Name: Andrew Choi
UID: 205348339

EE 209AS Pset 0

1. Linear Algebra

1.1.

1(a) $A = \begin{bmatrix} 1 & 2 & 3 & 2 \\ 3 & 6 & 9 & 5 \\ 2 & 4 & 6 & 9 \end{bmatrix} \xrightarrow[R_3-2R_1]{R_2-3R_1} \begin{bmatrix} 1 & 2 & 3 & 2 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 5 \end{bmatrix}$

$\xrightarrow[R_3-5R_2]{-1R_2} \begin{bmatrix} 1 & 2 & 3 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

$\boxed{\text{matrix } A \text{ is rank } 2}$

1(b) $Ax = b = \begin{bmatrix} 1 \\ 2 \\ b_3 \end{bmatrix}$ $\left[\begin{array}{cccc|c} 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & (b_3-7) \end{array}\right]$

$b_3 - 7 = 0$ $\boxed{b_3 = 7}$

1(c) $x_1 + 2x_2 + 3x_3 + 2x_4 = 1$
$x_4 = 1$

$$\boxed{\begin{aligned} x_1 + 2x_2 + 3x_3 &= -1 \\ x_4 &= 1 \end{aligned}}$$

1.2 difficulty: needed review & easy

2. Differential Equations

2.1 $\dot{y}(t) = z(t) + u(t)$

$\dot{z}(t) = 2y(t) + z(t) + 3u(t)$

2(a) $sY(s) = Z(s) + U(s)$

$sZ(s) = 2Y(s) + Z(s) + 3U(s)$

$(s-1)Z(s) = 2Y(s) + 3U(s)$

$Z(s) = \dfrac{2Y(s) + 3U(s)}{s-1}$

$sY(s) = \dfrac{2Y(s) + 3U(s)}{s-1} + U(s)$

$s(s-1)Y(s) = 2Y(s) + 3U(s) + (s-1)(U(s))$

$(s^2 - s - 2)Y(s) = (s+2)U(s)$

$H(s) = \dfrac{Y(s)}{U(s)} = \dfrac{s+2}{s^2-s-2}$

$$\boxed{H(s) = \frac{(s+2)}{(s-2)(s+1)}}$$

2(b)

$\dot{x} = Ax + Bu$ $\quad \dfrac{y}{u} = \dfrac{y}{z} \cdot \dfrac{z}{u}$ $\quad \dfrac{y}{z} = N(s)$

$y = Cx + Du$ $\quad \dfrac{z}{u} = 1/D(s)$

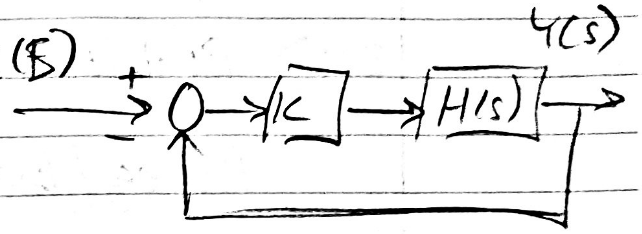
$x = \begin{bmatrix} \dot{z} \\ z \end{bmatrix}$ $\quad A = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix}$ $\quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$\ddot{z} = u + \dot{z} + 2z$

$y = \dot{z} + 2z$ $\quad C = \begin{bmatrix} 1 & 2 \end{bmatrix}$ $\quad D = 0$

2(c) not sure how to solve

2(d) $u(t) = Ky(t)$



$$G(s) = \frac{K\frac{s+2}{(s-2)(s+1)}}{1 + \frac{K(s+2)}{(s-2)(s+1)}}$$

$$G(s) = \frac{Ks + 2K}{s^2+2s-1+Ks+2K} = \frac{Ks + 2K}{s^2+(2+K)s+(2K+1)}$$

$$\begin{array}{lll} s^2 & 1 & 2K-1 \\ s^1 & 2+K & 0 \\ s^0 & 2K-1 & 0 \end{array}$$

