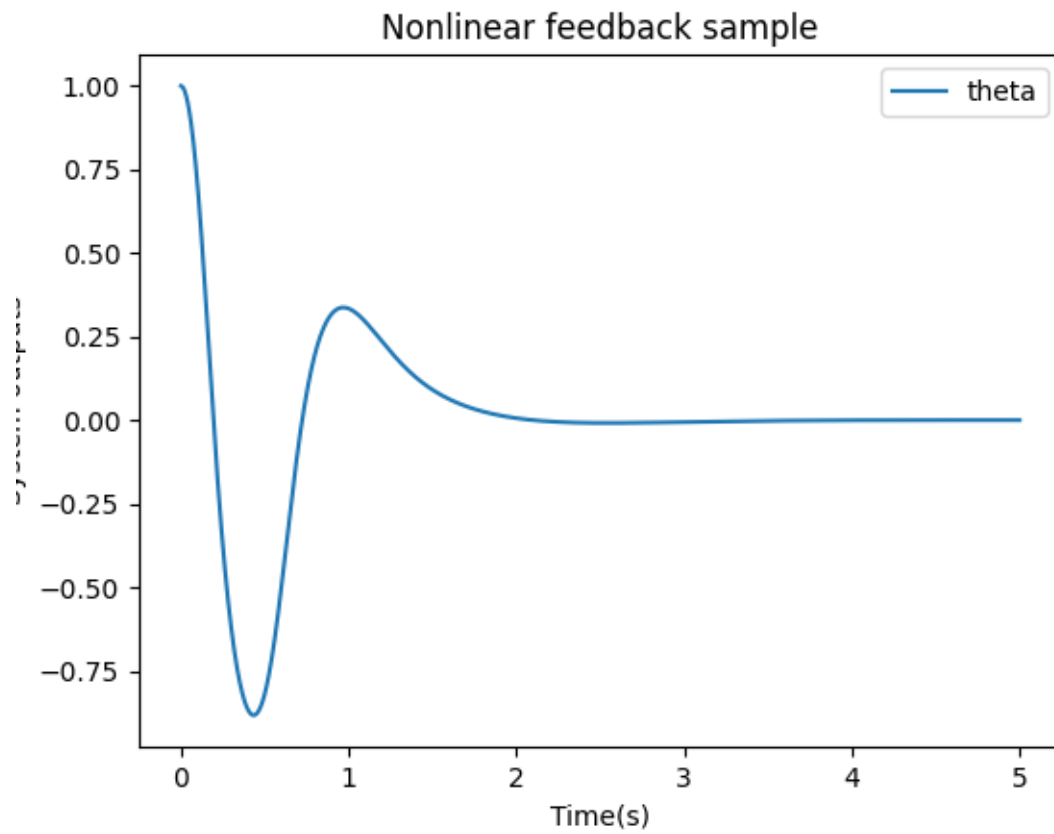
Figure 6: 3-3 system

For the limit, the system was considered stabalized if after 5 seconds the system state variable θ was within ± 0.01 of 0, and it had not fallen over it yet, the result of this is a theta limit of: 1.058 radians which is: 60.645 degrees

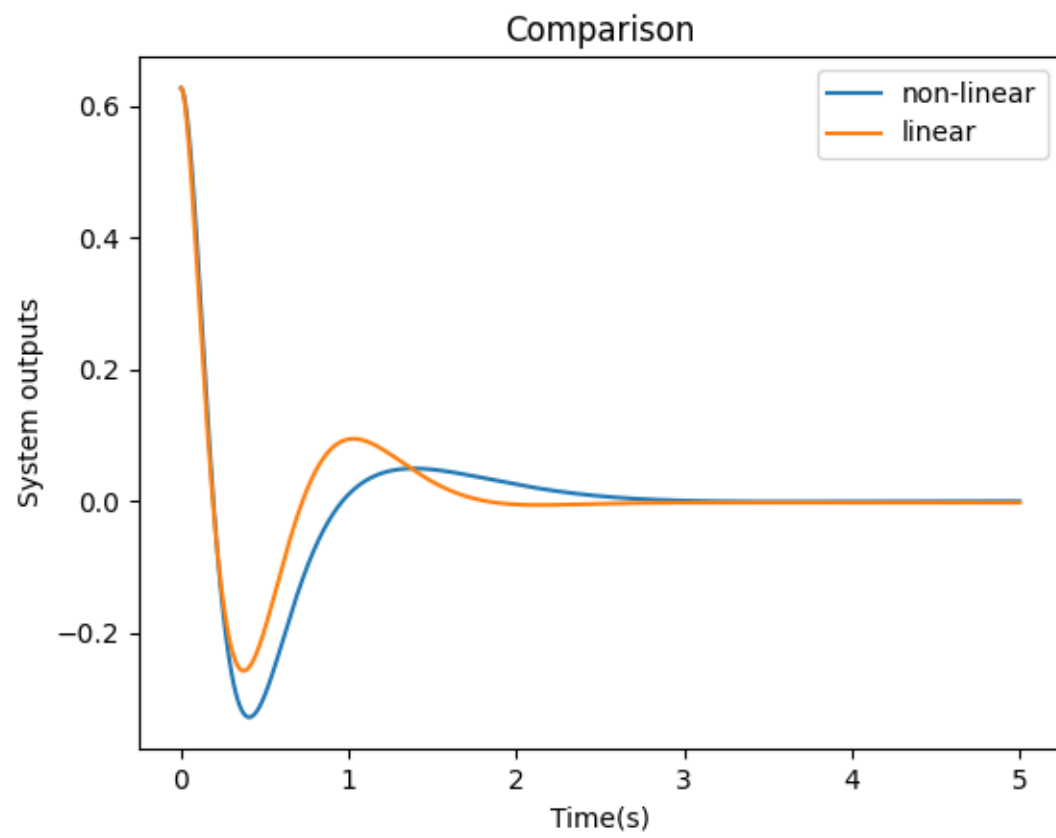4. The comparison betweeen the linear and nonlinear system can be seen in the system below in Figure 7

Figure 7: Comparison