Homework 4

Louis Lin

January 23, 2022

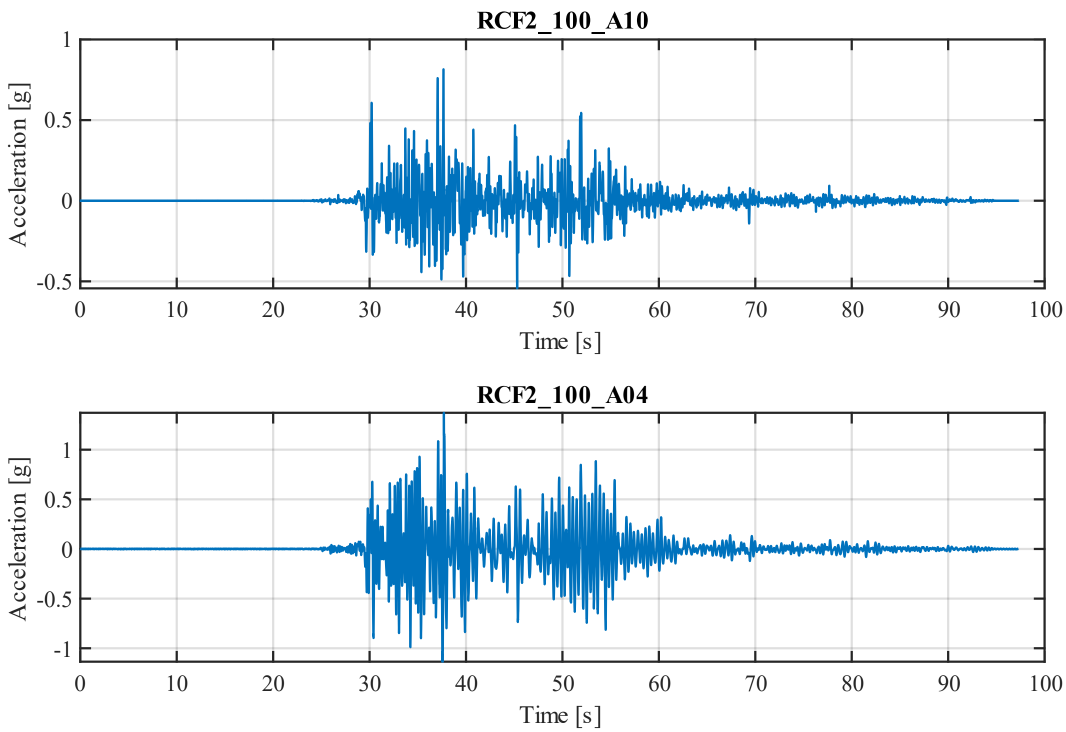
University of California, San Diego

Prof. Chin-Hsiung Loh

SE 267A Signal Processing and Spectral Analysis

Problem 4

The current homework examines the auto power spectral density function (PSD) and cross power correlation density function and the coherence between 2 signals. The two signals are from a reinforced concrete structure where the input signal comes from sensor RCF2_100_A10 and is considered base acceleration and output signal is from sensor RCF2_100_A04. The signals are given sampled at 200Hz.



The auto PSD and the cross PSD of the input and output signals are given below, where x denotes input and y denotes output. Two methods are used in the problem: using the Fourier transform of the respective correlation function given shown

$$\hat{S}_{xx}(f) = \int_{-P/2}^{P/2} \hat{R}_{xx}(\tau)\cos(2\pi f\tau)\,d\tau$$

$$\hat{S}_{xy}(f) = \int_{-P/2}^{P/2} \hat{R}_{xy}(\tau)\exp(-j2\pi f\tau)\,d\tau$$

