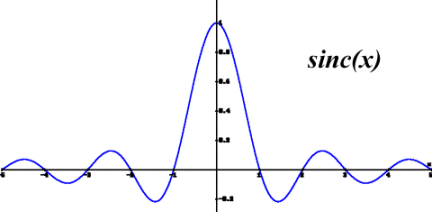
EE 419 - Project 7

FIR Filter Design Methods



|  |  |
| --- | --- |
| **Names: Chris Adams & Aiku Shintani** | **Lab Date: 2/26/19** |
| **Bench #: 9** | **Section: 2** |

**1) FIR Filter Design by Windowing Method – Hand Calculations**

**Design Specifications:**

FIR low pass filter

-6 dB cutoff frequency *fc*= 3200 Hz

Sampling rate of *fs* =16 KHz

**Design Parameters:**

Filter length M: 9

Window: Hamming

**Filter Design Steps:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Fc:** | **0.2** | cyc/samp |  |  |  |  |  |  |  |
| **n:** |  |  |  |  |  |  |  |  |  |
| **Ideal**  **h**[**n**]**Ideal** | **-0.0757** | **-0.0624** | **0.0935** | **0.3027** | **0.4** | **0.3027** | **0.0935** | **-0.0624** | **-0.0757** |
| **Window**  **w**[**n**] **(Hamming** | **0.08** | **0.2147** | **0.54** | **0.8653** | **1** | **0.8653** | **0.54** | **0.2147** | **0.08** |
| **Filter**  **h**[**n**] | **-0.00605** | **-0.0134** | **0.0505** | **0.2619** | **0.4** | **0.2619** | **0.0505** | **-0.0134** | **-0.00605** |

**Equations for the**

**IDEAL FILTER h**[**n**]**:** sin(2\*pi\*n\*Fc)/pi/n

**HAMMING WINDOW w**[**n**]**:** 0.54 + 0.46cos(2\*pi\*n/(M-1))

**DIFFERENCE EQUATION:**

**y**[**n**] **= -0.00605x**[**n**] **- 0.0134x**[**n-1**] **+ 0.0505x**[**n-2**] **+ 0.2619x**[**n-3**] **+ 0.4x**[**n-4**]  **+ 0.2619x**[**n-5**]  **+ 0.0505x**[**n-6**] **- 0.0134x**[**n-7**]  **- 0.00605x**[**n-8**]

**Filter Performance:**

1. Using your Matlab analysis programs, **plot the frequency response (linear scale vs. digital frequency) and pole/zero diagram** for your filter.

|  |  |
| --- | --- |
| **Frequency Response** | **Pole/Zero Diagram** |
|  |  |

1. **Based on the pole/zero diagram** and pole/zero locations (complex values), **explain** how can you tell that this filter is:
   1. **FIR : all poles are at the origin, #poles = #zeros so causal**
   2. **Linear Phase:**

Report the zero locations in polar form: |radius|  angle

|  |  |  |  |
| --- | --- | --- | --- |
| Radius |  | Angle (degrees) | 1 / Radius |
| 3.9452 |  | 0 | 0.2535 |
| 0.2535 |  | 0 | 3.9452 |
| 2.2854 |  | 148.0011 | 0.4376 |
| 2.2854 |  | -148.0011 | 0.4376 |
| 1 |  | 153.8422 | 1 |
| 1 |  | -153.8422 | 1 |
| 0.4376 |  | 148.0011 | 2.2854 |
| 0.4376 |  | -148.0011 | 2.2854 |

Explain how these confirm the linear phase behavior: **Each of the zeros are matched with their complex conjugate pair. Each of the zeros not on the unit circle are matched with their reciprocal radius (1/r) zero pair (at the same angle). Symmetrical h[n].**

* 1. **Low-Pass: Based off of the digital frequency response plot above, the magnitude response is unity gain at DC and the most attenuation occurs at F = 0.5. This response is low pass in nature. Based off of the pole/zero plot above, the system can be confirmed to be lowpass as well. The zeros are closest to the unit circle at high frequencies and furthest to the unit circle at low frequencies.**

1. From the Matlab filter analysis plots, report the following **actual performance measures** for your filter:
   1. Magnitude response at DC **= 0.9859 (linear units)**
   2. Stop-band attenuation (in dB) at the Nyquist Freq. = **41.83dB @ F = 0.5 cycles/sample**
   3. Magnitude response at the designed Cutoff Freq. = **0.499 (-6.038dB) @ Fc = 0.2 cycles/sample**

**2) FIR Filter Design by Windowing Matlab Program**

**function hn\_lp = FIR\_Filter\_By\_Window (M,Fc,window(M))**

M = the filter length (odd)

