Table of Contents

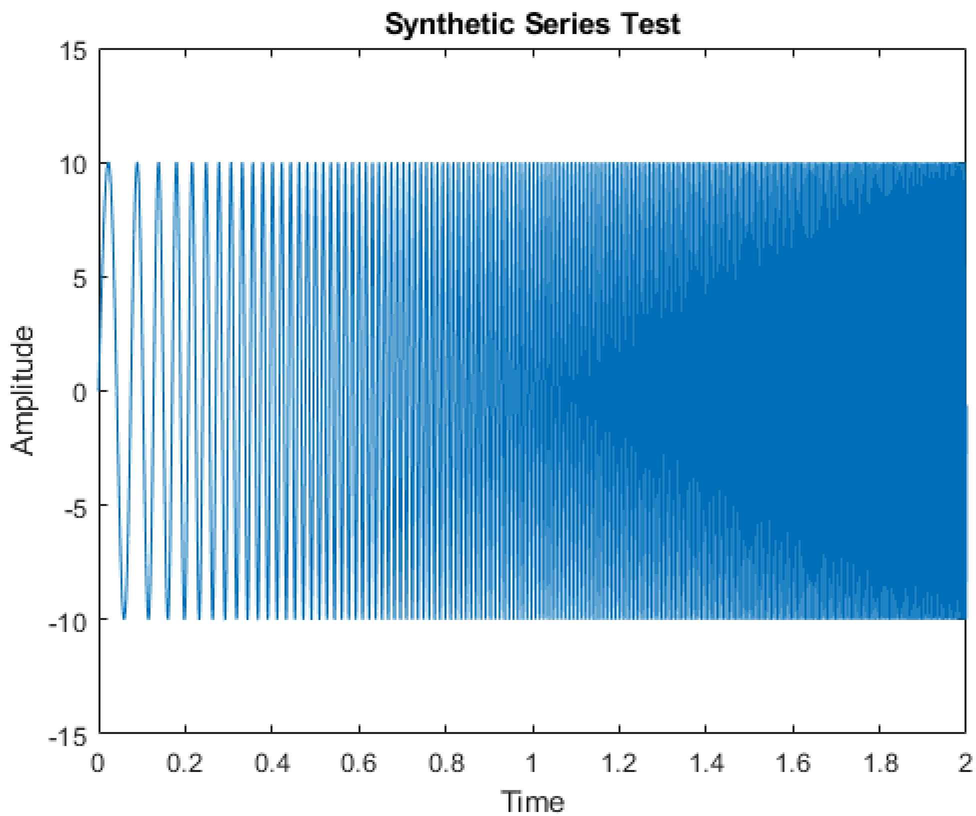

```
%Spectral Analysis Practice and Experimentation
%
% --- Objective List --- %
% 1) Create a function that creates a synethetic series to allow for
% experimentation. Allow inputs for amplitude, variable frequency, phase,
% linear trend, and SNR inputs. Test function works as expected by using a
% CWT.
%
% 2) Derive the auto-covariance function from a power spectrum and compare
%
% 3) Calculate the ACVF, amplitude spectrum, PSD, and least squares
% spectrum (in percentage variance, PSD, dB) and compare.
%
% 4) Add a linear trend, therby violating Fourier strict requirements, and
% compare against LSSA... which has no such strict requirements and
% investigate.
%
% 5) Investigate spectral response with varying frequency and high noise.
%
% 6) Introduce peak that will be aliased and determine the aliased peak.
%
% 7) Experiment with variable length of series and see when peak seperation
% can occur.
%
% 8) Experiment with windowing and see effect of smoothing ... peak
% seperation
%
% --- Question List -- %
```

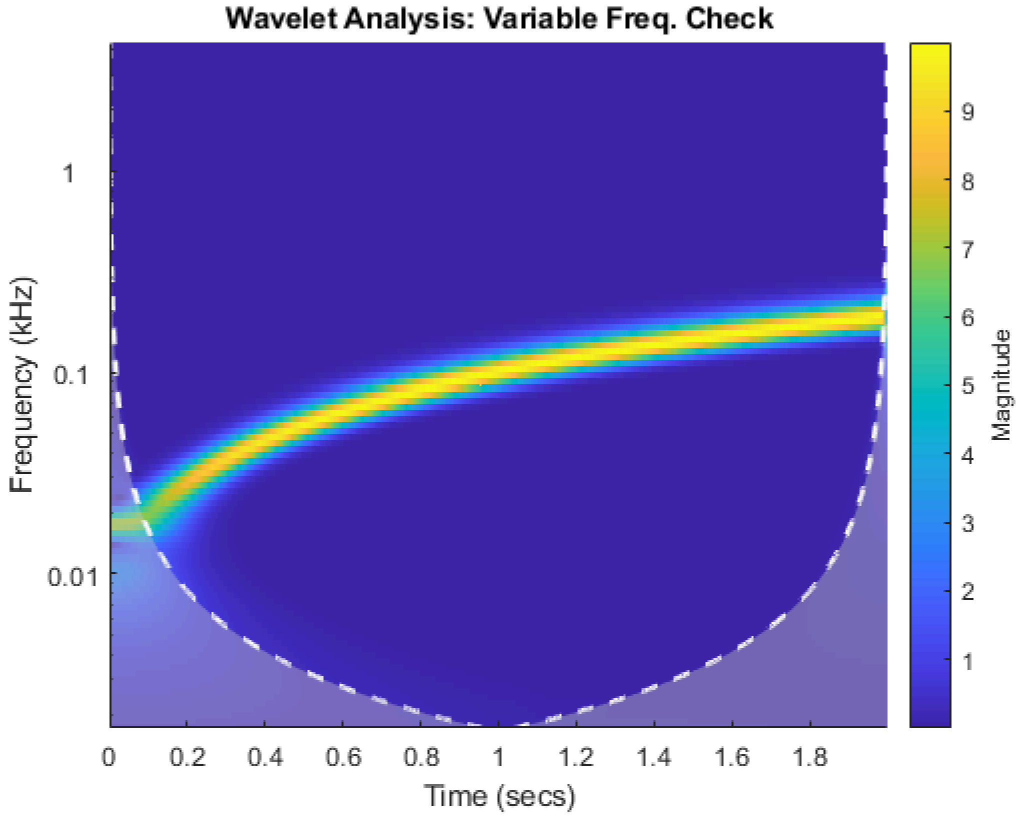
1) Test user defined synethetic series creation function

```
%--- Setting Parameters for synthetic_series
Fs = 10000; %sample freq
T = 2;      %duration of signal
```

```
freq_start = 10; freq_end = 100;
phase = 0; amp = 10; SNR_dB = 100; linear_trend_frac = 0;

%--- Call created function for developing synthetic series for experiem
[time, signal] = synthetic_series(amp, freq_start, freq_end, phase, T, Fs,
 SNR_dB,linear_trend_frac); %evaluated function

%--- Plot series, check SNR on constructed series, and use wavelet to see
 variable freq.
figure(1); plot(time, signal); title('Synthetic Series Test'); xlabel('Time');
 ylabel('Amplitude');
Evaluated_SNR = snr(signal);
fprintf('Evaluated SNR is: %i. Inputed SNR is %i.\n', round(Evaluated_SNR),
 SNR_dB);
figure(2); cwt(signal,Fs); title('Wavelet Analysis: Variable Freq. Check');
```

