EE 419 - Project 2

**Developing a Filter Analysis Program in Matlab**

|  |  |
| --- | --- |
| **Names: Chris Adams, Aiku Shintani** | **Lab Date: 1/15/19** |
| **Bench #: 9** | **Section: 2** |

**Learning Objectives:**

**Use Matlab to quantify the performance of a discrete-time filter design, including:**

* **Pole-zero diagram**
* **Frequency Response plots (linear and dB magnitude and phase)**
* **Unit Sample Response**
* **Algorithmic determination of filter type**
* **Numeric quantification of filter bandwidth**

**1) [Matlab] Creating a Unit Sample Response sequence**

**[hn, n]=unit\_sample\_response(*Bk, Ak, number\_of\_samples, figure\_number*);**

Where the values returned are:

hn – has the unit sample response sequence values

n – has the corresponding sample indices (starting at 0);

**Function Specifications:** Your Matlab function should

* Create a unit sample sequence using the function unit\_sample( ) you created last week.
* Apply the unit sample sequence to a filter defined by the Ak and Bk coefficients.
* Create a stem plot of the response hn vs. n, in Figure # = *figure\_number*
  + with small dots ‘.’ at ends of the stems
  + with a title and all axes labeled

**Test Case:** y[n] + 0.8 y[n-1] = x[n]

**Test Results:**

****

**Matlab Code:**

% This function takes 4 inputs: the Bk and Ak coefficients for a

% difference equation, the number\_of\_samples (n), and figure #.

% The function returns the unit sample response as a figure.

function [hn, n] = unit\_sample\_response(Bk, Ak, number\_of\_samples, figure\_number)

[dn, n] = unit\_sample(number\_of\_samples); %get arrays of sample index

hn = filter(Bk, Ak, dn); %filter the Z domain tf to the time domain

figure(figure\_number)

%make stem plot of unit sample response

stem(n, hn, '.', 'MarkerSize', 20, 'Linewidth', 2);

xlabel('Sample index')

ylabel('Unit Sample Response')

title('Unit Sample Response')

end

**2) [Matlab] Analyzing and Plotting the Frequency Response of a Filter**

**[poles,zeros,HF,Fd,hn,n] = show\_filter\_responses(*Ak,Bk,fsample,***

***num\_of\_f\_points,num\_of\_n\_points,figure\_num*);**

where the arguments are:

*Ak* = a list of the Ak coefficients of the filter difference equation (coefficients of the “y” terms)

*Bk* = a list of the Bk coefficients of the filter difference equation (coefficients of the “x” terms)

*fsample* = sampling frequency (samples / second)

*num\_of\_f\_points* = the # of points for the freq. response plot

*num\_of\_n\_points* = the # of points for the unit sample response plot

*figure\_num* = number of the 1st figure to use for plots

and the function returns the following information:

poles = a list of the complex pole locations (z values) for the Transfer Function (the roots of the H(z) denominator polynomial)

zeros = a list of the complex zero locations (z values) for the Transfer Function (the roots of the H(z) numerator polynomial)

HF = the complex DTFT frequency response values (linear scale)

Fd = digital frequencies that match the freq response values

hn – has the unit sample response sequence values

n – has the corresponding sample indices (0 to [*num\_of\_n\_points* – 1]);

This function should create the following system response plots:

* A “Pole/Zero” diagram
* Two Frequency Response plots of H(F) vs F

1. H(F) vs F (digital frequency): Using a digital frequency axis.
2. H(F) in dB vs *f*analog

* A Unit Sample Response Plot h[*n*] vs *n*

**Test Case:** y[n] + 0.1 y[n-1] = x[n]+ x[n-2] Assume a sampling rate of 1 KHz.

**Analysis:**

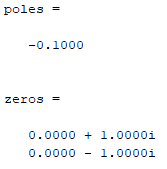
* What are the Filter Difference Equation Coefficient Ak and Bk values?

**Ak = [1, 0.1 ] Bk = [ 1, 0, 1 ]**

* What is the Transfer Function *H*(*z*) for this filter?



* What are the values of the Poles and Zeros of this filter?
  + Computed by hand, factoring H(z): **Poles @ z= [ 0, -0.1 ] Zeros @ z= [ j, -j ]**
  + Computed by your Matlab function:



**Test Results:**

(Paste all the Figures produced by your function for the test case HERE)

****

****

****

****

**3) [Lab / Matlab] Analyzing and Plotting the Frequency Response of a Filter**

**[Ak,Bk,HF,Fd,hn,n]=show\_filter\_responses\_pz(*poles,zeros,K,fsample,num\_of\_f\_points,***

***num\_of\_n\_points,figure\_num*);**

where the arguments are:

*poles* = a list of the Z plane locations (complex values) for all the POLES of the filter (a row vector)

*zeros* = a list of the Z plane locations (complex values) for all the ZEROS of the filter (a row vector)

*K* = Multiplier constant for the transfer function (which you should multiply H(z) by)

*fsample* = sampling frequency (samples / second)

