Fourier Analysis

Example

Given a noisy signal $y(t) = 80 \sin(2\pi f t) + 50\, n(t)$ where $f = 100$ Hz;

$n(t)$ is a normally distributed noise. Use MATLAB function *randn* to generate the noise signal. Use MATLAB to obtain the spectrum amplitudes of signal $y(t)$.

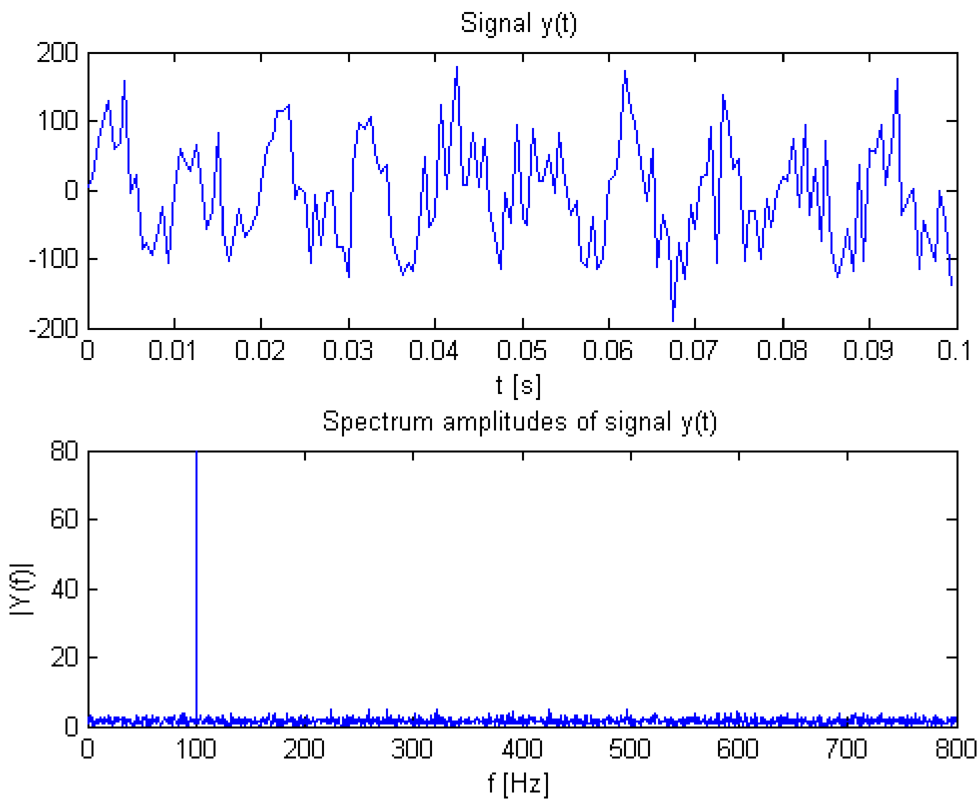
Solution

This example shows the use of the FFT function for spectral analysis. It is difficult to identify the frequency components from this signal; that is why spectral analysis is so popular. Finding the discrete Fourier transform of the noisy signal is easy; just take the fast-Fourier transform (FFT). The MATLAB program that can be used to plot the noisy signal $y(t)$ and obtain the spectrum amplitudes of signal $y(t)$ is

MATLAB Script

```
clear; clc;
f=100;    % basic signal frequency
fs=f*2^4 % sampling frequency (FFT requires that fs=f*2^number)
dt=1/fs;
n=fs*2    % total samples (FFT requires that n=2^number, e.g. n=fs*4)
t=[0:n-1]*dt; % vector of times
w=2*pi*f;
a=80*sin(w*t); % it generates the sine portion of signal
b=50*randn(size(t)); % it generates a normally distributed noise
y=a+b; % it generates the noisy signal
subplot(2,1,1);
% plots signal y(t) by means of the first 160 samples
plot(t(1:160),y(1:160));
title('Signal y(t)');
xlabel('t [s]');
Y=fft(y);      % performs FFT for samples of signal y(t)
fx=linspace(0,fs/2,n/2); % fx from 0 to fs/2, total of n/2 values
% It forms a frequency axis for the first half of points
% and uses it to plot the result. The remaining points are
% symmetric.
subplot(2,1,2);
plot(fx,2*abs(Y(1:n/2))/n);
title('Spectrum amplitudes of signal y(t)');
xlabel('f [Hz]'); ylabel('|Y(f)|');
```

The plots obtained from MATLAB are



Notice the peak at 100 Hz. This is the frequency of the original signal.