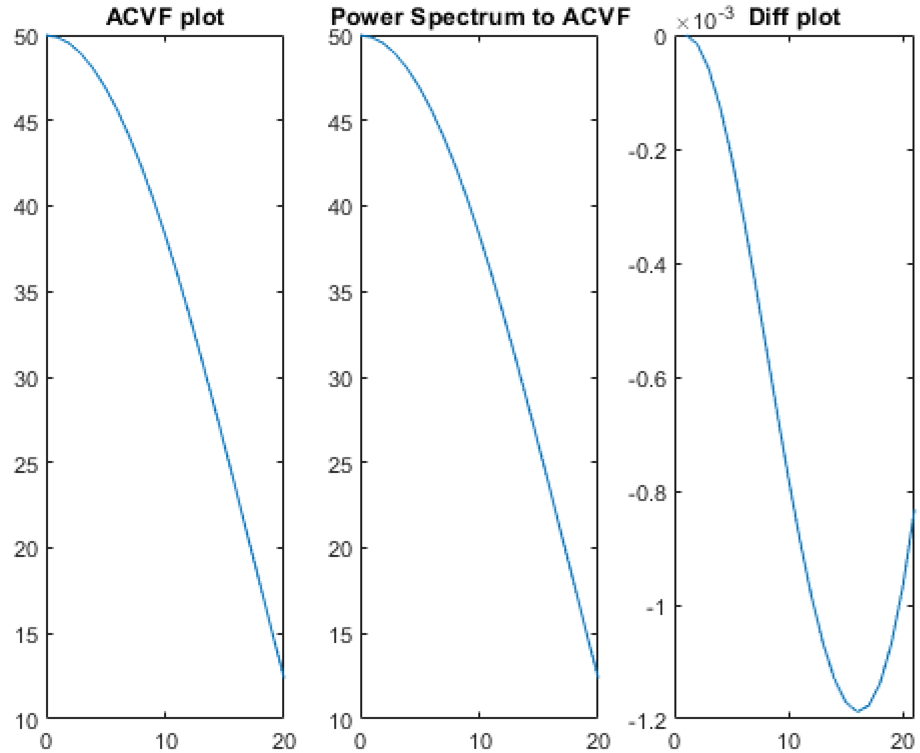
Evaluated SNR is: 100. Inputed SNR is 100.

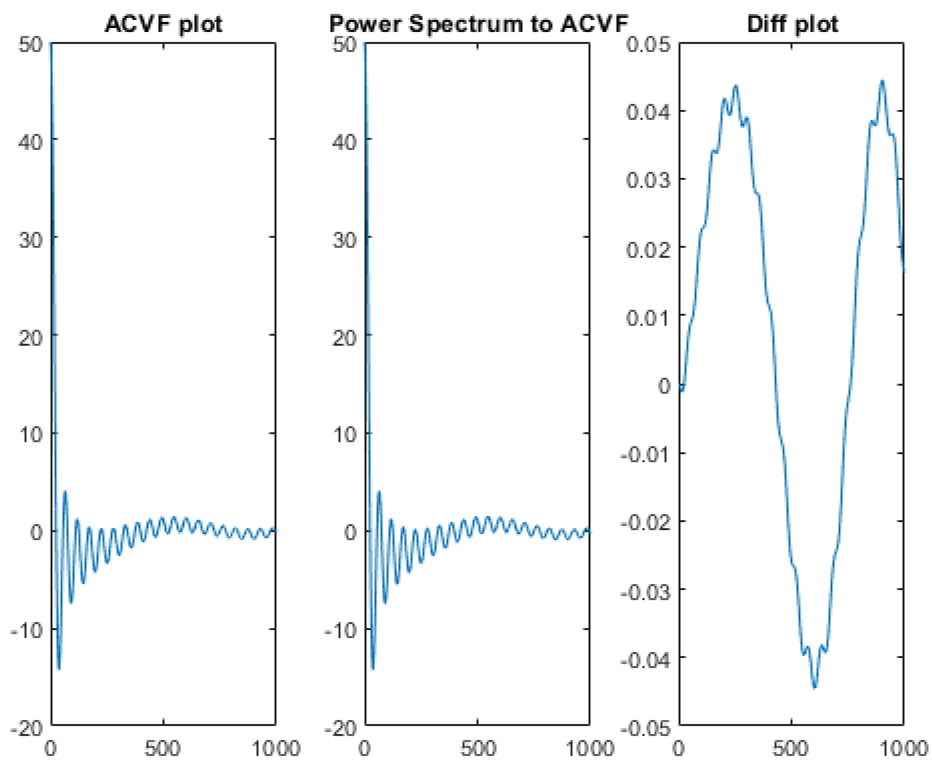




2) Test to see if I can derive ACVF from Power Spectrum

```
close all;
% --- Calculating Power Spectrum
n = length(signal);
spec = ((abs(fft(signal)).^2))/n; %2 sided spectrum
invspec = ifft(spec);

% --- Calculating ACVF Func
[acf, lags] = autocorr(signal, 'NumLags', 1000); acvf = acf*rms(signal)^2;

% --- Plotting with lots of lags and then without lots of lags
figure(3); subplot(1,3,1); plot(lags(1:21), acvf(1:21)); title('ACVF plot');
subplot(1,3,2); plot(lags(1:21),invspec(1:21)); title('Power Spectrum to
 ACVF');
subplot(1,3,3); plot(acvf(1:21)-invspec(1:21)); title('Diff plot');

figure(4); subplot(1,3,1); plot(lags, acvf); title('ACVF plot');
subplot(1,3,2); plot(lags,invspec(1:1001)); title('Power Spectrum to ACVF');
subplot(1,3,3); plot(acvf-invspec(1:1001)); title('Diff plot');
```

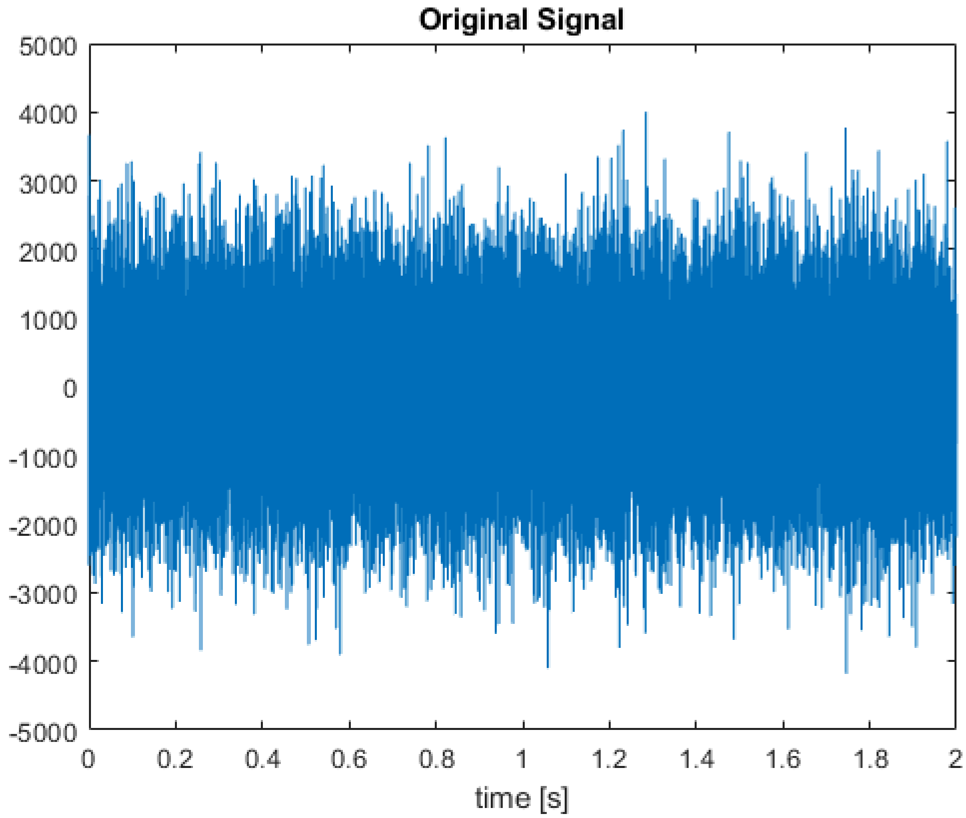

ACVF plot
Power Spectrum to ACVF
×10^-3 Diff plot
50
45
40
35
30
25
20
15
10
0
10
20
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2

