Digital Signal Processing (18-491/18-691)
Spring Semester, 2025

Problem Set 3

Issued: 1/30/25

Due: 2/7/25 at 0100 via Gradescope

Note: You will be submitting your homework using Gradescope. Write your homework on paper, scan your results, and upload them to the gradescope site.

Reading: During the past week we began our discussions of basic properties of z-transforms, focussing on the nature of the time function and the shape of the region of convergence (ROC), following Secs. 3.0-3.3 in the text. In the week to come we will continue this discussion, focussing on z-transform properties and the computation of inverse z-transforms using the partial fraction method and linear constant-coefficient difference equations, and the development of the magnitude and phase of the DTFT from the locations of poles and zeros in the z-plane. We also discussed the impact of pole/zero locations on frequency response, focusing on allpass systems, minimum and maximum-phase systems, zero-phase systems, and linear-phase systems. This material is discussed in OSYP Secs. 3.3-3.5 and 5.5-5.7, as well as the lecture notes on frequency response from pole/zero locations.

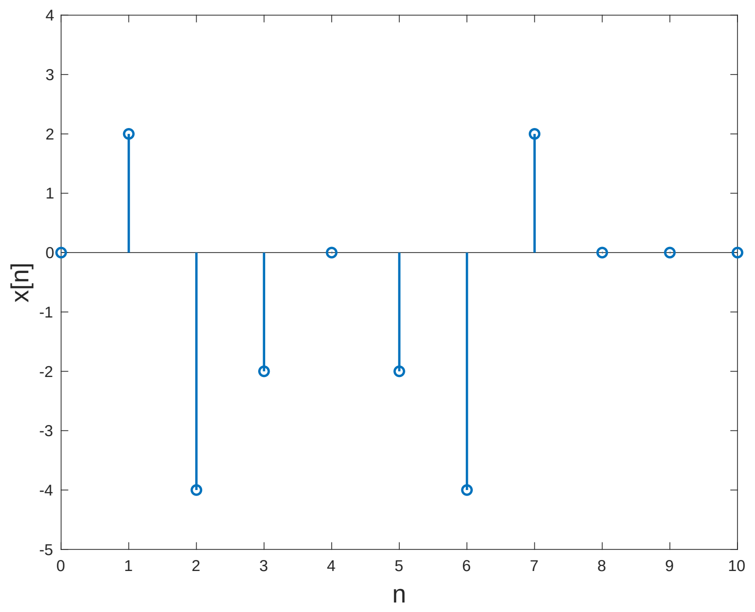
This material is also reviewed more compactly in the Notes on Z-Transform Properties and Inverses posted on the Lectures page of the course website. **Some reminders:**

- Remember that if the input of an LSI system is $x[n] = e^{j\omega_0 n}$, the output will be

  $$y[n] = e^{j\omega_0 n} H(e^{j\omega 0})$$

- While we will discuss z-transform properties in some detail, the properties below may be useful in working some problems on the problem set. Note that Table 3.2 on OSYP page 132 provides a more complete list of properties.
  - $x_1[n] + x_2[n] \Leftrightarrow X_1(z) + X_2(z)$ where the ROC is the overlap of the ROCs of $X_1(z)$ and $X_2(z)$.
  - $x[n-N] \Leftrightarrow z^{-N}$ where the ROC is is the same as the ROC of $x[n]$.
  - $x[n] * h[n] \Leftrightarrow X(z)H(z)$ where the ROC is the overlap of the ROCs of $X(z)$ and $H(z)$.

Problem 3.1:



Notes: All solutions in this problem exploit the symmetry properties of DTFTs. They also make use of some of the relationships below:

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|X(e^{j\omega})\right|^2 d\omega \qquad \text{(Parseval's theorem)}$$

$$x[0] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})d\omega$$

$$X(e^{j0}) = \sum_{n=-\infty}^{\infty} x[n]$$

Consider the time function $x[n]$ depicted above:

$$x[n] = 2\delta[n-1] - 4\delta[n-2] - 2\delta[n-3] - 2\delta[n-5] - 4\delta[n-6] + 2\delta[n-7]$$

Note that $x[n]$ can be thought of as an even function that is shifted in time. The following questions can be answered using DTFT properties including the ones cited above. There is no need to solve them directly.

