Digital Signal Processing

ECEN 5763
Fall 2001

IIR Filters

Design of Digital IIR Filters With Matlab

Purpose

The purpose of this assignment is for you to learn more about designing digital IIR filters using classical analog design techniques and the bilinear transform.

Introduction to IIR Design with Matlab

Matlab contains various routines for designing and analyzing digital IIR and FIR filters (and other z-domain functions). Most of these are part of the Signal Processing Toolbox, which must be available for you to complete this assignment. A selection of these functions is listed below. Not all of the listed functions will be useful in this assignment. For additional information, type *help signal* at the Matlab prompt to get a list of all the functions in the Signal Processing Toolbox. You should also refer to the on-line help for each function you consider using as this handout is not intended to be a tutorial in their use.

Note that in the list below are functions for performing each of the basic steps in IIR digital filter design (using the analog design technique). Rather than using these individual functions, you will use composite design functions which combine several design steps into a single command to simplify the process (the function, butter, for example) - see the example on the following page.

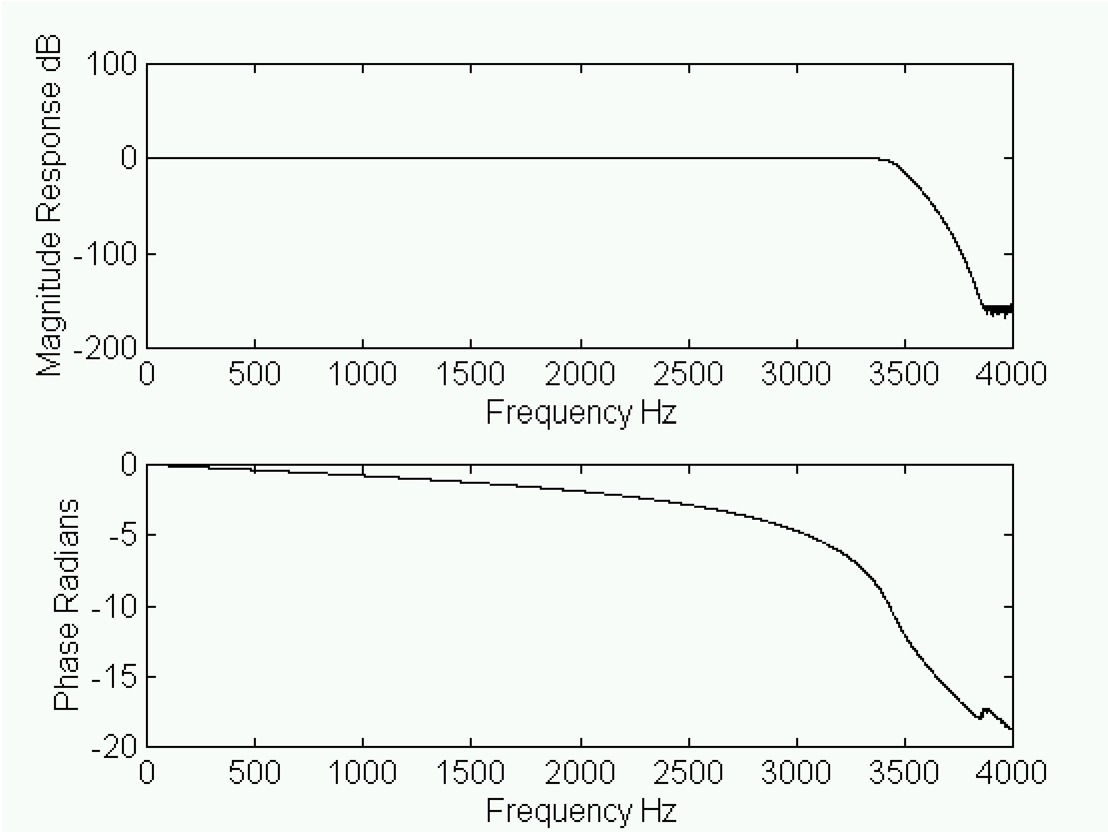
freqz	z-domain frequency response
buttord, cheb1ord, ellipord	IIR analog/digital filter order calculation
butter, cheby1, ellip	IIR analog/digital filter calculation
fir1	FIR digital design (window) - standard response
fir2	FIR digital design (window) - arbitrary response
firls	FIR digital design - arbitrary response w/t_band
remez	Parks-McClellan optimal FIR digital design
remezord	Parks-McClellan filter order calculation
abs, angle, unwrap	magnitude, phase, unwrapped phase
fft, ifft	forward and inverse discrete Fourier transform
bilinear	analog to digital bilinear transformation
impinvar	analog to digital impulse invariance
invfreqz	digital filter fit to frequency response
zplane	pole/zero plotting routine

As an example of IIR digital filter design using the bilinear transformation, consider the design of a digital Butterworth filter with the following specifications:

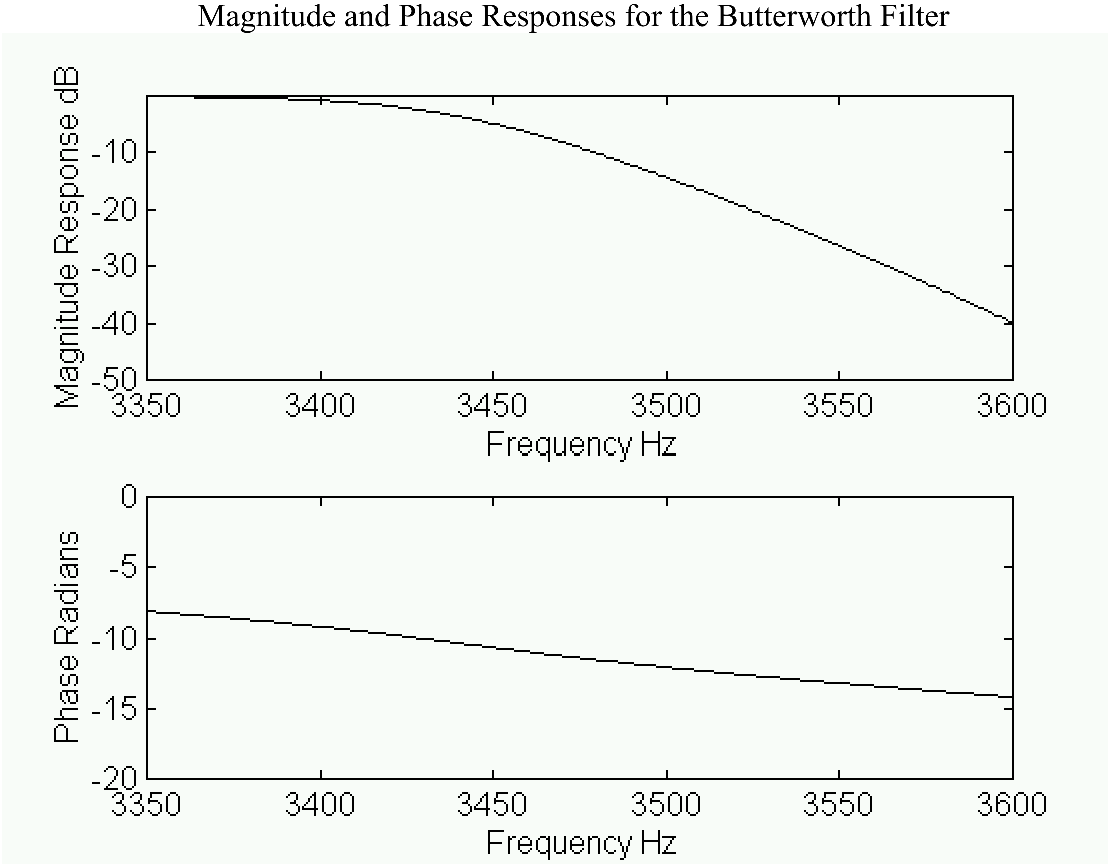
-1dB or less @ 3400 Hz and below	-40 dB or more @ 3600 Hz and beyond
sampling rate = 8000 Hz	

Note that Matlab uses a somewhat non-standard form for specifying frequencies for digital filters. The digital frequency axis which we usually consider as ranging over the interval $[0, 2\pi]$ is specified in Matlab digital filter design functions as $[0, 2]$, with 1.0 corresponding to half the sampling rate. The Matlab commands to perform this design and generate a sample plot of the magnitude and phase response are shown below. A magnitude and phase plot follow.

```
% Design a Butterworth Digital LPF using Matlab and Bilinear Transformation
%   No more than 1dB attenuation below 3400 Hz and at least 40 dB
%   attenuation at and beyond 3600 Hz. The sampling frequency is 8000 Hz.
% First compute the required order and scaling factor for the filter. Note
%   the manner in which the passband and stopband frequencies are specified.

[n,w]=buttord(3400/4000,3600/4000,1,40)

% Now design the actual filter using BUTTER. BUTTER hides most of
%   the design details from the user, returning the coefficients of H(z) directly.
%   Use the short exponential format to get reasonable printed output.

format short e
[b,a]=butter(n,w)

% Create a 100 point array of complex frequency response values and the
%   corresponding frequencies (in Hertz) for plotting and examination. Note
%   that the sampling frequency must be specified so that the actual analog
%   frequency can be determined from the digital response.

[h,f]=freqz(b,a,100,8000);

% List out the magnitude response array so that the response can be checked at
%   the passband and stopband edges (be sure those frequencies are in f).

[20*log10(abs(h)),f]

% Also create a magnitude and phase plot of H(z).

figure(1)
freqz(b,a,100,8000)

% Finally plot the poles and zeros.

figure(2)
zplane(b,a)

% Now plot the response over only the transition band (and a little more) to see
%    the response more clearly at the two critical frequencies.

figure(3)
f=3350:2:3650;
freqz(b,a,f,8000)
```



Magnitude and Phase Responses for the Butterworth Filter



Magnitude and Phase Response in the Transition Band

The IIR Filter Design Problem(s)

Use Matlab to design the digital IIR filters specified below. Perform the indicated analysis.

Problem IIR-1: Use Matlab to design three lowpass filters of types Butterworth, Chebyshev Type I, and elliptic lowpass filters to meet or exceed these specifications:

-1 dB or less at 1200 Hz and below (maximum passband ripple)	-50 dB or more at 1500 Hz and above (minimum stopband attenuation)
sampling frequency: 10,000 Hz	

For each filter, verify that the specifications were met at all critical frequencies by performing the following analysis. Plot the overall magnitude frequency response using a dB amplitude scale (*freqz* will create magnitude and phase plots together on the same figure). Plot the transition band(s) in dB with sufficient frequency resolution and numeric precision to verify the specifications were met (the Matlab *zoom* command may be helpful). Write directly on the transition band plot(s) the actual dB response value at each critical frequency (passband edge and stopband edge). Create a pole/zero plot (*zplane*). Finally, create an overall phase plot for each filter you designed. You are encouraged to use the *subplot* command, or cut and paste through the Windows clipboard to a word processor document, to minimize the number of pages you must plot.

Compare the three filters in terms of their performance by producing a table with three columns (filter type, order, minimum stopband attenuation achieved) and three rows (one for each filter type).

Compare the three phase plots. Which filter appears to produce the most linear phase within the passband? Which seems to produce the least linear phase within the passband?

Compare the three pole-zero plots. Observe and comment on the placement of poles and zeros and how they relate to production of a lowpass filter characteristic. Which filter appears to have its poles closest to the unit circle? Recall that if the poles/zeros are very close to the unit circle, this could produce problems in an implementation using reduced precision arithmetic. Note: your answer to this question might change if the design specifications were changed (i.e., your answer may not be true in general).

Problem IIR-2: Design a butterworth bandpass filter to meet these specifications:

-50 dB or more from 0 to 1200 Hz	-1 dB or less from 2,000 Hz to 4,000 Hz
sampling frequency: 16,000 Hz	-50 dB or more above 6,000 Hz

Consult the help file to see exactly how to specify a bandpass filter to Matlab. Note that you must choose the order properly - there may be more than one choice!

Carefully verify that the specifications were met at all critical frequencies as in the previous problem. Plot the overall magnitude frequency response using a dB amplitude scale. Plot the transition band(s) in dB with sufficient frequency resolution and numeric precision to verify the specs were met. Write directly on the transition band plot(s) the actual dB response value at each critical frequency (passband edge and stopband edge). List the filter order N. Create a pole/zero plot. Finally, create an overall phase plot.

Comment very briefly on the placement of the poles and zeros as their placement relates to the production of a bandpass response.

The Report

Turn in your filter designs and the corresponding analyses, complete with plots and all requested information from the questions above. Everything should be clearly labeled and organized in the order in which it was requested. Neatness is important! *Type* your answers or *print* clearly as requested directly on the plots. You are strongly encouraged to use Microsoft Word to produce your document as this allows direct importing of Matlab plots and greatly reduces the number of pages of plots that you must print and hand in.

This assignment is due on Friday, October 19, at the beginning of class.