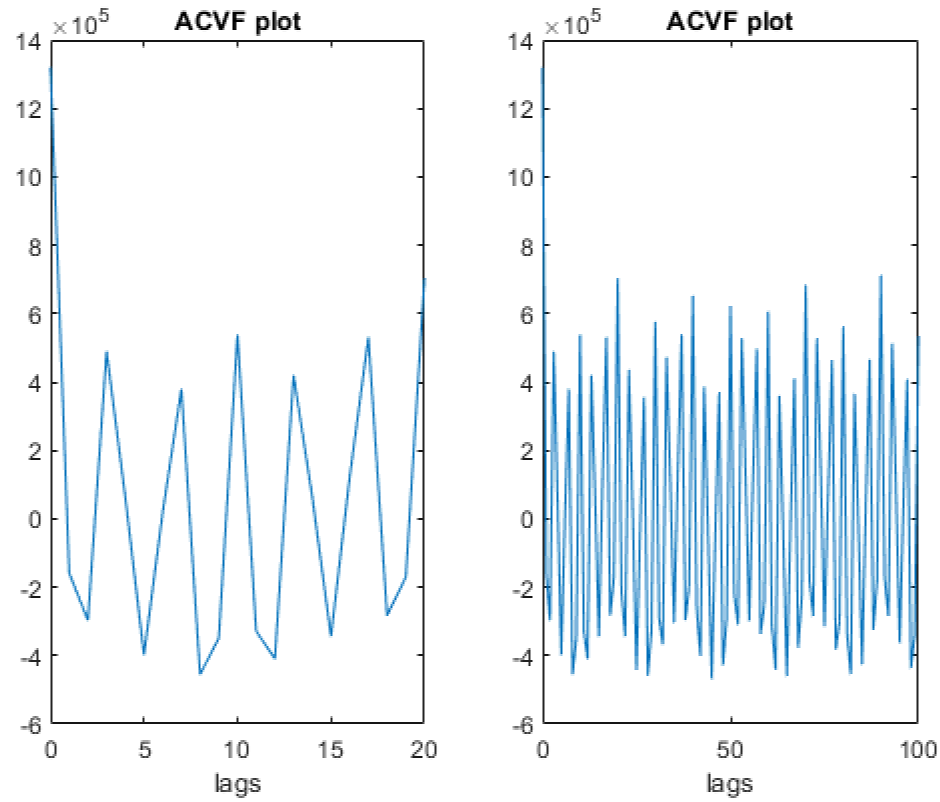

ACVF plot
Power Spectrum to ACVF
Diff plot
50
40
30
20
10
0
-10
-20
0
500
1000
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05

3) Test collection of series 1 on the 4 methods

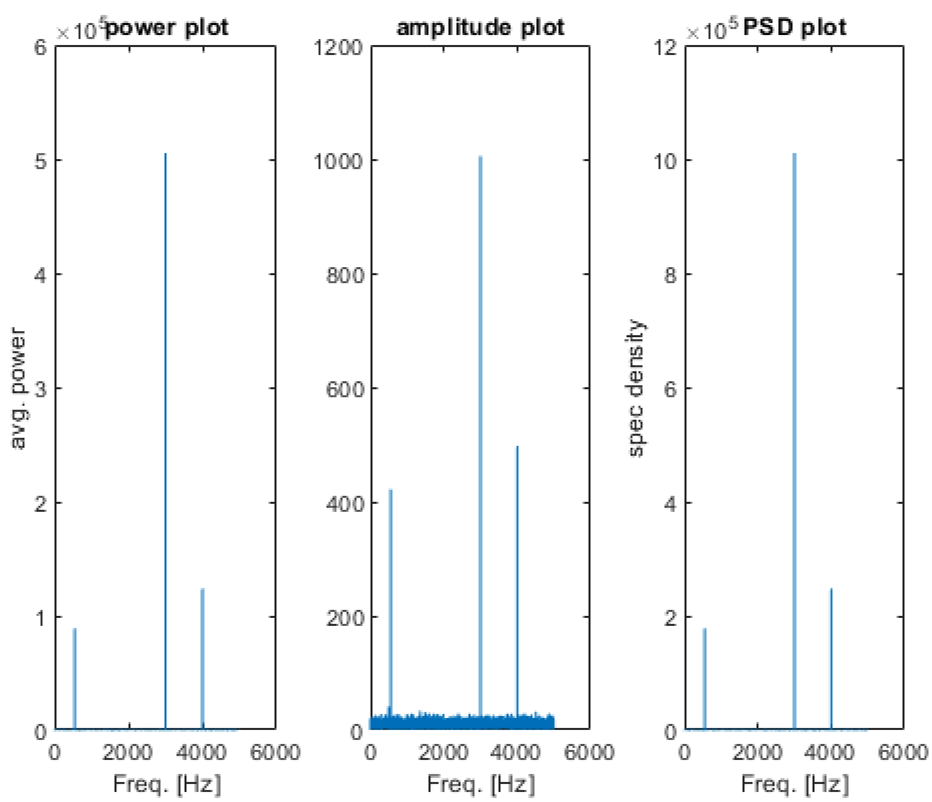
```
clearvars; close all

%--- Series has no linear trend, no variable freq.
%--- series has freq 500 and 550 with amp 20 and 420
%--- Testing smoothing effect of different plotting scales

%--- Series Parameters
amp = [10 20 420 1000 500];
phase = [0 60 20 15 10]; % degrees
freq_start = [300 500 550 3000 4000];
freq_end = freq_start;
Fs = 10000; %significantly over sampled
T = 2;
SNR_dB = 5;
linear_trend_frac = [0, 0, 0, 0, 0];

%--- Constructing Series
[time, signal] = multiple_synthetic_series(amp, freq_start, freq_end, phase,
 T, Fs, SNR_dB, linear_trend_frac);
figure(100);
plot(time,signal); title('Original Signal'); xlabel('time [s]');
%--- ACVF
[acf, lags] = autocorr(signal, 'NumLags', 100); acvf = acf*rms(signal)^2;
figure(1); subplot(1,2,2); plot(lags, acvf); title('ACVF plot');
 xlabel('lags');
subplot(1,2,1); plot(lags(1:21),acvf(1:21)); title('ACVF plot');
 xlabel('lags');

%--- Amplitude and Power and PSD in regular plot
figure(2);
[pxx,w] = periodogram(signal, rectwin(length(signal)), length(signal),
 Fs,'power');
[freq, amp] = my_FFT(time, signal);
subplot(1,3,1); plot(w,pxx); title('power plot'); ylabel('avg. power');
 xlabel('Freq. [Hz]');
subplot(1,3,2); plot(freq,amp); title('amplitude plot'); xlabel('Freq. [Hz]');
[pxx,w] = periodogram(signal, rectwin(length(signal)), length(signal), Fs);
subplot(1,3,3); plot(w,pxx); title('PSD plot'); ylabel('spec density');
 xlabel('Freq. [Hz]');

%--- Amplitude and Power and PSD in dB
figure(3);
[pxx,w] = periodogram(signal, rectwin(length(signal)), length(signal),
 Fs,'power');
subplot(1,3,1); plot(w,10*log10(pxx)); title('power dB plot'); ylabel('avg.
 power'); xlabel('Freq. [Hz]');
subplot(1,3,2); plot(freq,20*log10(amp)); title('amplitude dB plot');
 xlabel('Freq. [Hz]');
[pxx,w] = periodogram(signal, rectwin(length(signal)), length(signal), Fs);
subplot(1,3,3); plot(w,10*log10(pxx)); title('PSD dB plot'); ylabel('spec
 density'); xlabel('Freq. [Hz]');
```

```
%--- Least Squares Spectrum in regular plot
figure(4);
[spectrum, CritVal, CScoeff, res, norm_res, coeff, cov, Omega] = LSSA(time,
 signal);
PSD = LSSA_helper(CScoeff,'PSD', Omega);
amp = LSSA_helper(CScoeff,'amplitude');
subplot(1,3,2); plot(Omega,amp); title('amplitude plot (LSSA)'); xlabel('Freq.
 [Hz]');
subplot(1,3,3); plot(Omega,PSD); title('PSD plot (LSSA)'); ylabel('spec
 density'); xlabel('Freq. [Hz]');
subplot(1,3,1); plot(Omega,spectrum); title('Percentage Var. (LSSA)');
 ylabel('spec density'); xlabel('Freq. [Hz]');

%--- dB plot for PSD and Amplitude
figure(5);
subplot(1,2,1); plot(Omega,20*log10(amp)); title('amplitude dB plot (LSSA)');
 xlabel('Freq. [Hz]');
subplot(1,2,2); plot(Omega,10*log10(PSD)); title('PSD dB plot (LSSA)');
 ylabel('spec density'); xlabel('Freq. [Hz]');
```

