Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

Course: SEG3155
Semester: Winter 2012

Professor: Jiying Zhao
Room: STE 5019
Phone: (613)562-5800 x 6667
Email: jyzhao@site.uottawa.ca

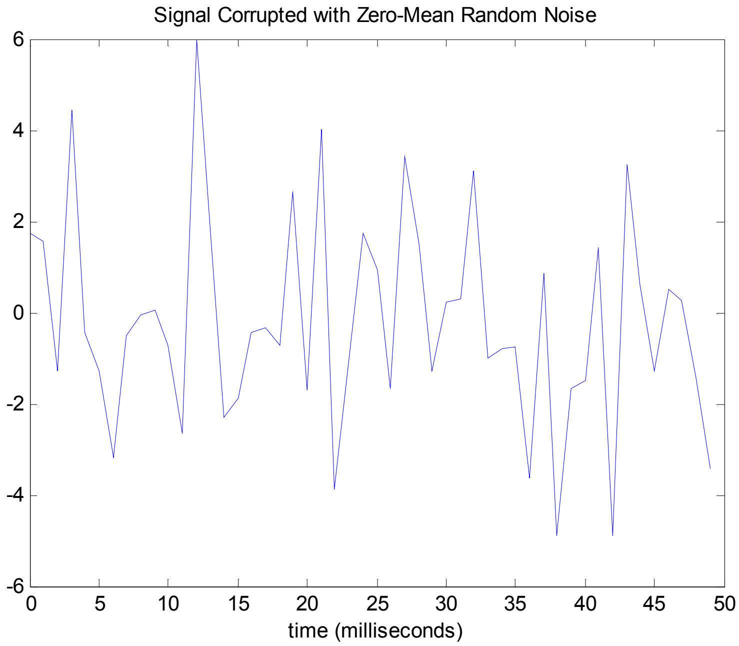
Lab 3
Fourier Transform

Week of February 6, 2012
Posted: February 6, 2012

Objectives: Get familiar with Fourier series and Fourier transform.

1. [Fourier series] Display the frequency spectrum of the following signals:
   a. $s(t) = 1 + (4/\pi)[\sin(2\pi ft)+(1/3)\sin(2\pi(3f)t)]$
   b. $s(t) = (4/\pi)[\sin(2\pi ft)+(1/3)\sin(2\pi(3f)t)+(1/5)\sin(2\pi(5f)t)]$
   c. $s(t) = (4/\pi)[\sin(2\pi ft)+(1/3)\sin(2\pi(3f)t)+(1/5)\sin(2\pi(5f)t)+(1/7)\sin(2\pi(7f)t)]$
2. [Fourier transform] A common use of Fourier transforms is to find the frequency components of a signal buried in a noisy time domain signal. Consider data sampled at 1000 Hz. Form a signal containing a 50 Hz sinusoid of amplitude 0.7 and 120 Hz sinusoid of amplitude 1 and corrupt it with some zero-mean random noise:

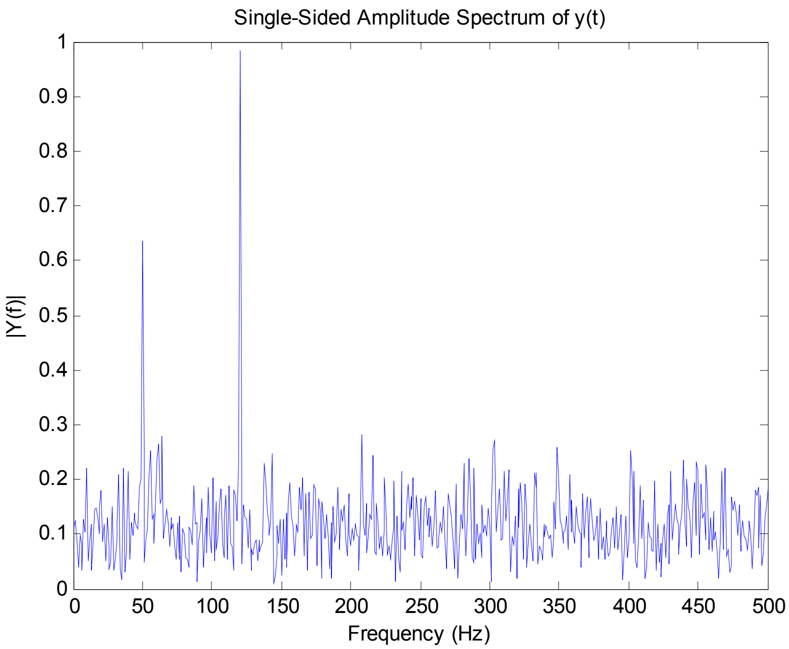
```
% from Matlab help file
Fs = 1000;                    % Sampling frequency
T = 1/Fs;                     % Sample time
L = 1000;                     % Length of signal
t = (0:L-1)*T;                % Time vector
% Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
x = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
y = x + 2*randn(size(t));     % Sinusoids plus noise
plot(Fs*t(1:50),y(1:50))
title('Signal Corrupted with Zero-Mean Random Noise')
xlabel('time (milliseconds)')
```



