INTRODUCTION

Impulse Response and Frequency Response

The response of a system can be specified in the time domain as well as in the frequency domain. In a recursive system, the impulse response is equal to the zero-state response of the system when the input $x(n) = \delta(n)$ and the system is initially relaxed. In the frequency domain, by knowing the frequency response, we are able to determine the response of the system to any sinusoidal input signal.

Poles and Zeros

By examining the transfer function of a filter, we can learn a lot about its magnitude and phase response. An easy way to achieve this is to plot the poles and zeros on the s-plane for a continuous time system or on the z-plane for a discrete time system. The magnitude and the phase response can then be determined by examining the positions of the poles and zeros relative to the frequency scale. In this laboratory, simple first order digital filter designs are explored. Pole-zero diagrams are also used to examine different filter designs, such as minimum phase, maximum phase, and all-pass filters.

Before attending this laboratory, you should understand the following concepts:

- Pole-zero plots in the s-domain and the z-domain (for continuous time and discrete time signals).
- Minimum phase, Mixed phase, Maximum phase filters.
- All-pass filters, Linear phase filters.
- Stability of system in relation to the pole positions.
- Magnitude and phase response obtained from pole-zero diagrams.
- Impulse response and difference equations.

LABORATORY PREPARATION

This laboratory is purely based on programming in MATLAB. You should try and program as much as possible before coming to the laboratory session. Some students might find it easier to save their initial program code in an m-file on a floppy disk and bring it to the laboratory for further work during the laboratory session.

MATLAB has a very informative Help Desk files in HTML format [Go to Help → Help Desk (HTML)]. You should be able to find help for the functions used in all our laboratory exercises. [Within the MATLAB Help Desk, one might find it easier to look for help under
the “MATLAB functions by Index” hyperlink, “Signal Processing Toolbox Ref” hyperlink (for discrete time signal processing functions) or “Control System Toolbox Ref” hyperlink (for some continuous time signal processing functions).

Alternatively, you can type `help` or `helpwin` or `doc` followed by the name of the function directly from the MATLAB Command Window. You also might want to refer to Lab 0 Matlab Exercises.

**LABORATORY WORK**

The following filter is to be implemented in MATLAB:

\[ \sum x(n) \rightarrow \begin{array}{c} -b_1 \ \ \ Z^{-1} \\ -b_2 \ \ \ Z^{-1} \end{array} \rightarrow y(n) \]

The difference equation for this filter is:

\[ y(n) = x(n) - b_1 y(n - 1) - b_2 y(n - 2) \]

For this experiment, \( b_1 \) and \( b_2 \) can take the following values:

<table>
<thead>
<tr>
<th>TYPE</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-1.343</td>
<td>0.9025</td>
</tr>
<tr>
<td>B</td>
<td>-1.414</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>-1.487</td>
<td>1.1025</td>
</tr>
</tbody>
</table>

**Section I: Impulse Response**

Calculate the impulse response of each of the above filter (i.e., Type A, B and C) by setting up the input as an impulse of 512 samples in length. Apply this to the filter and plot its impulse response.

*You must implement the filter using loop functions (e.g. using a `for loop` or a `while loop`).*

**Question 1:** Are they FIR filters or IIR filters? How can you tell by the MATLAB plots? How can you tell without the use of MATLAB plots?

**Question 2:** Comment on the shape of the impulse response in relation to the location of the poles. (You can plot the locations of the poles by using the `zplane` function). Are they all stable?

**Section I Checkpoint**

Marks: 2 out of 10
Section II: Frequency Response

The frequency response can be calculated by taking the Fourier Transform of the impulse response. Get the frequency response of the filter (Type A only) by taking the Fast Fourier Transform of the impulse response. Use MATLAB’s built-in function `fft()` to perform this task. Plot the output.

Note: `fft()` in MATLAB returns a complex number. The frequency response we are after is the magnitude response. Hence outputs of `fft()` have to be modified into the correct form before plotting. Use `abs` function to get the magnitude. Refer to Lab 0 Matlab Exercise.

**Question 3:** Comment on the shape of the frequency response in relation to the location of the poles.

Plot the phase response of each filter below the magnitude response. This can also be done by using `fft()`, again manipulation of the outputs is necessary before plotting. Remember to label all axes.

**MATLAB FUNCTIONS**

MATLAB has some very useful signal processing tools. To get a frequency response of a system, the `freqz()` command can be used. The prototype for the `freqz()` command is

\[ \text{freqz(numerator, denominator)} \]

where the `numerator` is the numerator of the system transfer function and the `denominator` is the denominator of the system transfer function.

Find the transfer function of the above filter. Use `freqz()` to plot the magnitude response for the Type A filter.

**Question 4:** What are the 2 plots that `freqz()` generates? What are the axes?
Section III: Filtering of Signals
Set up a square pulse that is 32 samples long in total. Make it high for 10 samples and zero elsewhere.

\[
\begin{array}{c}
\text{1} \\
\text{0}
\end{array}
\quad \begin{array}{c}
\text{32 samples} \\
\text{10 samples}
\end{array}
\]

Filter this signal through filter Type B. Use a for loop to perform the filtering task. Set the initial conditions as \( y(-1) = 0 \) and \( y(-2) = 0 \). Plot the output.

Question 5: In Section I, the impulse response of Type B is stable. However, this is not the case for all bounded signals, e.g., in this case, a square pulse. Examine the poles of this filter and comment on its stability.

Question 6: What are some practical uses of this filter?