Fc = filter cutoff digital frequency (-6dB) (cycles/sample)

window = the Matlab window function name

hn\_lp = windowed impulse response values for the Low-pass FIR filter

**Test Case:**

FIR low pass filter

-6 dB cutoff frequency *fc*= 3200 Hz

Sampling rate of *fs* =16 KHz

Filter length M: 9

Window: Hamming

**Test Results:**

**Matlab function call used for the test case:** FIR\_Filter\_By\_Window(9, 0.2, hamming(9))

**Filter Coefficients determined by your program for the test case:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **B0** | **B1** | **B2** | **B3** | **B4** | **B5** | **B6** | **B7** | **B8** |
| -0.006055 | -0.013392 | 0.05052 | 0.2619 | 0.4 | 0.2619 | 0.05052 | -0.013392 | -0.006055 |

**Listing of your Function m-file Code:**

function hn\_lp = FIR\_Filter\_By\_Window(M, Fc, window)

% This function takes three arguments: M is the filter length, Fc is the

% desired cutoff frequency, and window is the window type that is specified

% by the user as window\_name(M).

%test this func. using: hn\_lp = FIR\_Filter\_By\_Window(9, 0.2, hamming(9))

n = -(M-1)/2:(M-1)/2;

h\_ideal = sin(2\*pi.\*n\*Fc)/pi./n;

h\_ideal(((M-1)/2)+1) = 2\*Fc;

h\_window = window.';

hn\_lp = h\_ideal.\*h\_window;

end

**3) [Matlab] FIR Filter Design by Frequency Sampling**

**function [hn,HF,F]=FIR\_Filter\_By\_Freq\_Sample(HF\_mag\_samples,figurenum)**

% hn - impulse response of filter (same length as HF\_mag\_samples)

% HF - complex frequency response of filter

% F – digital frequency values corresponding to the estimated H(F)

% HF\_mag\_samples – H[k] Magnitude response samples for desired filter

% figurenum - Figure # to plot frequency responses

**Test Case:** [hn, HF, F] = FIR\_Filter\_By\_Freq\_Sample( [1 1 0 0 1], 100);



**Test Results:**

1. **The frequency response plot resulting from the test case** (should match the figure above)**.**

*NOTE: It is OK if the Phase Angle values when |H[k]|=0 are plotted as 0 instead of the values show.*



1. **The dB Magnitude response plot for the test case**

**

1. **A listing of your m-file.**

function [hn, HF, F] = FIR\_Filter\_By\_Freq\_Sample(HF\_mag\_samples, figure\_num)

%This function takes two input arugments: HF\_mag\_samples which correspond

%to the H\_k magnitude (positive only and for a DC gain of <1 all

%coefficients are <1) response that the user want and the figure #. The

%function returns the corresponding unit sample response (h[n]), the

%frequency response (HF), and the digital freq (F).

% part a

k = 0:length(HF\_mag\_samples)-1; %# of mag samples is M

M = length(k); % M is equal to # k samples for DFT

angle\_rad = -pi.\*k\*(M-1)/M; %compute angle argument (in radians)

%correct angles arguments so they're within -pi and pi

for x=1:length(angle\_rad)

while (angle\_rad(x) < -pi)

angle\_rad(x) = angle\_rad(x)+2\*pi;

end

while (angle\_rad(x) > pi)

angle\_rad(x) = angle\_rad(x)-2\*pi;

end

end

Hk\_angle = exp(j\*angle\_rad); %Hk\_angles

Hk = HF\_mag\_samples.\*Hk\_angle; %Hk in complex form

hn = real(ifft(Hk)); %get the unit sample response

% part b

HF\_no\_pad = fft(hn); %for use in low-res FFT ==> DFT, discrete

%compute non-padded HF

M\_pad = 2^12; %for use in high-res FFT ==> "DTFT", psuedo continuous

HF = fft(hn, M\_pad); %compute padded HF

F = 0:1/(M\_pad-1):1; %sample freq. spacing

Fk = 0:1/M:(M-1)/M; %for stem plots

HF\_mag = abs(HF); %compute the magnitude of padded HF

HF\_mag\_no\_pad = abs(HF\_no\_pad); %compute the magnitude of non-padded HF

HF\_ang = angle(HF)/pi; %compute the angle of padded HF

HF\_ang\_no\_pad = angle\_rad/pi; %compute the angle of non-padded HF

%plot digital frequency response

figure(figure\_num)

subplot(2,1,1)

plot(F, HF\_mag) %plot magnitude response (linear)

xlabel('Digital Frequency F (cycles/sample)')

ylabel('Magnitude Response')

title('Digital Frequency Response of Filter')

hold on

%superimpose non-padded DFT magnitude

stem(Fk, HF\_mag\_no\_pad, '.', 'MarkerSize', 20, 'Linewidth', 2);