or from the direct Fourier transformation, defined as

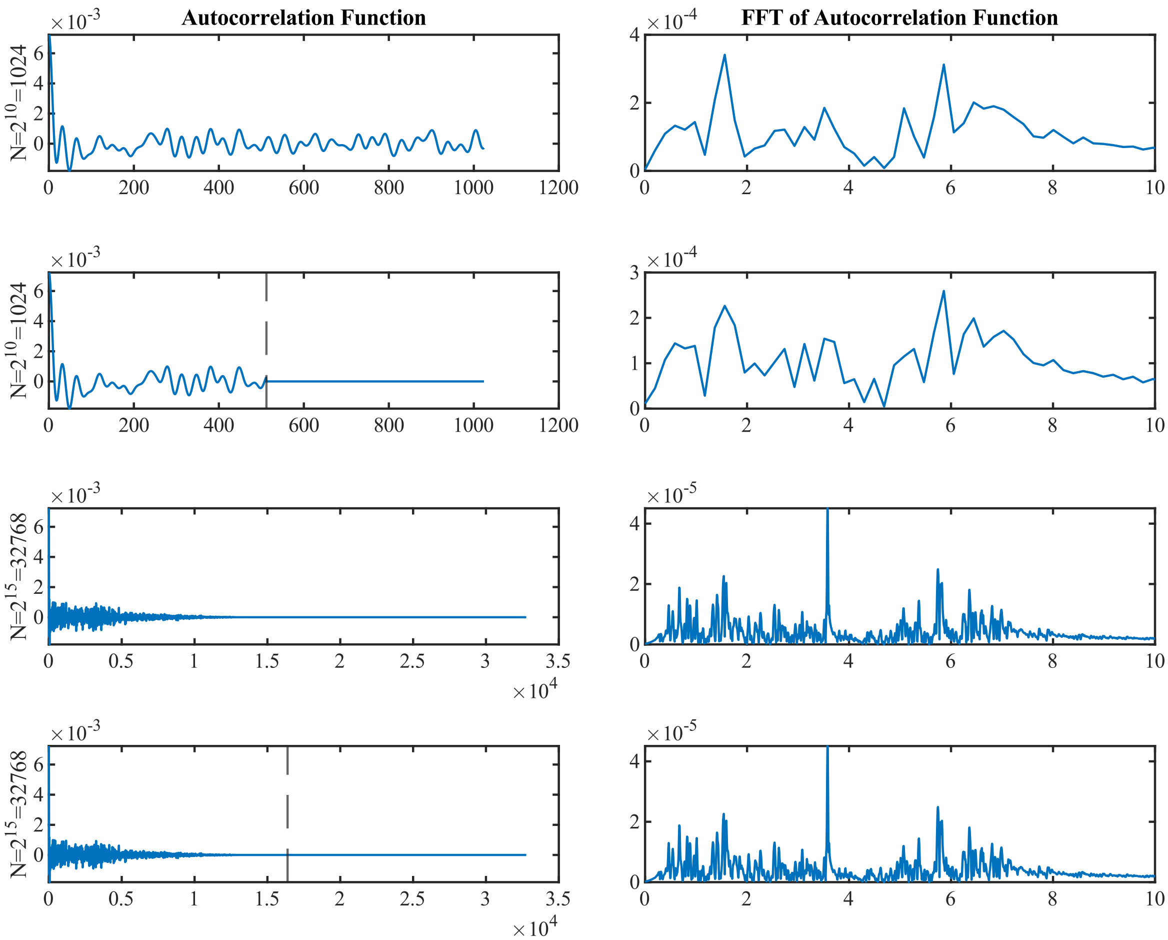
$$\hat{S}_{xx}(f) = \frac{1}{P}|X(f)|^2$$

$$\hat{S}_{xy}(f) = \frac{1}{P}|X^*(f)Y(f)|$$

where $X(f) = \int_{-\infty}^{\infty} x(\tau)e^{-j2\pi f\tau}d\tau = \mathcal{F}\{x(t)\}$ and $Y(f) = \int_{-\infty}^{\infty} y(\tau)e^{-j2\pi f\tau}d\tau = \mathcal{F}\{y(t)\}$ are the fast Fourier transforms of the $x(t)$ and $y(t)$ functions.

Using the correlation method to derive the PSD, the length of the correlation function determines the frequency resolution of the PSD. This gives the additional benefit of choosing the resolution for the PSD using the correlation method, but increases the step needed in comparing the two methods.

The signals from the test are themselves 19456 samples long. Compare the fast Fourier transform of the correlation function with length $2^{10} = 1024$ and $2^{15} = 32768$. It can be shown that increasing the length of the correlation function increases the frequency resolution of the PSD. If zeros are padded at ½ the length of the correlation function, it is also shown to increase the frequency resolution of the PSD. This is used in the following solution.

