Plotting Signals in Matlab

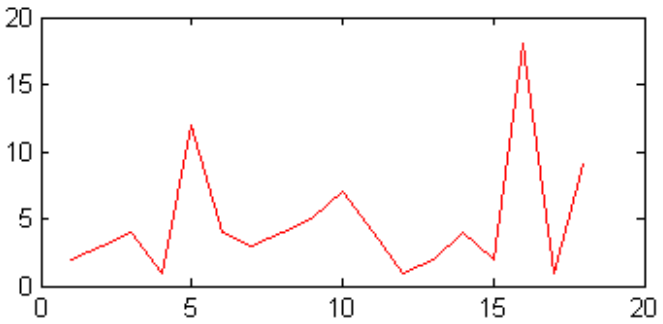
One of the most powerful tools available in matlab is the `plot` function, which helps engineers visualise and analyse signals and system behaviour. This document provides examples on how to use the plot command in different ways.

Contents

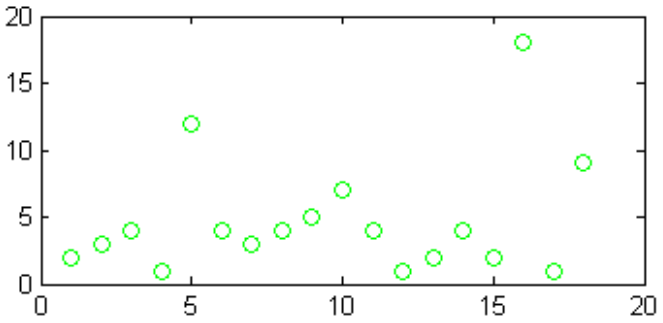
- Plot a signal using different colors and markers
- Label x and y axes, and add a title
- Plot a number of signals on the one plot
- Create a new figure for different plots
- Change the x-axis scale
- Create a file (jpg, gif, emf, bmp) for use in documentation
- Create multiple plots on the one figure using subplot

Plot a signal using different colors and markers

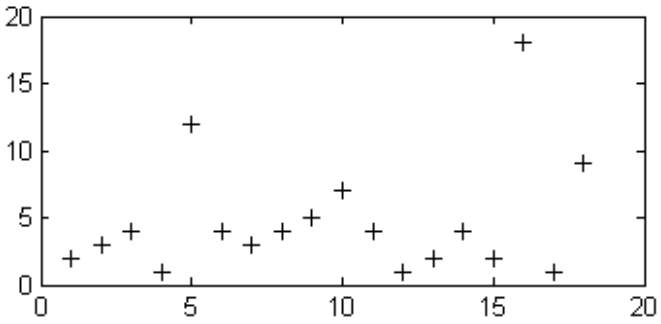
```
x = [2 3 4 1 12 4 3 4  5 7 4 1 2 4 2 18 1 9 ];
plot(x, 'r') % plot a signal in red - chnge r for g (green), y (yellow), k (black)
```




```
plot(x,'go') % use circles as the marker - color green
```

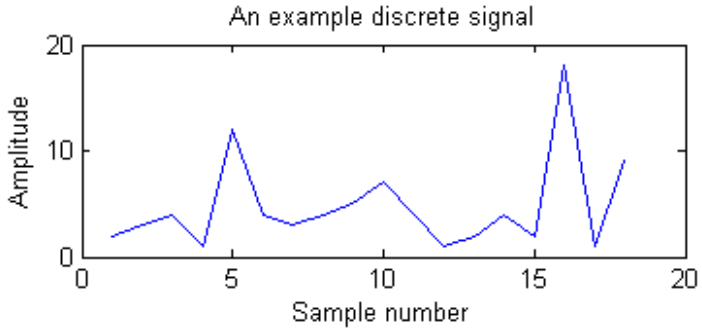



```
plot(x,'k+')  % use a + as the marker - color black
```



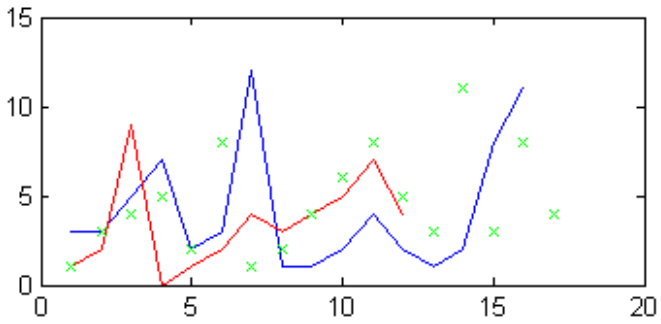
Label x and y axes, and add a title

```
plot(x)
ylabel('Amplitude');
xlabel('Sample number');
title('An example discrete signal')
```