$s^2 + (2+K)s + (2K+1) = 0$

$2+K > 0$ $\quad$ $2K+1 > 0$

$K > -2$ $\quad$ $K > -1/2$

$$\boxed{\text{system stable so long as } K > -0.5}$$

2.2. difficulty: needed review

3. Probability

3.1 sequence = TTTHTTTHT

3(a) $P(seq/fair) = 0.5^8$

3(b) $P(H/bias) = 0.25$

$P(seq/bias) = (0.25)^2(0.75)^6$

3(c) $P(bias) = 1/4$

3(d) $P(bias, sequence) = 0.25\, P(seq/bias)$
$= 0.0027809$

$P(fair, seq) = 0.75\, P(seq/fair)$
$= 0.002929$

$$P(bias/seq) = \frac{0.0027809}{0.0027809 + 0.002929}$$

$P(bias/seq) = 0.487 = 48.7\%$

3.2. difficulty: easy

4. Programming (using python)

4(a).

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

def coin_simulate(num_of_flips, coin_type):
    sequence = []
    if coin_type == "fair":
        sequence = ["H" if np.random.uniform() < 0.5 else "T" for _ in range(num_of_flips)]
    elif coin_type == "bias":
        sequence = ["H" if np.random.uniform() < 0.25 else "T" for _ in range(num_of_flips)]
    return sequence

for i in range(5):
    print("======== 40 Flip Sequence {} ========".format(i+1))
    print("============= Bias ==================")
    print(coin_simulate(40, "bias"))
    print("============= Fair ==================")
    print(coin_simulate(40, "fair"))
    print("\n")
```

Output:

======== 40 Flip Sequence 1 ========

============= Bias ==================

['H', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'T', 'T']

============= Fair ==================

['H', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'H', 'T', 'H', 'H', 'H', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'H', 'H', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'T']

======== 40 Flip Sequence 2 ========

============= Bias ==================

['T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T']

============= Fair ==================

['T', 'T', 'T', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'H', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T']

======== 40 Flip Sequence 3 ========

============= Bias ==================

['T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T']

============= Fair ==================

['T', 'H', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'H', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'H', 'H']

======== 40 Flip Sequence 4 ========

============= Bias ==================

['T', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'H']

============= Fair ==================

['H', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'H', 'H', 'H', 'H', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'H', 'H', 'H', 'H', 'H', 'T']

======== 40 Flip Sequence 5 ========

============= Bias ==================

['T', 'T', 'H', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H']

============= Fair ==================

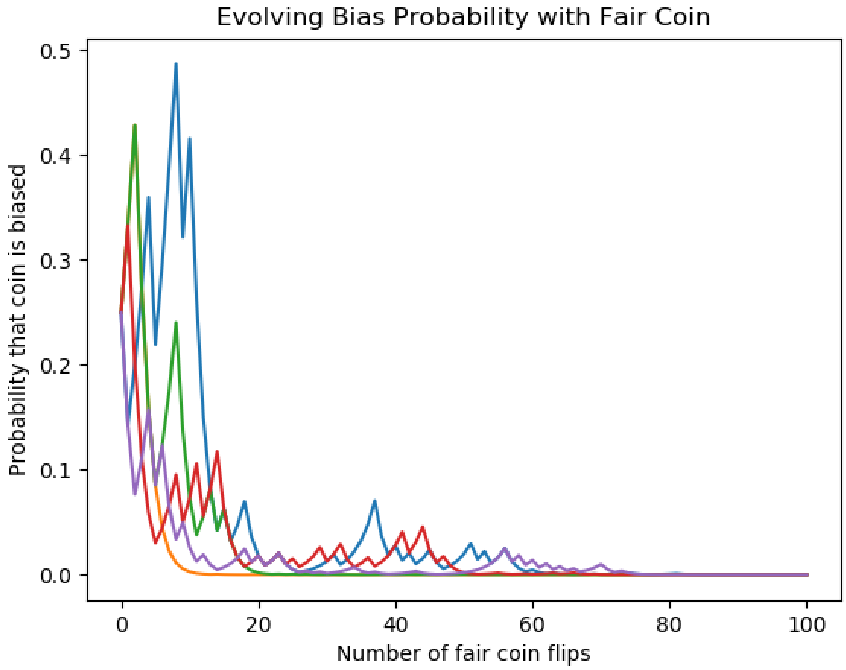
['T', 'T', 'H', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'H', 'H', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T']

4(b).

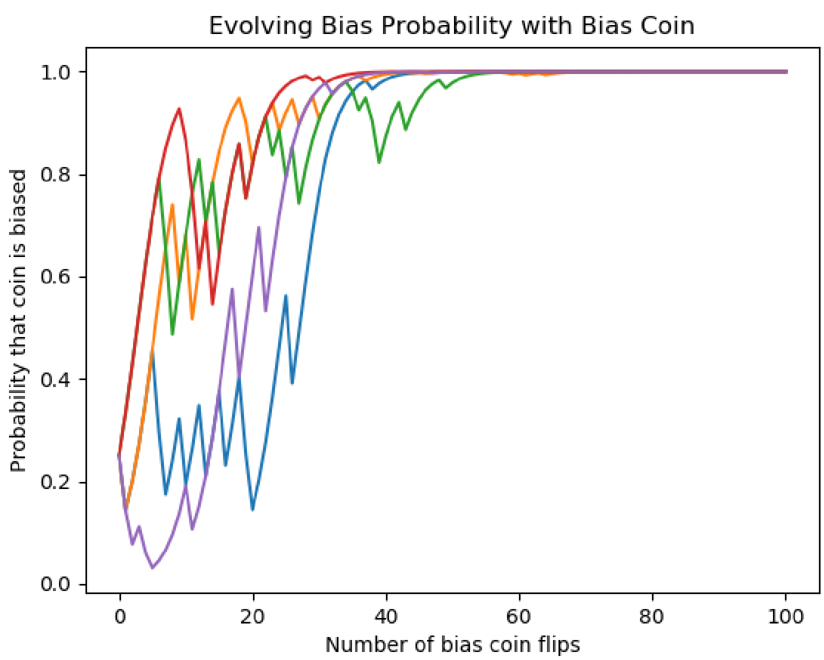
```python
def coin_simulate_with_likelihood(num_of_flips, coin_type="fair"):
    """
        probability of choosing biased coin is 1/4
        probability of choosing fair coin is 3/4
        for bias:
        choosing head is 1/4
        choosing tails is 3/4
    """
    sequence = []
    bias = [0.25] # starts at 0 flips
    fair = [0.75]
    p_h = 0.5 # probability of heads
    if coin_type == "bias": p_h = 0.25
    for i in range(num_of_flips):
        if np.random.uniform() < p_h:
            sequence.append("H")
            bias.append(bias[i]*p_h)
            fair.append(fair[i]*0.50)
        else:
            sequence.append("T")
            bias.append(bias[i]*(1-p_h))
            fair.append(fair[i]*0.50)