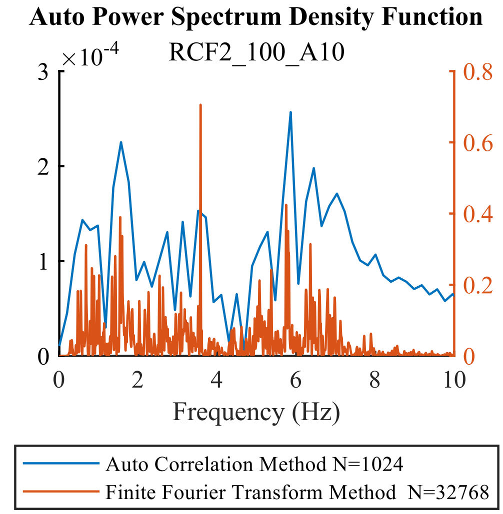


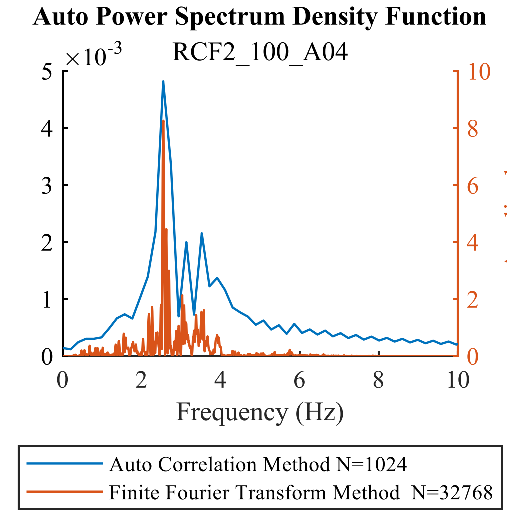
This frequency resolution is important to consider when comparing the results of the two methods; two methods are explored:

1) One is to match the resolution of the correlation function method PSD with that of the fourier transform method PSD. Say if 2^{15} points is used for the fast fourier transform, then the correlation function would use a maxlag of 2^{15} even though the signal length is not long enough. This would give 2^{14} useful points for both PSD.
2) Another is to pad the correlation fucntion with zeros to match the length of the fourier transform method. Although this will not add new information to the PSD, the magnitude of the two PSD will be now be comparable.

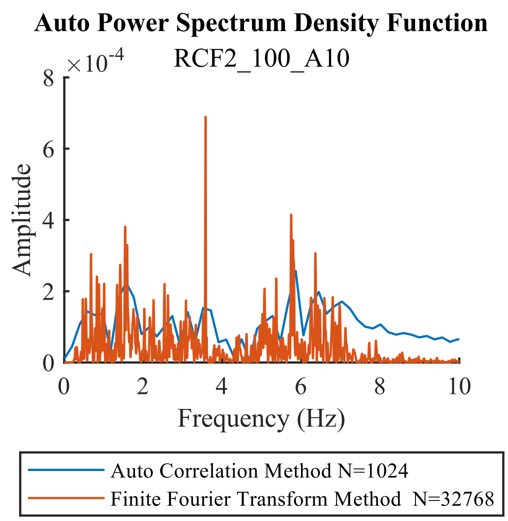
Regardless of the resolution between the PSD functions, in order to compare the power magnitude of the FFT method PSD function must be adjusted by resolution of the correlation function PSD function. E.g. if the 2^{15} points were used for the FFT method and a maxlag of 2^{10} was used for the correlation function, then the magnitude of the FFT method PSD function must be scaled down by 2^{10}.

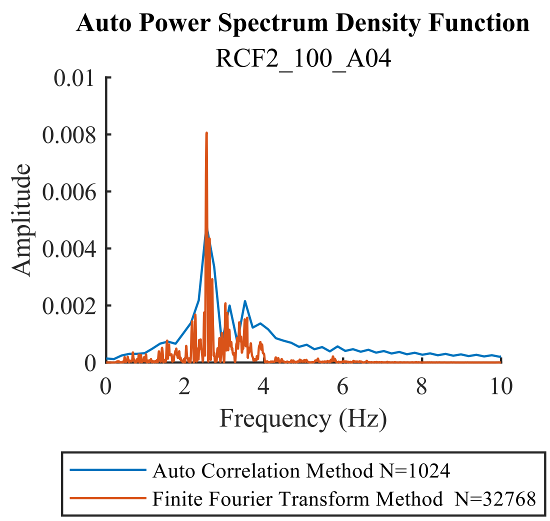
Below, I demonstrate using a maxlag for the correlation function of 2^{10} compared with the PSD using 2^{15} points for the FFT method.



