EE-3221 LABORATORY

Difference Equations, Impulse Response Functions, and Discrete Time Filters

Goal – Investigate discrete filtering in MATLAB using difference equations.

Materials - Laptop computer, speakers or headphones (recommended)

Background – The general form of a linear system represented by an Nth order difference equation is given by the following:

$$\boldsymbol{a_0 y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) - a_1 y(n-1) - \cdots - a_N y(n-N)}$$

The system response can be found in MATLAB for a given set of a and b vectors with the following:

```
y(n) = (1/a(1)) * (b(1)*x(n) + b(2)*x(n-1) + ... + b(M+1)*x(n-M)
                          - a(2)*y(n-1) - ... - a(N+1)*y(n-N));
```

1. *Explain why the a_0 term in the first equation becomes a(1) (higher index) in the second equation.*

A 6-point moving average FIR filter can be represented by the equation:

$$y(n) = \frac{1}{6}\big(x(n) + x(n-1) + \cdots + x(n-5)\big)$$

Let's apply this filter to a noisy audio clip we created in the Quantization Lab. The file "plumclip_4bit.wav" is the 4-bit (lowres) version of the Nutcracker clip. Download the file from the course website and use the "audioread" function to import the file into a vector in MATLAB called "lowres_clip". Be sure to also read the sampling rate fs.

To remind yourself of the beauty of this noisy clip, turn down the volume to a comfortable level and play the clip:

```
sound(lowres_clip, fs)
```

Now, pass the sound clip through the moving average filter. Use a "for" loop to do this. There is an immediate issue: how do we handle the input samples for n = 0,1,2,3, and 4? Notice that we don't have enough previous samples to fully implement the filter for these values of n! Let's take the simplest of approaches – ignore these samples and begin filtering at n = 5.

```
x = lowres_clip;         % x is filter input
y = zeros(size(x));      % y is filter output – initialize all values to zero
for ii=6:length(x)
     y(ii) = 1/6*(x(ii) + x(ii-1) + x(ii-2) + x(ii-3) + x(ii-4) + x(ii-5));
end
```

Listen to the filtered version of the clip:

```
sound(y, fs)
```

2. *Describe the difference you hear between the lowres_clip and the filtered version of lowres_clip.*

MATLAB has a built-in function to do filtering – filter(). To use it, you must specify the "a" and "b" coefficients of your filter. The 6-point moving average FIR filter can be implemented as follows:

```
a = 1;
b = (1/6)*[1 1 1 1 1 1];
```

An easier way to create the vector b in MATLAB is

```
b = (1/6)*ones(1,6);
```

The ones(m,n) function creates a matrix of all ones, with m rows and n columns.

```
filtered_clip = filter(b, a, lowres_clip);
```

Listen to the filtered version of the clip:

```
sound(filtered_clip, fs)
```

Examine the difference between the two methods of filtering, paying careful attention to the scale of the y axis:

```
plot(0:length(x)-1, filtered_clip-y)
```

3. *What can you conclude from the above graph of the differences?*

4. *Implement an 11-point moving average filter. Pass lowres_clip through this filter. Include the MATLAB code you wrote. How does the filter output compare to the 6-point moving average filter? Describe any differences you hear.*

Now, let's look more carefully at the 11-point moving average filter itself. Recall that the impulse function response function is the output when the input is an impulse. Use an impulse function as the input to verify the impulse response of the 11-point moving average filter. First, we create an impulse function in MATLAB:

```
n = -20:20;                          % create "n" vector
d = [zeros(1,20) 1 zeros(1,20)];     % d represents an impulse function
```

Create a "stem" plot in MATLAB to verify that d does in fact represent a discrete-time impulse.

Then, apply d as the filter input.

```
h = filter(b, a, d);
```

5. *Provide a* ***stem*** *plot of the impulse response of the 11-point moving average filter. Completely label the graph. The vertical axis should have a label of "Impulse response: h(n)", the horizontal axis should have a label of "Sample number: n". Provide an appropriate title.*

6. *Is this an FIR or an IIR filter? Explain your answer based on the plot.*

Now, consider a simple 1st order system characterized by the difference equation,

$$y(n) = x(n) + 0.5y(n-1)$$

7. *Write MATLAB code that implements the difference equation to calculate the impulse response of the filter. Do not use the "filter()" function, but rather implement the filter using a for loop. Include your MATLAB code. Provide a plot of the impulse response. Completely label the graph. The vertical axis should have a label of h(n), the horizontal axis should have a label of n. Provide an appropriate title. Include this with your submittal.*

8. *Is this an FIR or an IIR filter? Explain your answer based on the plot stem(n,h).*

Finally, we illustrate that the output of a discrete system is not always stable. As an example, compare the **step response** between two slightly different 1st order systems.

$$y1(n) = x(n) - 0.95y1(n-1)$$

and

$$y2(n) = x(n) - 1.05y2(n-1)$$

It is up to you to generate the unit step input signal!

9. *For the values of n specified in the MATLAB code below, write the line of code that creates a unit step function in vector u. Include a stem plot, with proper axes labels, showing your unit step function.*

```
n = -20:50;
u =
```

Create the **a** and **b** vectors for these systems and use "filter()" to determine their step response.

10. *Provide stem plots of the unit step response for both systems. Use the subplot function to illustrate these responses. What can you conclude?*