

Filtering noise with FFT

Jan 05, 2021

Contents

- [Example of periodic noise \(first ~1000 frames\)](#)
- [Plot FFT?](#)
- [TA-DA!!](#)

Loading in the relevant data trace.

```
rootdir = 'C:\Users\Windows 10\Dropbox\Sehgal Lab\In Vivo Imaging\Data Analyzed In vivo\MB077B\200814';
movementFile = 'fc2_save_2020-08-14-165328-_2Xspeed_fullMovementAndBrainSignal.mat';

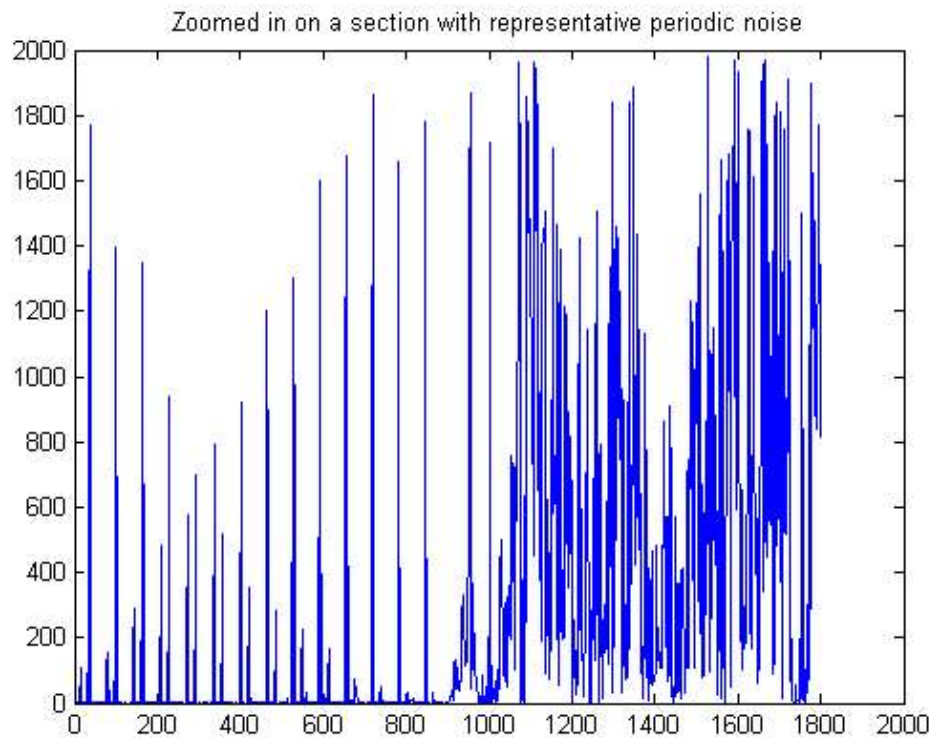
close all;
cd(rootdir);
moveDat = load(movementFile);
moveBrainDat = moveDat.fullMovementAndBrainSignal;

% Computations to get rid of NaNs.
movementDat_uninterpolated = moveBrainDat(:,1);
isNumIndices = find(~isnan(movementDat_uninterpolated));
movementDat = interp1(isNumIndices,movementDat_uninterpolated(isNumIndices),1:numel(movementDat_uninterpolated));
if(isnan(movementDat(1))),
    movementDat = movementDat(2:end);
end;
if(isnan(movementDat(end))),
    lastNumIndex = find(~isnan(movementDat),1,'last');

    numNanEndPadding = numel(movementDat)-lastNumIndex;
    movementDat = movementDat(1:lastNumIndex);
else,
    numNanEndPadding = 0;
end;
```

Example of periodic noise (first ~1000 frames)

```
sampleRange2Plot = (20.5*60*30):(21.5*60*30);
plot(movementDat(sampleRange2Plot));
title('Zoomed in on a section with representative periodic noise');
movementDat_subset = movementDat(sampleRange2Plot);
```



Plot FFT?

Taken almost directly from Matlab's sample code on their FFT documentation page.