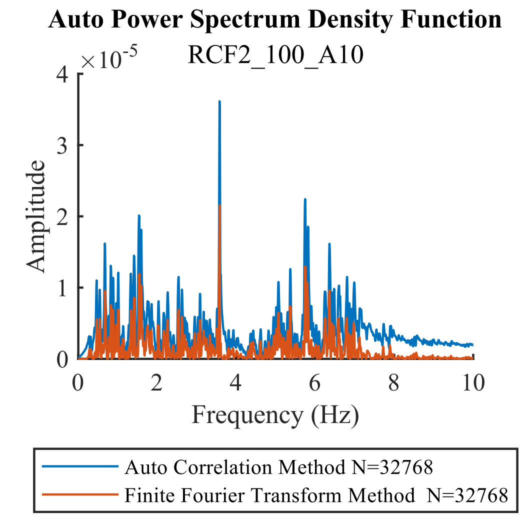


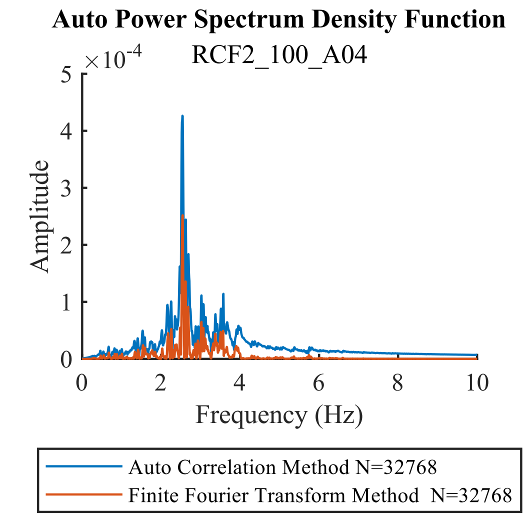
While the peaks of both spectrums are similar, it can be seen that the finite fourier transform method has a frequency resolution that is much higher than that of the correlation function method. The amplitudes of the PSD shown on the left and right y-axis are shown to not be comparable. If we divide the FFT PSD by 2^{10}, then the two can be plotted on the same y-axis as shown even though we are using different resolution for the methods.



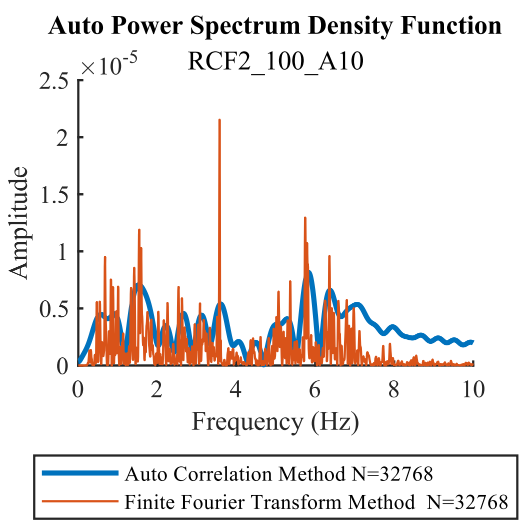


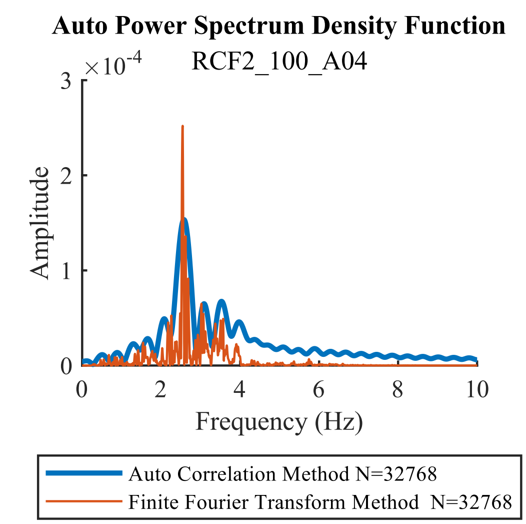
If we increase the resolution of the correlation function to have a maximum lag of 2^{15} to match that of the points used in the FFT method (and also adjust the FFT PSD by $1/(2^{15})$) then the resoltion between the two PSD can be shown to be comparable!