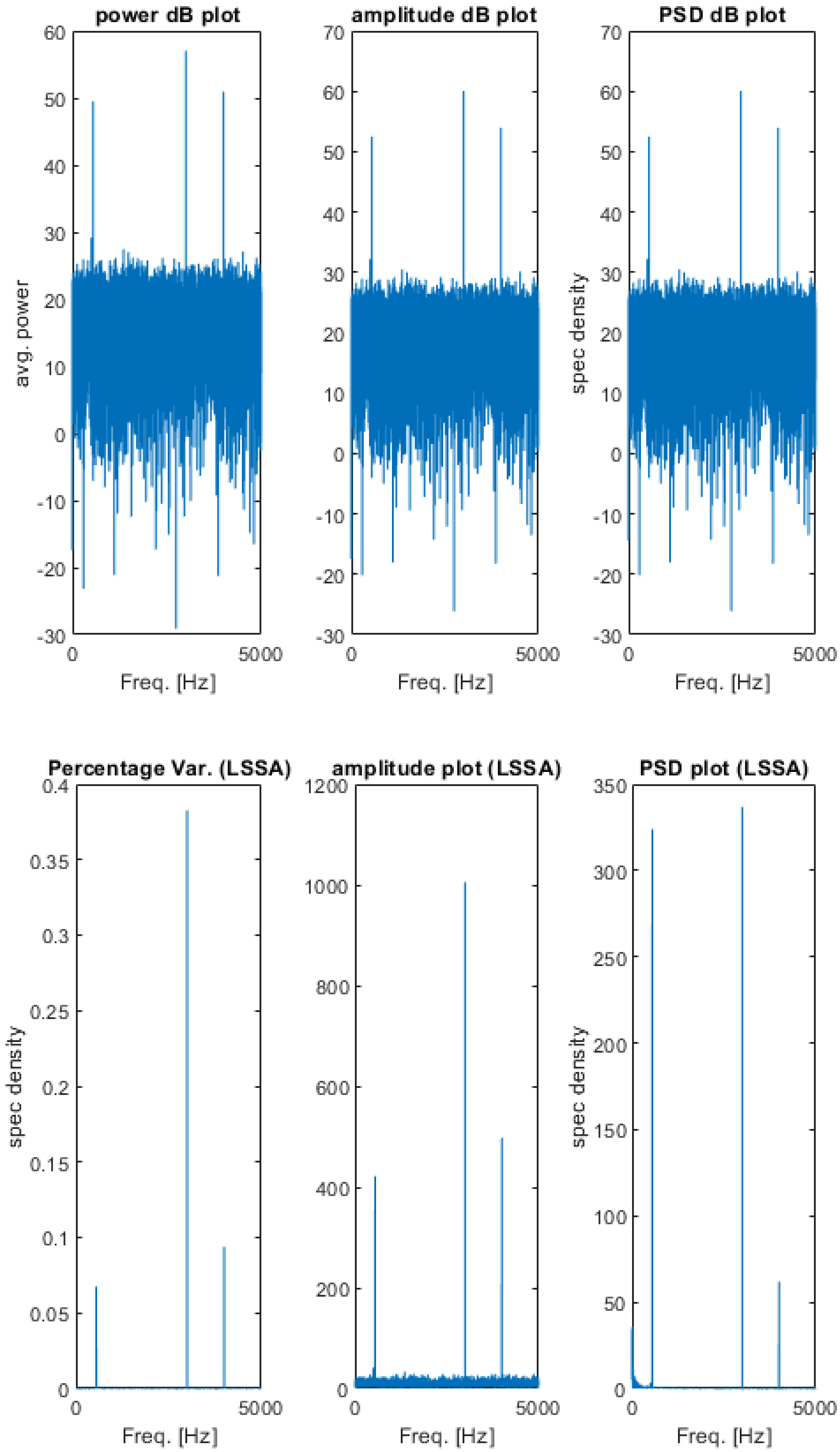


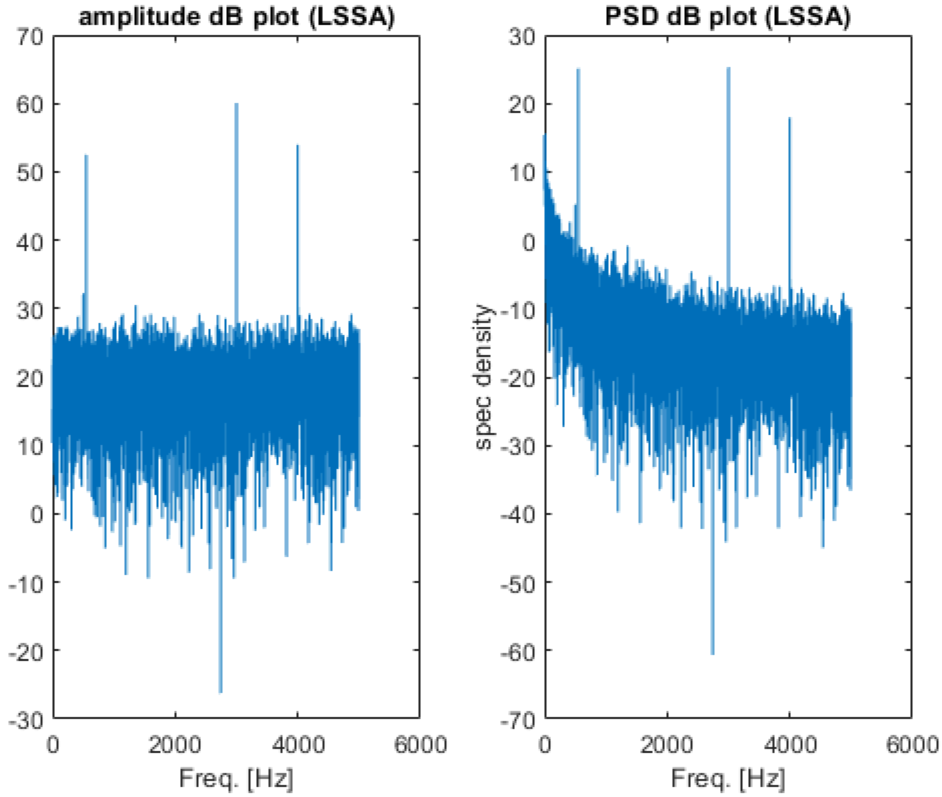


ACVF plot
ACVF plot
lags
lags



power plot
amplitude plot
PSD plot
avg. power
spec density
Freq. [Hz]
Freq. [Hz]
Freq. [Hz]

