ECE 351-51

April 12, 2022

Lab 11

Submitted By :
Kennedy Beach

Contents

1 Introduction

The purpose of this lab was to use Python functions and Christopher Felton's function to analyze a discrete system.

2 Equations

The following causal function was provided:

$$y[k] = 2x[k] - 40x[k-1] + 10y[k-1] - 16y[k-2]$$

The steps to find the transfer function are described below:

$$\begin{aligned}
y[k] - 10[k-1] + 16y[k-2] &= 2x[k] - 40x[k-1] \\
Y(z) - 10z^{-1}Y(z) + 16z^{-2} &= 2X(z) - 40z^{-1}X(z) \\
Y(z)(1 - 10z^{-1} + 16z^{-2} &= X(z)(2 - 40z^{-1} \\
\frac{Y(z)}{X(z)} &= \frac{2 - 40z^{-1}}{(1 - 10z^{-1} + 16z^{-2})} \\
H(z) &= \frac{2z(z-20)}{z^2 - 10z + 16}
\end{aligned}$$

The impulse response was found by using partial fraction expansion on the derived transfer function:

$$\begin{aligned}
\frac{H(z)}{z} &= \frac{2(z-2)}{z^2 - 10z + 16} \\
\frac{H(z)}{z} &= \frac{A}{z-8} + \frac{B}{z-2} \\
A &= \frac{2(z-20)}{z-8}|_{z=2} = 6 \\
B &= \frac{2(z-20)}{z-2}|_{z=8} = -4 \\
\frac{H(z)}{z} &= \frac{6}{z-8} - \frac{4}{z-2} \\
Z^{-1}\{\frac{H(z)}{z}\} = h[k] &= [6(-8)^k - 4(-2)^k]u[k]
\end{aligned}$$

3 Methodology

The first part was to verify the partial fraction results using scipy.signal.residuez(). This is seen below and the output is seen in the Results section.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig
import control
import control as con

num = [2, -40]
den = [1, -10, 16]

r, p, k = sig.residuez(num, den)

print("Task 3 Residue Results:\n r = {}\n p={}\n k = {}\n".format(r
    ,p,k))
```

Listing 1: scipy.signal.residuez()

The Z-plane function provided by Felton was used to create the pole-zero plot for the transfer function.

```
def zplane(b,a,filename=None):
    """Plot the complex z-plane given a transfer function.
    """
    import numpy as np
    import matplotlib.pyplot as plt
    from matplotlib import patches

    # get a figure/plot
    ax = plt.subplot(111)
    # create the unit circle
    uc = patches.Circle((0,0), radius=1, fill=False,
                        color='black', ls='dashed')
    ax.add_patch(uc)
    # The coefficients are less than 1, normalize the coeficients
    if np.max(b) > 1:
        kn = np.max(b)
        b = np.array(b)/float(kn)
    else:
        kn = 1
    if np.max(a) > 1:
        kd = np.max(a)
        a = np.array(a)/float(kd)
    else:
        kd = 1

```

```
    # Get the poles and zeros
    p = np.roots(a)
    z = np.roots(b)
    k = kn/float(kd)

    # Plot the zeros and set marker properties
    t1 = plt.plot(z.real, z.imag, 'o', ms=10,label='Zeros')
    plt.setp( t1, markersize=10.0, markeredgewidth=1.0)
    # Plot the poles and set marker properties
    t2 = plt.plot(p.real, p.imag, 'x', ms=10,label='Poles')
    plt.setp( t2, markersize=12.0, markeredgewidth=3.0)
    ax.spines['left'].set_position('center')
    ax.spines['bottom'].set_position('center')
    ax.spines['right'].set_visible(False)
    ax.spines['top'].set_visible(False)

    plt.legend()
    # set the ticks
    # r = 1.5; plt.axis('scaled'); plt.axis([-r, r, -r, r])
    # ticks = [-1, -.5, .5, 1]; plt.xticks(ticks); plt.yticks(ticks)
    if filename is None:
        plt.show()
    else:
        plt.savefig(filename)

    return z, p, k

z, p, k = zplane(num, den)
print('Zeros = ', z, '\nPoles = ',p)
```

Listing 2: Z-plane function

The last part involved plotting the magnitude and phase responses of the transfer function using scipy.signal.freqz(). Since the output of scipy.signal.freqz() was in Hertz, I converted the magnitude of the output to decibels.