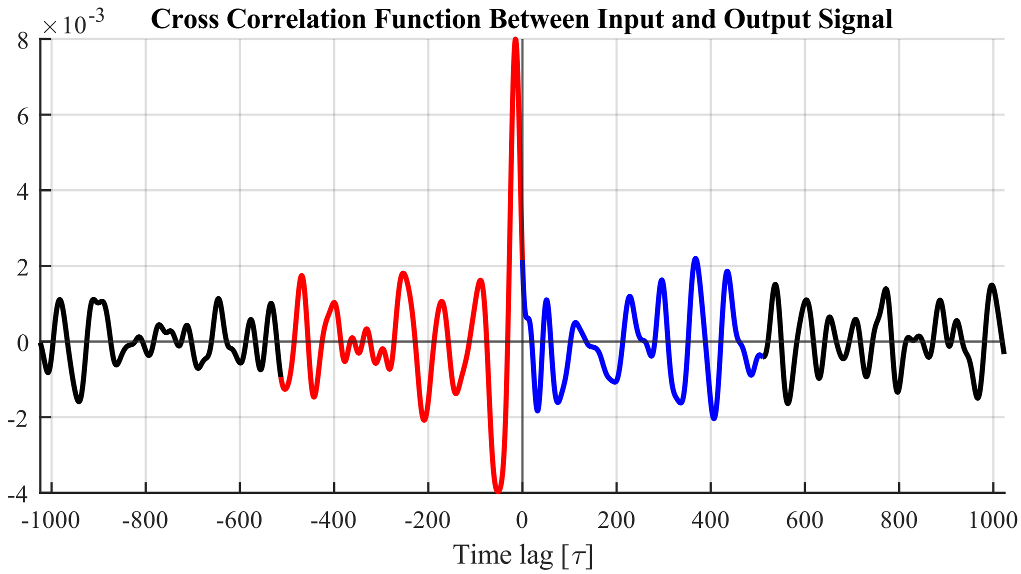


One last demonstration would be the padding zeros to the end of the correlation function to match the length of the FFT method with shows a much higher resolution PSD using the correlation function. Although the correlation function is 2^{15} long, it only has values for 2^{9} samples. The linewidth was changed to 2 to better show the correlation method.

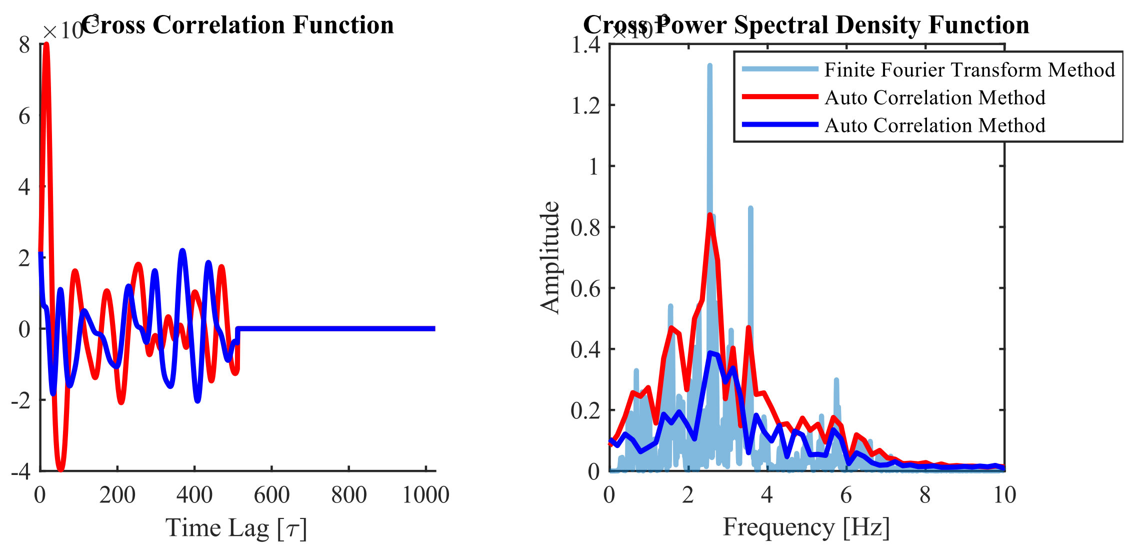




The formulation for the auto correlation function was given above, and shown below. The important thing to note is the autocorrelation between the signals is not symmetric so the portion of the autocorrelation function used is important.



In the following, the PSD using the red and blue portion of the cross correlation function is plotted afainst the FFT method.

