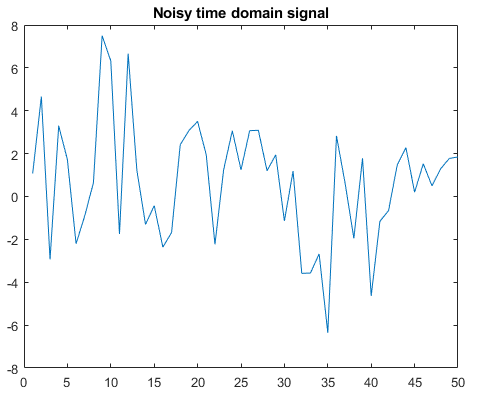
**FFT for Spectral Analysis**

Many times a transformation is performed to provide a better or clearer understanding of a phenomena. The time representation of a sine wave may be difficult to interpret. By using a Fourier series representation, the original time signal can be easily transformed and much better understood. Transformations are also performed to respresent the same data with significantly less information. Notice that the original time signal was defined by many discrete time points (ie, 1024, 2048, 4096 …) whereas the equivalent Fourier representation only requires 4 amplitudes and 4 frequencies.

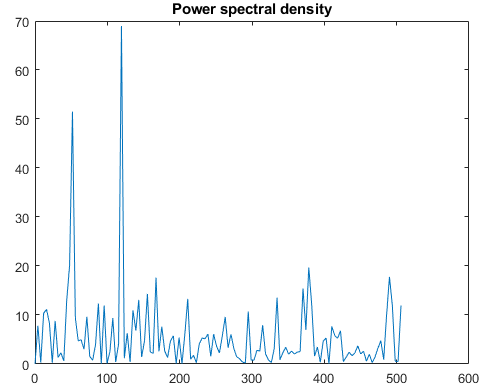
This example shows the use of the FFT function for spectral analysis. A common use of FFT's is to find the frequency components of a signal buried in a noisy time domain signal.

First create some data. Consider data sampled at 1000 Hz. Start by forming a time axis for our data, running from t=0 until t=.25 in steps of 1 millisecond. Then form a signal, x, containing sine waves at 50 Hz and 120 Hz.

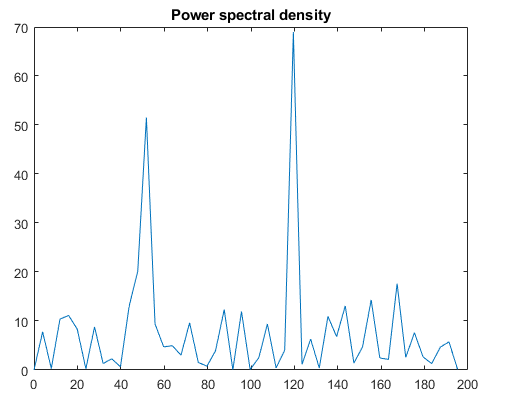


Clearly, it is difficult to identify the frequency components from looking at this signal; that's why spectral analysis is so popular.

Finding the discrete Fourier transform of the noisy signal y is easy; just take the fast-Fourier transform (FFT).



Zoom in and plot only up to 200 Hz. Notice the peaks at 50 Hz and 120 Hz. These are the frequencies of the original signal.



t = 0:.001:.25;

x = sin(2\*pi\*50\*t) + sin(2\*pi\*120\*t);

y = x + 2\*randn(size(t));

plot(y(1:50))

title('Noisy time domain signal');

%{

Y = fft(y,251);

Pyy = Y.\*conj(Y)/251;

f = 1000/251\*(0:127);

plot(f,Pyy(1:128))

title('Power spectral density')

xlabel('Frequency (Hz)');

plot(f(1:50),Pyy(1:50))

title('Power spectral density')

xlabel('Frequency (Hz)')

%}