It is difficult to identify the frequency components by looking at the original signal. Converting to the frequency domain, the discrete Fourier transform of the noisy signal y is found by taking the fast Fourier transform (FFT):

```
% from Matlab help file
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
plot(f,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
```



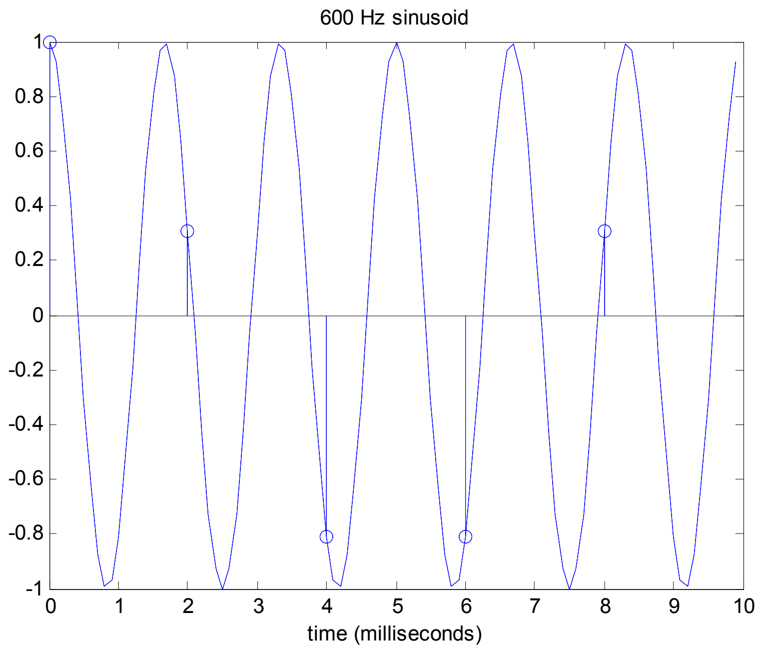
Circularly shift the 1024 (NFFT) sample points 100 samples to the right. Then draw the single-sided amplitude spectrum. Compare the resulting spectrum with the spectrum above.

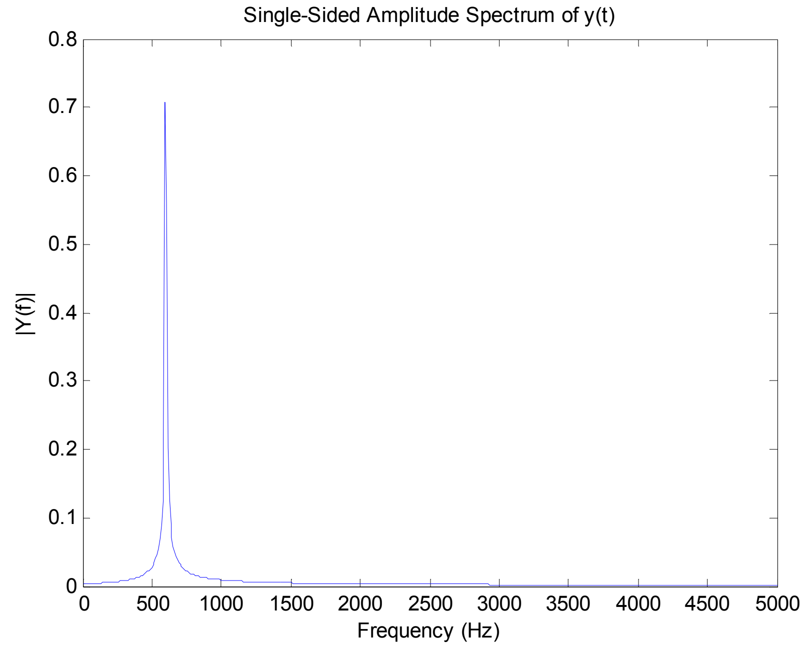
3. [Aliasing] A 600 Hz sinusoid is sampled at a sampling rate of 10000Hz. (fs=10000 sample/second) resulting in the sequence of samples x[n]=cos(2.4*pi*n). The following Matlab scripts display the sinusoid and its single-sided amplitude spectrum.

```
% see page 75 of signal processing first
% suppose
Fs = 10000;                    % Sampling frequency
T = 1/Fs;                      % Sample time
L = 1024;                      % Length of signal
t = (0:L-1)*T;                 % Time vector
% Sum of a 50 Hz sinusoid and a 120 Hz sinusoid
x = cos(2*pi*600*t);
figure(1);
plot(Fs/10*t(1:1:100),x(1:100));
hold on;
stem(Fs/10*t(1:20:100),x(1:20:100));
title('600 Hz sinusoid');
xlabel('time (milliseconds)');

figure(2);
y=x;
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
plot(f,2*abs(Y(1:NFFT/2+1)));
title('Single-Sided Amplitude Spectrum of y(t)');
xlabel('Frequency (Hz)');
ylabel('|Y(f)|');
```





Modify the Matlab scripts to sample the 600 Hz sinusoid with a sample rate of 500 Hz, then draw the corresponding single-sided amplitude spectrum.