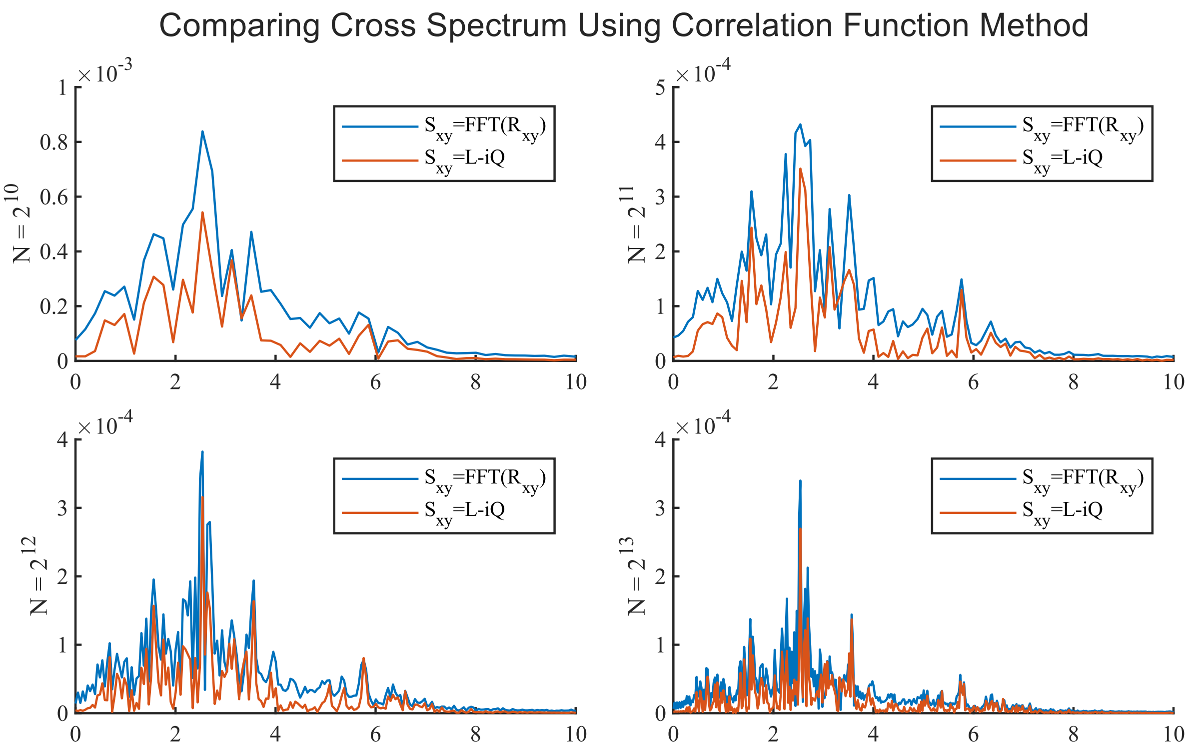


Comparing across the methods, it is shown that the red portion is the correct portion to select when calculaing the CSPD using the correlation function method.

The cross spectrum can also be calculated in two ways using the correlation method. One is calculate the co-spectrum and quadra-spectrum and combine them to form the cross power spectrum. Another is to directly fast fourier transform the cross correlation function.

The second method is straightfoward. For the first method, the procedure is briefly explained. The cross correlation between the input and output is calculated both ways, where the input and output are in the correct order and another where they are swapped. This this then used to find the co-cross correltaion and quadra cross correlation functions that are then transformed into the frequency domain to form the co-spectrum and quadra spectrums. The cross spectrum is the read part of the co-spectrum minus the imaginary part of the quadra spectrum.

I found that copmparing the two methods, the spectrum were slightly different and the results are shown below. In the following coherence calculation the $S_{xy}(f) = L(f) - iQ(f)$ was used.