(a) Determine the value of $X(e^{j\omega})$ for $\omega = 10\pi$

(b) Determine the value of $X(e^{j\omega})$ for $\omega = \pi/2$

(c) Determine $\angle X(e^{j\omega})$

(d) Determine the value of $\frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})d\omega$

(e) Determine the value of $\frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega})|^2 d\omega$

(f) Determine and sketch the signal whose DTFT is the real part of $X(e^{j\omega})$

Problem 3.2: Determine the z-transform of each of the following sequences. Sketch the pole-zero plot and indicate the region of convergence. Indicate whether or not the Fourier transform of the sequence exists. Use of the properties of the z-transforms will make many of the solutions much easier.

(a) $\left[\left(\frac{1}{5}\right)^n + \frac{1}{4}\left(\frac{-2}{3}\right)^n\right] u[n]$

(b) $3^n u[4-n]$

(c) $\left(\frac{1}{4}\right)^{n-3} u[n-3] + \left(\frac{1}{2}\right)^{-n+2} u[-n+2]$

(d) $n\left(\frac{1}{3}\right)^n u[n+3]u[1-n]$

Be sure to check both the transform tables and properties here. You may leave your answer in the form of the convolution of two functions, but you must end up with an analytical result, not in polynomial form.

(e) $-\,(2)^{-|n-4|}$

(f) $\left(\frac{1}{5}\right)^{n-4} \cos(5\pi(n-4)/4)u[n-4]$

Problem 3.3: Consider a linear shift-invariant system with transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^3 - 2}{\left(z - \frac{1}{5}\right)\left(z - \left(\frac{-1}{4} + j\frac{1}{4}\right)\right)\left(z - \left(\frac{-1}{4} - j\frac{1}{4}\right)\right)}$$

(a) Sketch the pole-zero diagram associated with this transfer function.

(b) List all the possible regions of convergence that could be associated with $H(z)$. For each possible ROC, indicate whether the corresponding unit sample response $h[n]$ is right-sided, left-sided, both-sided, or finite in duration.

(c) Suppose that you are told that the system is actually causal. Is it stable as well? Why or why not?

(d) As we have discussed in class, we can also write the transfer function as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 - 2z^{-3}}{\left(1 - \frac{1}{5}z^{-1}\right)\left(1 - z^{-1}\left(\frac{-1}{4} + j\frac{1}{4}\right)\right)\left(1 - z^{-1}\left(\frac{-1}{4} - j\frac{1}{4}\right)\right)}$$

By cross-multiplying the two expressions on the right and taking the inverse transform of the result of the cross-multiplication, obtain a difference equation that expresses the current output $y[n]$ as a linear combination of previous outputs and current and previous inputs. All the coefficients of your difference equation should be real.

Hint: Remember the property $x[n-N] \Leftrightarrow X(z)z^{-N}$

(e) Note that, the initial conditions for this difference equation must be zero in order for the system to be both linear and shift invariant. Specifically, if the system input $x[n]$ begins at $n = 0$ the initial conditions would be that all previous values of $y[n]$ that are needed to calculate the output $y[n]$. (This applies to only a finite number of previous values of $y[n]$.)

If the input to this system is applied at $n = 0$, what are the required initial conditions?

(f) As you know, the unit sample response of the system $h[n]$ is the inverse z-transform of the transfer function $H(z)$. We began a discussion Monday about a set of general procedures involving partial fractions that enable us to compute inverse z-transforms when the transfer function is of the form of a ratio of polynomials in z or z^{-1} as is the case in the present example (and most situations that you will encounter in this course).

An alternate way of obtaining $h[n]$ on a sample-by-sample basis is to calculate the system output $y[n]$ when the input is $x[n] = \delta[n]$. This can be done using *iteration*, which is described in Sec. III of the z-transform notes and which was reviewed briefly last Monday. Using the iteration techniques, obtain numerical values of $h[n]$ for n equal to 0 through 2.

