MATLAB Programming exercises

1. Sinusoids and their spectral representation

Problem parameters:

Sample Rate: fS = 80 kHz,   
Frequency: f0 = 5.0 kHz  
Number of Samples: N = Number required to plot 20 cycles of sinusoid

For each signal, a figure with 4-subplots  
subplot(4,1,1) 100 samples of Real and Imaginary Part of time Series  
subplot(4,1,2) real part of N-Point FFT of N-point sequence  
subplot(4,1,3) imaginary part of N-Point FFT of N-point sequence  
subplot(4,1,4) 20 Log Mag of N-Point FFT of N-point sequence

1. x1 = cos(2\*pi\*(0:N-1)\*f\_0/f\_S)
2. x2 = sin(2\*pi\*(0:N-1)\*f\_0/f\_S)
3. x3 = cos(2\*pi\*(0:N-1)\*f\_0/f\_S) + sin(2\*pi\*(0:N-1)\*f\_0/f\_S)
4. x4 = j\*sin(2\*pi\*(0:N-1)\*f\_0/f\_S)
5. x5 = cos(2\*pi\*(0:N-1)\*f\_0/f\_S)+j\*sin(2\*pi\*(0:N-1)\*f\_0/f\_S)
6. x5 = cos(2\*pi\*(0:N-1)\*f\_0/f\_S + theta)+j\*sin(2\*pi\*(0:N-1)\*f\_0/f\_S + theta), theta=2pi/8

Example:



1. Digital Recursive (IIR) Filters, their spectra and time responses.

Design 4 recursive filters, Butterworth, Chebyshev, Inverse Chebyshev, & Elliptic  
Matlab scripts, butter(N,W1), cheby1(N,Rp,W1), cheby2(N,Rs,W1), ellip(N,Rp,Rs,W1)

Problem parameters:

Sample Rate: fS = 80.0 kHz,   
Passband Frequency: f1 = 5.0 kHz  
Stopband Frequency: f2 = 10.0 kHz  
Passband Ripple: 0.1 dB  
Stopband Ripple: 60.0 dB  
Filter Order: N (determined by iteration)

For each filter, a figure with 6-subplots  
subplot(4,1,1) 100 samples of impulse response  
subplot(4,2,3) Log mag frequency response with spectral mask (1-K fft(--))  
subplot(4,2,4) Zoom to passband ripple (1-K fft)  
subplot(4,2,5) Phase shift frequency response (unwrap(angle(fft(--))))  
subplot(4,2,7) Group delay response (grpdelay(b,a,N,’whole’))  
subplot(2,2,4) Pole Zero plot

Compare Impulse responses and identify which filter has the shortest Impulse response and which filter has the smallest group delay spread over its bandwidth





3. Digital Recursive (IIR) Filters, Narrow bandwidth, spectra and time responses.   
Coefficient quantization and finite arithmetic effects

Design two 7-th order Inverse Chebyshev filters  
Matlab scripts cheby2(N,Rs,W1), [sos,g] = tf2sos(bb,aa)

Convert the transfer functions to second order bi-quads, determine the system responses using floating point then quantized coefficients. Use 16 bits for the wideband filter direct and its bi-quad, Use 36 bits for the wideband direct and 16 bits for its bi-quad. Compare the quantized coefficient list to the non-quantized list. Compare their frequency responses.

Determine the impulse response of the wideband direct denominator and the cascade impulse response of the bi-quad denominators:

g1= filter(1,aa1,[1 zeros(1,400)]) and

g11=filter(1,sos(1,4:6),[1 zeros(1,400)]),

g12=filter(1,sos(2,4:6),g11)/sum(g11)

g13=filter(1,sos(3,4:6),g12)/sum(g12)

g14=filter(1,sos(4,4:6),g13)/sum(g13)

Plot the impulse response and cascade of impulse responses for both the denominators of the wideband and narrowband filter in the direct and the cascade second order forms. We want to compare size of internal register states

Problem parameters: Filter 1 Filter 2

Sample Rate: fS = 80.0 kHz, 80.0 kHz  
Passband Frequency: f1 = 5.0 kHz 0.5 kHz   
Stopband Frequency: f2 = 10.0 kHz 1.0 kHz  
Passband Ripple: 0.1 dB 0.1 dB  
Stopband Ripple: 60.0 dB 60.0 dB  
Filter Order: 7 7

For each filter configuration, form a figure with 6-subplots  
subplot(4,1,1) 100 samples of impulse response  
subplot(4,2,3) Log mag frequency response with spectral mask (1-K fft(--))  
subplot(4,2,4) Zoom to passband ripple (1-K fft)  
subplot(4,2,5) Phase shift frequency response (unwrap(angle(fft(--))))  
subplot(4,2,7) Group delay response (grpdelay(b,a,N,’whole’))  
subplot(2,2,4) Pole Zero plot