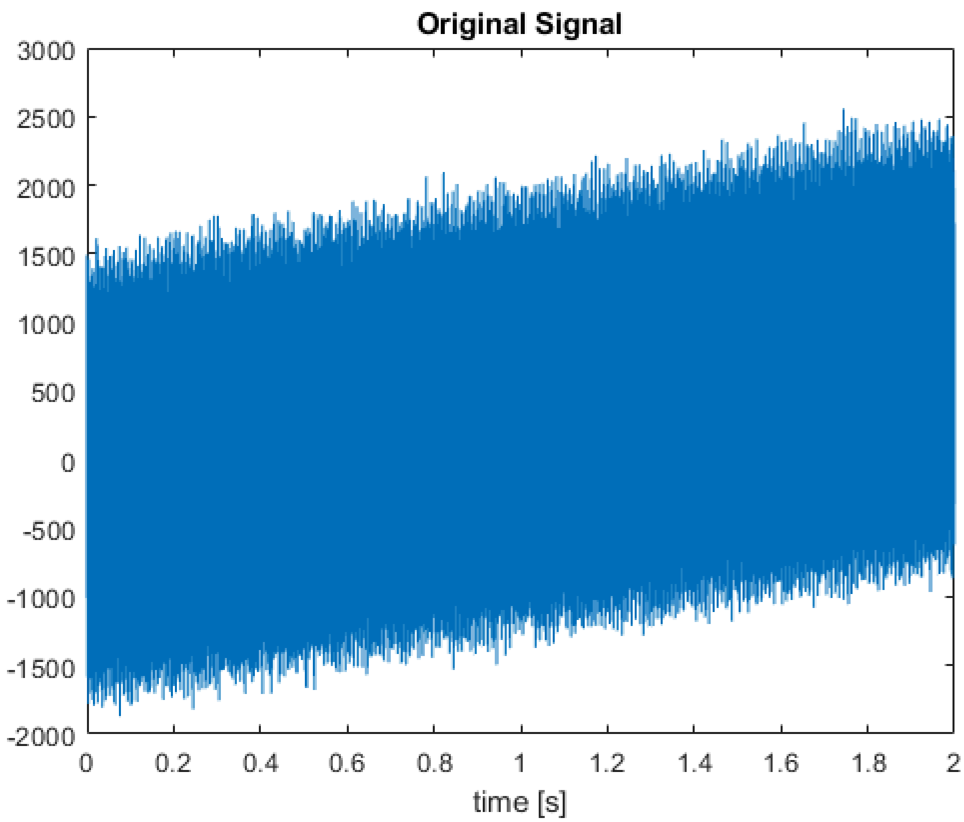

power dB plot
avg. power
Freq. [Hz]
amplitude dB plot
Freq. [Hz]
PSD dB plot
spec density
Freq. [Hz]
Percentage Var. (LSSA)
spec density
Freq. [Hz]
amplitude plot (LSSA)
Freq. [Hz]
PSD plot (LSSA)
spec density
Freq. [Hz]

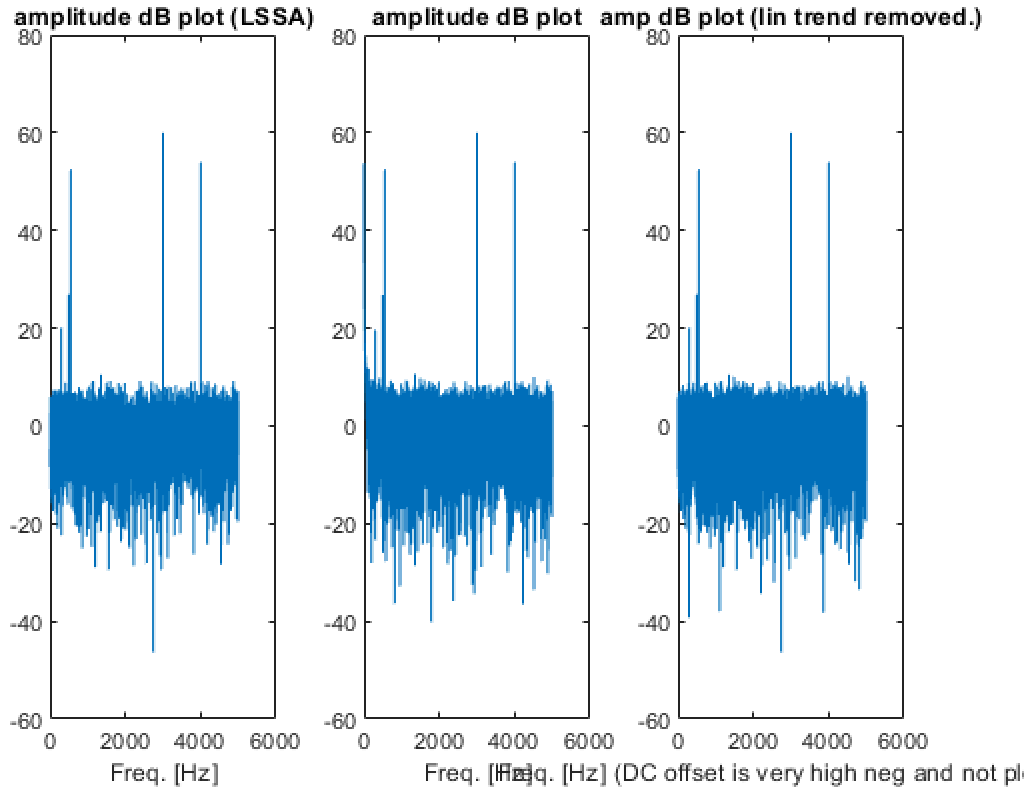


4) Test collection of series 2

```
clearvars; close all
%--- Series has linear trend, no frequency

%--- Test will use LSSA amplitude plot (dB)
%--- Test will use FFT amplitude plot (dB)
%--- Issue with PSD dB plot position
%--- Little difference between PSD and Amplitude in dB anyway

%--- Series parameters and construction
amp = [10 20 420 1000 500];
phase = [0 60 20 15 10]; % degrees
freq_start = [300 500 550 3000 4000];
freq_end = freq_start;
Fs = 10000;
T = 2;
SNR_dB = 25;
linear_trend_frac = repmat(0.5,[1,5]);
[time, signal] = multiple_synthetic_series(amp, freq_start, freq_end, phase,
 T, Fs, SNR_dB, linear_trend_frac);

%--- Plotting Original Signal
figure(1);
plot(time,signal); title('Original Signal'); xlabel('time [s]');
```

```
%--- LSSA amp (dB)
[spectrum, CritVal, CScoeff, res, norm_res, coeff, cov, Omega] = LSSA(time,
 signal, 'linear');
amp = LSSA_helper(CScoeff,'amplitude');
figure(2);
subplot(1,3,1); plot(Omega,20*log10(amp)); title('amplitude dB plot (LSSA)');
 xlabel('Freq. [Hz]');
%--- FFT amp (dB)
[freq, amp] = my_FFT(time, signal);
subplot(1,3,2); plot(freq,20*log10(amp)); title('amplitude dB plot');
 xlabel('Freq. [Hz]');
%--- FFT amp (dB) w/ linear trend removed
[freq, amp] = my_FFT(time, detrend(signal));
subplot(1,3,3); plot(freq(2:end),20*log10(amp(2:end))); title('amp dB plot
 (lin trend removed.)'); xlabel('Freq. [Hz] (DC offset is very high neg and
 not plotted)');
```