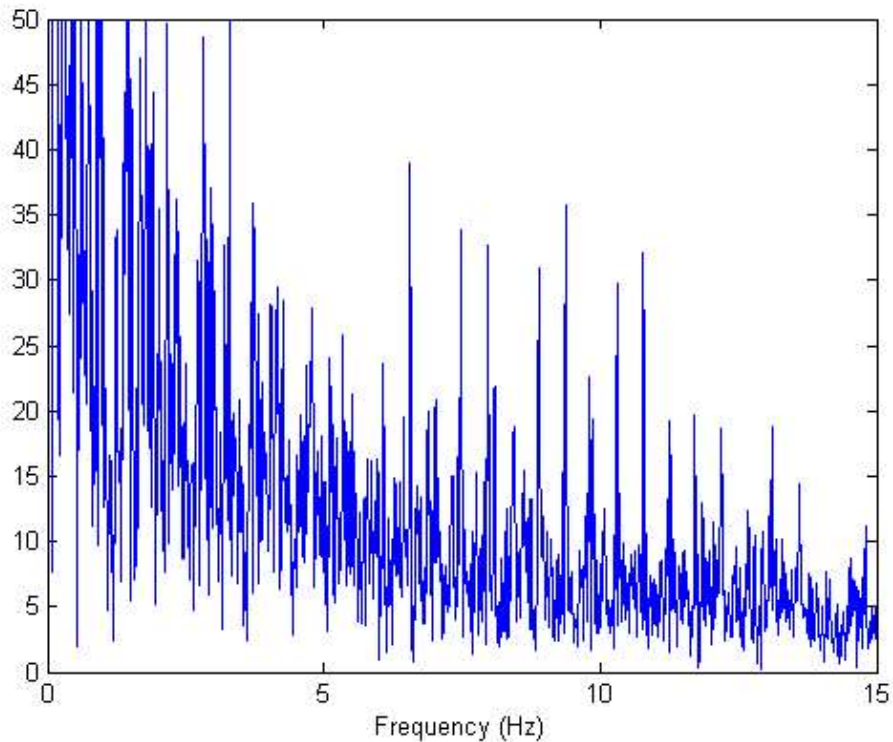
```

Fs = 30;           % Sampling frequency (frames per second)
T = 1/Fs;          % Sampling period
L = numel(movementDat_subset); % Length of signal
t = (0:L-1)*T;     % Time vector
Y = fft(movementDat_subset);
negative_fftIndices = find(Y<0);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
figure;
f = Fs*(0:(L/2))/L;

% figure(1);
% subplot(1,2,2);
plot(f,P1);
ylim([0 50]);
xlabel(['Frequency (Hz)']);

```

Warning: Integer operands are required for colon operator when used as index



Here is my attempt to filter out the signal based on frequency:

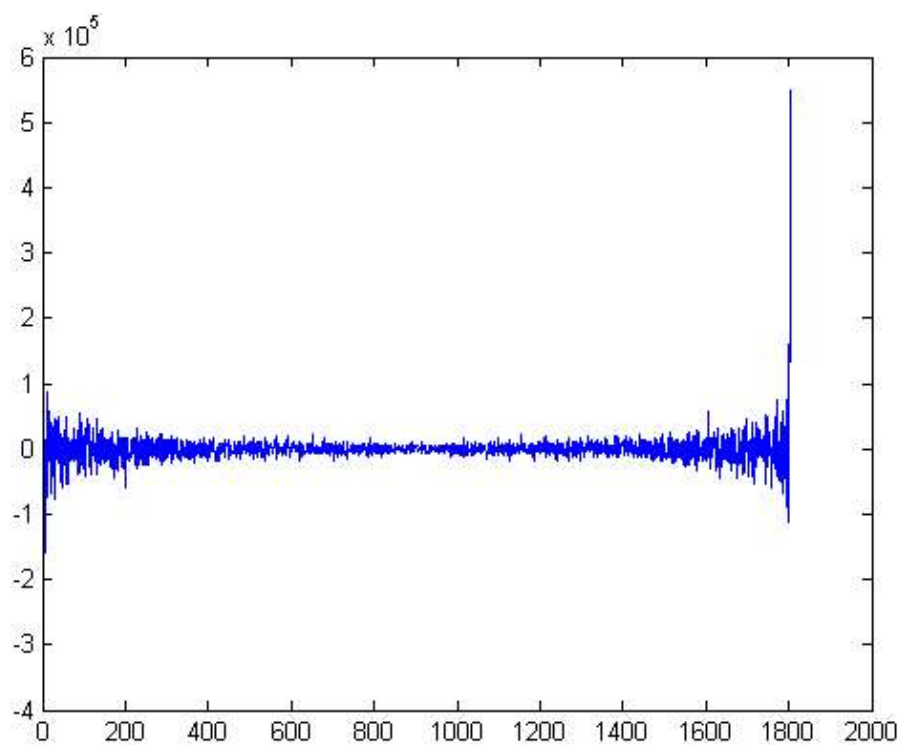
```
amplitudeThresh = 25;
freqThresh = 5;

aboveThreshIndices = find(P1>amplitudeThresh);
freq_aboveThreshIndices = find(f(aboveThreshIndices)>freqThresh);
P1(aboveThreshIndices(freq_aboveThreshIndices)) = 0;

%Now need to reverse the steps we took to get here somehow?
P1(2:end-1) = 0.5*P1(2:end-1);
P2 = [flipud(P1(:)); P2((L+1)/2); P1(:)];
P2(negative_fftIndices) = P2(negative_fftIndices)*-1;
inverseMovementSample = ifftshift(P2*L);
```

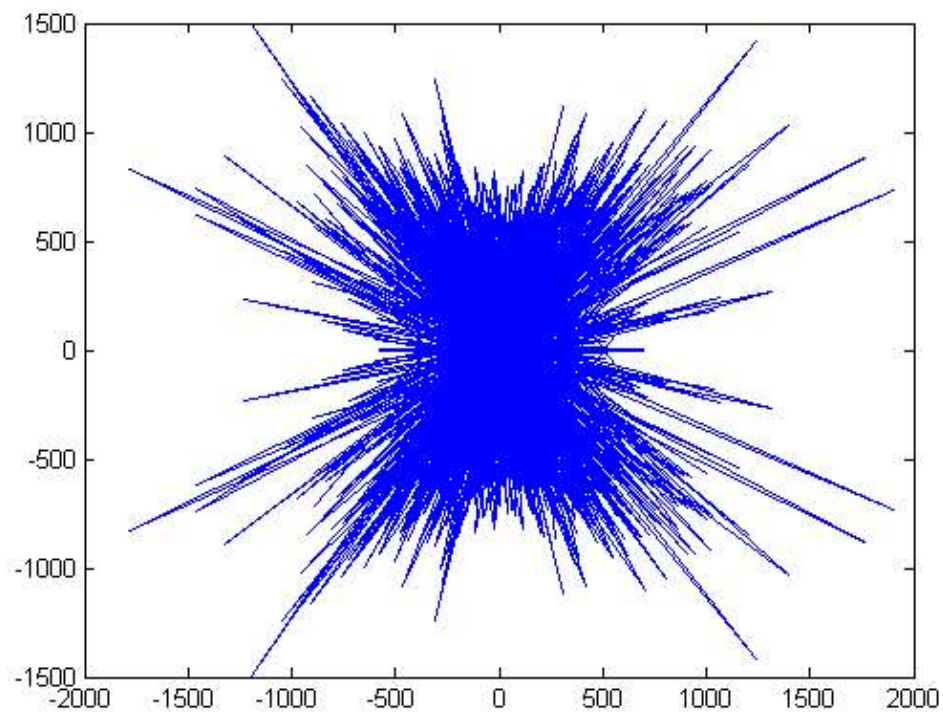
TA-DA!!

```
plot(inverseMovementSample);
```



Also tried using ifft instead of ifftshift:

```
plot(ifft(P2*L));
```



.....

Published with MATLAB® R2014a