4. Digital Non-Recursive (FIR) Filters, Low pass filter, Remez (or firpm) equal ripple design,  
 Modified Remez, equal ripple pass band ripple and 1/f stopband ripple, and windowed sinc  
 function design. Estimate approximate filter length using harris approximation. Design a low  
 pass FIR filter meeting following specifications. Note that 0.1 dB is approx. 1/100 and 60 dB is  
 1/1000 so the stop band ripple is 1/10 of passband ripple this drives penalty function weights

Filter type FIR

f\_s 80 kHz f\_ss = f\_s/2 = 40 kHz

f\_pass 5 kHz

f\_stop 10 kHz

Passband Ripple 0.1 dB

Stopband Ripple 60.0 dB

N ≅ (fs/df)\*atten(db)/22 = 80/(10-5)\*60/22 = 44

h1=firpm(N,[0 f\_ps f\_stp f\_s/2]/fs/2,[1 1 0 0],[1 10]);

h2=firpm(N,[0 f1 f2 fss/fss,{’myfrf’,[1 1 0 0]},[1 10]);

cc=1.45;

h3=sinc((-4-1/8:1/8:4+1/8)\*cc).\*kaiser(57,5.8)';

scl=8/cc;

h3=h3/scl;

For each filter configuration, form a figure with 4-subplots  
subplot(3,1,1) samples of impulse response  
subplot(3,1,2) Log mag frequency response with spectral mask (1-K fft(--))  
subplot(3,2,5) Zoom to passband ripple (1-K fft)  
subplot(3,2,6) Pole Zero plot



5. Digital Non-Recursive (FIR) Filters, Low pass filter, Modified Remez, equal ripple pass band ripple and 1/f stopband ripple. Estimate approximate filter length using harris approximation.   
Estimate increase in filter length when we increase stop band ripple, or reduce passband ripple, or do both, increase stop band ripple and decrease passband ripple

Design a low pass FIR filter meeting following specifications. Note that 0.1 dB is approx. 1/100 and 60 dB is 1/1000 and 80 dB i1 1/10000, so the stop band ripple is 1/10 and 1/100 of passband ripple this drives penalty function weights.

Filter types 1 2 3 4

f\_S 80 kHz 80 kHz 80 kHz 80 kHz

f\_pass 5 kHz 5 kHz 5 kHz 5 kHz

f\_stop 10 kHz 10 kHz 10 kHz 10 kHz

Passband Ripple 0.1 dB 0.1 dB 0.01 dB 0.01 dB

Stopband Ripple 60 dB 80 dB 60 dB 80 dB

N1 ≅ (fs/df)\*atten(db)/22 = 80/(10-5)\*60/22 = 44

h1=firpm(N1,[0 f\_ps f\_stp f\_s/2]/fs/2,[1 1 0 0],[1 10]);

N2 ≅ (fs/df)\*atten(db)/22 = 80/(10-5)\*80/22 = 56

H2=firpm(N2,[0 f\_ps f\_stp f\_s/2]/fs/2,[1 1 0 0],[1 100]);

For each filter configuration, form a figure with 4-subplots  
subplot(3,1,1) samples of impulse response  
subplot(3,1,2) Log mag frequency response with spectral mask (1-K fft(--))  
subplot(3,2,5) Zoom to passband ripple (1-K fft)  
subplot(3,2,6) Pole Zero plot



6. Digital Non-Recursive (FIR) Filters, Low pass filter, Modified Remez, equal ripple pass band ripple and 1/f stopband ripple. Estimate approximate filter length using harris approximation.   
Estimate increase in filter length when we increase both stop band ripple and decrease passband ripple

Design a low pass FIR filter meeting following specifications. Note that 0.1 dB is approx. 1/100 and 60 dB is 1/1000 and 80 dB i1 1/10000, so the stop band ripple is 1/10 and 1/100 of passband ripple this drives penalty function weights.

Filter types 1 2

f\_S 80 kHz 80 kHz

f\_pass 5 kHz 5 kHz

f\_stop 10 kHz 10 kHz

Passband Ripple 0.1 dB 0.01 dB

Stopband Ripple 60 dB 80 dB

For the two filters, form figures with filters using floating point coefficients (default), quantized to (60dB)/(5db/bit) or (80dB)/(5dB/bit) bits without scaling coefficients, and with scaling, quantizing, and scaled coefficients

For each filter configuration, form a figure with 4-subplots  
subplot(3,1,1) samples of impulse response  
subplot(3,1,2) Log mag frequency response with spectral mask (1-K fft(--))  
subplot(3,2,5) Zoom to passband ripple (1-K fft)  
subplot(3,2,6) Pole Zero plot





7. Digital Non-Recursive (FIR) Filters, Low pass filter, Modified Remez, equal ripple pass band ripple and 1/f stopband ripple. Estimate approximate filter length using harris approximation.   
First option, Fix Transition bandwidth and increase Passband bandwidth; Filter length and ripple specifications remain fixed.