Problem 3.4:

In Problem 1.4c and again in Problem 2.2 we considered the convolution of the following two functions:

$$x[n] = (1/3)^{n-1}u[n-1] \text{ with } h[n] = (1/3)^{-n+1}u[-n+1]$$

As you know from the homework solutions, the result of this convolution is

$$y[n] = \begin{cases} \left(\frac{1}{8}\right)\left(\frac{1}{3}\right)^{-n}, & < 2 \\ \left(\frac{81}{8}\right)\left(\frac{1}{3}\right)^{n}, & n \geq 2 \end{cases}$$

(a) Obtain $X(z)$ and $H(z)$, the z-transforms for $x[n]$ and $h[n]$. Be sure to specify the regions of convergence.

(b) Obtain $Y(z)$, the z-transform of $y[n]$, and its region of convergence.

(c) Show that $Y(z) = X(z)H(z)$ with the appropriate region of convergence.

Problem 3.5: Listed below are three z-transforms:

1. $$X_1(z) = \frac{1}{1 + \frac{1}{4}z^{-1}} \text{ for } |z| > \frac{1}{4}$$

2. $$X_2(z) = \frac{1}{1 + \frac{1}{4}z^{-1}} \text{ for } |z| < \frac{1}{4}$$

3. $$X_3(z) = \frac{1 - 3z^{-1}}{\left(1 + \frac{1}{4}z^{-1}\right)\left(1 - \frac{1}{3}z^{-1}\right)} \text{ for } \frac{1}{4} < |z| < \frac{1}{3}$$

For each of the z-transforms above, determine the inverse z-transform using partial fraction expansion. (This is trivial in the first two cases.) For the first two functions, also determine the first three nonzero terms of the inverse z-transform using the "long-division" method. Convince yourself that the two methods produce equivalent answers.

MATLAB Problems

Note that the MATLAB functions `zplane`, `roots`, `poly`, `residuez`, and `freqz` are especially useful in dealing with some of the issues addressed in this weeks problem set. Look over the help files for these MATLAB functions as they will be very useful in the weeks to come.

For Python users: The recommended general local Python configuration is

```
- Python Version: v3.10.x
- Libraries & Recommended Versions
   - SciPy: 1.10.0
   - NumPy: 1.24.1
   - Matplotlib: 3.6.3
```

Some general recommendations for Python users include:

- Store signals as `numpy` arrays
- Read the online documentations of the functions we use to understand the differences between the MATLAB and Python implementations (although they are mostly like-for-like, nuanced differences may exist for a few of these functions)
- You would be able to find Python equivalents for most MATLAB functions by searching for `Python equivalent for <MATLAB_function_name>` online. `Stack overflow` often helps as well.

Problem C3.1:

In this short problem we will compensate for simple echos in audio signals using an inverse filter.

A very simple model of a single echo is the expression

$$x_e[n] = x[n] + \alpha x[n - N]$$

Included in the `h3_extras` folder are two files, `welcome16.wav` and `xe.wav`, where `xe.wav` was generated with the parameter `N` corresponding to 40 ms and $\alpha = .4$. `xe.wav` was generated with the following script:

```
[x,fs] = audioread('welcome16.wav');

% set and echo amplitude parameters
alpha = .4;
N = round(.04*fs); % number of samples in .04 seconds

h = [1 zeros([1,N-1]) alpha]; % effective unit sample response of echo "filter"
xe = conv(x,h);
audiowrite('xe.wav',xe,fs);
soundsc(xe,fs)
```

Alternate starter code for Python users:

```
C3.1a
   import numpy as np
   from scipy.io import wavfile

   fs, x = wavfile.read('welcome16.wav')
   # set echo and amplitude parameters
   x_max = x.max()
   alpha = 0.4
   N = np.rint(0.04 * fs).astype(np.int32)  # number of samples in .04 seconds
   h = np.hstack((1.0, np.zeros(N - 2), alpha))  # effective unit sample response of echo "fil
   xe = np.convolve(x, h).astype(np.int16)  # converting from float32 to int16 to retain origi
   wavfile.write('xe.wav', fs, xe)  # saving the file instead of playing it
```

An *inverse filter* is an LSI system that is the reciprocal of a given filter:

$$H_I(z) = \frac{1}{H(z)}$$

where $H_I(z)$ is (you guessed it) the inverse of $H(z)$. Because the product of $H(z)$ and $H_I(z)$ is 1, the result of passing a signal through the cascade of any filter and its inverse is in principle the original signal once again. In practice though, there is no guarantee that the inverse of a causal and stable LSI system is also stable and causal.