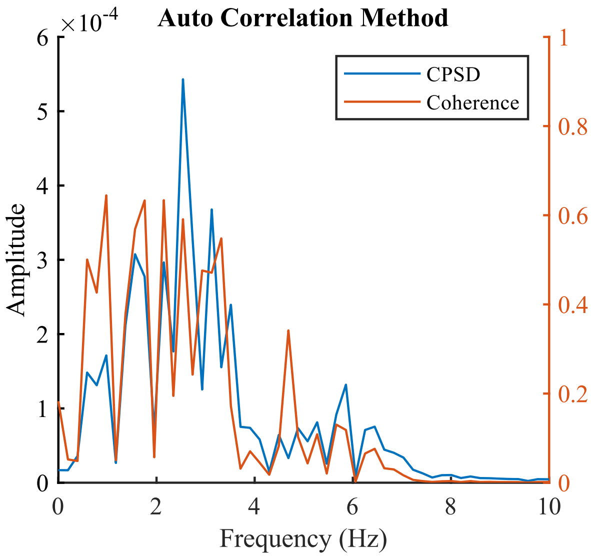


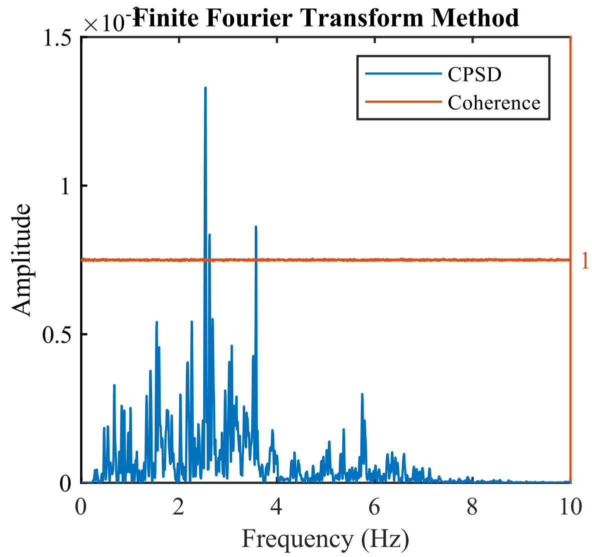
The coherence function between the input $x(t)$ and ouptut $y(t)$ is defined as

$$\gamma_{xy}^2(f) = \frac{|G_{xy}(f)|^2}{G_{xx}(f)G_{yy}(f)} = \frac{|S_{xy}(f)|^2}{S_{xx}(f)S_{yy}(f)}$$

where

$$0 \le \gamma_{xy}^2 \le 1$$





It can seen that the coherence using the finite fourier transform method is unusable since the transformation cancel each other out when the coherence spectrum is calculated. The coherence of the autocorrelation method is seem to be around 0.6 which is pretty low for the signal. The coherence can be less than 1 due to extraneous noise in the measurements, the system relating $x(t)$ and $y(t)$ is nonlinear, or $y(t)$ is an output due to more than just input $x(t)$. Most like it's because the excitation was not a white noise input and not all frequency

```
%% Load Data
clear; clc;
data = readtable("Homework-2 data set-RCF-Four Specimen Test Data.xlsx");

fs = 200;

%% Plot the Auto Power Spectrum Desntiy Function

figure(1); clf; tiledlayout(1,2);
ii = 1;

for channels = ["RCF2_100_A10","RCF2_100_A04"]
   figure(1); nexttile(ii); hold on;  ii = ii+1;
   signal = data.(channels);

   % Auto Correlation Function Method
   maxlag = 2^10;
   Rxx = xcorr(signal, maxlag, 'unbiased'); % autocorrelation
   Rxx = flip(Rxx(1:maxlag)); % Select only the relevant portion of the spectrum
   Rxx(maxlag/2:end) = 0; % Increase frequency resolution while also trimming unreliable data
%        Rxx(maxlag/2:2^15) = 0; % Increase frequency resolution while also trimming unreliable data
   Sxx = abs(fft(Rxx)); % FFT to obtain auto-spectrum
   npt = length(Sxx);
   fxx = (0:npt-1)* fs /npt; %
   id = (1:npt/2);
   plot(fxx(id) ,Sxx(id)*2/npt, DisplayName=["Auto Correlation Method N="+npt], LineWidth=2); %

   % Finite Fourier Transformation Method
%     yyaxis right
   npt = length(signal);
   npt = 2^nextpow2(npt);
   X = fft(signal,npt);
   Sxx = X.*conj(X)/npt;
   Sxx = Sxx/maxlag;
   f = (0:npt-1)*fs/npt;
   id = (1:npt/2); % First half
   plot(f(id), Sxx(id),DisplayName="Finite Fourier Transform Method  N="+npt);
   xlabel ("Frequency (Hz)"); %
   ylabel ('Amplitude'); %
   title('Auto Power Spectrum Density Function',channels,Interpreter='none'); %
   xlim([0,10]);
   legend(Location="southoutside")
end

%% Cross Power Spectrum Density Function

% Plot the Auto Power Spectrums
figure(2); clf;  hold on;
```

```
x = data.RCF2_100_A10;
y = data.RCF2_100_A04;

% Auto Correlation Function Method
maxlag = 2^10;
Rxy = xcorr(x, y, maxlag, 'unbiased'); % autocorrelation
Rxy = flip(Rxy(1:maxlag)); % Select only the relevant portion of the spectrum
Rxy(maxlag/2:end) = 0; % Increase frequency resolution while also trimming unreliable data
Sxy = abs(fft(Rxy)); % FFT to obtain auto-spectrum
npt = length(Sxy);
fxy = (0:npt-1)* fs /npt; %
id = (1:npt/2);
plot(fxy(id) ,Sxy(id)*2/npt, DisplayName="Auto Correlation Method"); %

% Finite Fourier Transformation Method
npt = length(x);
npt = 2^nextpow2(npt);
X = fft(x,npt);
Y = fft(y,npt);
Sxy = abs(Y.*conj(X));
Sxy = Sxy/npt;
Sxy = Sxy/maxlag;
f = (0:npt-1)*fs/npt;
id = (1:npt/2); % First half
plot(f(id), Sxy(id),DisplayName="Finite Fourier Transform Method");

xlabel ("Frequency (Hz)"); %
ylabel ('Amplitude'); %
title('Cross Power Spectrum Density Function',Interpreter='none'); %
xlim([0,10]);
legend()
```