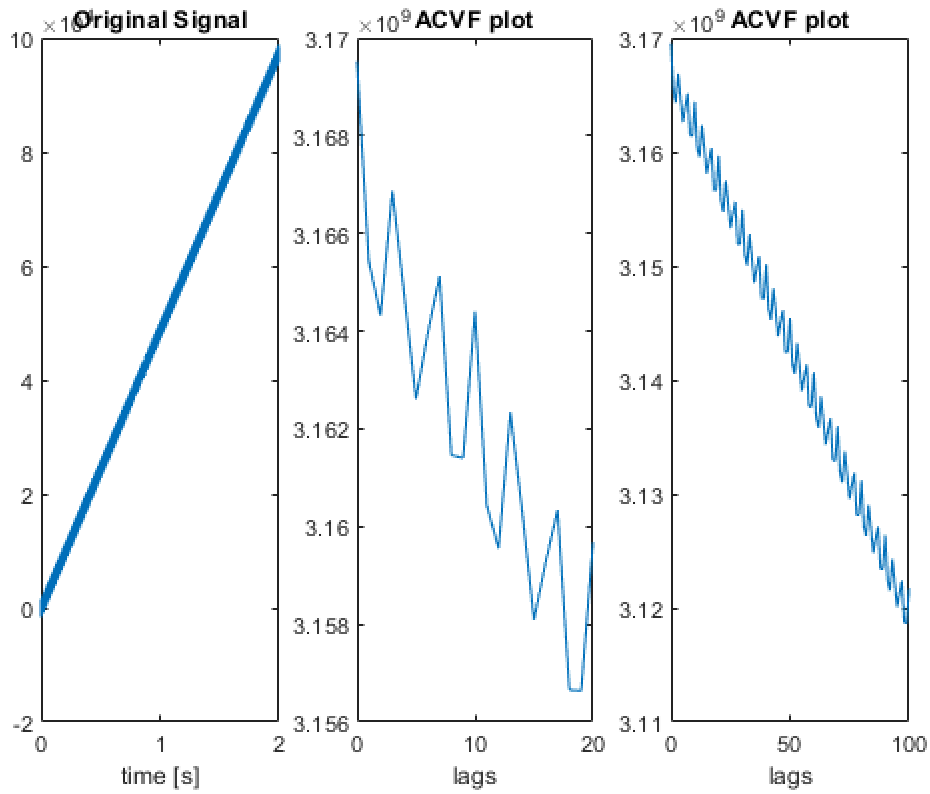


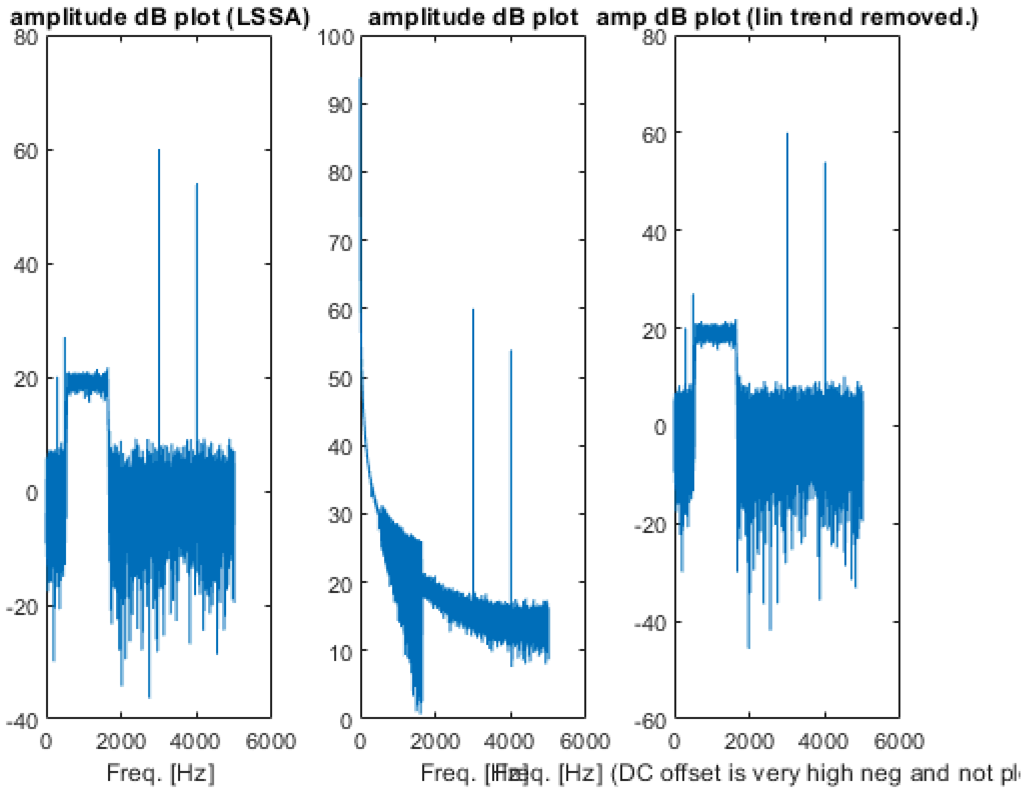


5) Test collection of series 3

```
clearvars; close all
%--- Series has linear trend, one frequency varying
%--- wave of amplitude 420 has varying freq 550 to 1100
%--- Test will use LSSA amplitude plot (dB)
%--- Test will use FFT amplitude plot (dB)
%--- Series parameters and construction

amp = [10 20 420 1000 500];
phase = [0 60 20 15 10]; % degrees
freq_start = [300 500 550 3000 4000];
freq_end = [300 500 1100 3000 4000];
Fs = 10000;
T = 2;
SNR_dB = 25;
linear_trend_frac = repmat(50,[1,5]);
[time, signal] = multiple_synthetic_series(amp, freq_start, freq_end, phase,
 T, Fs, SNR_dB, linear_trend_frac);

%--- Plotting Original Signal
figure(1);
subplot(1,3,1);
plot(time,signal); title('Original Signal'); xlabel('time [s]');
```

```
%--- ACVF
[acf, lags] = autocorr(signal, 'NumLags', 100); acvf = acf*rms(signal)^2;
subplot(1,3,3); plot(lags, acvf); title('ACVF plot'); xlabel('lags');
subplot(1,3,2); plot(lags(1:21),acvf(1:21)); title('ACVF plot');
 xlabel('lags');

%--- LSSA amp (dB)
[spectrum, CritVal, CScoeff, res, norm_res, coeff, cov, Omega] = LSSA(time,
 signal, 'linear');
amp = LSSA_helper(CScoeff,'amplitude');
figure(3);
subplot(1,3,1); plot(Omega,20*log10(amp)); title('amplitude dB plot (LSSA)');
 xlabel('Freq. [Hz]');
%--- FFT amp (dB)
[freq, amp] = my_FFT(time, signal);
subplot(1,3,2); plot(freq,20*log10(amp)); title('amplitude dB plot');
 xlabel('Freq. [Hz]');
%--- FFT amp (dB) w/ linear trend removed
[freq, amp] = my_FFT(time, detrend(signal));
subplot(1,3,3); plot(freq(2:end),20*log10(amp(2:end))); title('amp dB plot
 (lin trend removed.)'); xlabel('Freq. [Hz] (DC offset is very high neg and
 not plotted)');

%--- tracks frequency well but amplitude component for varying frequency
 severely underestimated for all methods
```

