Table of Contents

1

read wave file 1

run fft 1

cut irrelevant frequencies 1

run inverse fft 1

plot magnitude of fft result 1

plot magnitude dB 2

plot relevant frequencies 3

plot original signal and filtered signal 4

display text for relevant frequencies 5

```
clear all;
close all;
```

read wave file

```
[XN,FS,NBITS] = wavread('sineWavesInNoise.wav');
L_XN = length(XN);
L_XN_PADDED = 2^nextpow2(length(XN));   % padded sample size for fft.
```

run fft

```
XN = XN .* hamming(L_XN); % hamming window
XF = fft(XN, L_XN_PADDED);  % L_XN_PADDED size fft.
% f = L_XN_PADDED * linspace(0,1,L_XN_PADDED);
f = (0:L_XN_PADDED-1)*FS/L_XN_PADDED;
```

cut irrelevant frequencies

Cut off the frequencies which has lower than dB_threshold

```
dB_threshold = -6; % -6 dB
XF_syn = XF;
XF_dB = 20 * log10(abs(XF_syn)/max(abs(XF_syn)));
XF_syn(XF_dB<dB_threshold) = 0;     % set zero magnitude for cut off frequencies.
```

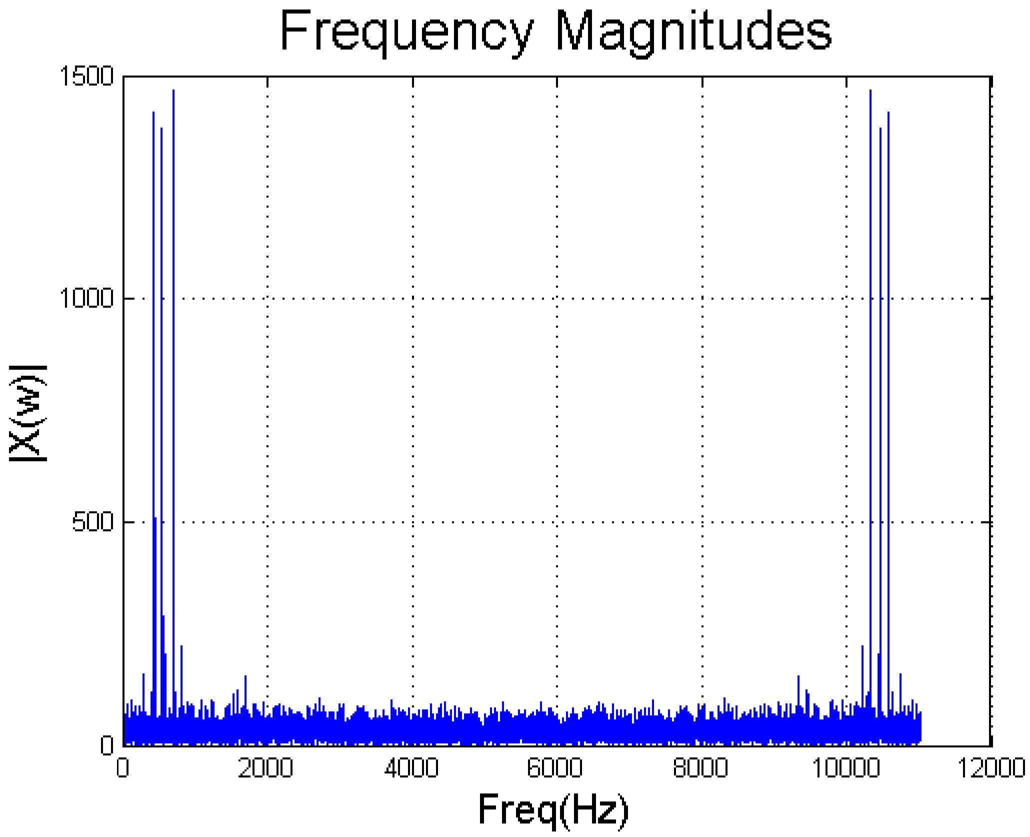
run inverse fft

```
x_synthesis = ifft(XF_syn);% inverse fast fourier transform
```

