**Solution Set for Homework #6**

*By Prof. Brian Evans and Mr. Firas Tabbara*

October 22, 2021 ***Version 4.0***

**PROBLEM 1: FREQUENCY AND STEP RESPONSES**

For each of the following linear time-invariant (LTI) systems, determine the impulse response, step response, and frequency response. Plot the magnitude and phase of the frequency response using freqz:

1. First-order unnormalized averaging filter (lowpass filter): and the initial condition
2. First-order difference filter (highpass filter): and the initial condition
3. Second-order difference filter (highpass filter): and the initial condition

Chart

Description automatically generated**Solution for part (a) subpart 1:** The impulse response is the output (response) when the input is the discrete-time impulse . When , the output is

n = -2 : 5;   
h = [ 0 0 1 1 0 0 0 0 ];   
stem(n, h)   
ylim( [ -0.2 1.2] );

We compute several impulse response values:

*h*[-2] = d[-2] + d[-3] = 0 + 0 = 0

*h*[-1] = d[-1] + d[-2] = 0 + 0 = 0

*h*[0] = d[0] + d[-1] = 1 + 0 = 1

*h*[1] = d[1] + d[0] = 0 + 1 = 1

Chart

Description automatically generated  
**Solution for part (a) subpart 2:** The step response *y*step[*n*] is the output (response) when the input is the discrete-time step function *u*[*n*]. The unit step is *u*[*n*] = 1 for and 0 otherwise. When , the output is

Matlab code:

n = -2 : 5;   
unitstep = ( n >= 0 );   
% Output: 0 0 1 1 1 1 1 1   
unitstepdelayed = ( (n-1) >= 0 );   
ystep = unitstep + unitstepdelayed;   
stem(n, ystep);   
ylim( [ -0.2 2.2 ] );   
xlim( [ -2.2 5.2 ] );

**Solution for part (a) subpart 3:** The frequency response of the LTI system is the discrete-time Fourier transform of the impulse response:

We can factor the frequency response into magnitude-phase form per lecture slide 9-9 by factoring out the phase shift corresponding to the delay equal to the index of the midpoint of the impulse response, which is at an index of ½, and hence the phase shift is :

Chart

Description automatically generatedMagnitude response: which is non-negative for .

Phase response: which is a line of slope that passes through the origin.

The MATLAB command:

freqz( [1 1] );

will plot the magnitude and phase response for an LTI system with impulse response

The magnitude response, by default, will be plotted in decibels vs. discrete-time frequency, where *AdB* = 20 log10 |*A*| for , as shown above. It is a lowpass filter.

Chart, box and whisker chart

Description automatically generated**Solution for part (b) subpart 1:** The impulse response is the output (response) when the input is the discrete-time impulse . When , the output is

We compute several impulse response values:

*h*[-2] = d[-2] - d[-3] = 0 - 0 = 0

*h*[-1] = d[-1] - d[-2] = 0 - 0 = 0

*h*[0] = d[0] - d[-1] = 1 - 0 = 1

*h*[1] = d[1] - d[0] = 0 - 1 = -1

MATLAB code for the plot of :

n = -2 : 5;   
h = [ 0 0 1 -1 0 0 0 0 ];   
stem(n, h)   
ylim( [ -1.2 1.2] );

Chart

Description automatically generated**Solution for part (b) subpart 2:** The step response *y*step[*n*] is the output (response) when the input is the discrete-time step function *u*[*n*]. When , the output is

Matlab code:

n = -2 : 5;   
unitstep = ( n >= 0 );  
unitstepdelayed = ( (n-1) >= 0 );   
ystep = unitstep - unitstepdelayed;   
stem(n, ystep);   
ylim( [ -0.2 2.2 ] );   
xlim( [ -2.2 5.2 ] );

**Solution for part (b) subpart 3:** The frequency response of the LTI system is the Fourier transform of the impulse response:

Chart

Description automatically generatedWe can factor the frequency response into polar form per lecture slide 9-9:

freqz( [1 -1] );

Using the fact that ,

Amplitude Response: is non-positive for (which is not a magnitude) and non-negative for (which is a magnitude).

Phase Response: which is a line of slope that intercepts the y-axis at .

We can put into magnitude-phase form for for by multiplying the amplitude term by -1 and the phase term by :

A student noticed the similarity between the impulse responses between parts (a) and (b):

This is amplitude modulation which shifts the frequency content by p and -p.

Chart, box and whisker chart

Description automatically generated**Solution for part (c) subpart 1:** The impulse response is the output (response) when the input is the discrete-time impulse . When , the output is

We compute several impulse response values:

*h*[-2] = d[-2] – 2 d[-3] + d[-4] = 0 - 0 + 0 = 0   
*h*[-1] = d[-1] – 2 d[-2] + d[-3] = 0 - 0 + 0 = 0  
*h*[0] = d[0] – 2 d[-1] + d[-2] = 1 - 0 + 0 = 1

*h*[1] = d[1] – 2 d[0] + d[-1] = 0 - 2 + 0 = -2

*h*[2] = d[2] – 2 d[1] + d[0] = 0 - 0 +1 = 1

MATLAB code for the plot of :

n = -2 : 5;   
h = [ 0 0 1 -2 1 0 0 0 ];   
stem(n, h)   
ylim( [ -2.2 1.2] );