Second option, fix passband bandwidth and decrease transition bandwidth by factors of 2; filter length increases by factors of 2 and ripple specifications remain fixed.

Design a low pass FIR filter meeting following specifications. Note that 0.1 dB is approx. 1/100 and 60 dB is 1/1000 and 80 dB i1 1/10000, so the stop band ripple is 1/10 and 1/100 of passband ripple this drives penalty function weights.

Option 1, Change Passband Bandwidth

Filter types 1 2 3 4

f\_S 80 kHz 80 kHz 80 kHz 80 kHz

f\_pass 5 kHz 10 kHz 15 kHz 20 kHz

f\_stop 10 kHz 15 kHz 20 kHz 25 kHz

Passband Ripple 0.1 dB 0.1 dB 0.1 dB 0.1 dB

Stopband Ripple 60 dB 60 dB 60 dB 60 dB

Option 2, Change Transition Bandwidth

Filter types 1 2 3 4

f\_S 80 kHz 80 kHz 80 kHz 80 kHz

f\_pass 5 kHz 5 kHz 5 kHz 5 kHz

f\_stop 10 kHz 7.5 kHz 6.25 kHz 5.625 kHz

Passband Ripple 0.1 dB 0.1 dB 0.1 dB 0.1 dB

Stopband Ripple 60 dB 60 dB 60 dB 60 dB

For each filter option, form two figures with 4-subplots  
One showing samples of impulse response  
One showing Log mag frequency response with spectral mask (1-K fft(--))





8. Digital (IIR) Recursive and Non-Recursive (FIR) Filters and their effect on signals they process.

We first design 4 filters and then pass a communication signal through them to see the effect of their in band distortion characteristics. These include deviation from a constant in band magnitude response and deviation from an in band group delay response.

Filter Type Sample Rate Pass band Stop band in band Ripple Stop band atten

1. 5-th pole elliptic 40 kHz 0-7.5 kHz 12 kHz 0.1 dB 60 dB   
 2. 7-th pole inv cheby 40 kHz 0-7.5 kHz 12 kHz 0.1 dB 60 dB  
 3. 25 tap Remez FIR 40 kHz 0-7.5 kHz 12 kHz 0.1 dB 60 dB  
 4. 31 tap Remez FIR 40 kHz 0-7.5 kHz 12 kHz 0.01 dB 60 dB  
  
 For each filter configuration, form a figure with 6-subplots  
 subplot(4,1,1) 100 samples of impulse response  
 subplot(4,2,3) Log mag frequency response with spectral mask (1-K fft(--))  
 subplot(4,2,4) Zoom to passband ripple (1-K fft)  
 subplot(4,2,5) Phase shift frequency response (unwrap(angle(fft(--))))  
 subplot(4,2,7) Group delay response (grpdelay(b,a,N,’whole’))  
 subplot(2,2,4) Pole Zero plot

Now implement a SQRT Nyquist filter using sqrt\_nyq\_y2(4,0.5,6,0)   
gg = sqrt\_nyq\_y2(sps, alpha, delay, plot-flag)   
 sps: 4 samples per symbol, alpha: 50 % excess BW  
 delay: 6 symbols from sample 1 to filter center, plot\_flag: 0 for no figures

Form 1000 random QPSK symbols, shape the symbols with a 1-to-4 up-sampling filter. Pass the shaped signal sequence through the 4 filters, and then pass the filtered signal sequence through the scaled shaping filter (now a matched filter). On three subplots, plot the real part of 100 symbols from the two IIR filters and from the first FIR filter. enter the hold command and then plot every 4-th sample with a ‘ro’ marker. You have to determine which starting sample initiates the plotted samples as seen in this example: plot (0:4:400,x4(1:4:401),’ro’). We may have to pass the two IIR signals through a delay line to align the output samples with the correct sample position. Use this delay line option

vv=remez(158,[0 7.5 32.5 640]/640,[1 1 0 0]);

vv2=32\*reshape([vv 0],32,5);

delayed version of matched filter output

x4a=filter(gg,1,x3a)/(gg\*gg');

x4a=filter(vv2(31,:),1,x4a);

Finally plot the constellation points output from the matched filters for the two IIRs and the first FIR. Last plot and compare the constellations formed by the signals passed through the two FIR filters and the matched filter and of the signal processed by the matched filter without passing through any of the 4 filters.