plot magnitude of fft result

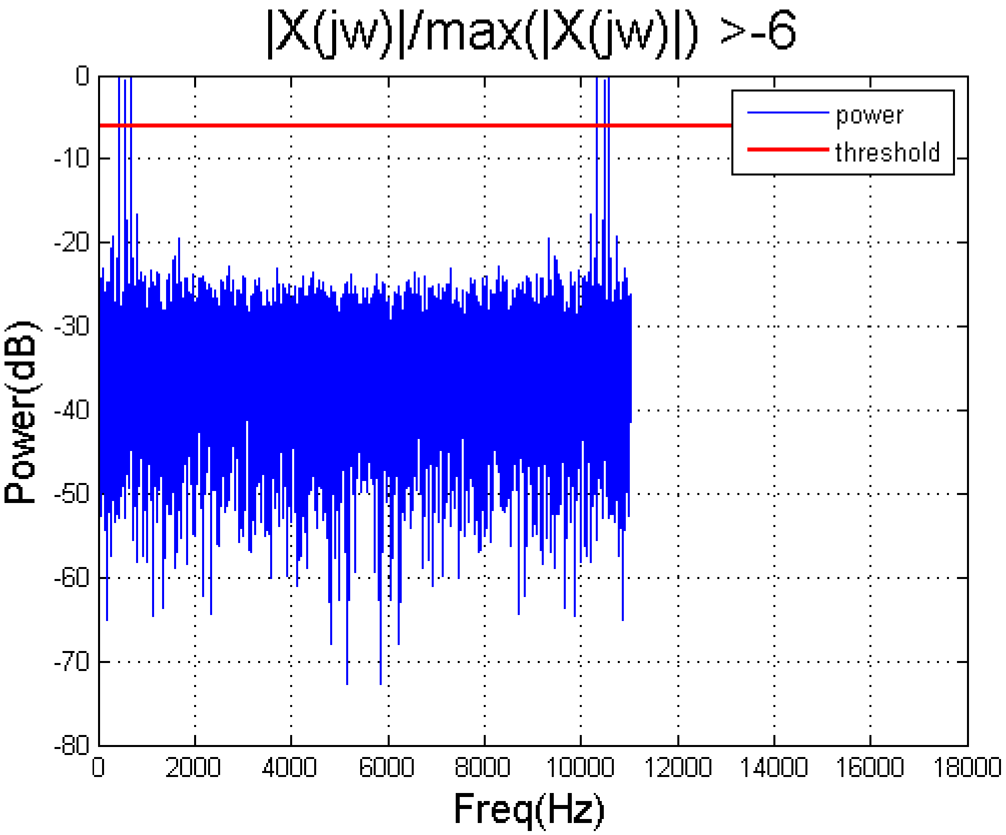
```
figure()
```

```
plot(f, abs(XF)); grid on
xlabel('Freq(Hz)','fontsize',15);
ylabel('|X(w)|','fontsize',15);
title('Frequency Magnitudes','fontsize',20);
```



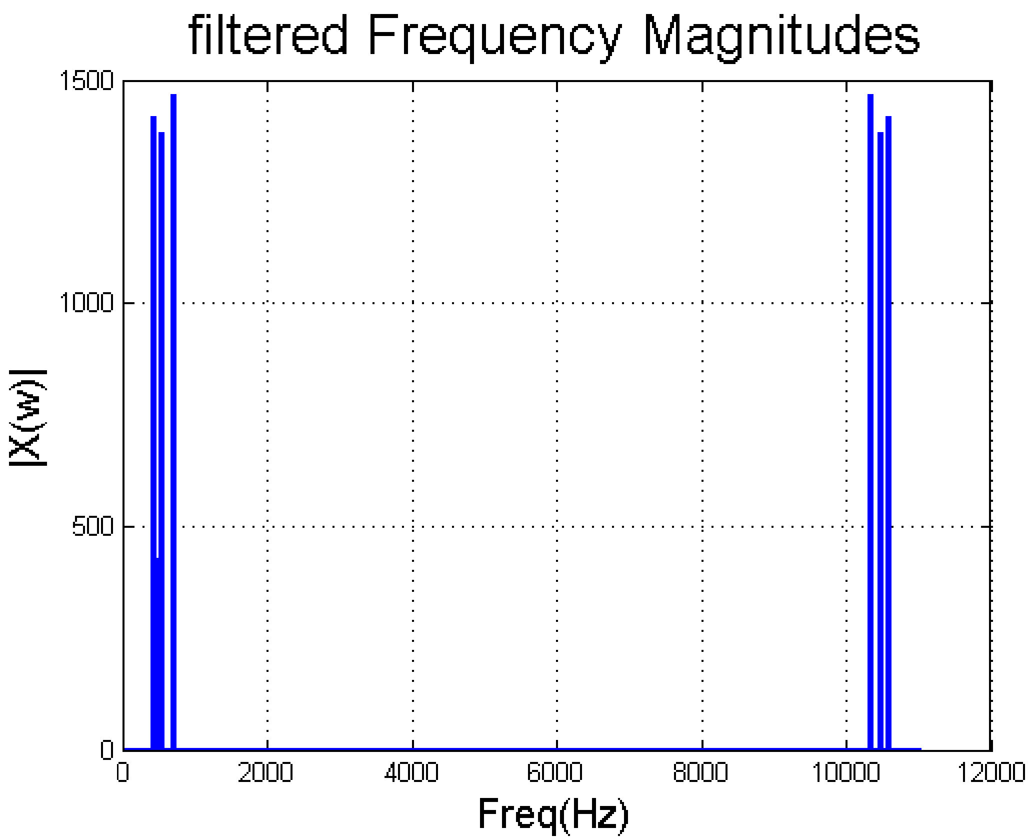
plot magnitude dB

```
figure()
plot(f, XF_dB); grid on; hold on;
line([0 L_XN_PADDED],[dB_threshold dB_threshold],'color','r','linewidth',2);
hold off;
xlabel('Freq(Hz)','fontsize',15);
ylabel('Power(dB)','fontsize',15);
title(strcat('|X(jw)|/max(|X(jw)|) > ',num2str(dB_threshold)),'fontsize',20);
legend('power', 'threshold')
```