    likelihood_bias = [bias[i]/(bias[i]+fair[i]) for i in range(len(bias))]
    return sequence, likelihood_bias
```

```python
plt.figure(1)
flips = [i for i in range(101)]
for _ in range(5):
    coin_sequence, bias_prob = coin_simulate_with_likelihood(100, "fair")
    plt.plot(flips, bias_prob)
plt.title("Evolving Bias Probability with Fair Coin")
plt.ylabel("Probability that coin is biased")
plt.xlabel("Number of fair coin flips")

plt.figure(2)
for _ in range(5):
    coin_sequence, bias_prob = coin_simulate_with_likelihood(100, "bias")
    plt.plot(flips, bias_prob)
plt.title("Evolving Bias Probability with Bias Coin")
plt.ylabel("Probability that coin is biased")
plt.xlabel("Number of bias coin flips")

plt.show()
```

4(c).



4(d).



4.2. difficulty: easy

5.1

I found the probability and programming sections to be easy and did not need any review. A little under 1 hour to complete both at a leisurely pace.

The linear algebra portion required a little bit of review but was easy after brushing up for approximately 45 minutes. Took only 10ish minutes to finish afterwards.

The differential equations portion took me a bit to solve and required significant review (~2.5 hours). My last controls class was about 2 years ago, so I had to brush up on Laplace transforms and state-space representations. Was unable to come up with an answer for 2(c).