

Example 8-4: Trumpet Notch Filter.

Purpose:

A signal generated by an actual trumpet playing note B (which has frequency harmonics of 494 Hz) had a interfering 1200-Hertz sinusoid added to it. The trumpet-plus-sinusoid signal was sampled at the standard CD sampling rate of 44100 samples/s. Design and implement a discrete-time notch filter to eliminate the interfering sinusoid. Use $a = 0.99$.

Inputs:

Signal of an actual trumpet playing note B from the file `trumpet.mat`.

$aa=a$ =radius of poles.

f =frequency in Hz of tone to be eliminated.

fs =Sampling rate in samples per s used.

Outputs:

Plot and sound of trumpet+tone.

Plot and sound of filtered trumpet+tone.

Spectrum of trumpet-plus-tone.

Frequency response H of notch filter.

Comments:

- The first sound is the trumpet+tone. Hit any key to hear the filtered signal.
- The trumpet signal is an actual trumpet playing note B, which has a fundamental frequency of 494 Hz and period $\frac{1}{494} \approx 2$ ms. This is apparent in its waveform plot.
- The Chapter 8 examples display different time intervals than Chapter 6 examples. This distinguishes them from each other.
- Frequency response of notch in red. Spectrum of trumpet+tone in blue.
- Duration of transient response is 315.

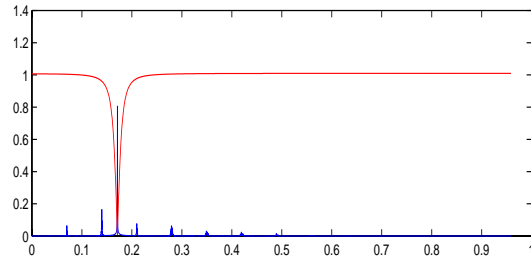


Figure 1: Frequency response of notch filter (red) and spectrum of trumpet-plus-tone signal (blue), plotted vs. frequency Ω .

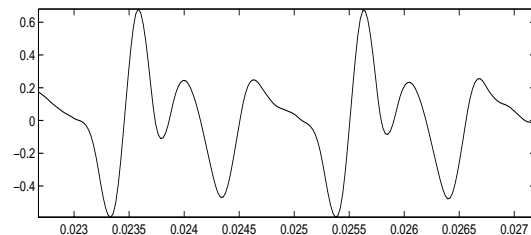
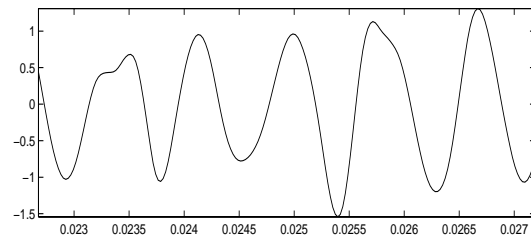


Figure 2: Trumpet-plus-tone (top), filtered trumpet-plus-tone (bottom).

Program:

```
load 'trumpet.mat'
aa=0.99;fs=44100;f=1200;
W0=2*pi*f/fs;N=length(X);
X=X+cos(W0*[1:N]);%add tone.
B=[1 -2*cos(W0) 1];
A=[1 -2*aa*cos(W0) aa*aa];
W=2*pi/N*[0:4999];EW=exp(j*W);
H=polyval(B,EW)./polyval(A,EW);
FX=2*abs(fft(X))/N;subplot(211),
plot(W,FX(1:5000),W,abs(H),'r')
Y=filter(B,A,X);
I=[1000:1199];T=I/fs;figure,
subplot(211),plot(T,X(I))
subplot(212),plot(T,Y(I))
soundsc(X,fs),pause,soundsc(Y,fs)
```