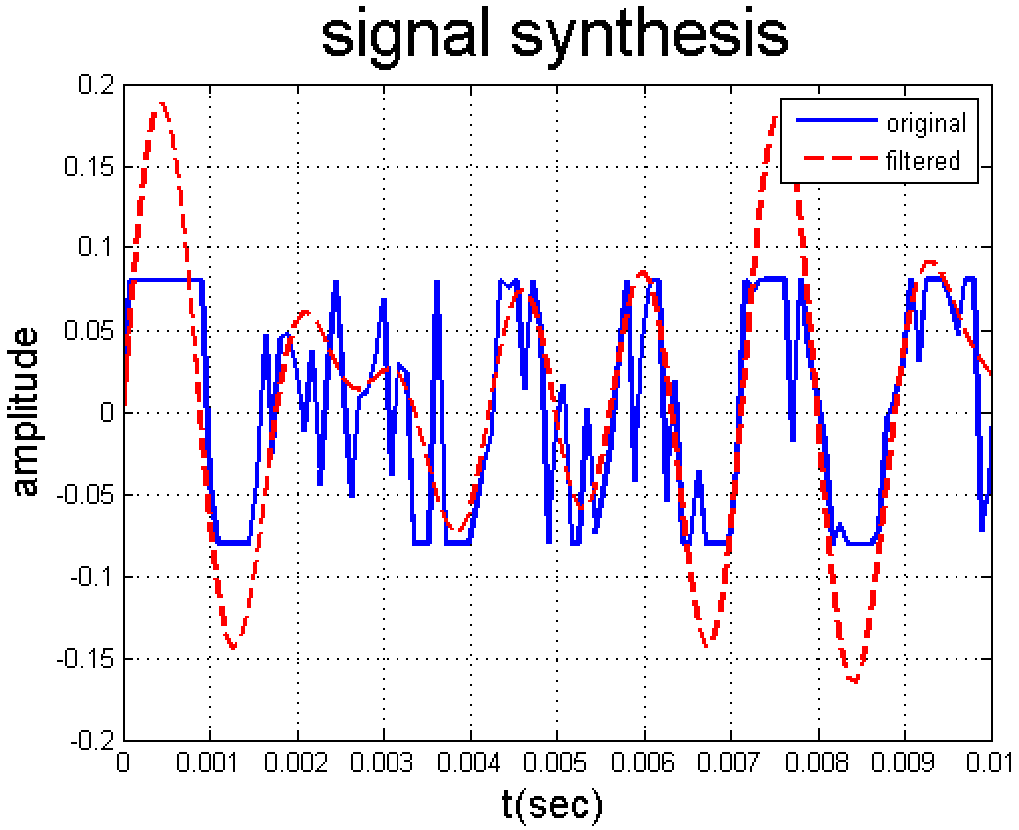
plot relevant frequencies

```
figure();
plot(f, abs(XF_syn),'linewidth',2); grid on
xlabel('Freq(Hz)','fontsize',15);
ylabel('|X(w)|','fontsize',15);
title('filtered Frequency Magnitudes','fontsize',20);
```



plot original signal and filtered signal

```
figure();
plot(1/FS*(0:1:L_XN-1), XN,'b','linewidth',2);% original signal
hold on;
plot(1/FS*(0:1:L_XN_PADDED-1), x_synthesis,'--r','linewidth',2); % synthesized sig
grid on; hold off
xlim([0,0.01])
legend('original','filtered')
title('signal synthesis','fontsize',25)
xlabel('t(sec)','fontsize',15);
ylabel('amplitude','fontsize',15)
```



display text for relevant frequencies

```
idx_filtered = find(XF_syn(1:floor(L_XN_PADDED/2)));% indices for dominant freq
mag_filtered = XF_syn(idx_filtered);   % magnitudes of dominant freq
[mag_filtered,idx] = sort(mag_filtered,1,'descend'); % sort magnitudes
idx_filtered = idx_filtered(idx);   % sort indices too
f_dominant = f(idx_filtered);   % get sorted dominant frequencies
fprintf('%d of frequencies are dominant \n\n', length(f_dominant))
for i = 1:length(f_dominant)
    fprintf('%d : %4.3f Hz \n', i,f_dominant(i))
end
```

```
9 of frequencies are dominant

1 : 699.829 Hz
2 : 440.085 Hz
3 : 549.770 Hz
4 : 700.502 Hz
5 : 550.443 Hz
6 : 439.412 Hz
7 : 440.758 Hz
8 : 699.156 Hz
9 : 549.097 Hz
```

Published with MATLAB® R2013b