Math 240 (E1): Scientific Computation
Final Solution Code with Steps Implemented

Zachary Barnes

June 13, 2018

Steps Taken to Filter Signal:

1. Load the data by `audioread()`.
2. Assign variable `S` to the sound file.
3. Look at pressure vs. time and notice the interference in amplitude when unwanted tones are played.
4. Perform Fast Fourier Transform to move to frequency domain.
5. Plot power vs. frequency and notice the large spikes around certain frequencies. These are the tones which need to be removed.
6. The filtering process iterates through and removes any of those large spikes in frequency. I arrived at `2500` iterations from some experimentation, and `2500` seemed to be the optimal choice for minimizing computation time while maximizing tone removal.
7. Plot the new power vs. frequency of the filtered data in the frequency domain.
8. Perform inverse Fourier Transform to arrive back in time domain.
9. Plot the new pressure vs time signal with tones removed.
10. Write the filtered sound wave by using `audiowrite()`.

```
audioinfo('prob.wav')

S=audioread('prob.wav');

figure(1)
plot(S)
title('Unfiltered Wave')
xlabel('Sample Number');
ylabel('Amplitude');

FSF=fft(S);

N=length(FSF);
power = abs(FSF(1:N)).^2;
freq = (1:N)/N;

figure(2)
plot(FSF,'ro')
title('Fourier Coefficients in the Complex Plane');
xlabel('Real Axis');
ylabel('Imaginary Axis');

figure(3)
plot(freq,power)
xlabel('Cycles/Sample Interval')
ylabel('Power');
title('Periodogram')


index=find(power==max(power));
period=1./freq;
%Herz=44100/period(index)
mainPeriodStr=num2str(period(index));
figure(4)
plot(period(index),power(index),'r.', 'MarkerSize',25);
text(period(index)+0.01*period(index),power(index),['Period = ...
    ',mainPeriodStr]);

tic
for i = 1:2500
   power = abs(FSF(1:N)).^2;
   index=find(power==max(power));
   FSF(index(1)) = 0;
   FSF(index(2))=0;
end
toc

 power = abs(FSF(1:N)).^2;
 plot(freq,power)

INFT = ifft(FSF);
figure(5)
plot(INFT)
title('Filtered Wave')
xlabel('Sample Number');
ylabel('Amplitude');
```

```
audiowrite('FilteredSound.wav',INFT,44100)
```

Note: The time to filter the data on the last trail run of the `M−File` was `61.095220 seconds`