6

z-Transform II

6.1 Purpose

The purpose of this lab is to designing a system to remove a corrupting tone from a recording made by the Perseverance Mars Rover.

6.2 Background

NASA's Perseverance Mars Rover (https://mars.nasa.gov/mars2020/) was launched in July, 2020 and landed in the Jezero Crater of Mars 6.5 months later, on 18 Feb, 2021. Its mission is to collect and analyze soil and rock samples taken from the surface of the planet. In addition to its many scientific instruments, the rover had a pair of microphones sampling at 11025 Hz that capture sounds as the rover crunched along the planet's surface. Unfortunately, the recordings are corrupted by a continuous tone at somewhere between 500 and 1000 Hz, probably due to electronic interference from the motors. Your job is find the frequency of the tone and create a filter that will remove it from the recording.

To prepare for this exercise, review Chapters 4 and 5 of the book, particularly section 5.5

6.3 Assignment

The assignment has three parts:

6.3.1 Find the frequency of the corrupting tone

Load the sound file

```
[x, fs] = audioread('persevere_bad.wav');
```

Use the DTFT/FFT to find the frequency of the tone.

One way to do this is to take the FFT of several frames of the signal, say 1024 points long, and average the magnitudes of the FFTs. (Why is it a bad idea just to average the frames of sound data and then take the magnitude of the FFT of that average?). Then plot the average FFT. You'll see that most of the energy is at low frequencies, but that there is a small sharp peak at a somewhat higher frequency. That's the corrupting tone you need to remove. You can use Matlab's `max` command to find the peak in the range of points you see in the plot that corresponds to this tone. Then, knowing the sampling frequency, you can convert this point number into a frequency. To do this, you need to remember

1) The relation between a sample frequency and a discrete-time frequency ω. The sample frequency of the analog signal f_s corresponds to the discrete-time frequency $\omega = 2\pi$.
2) The relation between the point number of the FFT and frequency ω. Remember that Point #1 in Matlab's `fft` corresponds to the frequency $\omega = 0$. The last point of the *N*-pt FFT corresponds to the frequency $\omega = 2\pi(N-1)/N$.

Deliverables:

I: Write a program that will

a) Plot of the spectrum (the magnitude of the DTFT/FFT) showing the peak at the corrupting tone

b) Find the frequency of the corrupting tone in Hz and in rad.

6.3.2 Find the poles and zeros of the filter that will remove the tone

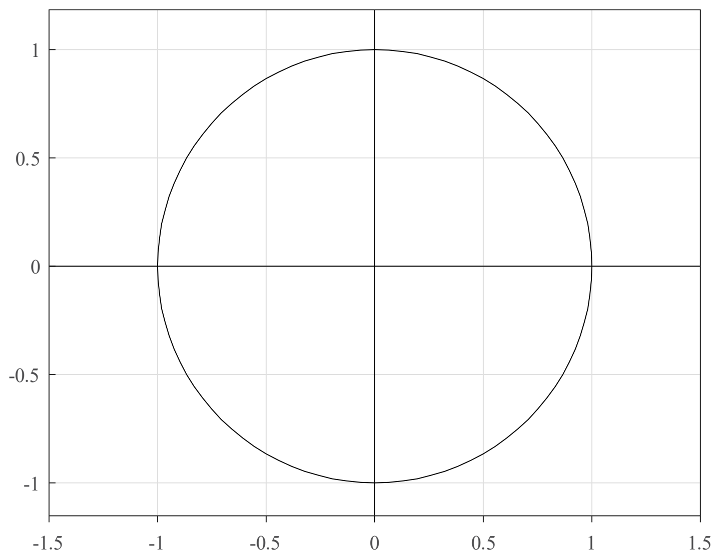
In class we designed a notch filter to remove a corrupting cosine tone and that's what is required here as well.

Deliverables:

I: Write a program that will

c) Sketch the pole-zero plot of a notch filter that will remove the tone.

d) Give the precise angle and magnitude of the poles and zeros.



6.3.3 Design the filter to remove the tone

Deliverables:

II: Write a program that will

e) Find values of *b* and *a* of a filter that will remove the tone from the sound file. Provide those values.

6.3.4 Test your filter

Test your filter with Matlab to confirm that your filter worked

```
y = filter(b, a, x);
soundsc(y, fs);
```

Upload printouts of all the code you used to plot the spectrum, find the frequency of the tone, find the *a* and *b* values of the filter.