```
w, h = sig.freqz(num, den, whole=True)

plt.figure(figsize = (10, 7))
plt.subplot(2, 1, 1)
plt.ylabel('|H(j[U+FFFD])| (dB)')
plt.semilogx(w, 20*np.log10(np.abs(h)))
plt.grid()
plt.subtitle('Task 5 - Bode Plot of H(z) via sig.freqz() function')
plt.subplot (2, 1, 2)
plt.ylabel('/_H(j[U+FFFD])')
plt.semilogx(w, np.angle(h))
```

```
plt.grid()
```

Listing 3: Magnitude and Phase Responses

4 Results

```
Task 3 Residue Results:
 r = [ 6. -4.]
 p=[2. 8.]
 k = []
```

Figure 1: Residue Output

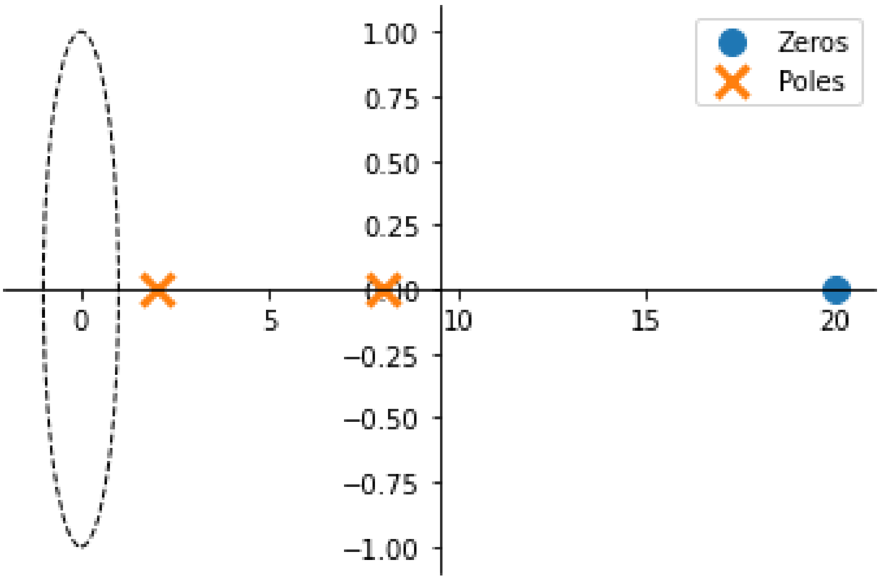

Figure 2: Z-plane Function Output

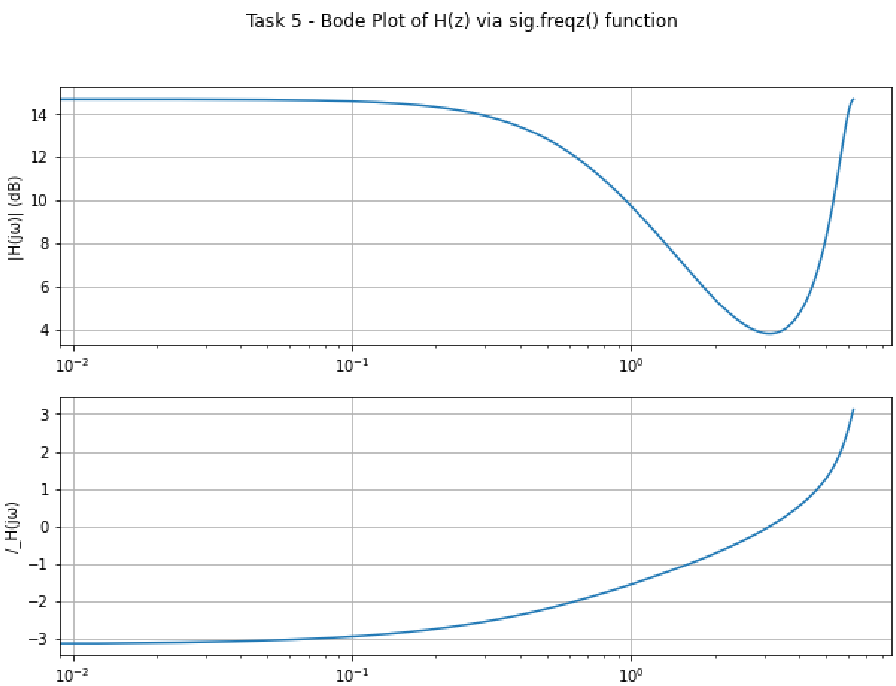

Figure 3: Bode Plot using scipy.signal.freqz()

5 Error Analysis

There may be some potential errors with the bode plot since it is was not checked to make sure it is correct.

6 Questions

1. Looking at the plot generated in Task 4, is H(z) stable? Explain why or why not.

H(z) is not stable since the poles and zeros are not within the unit circle.

2. Leave any feedback on the clarity of the expectations, instructions, and deliverables.

Everything was clear on what needed to be done and turned in.

7 Conclusion

This lab delved into analyzing systems using the Z-domain and more of Python's capabilities. The Python and LaTeX code are seen in `https://github.com/Eniac618/ECE351_Code` and `https://github.com/Eniac618/ECE351_Reports` respectively.