(a) In this section we will explore some attributes of inverse filtering in this case.

1. Suppose an arbitrary system $H(z)$ is known to be causal and stable. What is the constraint on the location of the poles and zeros of $H(z)$ that guarantees that its inverse is also causal and stable?
2. What is the z-transform of the filter $H(z)$ that takes $x[n]$ as the input and produces $x_e[n]$ as the output?
3. What is the z-transform of the *inverse filter* $H_I(z)$ that takes $x_e[n]$ as the input and produces $x_r[n]$ as the output, where $x_r[n]$ is intended to be a close approximation to $x[n]$?
4. Write the difference equation of the system defined by $H_I(z)$.
5. Use the MATLAB `filter` command to implement the inverse filter $H_I(z)$, taking `xe.wav` as the input and outputting `xr.wav` as the output. (Be sure to read the MATLAB `help` file for `filter` to specify the arguments correctly.) **Python users:** You can use Python's `scipy.signal.lfilter` instead of MATLAB's `filter` routine.
6. Use the MATLAB `soundsc` routine to play the recovered signal. To what extent does $x_r[n]$ sound like the original $x[n]$? **Python users:** You can use `scipy.io.wavfile.write(destination file name.wav,fs,x)` instead of MATLAB's `soundsc`.

(b) The approach above relied on our having knowledge of the echo amplitude and delay parameters. In this part of the problem we will explore the use of the *autocorrelation function* of $x_e[n]$ to estimate the time delay of the echo. The autocorrelation function of a finite-energy deterministic signal $x[n]$ is defined to be

$$\phi_{xx}[m] = \sum_{n=-\infty}^{\infty} x[n]x[n+m]$$

The autocorrelation function $\phi_{xx}[m]$ is a measure of how much the time function $x[n]$ resembles itself in the past and in the future. It is sometimes convenient to think of the autocorrelation function as the convolution of $x[n]$ with $x[-n]$. **Python users:** You can use `numpy.correlate(x,y,"full")` instead of MATLAB's `xcorr` routine. Do not change the third argument. The default value for this argument will not give the same result as MATLAB's `xcorr`.

The autocorrelation function of $x[n]$ can be implemented directly in MATLAB using `conv` or by using the MATLAB function `xcorr`. For example, you can plot the first 10,000 lags of the autocorrelation function of $x[n]$ using the command

```
[c,lags] = xcorr(x,x,10000);
stem(lags,c)
```

1. Plot the autocorrelation functions of $x[n]$ and $x_e[n]$ and compare them carefully. Is there anything in the nature of the plots that can enable you to estimate the delay parameter N?

2. Is there anything in the nature of the plots that can enable you to estimate the attenuation parameter α?

3. A second version of the "Welcome" utterance, `xe2.wav` is also included in `h3_extras`. This signal also includes an echo, but with different and unknown delay and attenuation parameters. Estimate the delay and attenuation parameters as best you can from the autocorrelation function of $x_{e2}[n]$. Pass the echoed signal `xe2.wav'` through an appropriate inverse filter to produce the restored signal $x_{r2}[n]$. Use MATLAB to generate and save the signal `xr2.wav`. To what extent does $x_{r2}[n]$ sound like $x[n]$?

4. Do you think that this approach will work for all values of N and α? What are some practical limitations of its utility?

What to turn in: Turn in to Gradescope the following:

- **Written answers** for parts (a) 1, 2, 3, 4, and 6, and (b) 1, 2, and 4. Also include the plots for part (b) 1.
- **Code** for the answers for parts (a) 5 and (b) 3
- **.wav files** for `xr.wav` and `xr2.wav`