Chart, bubble chart, box and whisker chart

Description automatically generated**Solution for part (c) subpart 2:** The step response *y*step[*n*] is the output (response) when the input is the discrete-time step function *u*[*n*]. When , the output

Matlab code:

n = -2 : 5;

unitstep = ( n >= 0 );

unitstepdelayed1 = ( (n-1) >= 0 );

Chart, line chart

Description automatically generatedunitstepdelayed2 = ( (n-2) >= 0 );   
ystep = unitstep – 2\*unitstepdelayed1 + unitstepdelayed2;   
stem(n, ystep);   
ylim( [ -2.2 2.2 ] );   
xlim( [ -2.2 5.2 ] );

**Solution for part (c) subpart 3:** The frequency response of the LTI system is the Fourier transform of the impulse response:

The MATLAB command:

freqz( [1 -2 1] );

**PROBLEM 2: CASCADE OF THREE SYSTEMS**

Suppose that three systems are connected in cascade; i.e., the output of is the input to , and the output of is the input to . The three systems are specified as follows

where the output of is and its input is .

1. Determine the equivalent system that is a single operation from the input (into ) to the output , which is the output of . Thus, is and is .
2. Use the frequency response to write one difference equation that defines the overall system in terms of and only.

**Solution to (a):** Here’s a block diagram of the cascade of the three systems.

**Rectangle

Description automatically generated with low confidence**

*Approach #1 Using Impulse Responses.* We find the impulse response of the overall system by letting . The output of LTI System #1 will be its impulse response . The signal is the input to LTI System #2 and the output is where \* denotes convolution. (See Lecture Slides 8-8 and 8-11 and SP First Sections 5-6 and 5-7.) The response is the input to LTI System #3 and the output is The impulse responses are

We know that the length of the signal that results from the convolution of two finite-length signals is the sum of the lengths of the two finite-length signals. Here, is of length 2 samples and is of length 3 samples, and hence, the result will have 4 samples:

The discrete-time impulse is 1 for and 0 otherwise. The product is 1 when and the product simplifies to :

Convolution for finite-length signals is reminiscent of polynomial multiplication. In fact, the Matlab command conv can also be used to compute the product of two polynomials.

We could check our answers for the convolution calculations in MATLAB:

h1Convh2 = conv( [1 -1], [1 0 1] ); %% gives 1 -1 1 -1

h = conv(h1Convh2, [0 1 1] ); %% gives 0 1 0 0 0 -1

which gives [ 0 1 0 0 0 -1 ] or equivalently .

*Approach #2 Rewriting Equations*. The output of the second system is the input of the third system, :

The output of the first system is the input of the second system:

The equivalent system that is a single operation from the input (into ) to the output , which is the output of , is the following:

**Solution for part (b):** The frequency response form of the systems 1, 2, and 3 are shown below:

The three systems are connected in cascade. Therefore,

Converting back:

**Epilogue**: In the time domain, the response (output) of an LTI system will be the convolution of the input signal and the system’s impulse response. In the frequency domain, the output of the LTI system will be the product of the input frequency content and the system’s frequency response. The discrete-time Fourier transform can compute the frequency content of a periodic or aperiodic signal. It is a generalization of the discrete-time Fourier series.

The convolution of two finite-length signals can be computed using polynomial multiplication. That is, each finite-length signal can be represented as a finite-length vector of amplitude values, and the process of convolving these vectors is the same as polynomial multiplication when interpreting the vectors as polynomial coefficients. See the Epilogue for Problem 3 which talks about the same connection in the z-domain, where it might be easier to see.

**PROBLEM 3: TRANSFER FUNCTIONS IN THE DOMAIN**

For each of the following linear time-invariant (LTI) systems, derive the transfer function, compute the poles and zeros, and plot the poles and zeros using zplane:

1. First-order unnormalized averaging filter (lowpass filter): and the initial condition
2. First-order difference filter (highpass filter): and the initial condition
3. Second-order difference filter (highpass filter): and the initial condition

**Solution for part (a):**  for and as a necessary condition for the system to be at rest.The impulse response is:

By performing the z-transform of the impulse response, we can calculate the transfer function:

The pole (root of the denominator) is at *z* = 0, and the zero (root of the nominator) is at *z* = -1.

Using zplane, we can plot zeros and poles:

zplane([1 1])

In the plot above, a pole is shown by × and a zero is depicted by o; hence, the system has one pole at *z* = 0 and one zero at *z* = -1

**Solution for part (b):** With



the impulse response is

.

The transfer function is:

Therefore, system has one pole at *z* = 0,

and one zero at *z* = 1.

MATLAB code:

zplane([1 -1])

****

**Solution for part (c):** With



the impulse response is



and the transfer function is:

System has two poles at *z* = 0, and two zeros

at *z* = 1:

zplane([1 -2 1])

**Epilogue:** Armed with the z-transform, we’ll take another look at the connection between convolution and polynomial multiplication mentioned in the Epilogue in Problem 2. From Problem 2, let’s compute the convolution of and using *z*-domain techniques: