

ECE 535 DIGITAL SIGNAL PROCESSING
Matlab Project 1: Notch Filter
Spring 2003

Issued: Sunday, February 2, 2003

Due: Tuesday, February 25, 2003

Your solutions to this assignment should include plots and Matlab code for each exercise along with any necessary explanations or answers to questions. The latter may be handwritten or typed, whichever you prefer.

Note: This project is based on a similar one in *Computer-Based Exercises for Signal Processing Using Matlab 5* by McClellan *et al.*

Notch filters are used to remove one particular frequency from a signal. For example, assume that a bandlimited continuous-time signal is known to contain a sinusoidal component, which we want to remove by processing with the system shown below:

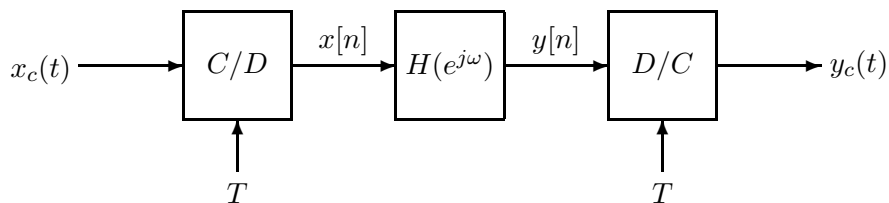


Figure 1: Discrete-time processing system

A simple discrete-time notch filter is defined by the frequency response:

$$H(e^{j\omega}) = \frac{(1 - e^{-j(\omega - \omega_0)})(1 - e^{-j(\omega + \omega_0)})}{(1 - re^{-j(\omega - \omega_0)})(1 - re^{-j(\omega + \omega_0)})} \quad (1)$$

where r and ω_0 are constants.

1 Preliminaries

- (a) Write a difference equation for the notch filter defined in Equation 1.
- (b) Pick trial values of r and ω_0 and use the Matlab command `zplane` to sketch the pole-zero plot. What are the restrictions on r and ω_0 if you want the system to be stable and causal?
- (c) Pick several trial values of r , ω_0 and use Matlab's `freqz` function to plot the magnitude and phase of $H(e^{j\omega})$. Discuss how the frequency response varies as a function of r and ω_0 . (You only need to include one or two plots, but you may want to experiment with more.)

2 Removing a sinusoidal component

Assume that an analog signal containing an undesired 180 Hz component is sampled at a rate of 8192 Hz.

- (a) What is the highest frequency that the analog signal can contain if we want to avoid distortion due to aliasing?
- (b) What value of ω_0 should you use in order for the notch filter to eliminate the 180 Hz component?

- (c) In Matlab generate 400 samples of a 180 Hz sine wave, sampled at a rate of $f_s = 1/T=8192$ Hz. Determine a vector of times to plot against (assuming the first time sample occurs at $t = 0$). Make a plot of the 180 Hz sine wave versus time. Note that, although the signal you have created is discrete, it will probably be easier to interpret your graph if you use the `plot` command (which connects the discrete points with lines) rather than the `stem` command.
- (d) The Matlab function `filter` can be used to implement the discrete-time notch filter using the difference equation you determined in part (a). Using the value of ω_0 you determined in part b and a value of r of your own choosing, define the `a` and `b` vectors for the notch filter. Process the 180-Hz signal using the `filter` command (using your `a` and `b` vectors). Plot the output of the filter on the same axes as the input signal (you may want to use the `hold` command to do this). Does your notch filter eliminate the 180 Hz sine wave?
- (e) You should observe a transient before the filter rejects the 180 Hz sinusoid completely. What governs the length of this transient response? You may want to experiment with different values of r and ω_0 to determine this.
- (f) Justify your answer in part (e) by determining the analytical form of the response to a sinusoidal input with frequency ω_0 .

3 Applying the notch filter to a music signal

The file `noisy_sig.mat` contains a music signal (an approximately 9 second segment of the Hallelujah Chorus) that has been corrupted with a 180 Hz tone. Load this signal into Matlab by typing `load noisy_sig`. (The file must be located in a directory contained in Matlab's path; the current working directory is probably the best place to put it.) The file contains the signal in the `sig` vector and the sampling frequency in the variable `fs`. The sampling frequency is 8192 Hz.

- (a) Use Matlab's `soundsc` function to play the segment. Note the presence of the 180 Hz tone.
- (b) Apply the notch filter designed with the value of ω_0 chosen in the previous section to the noisy signal and $r = 0.95$. Play the results using the `soundsc` function. Does the filter eliminate the 180 Hz tone?
- (c) Try using several different values for r , e.g., $r = 0.6$, $r = 0.9999$. Listen to those results. (You may want to experiment with more values.) Discuss what you hear and explain the results in terms of the time and frequency analysis done in the earlier sections of this project.