Z - Transform Operations

Lab 11

Spring 2021

1 Purpose

Analyze a discrete system using Python's built-in functions and a function developed by Christopher Felton.

2 Deliverables Overview

2.1 Part 1

- Typed derivation from **Task 1**.
- Typed derivation from **Task 2**.
- Printed output from **Task 3**.
- Plot from **Task 4**.
- Plot from **Task 5**.

As usual, plots and equations need to be thoroughly discussed in your report.

3 Part 1

3.1 Purpose

Consider the causal function,

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

where $y[k]$ is the output and $x[k]$ is the input. Assume that the system is initally at rest.

3.2 Deliverables

1. Typed derivation from **Task 1**. Show important steps and the final derivation in the **Methodology** section of your report.
2. Typed derivation from **Task 2**. Show important steps and the final derivation in the **Methodology** section of your report.

1. $y[k] = \left(2x[k] - 40x[k-1] + 10y[k-1] - 16y[k-2]\right)u[k]$

$$Y(z) = 2X(z) - 40\left(z^{-1}X(z) + \cancel{x[-1]}\right) + 10\left(z^{-1}Y(z) + \cancel{Y[-1]}\right) - 16\left(z^{-2}Y(z) + \cancel{z^{-1}Y[-1]} + \cancel{Y[-2]}\right)$$

No initial conditions

$$Y(z) = 2X(z) - 40z^{-1}X(z) + 10z^{-1}Y(z) - 16z^{-2}Y(z)$$

$$Y(z)\left(1 - 10z^{-1} + 16z^{-2}\right) = X(z)\left(2 - 40z^{-1}\right)$$

$$\Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{(-40z^{-1}+2)}{(16z^{-2}-10z^{-1}+1)} = \frac{(-40z^{-1}+2)}{(-8z^{-1}+1)(-2z^{-1}+1)}$$

2. $$H(z) = \frac{A}{(-8z^{-1}+1)} + \frac{B}{(-2z^{-1}+1)}$$

$$\Rightarrow A = \frac{-40z^{-1}+2}{(-2z^{-1}+1)} \text{ for } z=8$$
$$= \frac{-40/8+2}{(-1/4+1)} = \frac{-3}{.75} = -4$$

$$B = \frac{-40z^{-1}+2}{(-8z^{-1}+1)} \text{ for } z=2$$
$$= \frac{-20+2}{(-4+1)} = \frac{-18}{-3} = 6$$

$$\Rightarrow H(z) = \frac{6}{(-2z^{-1}+1)} - \frac{4}{(-8z^{-1}+1)}$$

$$\boxed{H(z) = \frac{6z}{(z-2)} - \frac{4z}{(z-8)}}$$

$$\Rightarrow H[k] = \left(6(2^k) - 4(8^k)\right)u[k]$$

3. Printed output from **Task 3** to be attached to the end of your report as an **appendix**.
4. Plot from **Task 4** to be included in the **Results** section of your report.
5. Plot from **Task 5**to be included in the **Results** section of your report.

3.3 Tasks

1. By hand, find $H(z)$.
2. By hand, find $h[k]$ by partial fraction expansion.
3. Use `scipy.signal.residuez()` to verify your partial fraction expansion.
4. Use the provided `zplane()` function to obtain the pole-zero plot for $H(z)$.
5. Use `scipy.signal.freqz()` to plot the magnitude and phase responses of $H(z)$. *Note: You must set* `whole = True` *within the* `scipy.signal.freqz()` *command. (See function documentation for details).*

4 Questions

1. Looking at the plot generated in **Task 4**, is $H(z)$ stable? Explain why or why not.
2. Leave any feedback on the clarity of lab tasks, expectations, and deliverables.

$x \text{ rad/s} \cdot \frac{1 \text{ Hz}}{2\pi \text{ rad/s}} = \frac{x}{2\pi} \text{ Hz}$

5 Appendix

5.1 zplane() function

```python
#
# Copyright (c) 2011 Christopher Felton
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU Lesser General Public License for more details.
#
# You should have received a copy of the GNU Lesser General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.
#
# The following is derived from the slides presented by
# Alexander Kain for CS506/606 "Special Topics: Speech Signal Processing"
# CSLU / OHSU, Spring Term 2011.
#
#
#
# Modified by Drew Owens in Fall 2018 for use in the University of Idaho's
# Department of Electrical and Computer Engineering Signals and Systems I Lab
# (ECE 351)
#
# Modified by Morteza Soltani in Spring 2019 for use in the ECE 351 of the U of
#  I.
#
# Modified by Phillip Hagen in Fall 2019 for use in the University of Idaho's
# Department of Electrical and Computer Engineering Signals and Systems I Lab
# (ECE 351)

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(1, 1, 1)

    # 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 coefficients
    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
```