%plot phase response

subplot(2,1,2)

plot(F, HF\_ang)

xlabel('Digital Frequency F (cycles/sample)')

ylabel('Phase Response/pi')

hold on

%superimpose non-padded DFT phase

stem(Fk, HF\_ang\_no\_pad, '.', 'MarkerSize', 20, 'Linewidth', 2);

%part c, plot magnitude response (in dB this time)

F\_c = 0:1/(M\_pad-1):0.5; %F = 0-0.5 instead of F =0-1 like before

figure(figure\_num + 1)

plot(F\_c, 20\*log10(HF(0:0.5\*(M\_pad - 1)))) %only take up to F = 0.5 worth of HF

xlabel('Digital Frequency F (cycles/sample)')

ylabel('Magnitude Response in dB')

title('Digital Frequency Response of Filter')

end

**4a) FIR By Frequency Sampling - Matlab & Manual Design**

**Design Specifications:**

FIR Low Pass Filter

-6 dB Cutoff Frequency *fc*= 12 KHz

Sampling Rate of *fs* =48 KHz

**Design Parameters:**

Filter length M: 9

Method: Frequency Sampling

Constraint: |H[k]| = 0 or 1

**Filter Design:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sample**  **Index**  **k** | **Digital Frequency**  **Fk** (cycl/samp) | **Magnitude**  **|Hd[k]|** | **Phase Angle**  **Hd[k]** | **Complex H[k]** | **Unit Sample Response**  **h**[**n**] |
| **0** | **0** | **0** | **0** | **0** | **-0.0725** |
| **1** | **1/9** | **0** | **-8\*pi/9** | **0** | **0.1111** |
| **2** | **2/9** | **0** | **-16\*pi/9** | **0** | **0.0591** |
| **3** | **3/9** | **1** | -24\*pi/9 +18\*pi/9 = **-6\*pi/9** | **-0.5 – j0.866** | **-0.3199** |
| **4** | **4/9** | **1** | -32\*pi/9 + 36\*pi/9 = **4\*pi/9** | **0.1736 + j0.9848** | **0.4444** |
| **5** | **5/9** | **1** | -40\*pi/9 + 36\*pi/9 = **-4\*pi/9** | **0.1736 - j0.9848** | **-0.3199** |
| **6** | **6/9** | **1** | -48\*pi/9 + 54\*pi/9 = **6\*pi/9** | **-0.5 + j0.866** | **0.0591** |
| **7** | **7/9** | **0** | -56\*pi/9 + 72\*pi/9 = **16\*pi/9** | **0** | **0.1111** |
| **8** | **8/9** | **0** | -64\*pi/9 + 72\*pi/9 = **8\*pi/9** | **0** | **-0.0725** |

**The difference equation for the filter designed:**

**y[n] = -0.0725x[n] + 0.1111x[n-1] + 0.0591x[n-2] - 0.3199x[n-3] + 0.4444x[n-4] - 0.3199x[n-5] + 0.0591x[n-6] + 0.1111x[n-7] - 0.0725x[n-8]**

**Filter Performance:**

**Plot of the magnitude and phase of the filter’s frequency response (**using plotfrom FIR\_Filter\_By\_Freq\_Sample( ))



**Plots of the magnitude and phase of the filter’s frequency** **response** (from your Matlab filter analysis program).

**

**

**Report the percent errors in magnitude (mean absolute error and mean squared error) between the M=9 filter and an ideal filter** (computed by the M-File function: magnitude\_response\_error())

**

**mean\_abs\_error = 12.9% mean\_sq\_error = 4.55%**

**Quantify the performance of the filter, based on the frequency response plots.**

1) Maximum Passband Ripple **= 0.205**

2) Maximum Passband Attenuation Ap **= 3.61**dB

3) Cutoff Frequency (-6 dB) **Fc = 0.2831**cycle/sample; ***f*c** = 13.589KHz

4) Maximum Stopband Ripple **= 0.09144**

5) Minimum Stopband Attenuation As **= 20.78**dB

**4b) Improved Design By Iteration With Matlab**

**Design Goals:**

* minimize filter’s frequency response mean absolute error percentage
* reduce the magnitude response overshoots and ripple
* achieve closer to the desire -6dB cutoff frequency.