Section III Checkpoint
Marks: 1 out of 10

Section IV(a): First order Digital Filter Design

Two first order digital filters are given as follows:

\[
H(z) = \frac{z \cdot a}{z}
\]

\[
G(z) = \frac{z}{z \cdot a}
\]

There are many MATLAB functions that facilitate digital filter designs. Use the following commands to plot the pole-zero diagrams of the above filters, \( H(z) \) and \( G(z) \), for both \( a=0.9 \) and \( a=-0.9 \).

- \texttt{zplane(zeros,poles)}
- \texttt{freqz(numerator,denominator)}

Examine the MATLAB frequency plots and classify each of the above filters as Low Pass Filter (LPF) or High Pass Filter (HPF) for both \( a = 0.9 \) and \( a = -0.9 \).
**Question 7:** Which ones are Finite Impulse Response (FIR) filters and which ones are Infinite Impulse Response (IIR) filters?

<table>
<thead>
<tr>
<th></th>
<th>H(z)</th>
<th>G(z)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a = 0.9</td>
<td>Pole / Zero Diagram</td>
<td>LPF or HPF, FIR or IIR</td>
</tr>
<tr>
<td>a = -0.9</td>
<td>Pole / Zero Diagram</td>
<td>LPF or HPF, FIR or IIR</td>
</tr>
</tbody>
</table>

**Section IV(a) Checkpoint**

**Marks: 1 out of 10**

**Section IV(b): All Pass Filter / Linear Phase Filter**

(i) **All Pass Filter**

In this example, the digital filter has 4 poles and 4 zeros. Their locations are given as follows:

<table>
<thead>
<tr>
<th>Poles</th>
<th>Zeros</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75 ± j 0.25 ; -0.5 ± j 0.5</td>
<td>1.2 ± j 0.4; -1 ± j</td>
</tr>
</tbody>
</table>

Use the commands in Section I to generate a pole-zero plot and a frequency response plot. Comment on the magnitude response of this filter.

*Hint: You might find the MATLAB function `poly()` useful here and in all of the remaining sections.*

**Question 8:** What is the relationship between the poles and zeros?
(ii) Linear Phase Filter
In this example, the digital filter has 8 zeros (and 8 trivial poles located at 0). Their locations are given as follows:

<table>
<thead>
<tr>
<th>Zeros</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75 ± j 0.25 ; -0.5 ± j 0.5; 1.2 ± j 0.4; -1 ± j</td>
</tr>
</tbody>
</table>

Use the commands in Section I (a) to generate a pole-zero plot and a frequency response plot. Comment on the phase response of this filter.

Question 3: What is the special relationship in regards to the location of the zeros?

Question 4: Why is this class of filters useful in practice?

Section IV (b) Checkpoint  
Marks: 2 out of 10

Section IV (c): Minimum Phase / Mixed Phase / Maximum Phase Filters
In the following examples, the digital filters have 4 zeros each (and 4 trivial poles located at 0). Their locations are given as follows:

<table>
<thead>
<tr>
<th>Zeros Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter A</td>
</tr>
<tr>
<td>0.75 ± j 0.25 ; -0.5 ± j 0.5</td>
</tr>
<tr>
<td>Filter B</td>
</tr>
<tr>
<td>1.2 ± j 0.4 ; -0.5 ± j 0.5</td>
</tr>
<tr>
<td>Filter C</td>
</tr>
<tr>
<td>1.2 ± j 0.4; -1 ± j</td>
</tr>
</tbody>
</table>

Use the commands in Section I (a) to generate a pole-zero plot and a frequency response plot for each of the filters (i.e. Filter A, Filter B and Filter C).

Question 9: Are there any differences in their magnitude responses?

Question 10: Comment on the phase lag of each of the filter.

Question 11: Classify each of the above filters as either Maximum Phase, Mixed Phase of Minimum Phase filter.

Section IV (c) Checkpoint  
Marks: 1 out of 10
Section V: Continuous Time System – The s-plane

In Section IV, we examined the discrete time domain and explored how pole-zero plots on the z-plane can facilitate us in digital filter designs.

MATLAB has some built-in functions to generate pole-zero plots on the s-plane that facilitate analogue filters design in the continuous time domain as well. The functions of interests are as follows:

- \( \text{tf}(\text{num}, \text{den}) \)
- \( \text{pzmap}(H) \)
- \( \text{freqs}(\text{num}, \text{den}) \)

Note that, please type \text{help function\_name} if you do not understand how to use those functions.

In the following example, the analogue filter has 2 poles and 2 zeros. Their locations are given below:

<table>
<thead>
<tr>
<th>Poles</th>
<th>-0.5 ± j 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zeros</td>
<td>-0.1 ± j</td>
</tr>
</tbody>
</table>

The transfer function can also be expressed as:

\[
H(s) = \frac{s^2 + 0.2s + 1.01}{s^2 + s + 16.25}
\]

Use the above commands to generate a pole-zero plot and a frequency response plot for the above analogue filter.

Question 12: Comment on the location of the poles and the zeros in relation to the peaks and troughs of the frequency response.

Section V Checkpoint

MODULE OUTCOMES:

At the end of the laboratory, students should have a thorough understanding of the following concepts:

- Low Pass Filters; High Pass Filters; All Pass Filters.
- Linear Phase Filters; Minimum / Mixed / Maximum Phase Filters.
- MATLAB implementation of impulse and frequency response.
- Three types of system stability: Stable, Marginal stable and Unstable and the relationship with the filter’s pole positions.
- Preliminary concept of a digital oscillator.