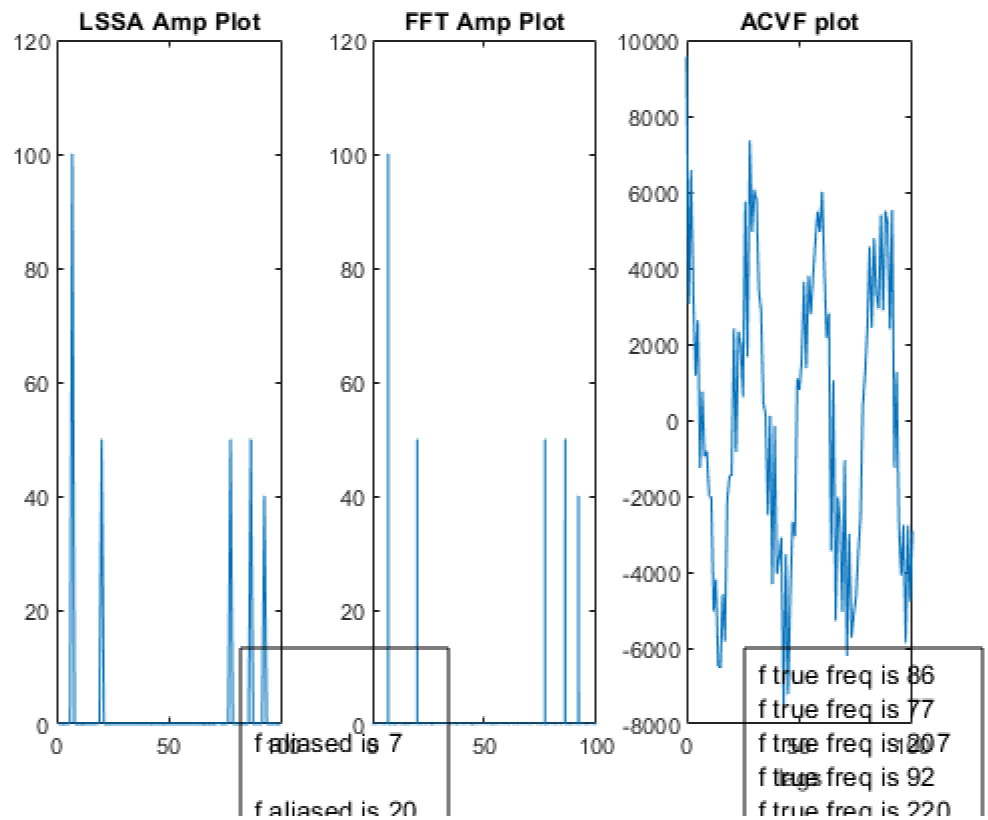




6) Find aliased peak

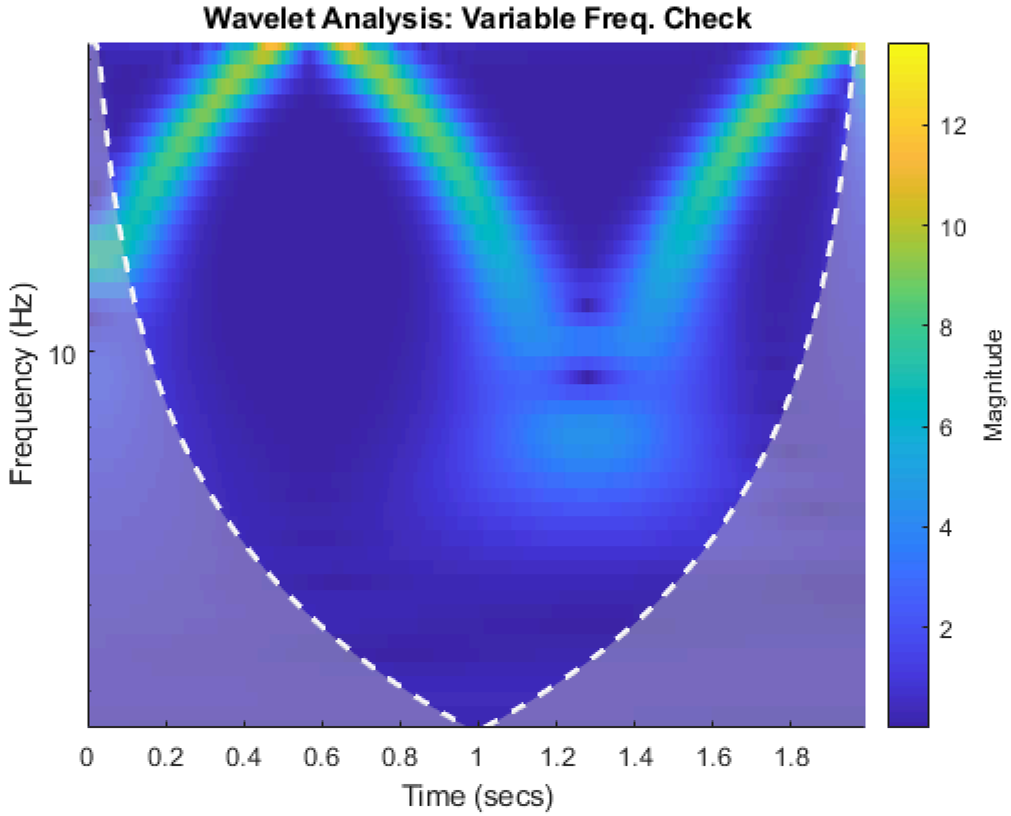
```
close all; clearvars
% --- Function is pasted here
%Function finds aliased peak
%Assuming only looking at positive frequency components
%Must know the sampled frequency and the true frequency (i.e., before is
 aliased)
%
% function f_aliased = aliased_peak(f_true, f_sample)
%
%     f_nyquist = f_sample/2;
%     f_aliased_unfolded = f_true - f_sample;
%     f_aliased = mod(f_aliased_unfolded,f_nyquist);
%
% end

%--- Series will have no linear trend or variable freq. (checked those
 scenarios above already)
%--- Will see aliased peakl at the same alias in all PSD, Power, Amp)
%--- Will construct a random frequencies (10, 600) and will use function check
 which are aliased

amp = [50 50 100 40 50];
phase = [0 30 42 12 30]; % degrees
```

```
freq_start = [86, 77, 207, 92, 220];
freq_end = freq_start;
Fs = 200;
T = 40;
SNR_dB = 40;
linear_trend_frac = repmat(0,[1,5]);
[time, signal] = multiple_synthetic_series(amp, freq_start, freq_end, phase,
 T, Fs, SNR_dB, linear_trend_frac);

%--- Finding Aliased Peak For All freq
for i = 1:length(freq_start)
    f_aliased(i) = aliased_peak(freq_start(i), Fs);
end

%--- LSSA amp
[spectrum, CritVal, CScoeff, res, norm_res, coeff, cov, Omega] = LSSA(time,
 signal, 'linear');
figure(1);
amp = LSSA_helper(CScoeff,'amplitude');
subplot(1,3,1);
plot(Omega, amp); title('LSSA Amp Plot');
subplot(1,3,2);
[freq, amp] = my_FFT(time, signal);
plot(freq, amp); title('FFT Amp Plot');
annotation('textbox', [0.3, 0.1, 0.1, 0.1], 'String', "f aliased is " +
 f_aliased)
annotation('textbox', [0.75, 0.1, 0.1, 0.1], 'String', "f true freq is " +
 freq_start)
subplot(1,3,3);
[acf, lags] = autocorr(signal, 'NumLags', 100); acvf = acf*rms(signal)^2;
plot(lags, acvf); title('ACVF plot'); xlabel('lags');
```