Figure 1 Auto and Cross Spectrums

```
%% Load Data
clear; clc;
data = readtable("Homework-2 data set-RCF-Four Specimen Test Data.xlsx");
x = data.RCF2_100_A10;
y = data.RCF2_100_A04;

fs = 200;

figure(1); clf;
tiledlayout(1,2,"TileSpacing","compact",Padding="compact")
% Auto Correlation Function Method
nexttile(); hold on;
maxlag = 2^12;

% Autocorrelation of input
Rxx = xcorr(x, maxlag, 'unbiased'); % autocorrelation
Rxx = flip(Rxx(1:maxlag)); % Select only the relevant portion of the spectrum
Rxx(maxlag/2:end) = 0; % Increase frequency resolution while also trimming unreliable data
Sxx = abs(fft(Rxx)); % FFT to obtain auto-spectrum
% Autocorrelation of output
Ryy = xcorr(y, maxlag, 'unbiased'); % autocorrelation
Ryy = flip(Ryy(1:maxlag)); % Select only the relevant portion of the spectrum
Ryy(maxlag/2:end) = 0; % Increase frequency resolution while also trimming unreliable data
Syy = abs(fft(Ryy)); % FFT to obtain auto-spectrum
% Finding Cross Spectrum
[Rxy] = xcorr(x, y,maxlag, "unbiased");
[Ryx] = xcorr(y,x, maxlag, 'unbiased');
Rxy = flip(Rxy(1:maxlag)); % select only half of data
Ryx = flip(Ryx(1:maxlag)); %select only half of data

Rxy(maxlag/2:end) = 0; % zeros out the unreliable data
Ryx(maxlag/2:end) = 0; %select only half of data

lxy = 0.5* (Rxy+Ryx); % Co-Correlation
qxy = 0.5* (Rxy-Ryx); % Quadra-Correlation

Lxy = real(fft(lxy)); %Co-Spectrum
Qxy = imag(fft(qxy)); % Quadra-Spectrum
Sxy = Lxy-Qxy*1i; % Cross Spectrum
%
% Rxy = xcorr(x, y, maxlag, 'unbiased'); % autocorrelation
% Rxy = flip(Rxy(1:maxlag)); % Select only the relevant portion of the spectrum
% Rxy(maxlag/2:end) = 0; % Increase frequency resolution while also trimming unreliable data
% Sxy = abs(fft(Rxy)); % FFT to obtain auto-spectrum

npt = length(Sxy);
fxy = (0:npt-1)* fs /npt; %
id = (1:npt/2);
```

```
plot(fxy(id) , abs(Sxy(id))*2/npt, DisplayName="CPSD "); %

ylabel ('Amplitude');
yyaxis right
gamma = abs(Sxy).^2./(Sxx.*Syy);
% % gamma(1)=1;
plot(fxy(id), gamma(id), DisplayName="Coherence"); %
xlabel ("Frequency (Hz)"); %
title('Auto Correlation Method',Interpreter='none'); %
ylim([0,1] * 1)
legend()
% yline(1,"--","Theoretical Limit","HandleVisibility","off")
xlim([0,10])
legend(Location="northeast")

%%

nexttile()
npt = 2^nextpow2(length(x));
X = fft(x,npt);
Sxx = X.*conj(X)/npt/maxlag;

npt = 2^nextpow2(length(y));
Y = fft(y,npt);
Syy = Y.*conj(Y)/npt/maxlag;

Sxy = abs(Y.*conj(X));
Sxy = Sxy/npt/maxlag;

id = (1:npt/2); % First half
f = (0:npt-1)*fs/npt;
plot(f(id), Sxy(id),DisplayName="CPSD");
ylim([0,1] * 0.0015)
xlabel ("Frequency (Hz)"); %
ylabel ('Amplitude'); %

yyaxis right
gamma = Sxy.^2./(Sxx.*Syy);
plot(f(id), gamma(id), DisplayName="Coherence"); %

% yline(1,"--","Theoretical Limit","HandleVisibility","off")
title('Finite Fourier Transform Method',Interpreter='none'); %
xlim([0,10]);
yticks(1)
legend(Location="northeast")
```

Figure 2 Coherence Plots