**Final Filter Design:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index **k, n** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Magnitude  Samples  **|Hd[k]|** | **0** | **0** | **0.179** | **0.912** | **1** | **1** | **0.912** | **0.179** | **0** |
| Unit Sample Response  **h[n]** | **-0.0323** | **0.0717** | **0.0315** | **-0.3032** | **0.4647** | **-0.3032** | **0.0315** | **0.0717** | **-0.0323** |

**The revised difference equation for the improved filter:**

**y[n] = -0.0323x[n] + 0.0717x[n-1] + 0.0315x[n-2] - 0.3032x[n-3] + 0.4647x[n-4] - 0.3032x[n-5] + 0.0315x[n-6] + 0.0717x[n-7] - 0.0323x[n-8]**

**Final Filter Performance:**

**Plot of the magnitude and phase of the filter’s frequency response (**using plotfrom FIR\_Filter\_By\_Freq\_Sample( ))

**

**Plots of the magnitude and phase of the filter’s frequency** **response** (from your Matlab filter analysis program).

**

**

**Report the percent errors in magnitude (mean absolute error and mean squared error) between the M=9 filter and an ideal filter** (computed by the M-File function: magnitude\_response\_error())

**

**mean\_abs\_error = 10.1% mean\_sq\_error = 3.3%**

**Quantify the performance of the filter, based on the frequency response plots.**

1) Maximum Passband Ripple **= 0.073**

2) Maximum Passband Attenuation Ap **= 1.27**dB

3) Cutoff Frequency (-6 dB) **Fc = 0.2733**cycle/sample; ***f*c** = 13.118KHz

4) Maximum Stopband Ripple **= 0.003831**

5) Minimum Stopband Attenuation As **= 48.33**dB

**5) [Matlab] FIR Filter Design Comparison**

All designs should use the following parameters:

Target Cutoff Frequency (-6 dB) Fc = 0.20 cycles/sample

Transition Width FTransition = 0.133 cycles / sample

Filter Length M = 15

Linear Phase Response

Real, Symmetric Filter Coefficients

Sampling Frequency of *fs* = 1 KHz

The cases to be evaluated are:

1. FIR Design by Windowing (3 cases)
   1. Rectangular Window
   2. Hamming Window
   3. Kaiser Window (s =p = 0.03)
2. FIR Design By Frequency Sampling (No windowing)
3. FIR Optimal Design By Parks-McClellan Algorithm

**Alternative Designs:**

1. **Listing of Matlab commands used to create each design.** (Analysis steps do not need to be shown).
2. **Resulting Filter Designs:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Filter Coefficients** | **Windowing – Rectangular** | **Windowing – Hamming** | **Windowing – Kaiser** | **Frequency Sampling** | **Parks - McClellan** |
| **B0** |  |  |  |  |  |
| **B1** |  |  |  |  |  |
| **B2** |  |  |  |  |  |
| **B3** |  |  |  |  |  |
| **B4** |  |  |  |  |  |
| **B5** |  |  |  |  |  |
| **B6** |  |  |  |  |  |
| **B7** |  |  |  |  |  |
| **B8** |  |  |  |  |  |
| **B9** |  |  |  |  |  |
| **B10** |  |  |  |  |  |
| **B11** |  |  |  |  |  |
| **B12** |  |  |  |  |  |
| **B13** |  |  |  |  |  |
| **B14** |  |  |  |  |  |

**Test Results and Performance Comparison:**

**Composite Magnitude Response Plot (linear scale)**

*(Insert Plot HERE)*

**Composite Magnitude Response Plot (log (dB) scale)**

*(Insert Plot HERE)*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Design Method** | **Maximum Passband Ripple**  **p** | **Maximum Stopband Ripple**  **s** | **Minimum Stopband Attenuation**  **As (dB)** | **|H(F)| at Fc = 0.2**  **cyc/spl** | **Passband**  **Edge Freq. Fp**  **(cyc/spl)** | **Stopband**  **Edge Freq. Fs**  **(cyc/spl)** | **Transition**  **Bandwidth**  **FT**  **(cyc/spl)** | **Mean**  **Absolute**  **Error**  **(%)** |
| **Windowing**  **(Rectangle)** |  |  |  |  |  |  |  |  |
| **Windowing**  **(Hamming)** |  |  |  |  |  |  |  |  |
| **Windowing**  **(Kaiser)** |  |  |  |  |  |  |  |  |
| **Frequency**  **Sampling** |  |  |  |  |  |  |  |  |
| **Parks- McClellan** |  |  |  |  |  |  |  |  |

**Conclusions of your comparison between methods and their performance. Identify relative strengths / weaknesses of each approach, and the tradeoffs involved in selecting an FIR filter design method.**

**Project Conclusions:**

*Summarize one or two learnings about FIR Filter Design that this project helped you understand better. Also describe any particular challenges that you had to overcome, and at least one suggestion for improvement of this lab in the future.*

**Name: Aiku Shintani**

**Conclusions:** Through the course of this project, different methods of FIR filter design were investigated. The limitations of the FIR design via windowing and frequency sampling were clear when compared to the Parks McClellan method. The Parks McClellan method takes advantage of statistical theories which optimize the filter design process. The frequency sampling method is relatively useful but requires quite a bit of tuning until the desired response is achieved. It is clear from these observations that in practical FIR design, the Parks McClellan method is the most efficient in achieving quick results.

**Name:**

**Conclusions:**