*num\_of\_f\_points* = the # of points for the freq. response plot

*num\_of\_n\_points* = the # of points for the unit sample response plot

*figure\_num* = number of the 1st figure to use for plots

and the function returns the following information:

Ak = a list of the Ak coefficients of the filter difference equation (coefficients of the “y” terms)

Bk = a list of the Bk coefficients of the filter difference equation (coefficients of the “x” terms)

HF = the DTFT frequency response values (linear scale)

Fd = digital frequencies that match the freq response values

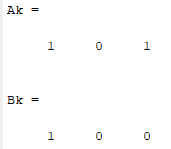
hn – has the unit sample response sequence values

n – has the corresponding sample indices (0 to [*num\_of\_n\_points* – 1]);

**Test Case:** y[n] = x[n] + x[n-2] Assume a sampling rate of 1 KHz.

**Analysis:**

* What are the Filter Difference Equation Coefficient Ak and Bk values?
  + From the difference equation: **Ak = [ 1, 0, 0 ] Bk = [ 1, 0, 1 ]**
  + Computed by your Matlab function:



* What is the Transfer Function *H*(*z*) for this filter?



* What is the overall Transfer Function Gain (‘K’) value? **K =** 1
* What are the values of the Poles and Zeros of this filter?
  + Computed by hand, factoring H(z): **Poles @ z= [ 0, 0 ] Zeros @ z= [j, -j ]**

**Test Results:**

****

****





**4) [Matlab] Additional Analysis of the Responses of a Digital Filter**

**Filter Analysis Features**

* + Find the **Peak magnitude response value**, and the **digital frequency** at which it occurs
  + Find the **Minimum magnitude response value,** and its **digital frequency**
  + **Maximum attenuation** of the filter (dB difference between max and min responses)
  + **Magnitude response at the 3 dB cutoff frequency** (linear magnitude value that is 3 dB below the peak magnitude value).
  + **Determine the type of filter**: *i.e.,* low-pass, high-pass, band-pass or band-stop (notch) filter.
  + **Determine the 3 dB cutoff frequency** (or frequencies) F3dB
  + Determine the (one-sided) **3dB bandwidth** of the filter (as appropriate for the filter type).

**Test Cases**:

**a) y[n] = x[n] - 0.9 y[n-1]**

**b) y[n] = 0.5 x[n] + 0.5 x[n-2]**

**c) y[n] = 0.5 x[n] + 0.5 x[n-1]**

**d) Poles at z=[0.7\*j -0.7\*j], Zeros at z=[-1 1], K=1**

**Summarized Test Results:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Filter** | **Type** | **Peak**  **Response** | **Peak**  **Freq**  **(cyc/samp)** | **Min**  **Response** | **Min**  **Freq**  **(cyc/samp)** | **Max**  **Atten.**  **(dB)** | **3dB**  **Mag** | **3dB**  **Freq**  **(cyc/samp)** | **3dB**  **BW**  **(cyc/samp)** |
| **a** | **HPF** | **9.982** | **0.5** | **0.5263** | **0** | **25.56** | **7.07** | **0.485** | **0.015** |
| **b** | **Notch** | **1** | **0** | **0** | **0.2505** | **Inf** | **0.7079** | **0.1253, 0.3758** | **0.2505** |
| **c** | **LPF** | **1** | **0** | **0.0031** | **0.5** | **50.06** | **0.7079** | **0.2495** | **0.2495** |
| **d** | **BPF** | **3.922** | **0.2505** | **0** | **0** | **Inf** | **2.776** | **0.1984, 0.3026** | **0.1042** |

**Individual Test Case Results:** *(Frequency Response plots with data cursor displays confirming above results)*

**Test Case** **a: y[n] = x[n] - 0.9 y[n-1]**

*(Paste Plot(s) with Data Cursor annotations HERE)*

**Test Case** **b: y[n] = 0.5 x[n] + 0.5 x[n-2]**

*(Paste Plot(s) with Data Cursor annotations HERE)*

**Test Case** **c: y[n] = 0.5 x[n] + 0.5 x[n-1]**

*(Paste Plot(s) with Data Cursor annotations HERE)*

**Test Case** **d: Poles at z=[0.7\*j -0.7\*j], Zeros at z=[-1 1], K=1**

*(Paste Plot(s) with Data Cursor annotations HERE)*

**Final (Complete) Matlab Code for Each Function:**

(Paste Matlab Code HERE)

**5) [Matlab] Using Matlab’s Filter Analysis Tool**

fvtool(B,A) launches the Filter Visualization Tool and computes the

Magnitude Response for the filter defined by numerator and denominator

coefficients in vectors B and A.

**Test Case:**  **y[n] = ¼ x[n] - ¼ x[n-2] + ¼ y[n-1] – ½ y[n-2]**

**Test Results** (plots from the fvtool):

* Frequency Response Magnitude and Phase Responses (on single plot)

**

* Group Delay



* Step Response

**

* Pole/Zero Diagram

**