Do CWT for one frequency with variable frequency that will fold over

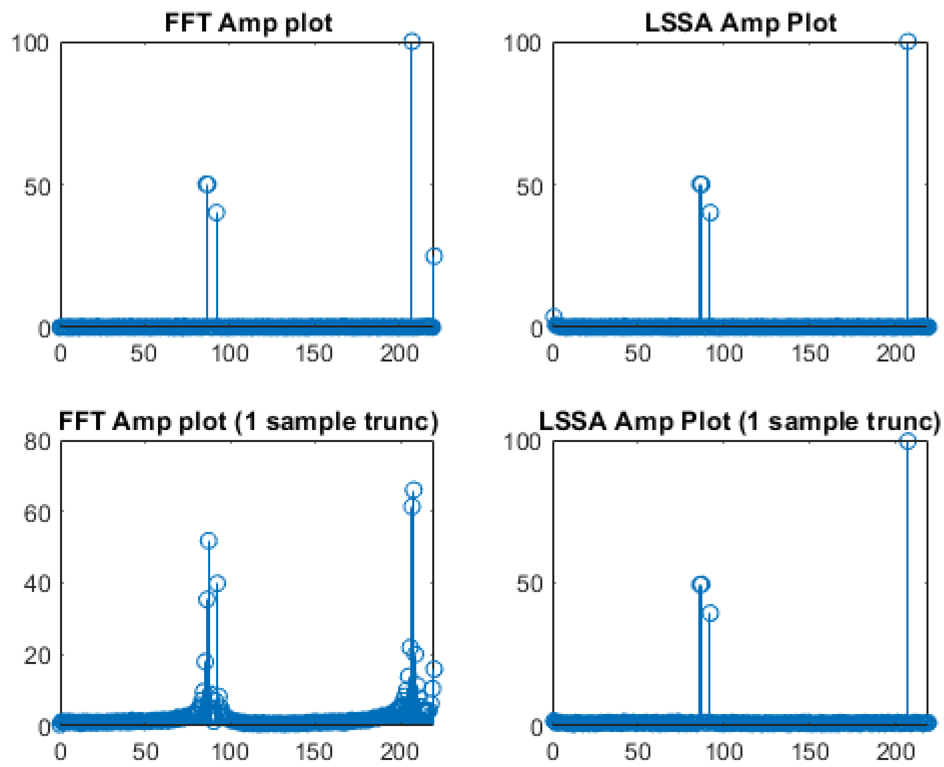
```
close all; clearvars
%--- Setting Parameters for synthetic_series
Fs = 100; %sample freq
T = 2;      %duration of signal
freq_start = 10; freq_end = 80;
phase = 0; amp = 10; SNR_dB = 100; linear_trend_frac = 0;

%--- Call user function
[time, signal] = synthetic_series(amp, freq_start, freq_end, phase, T, Fs,
 SNR_dB,linear_trend_frac); %evaluated function
figure(2); cwt(signal,Fs); title('Wavelet Analysis: Variable Freq. Check');
```



Experiment with variable length series

```
%--- Frequency are 86 and 87
%--- series must be of order 1cps (based on rule from book)
%--- then sample freq must be at least 440 to avoid aliasing
%--- freq can be resolved
close all; clearvars;
amp = [50 50 100 40 50];
phase = [0 30 42 12 30]; % degrees
freq_start = [86, 87, 207, 92, 220];
freq_end = freq_start;
Fs = 440;
T = 1;
SNR_dB = 40;
linear_trend_frac = repmat(0,[1,5]);
[time, signal] = multiple_synthetic_series(amp, freq_start, freq_end, phase,
 T, Fs, SNR_dB, linear_trend_frac);
subplot(2,2,1);
[freq, amp] = my_FFT(time, signal);
stem(freq, amp); title('FFT Amp plot'); subplot(1,2,2);
[spectrum, CritVal, CScoeff, res, norm_res, coeff, cov, Omega] = LSSA(time,
 signal);
amp = LSSA_helper(CScoeff,'amplitude');
subplot(2,2,2);
stem(Omega, amp); title('LSSA Amp Plot');
```

```
[freq, amp] = my_FFT(time(1:end-1), signal(1:end-1));
subplot(2,2,3); stem(freq, amp); title('FFT Amp plot (1 sample trunc)');
subplot(2,2,4); [spectrum, CritVal, CScoeff, res, norm_res, coeff, cov, Omega]
 = LSSA(time(1:end-1), signal(1:end-1));
amp = LSSA_helper(CScoeff,'amplitude'); stem(Omega, amp); title('LSSA Amp Plot
 (1 sample trunc)');
```

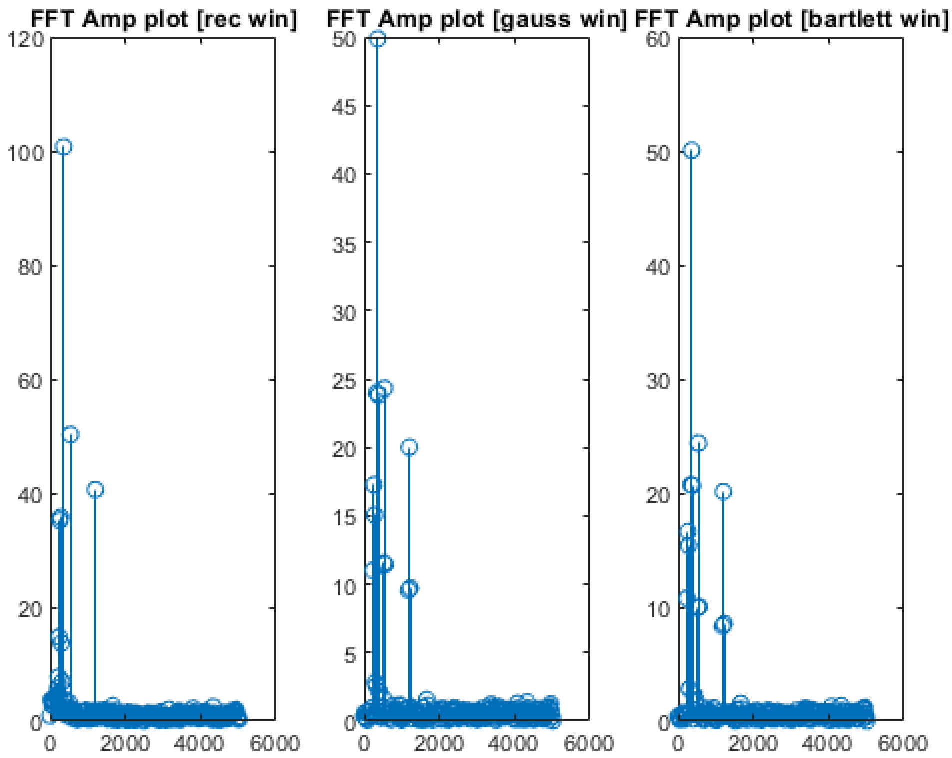


Experiment with windowowing

expecting ripples short length series for rectangular no aliasing will be present will see effect of smoothing 2 frequencies close together

```
close all; clearvars;
amp = [50 50 100 40 50];
phase = [0 30 42 12 30]; % degrees
freq_start = [258, 270, 360, 1200, 540];
freq_end = freq_start;
Fs = 10000;
T = 0.05;
SNR_dB = 25;
linear_trend_frac = repmat(0,[1,5]);
[time, signal] = multiple_synthetic_series(amp, freq_start, freq_end, phase,
 T, Fs, SNR_dB, linear_trend_frac);
[freq, amp] = my_FFT(time, signal);
subplot(1,3,1);
stem(freq, amp); title('FFT Amp plot [rec win]');
```

```
subplot(1,3,2);
gauss_window = gausswin(length(signal));
[freq, amp] = my_FFT(time, prod([signal; gauss_window']));
stem(freq, amp); title('FFT Amp plot [gauss win]');
subplot(1,3,3);
triangle_window = triang(length(signal));
[freq, amp] = my_FFT(time, prod([signal; triangle_window']));
stem(freq, amp); title('FFT Amp plot [bartlett win]');
```



Published with MATLAB® R2022a