Plot a number of signals on the one plot

```
x = [3 3 5 7 2 3 12 1 1 2 4 2 1 2 8 11 ]; %signal 1
y = [ 1 2 9 0 1 2 4 3 4  5 7 4]; % signal 2
w = [ 1 3 4 5 2 8 1 2 4 6 8 5 3 11 3 8 4]; % signal 3
plot(x)
hold on
plot(y,'r')
plot(w,'gx')
hold off
```

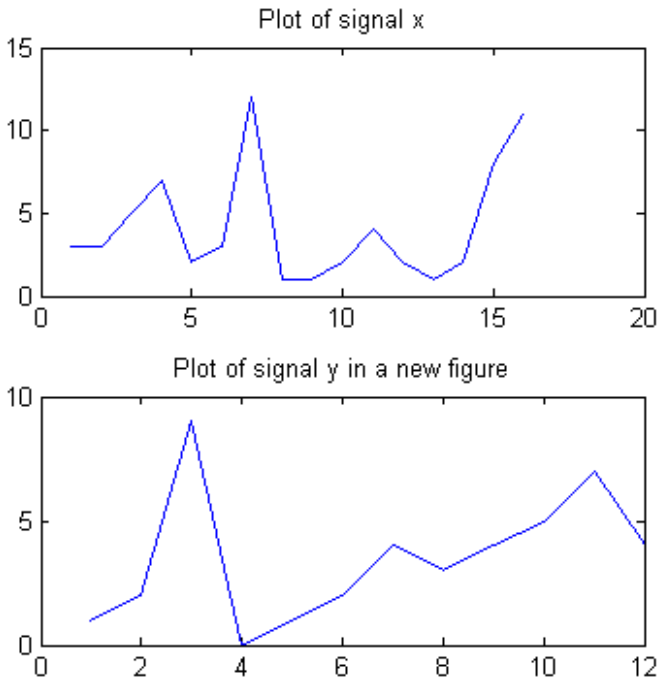


Create a new figure for different plots

The figure command creates an entirely new figure

```
x = [3 3 5 7 2 3 12 1 1 2 4 2 1 2 8 11 ]; %signal 1
y = [ 1 2 9 0 1 2 4 3 4  5 7 4]; % signal 2
plot(x)
title('Plot of signal x')
figure
```

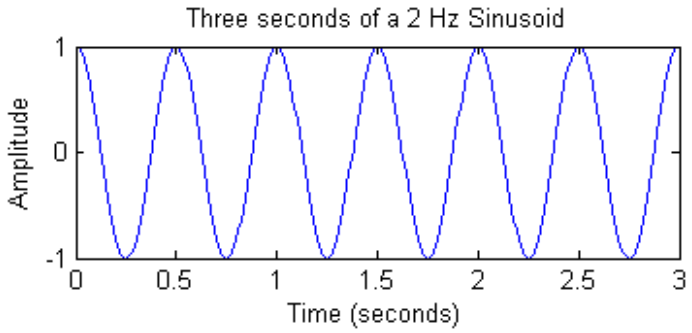
```
plot(y)
title('Plot of signal y in a new figure')
```



Change the x-axis scale

The default is to plot the sample number of the x-axis. However this example shows how a sinusoid can be plotted against time rather than sample number.

```
T = 0.01; %samling period
n = 0 : 300; % sample number
frequency = 2; % frequency of a sinusoid in hertz
x = cos(2*pi*frequency*n*T); % create three seconds of a cosine sinusoid

sample_times = n*T;
plot(sample_times, x);
xlabel('Time (seconds)')
ylabel('Amplitude')
title('Three seconds of a 2 Hz Sinusoid');
```

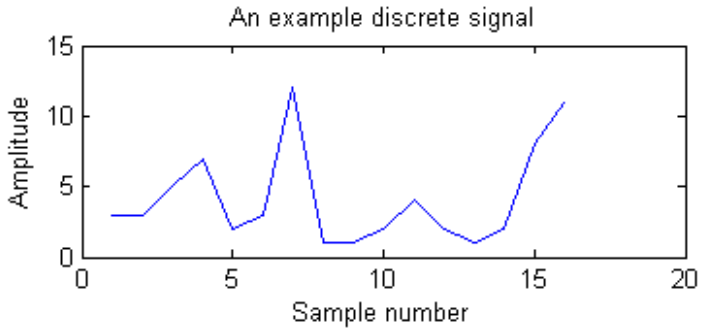


Create a file (jpg, gif, emf, bmp) for use in documentation

When writing reports it can be vey useful to create plots in matlab and then export the plot to a standard image file.

```
x = [3 3 5 7 2 3 12 1 1 2 4 2 1 2 8 11 ]; %signal 1
plot(x)
ylabel('Amplitude');
xlabel('Sample number');
```

```
title('An example discrete signal')

%create three different images - Note: - the gcf varaible is a handle for
%the current figure. gcf means Get Current Figure handle.
saveas(gcf, 'example_signal.emf', 'emf'); % good format for word docs
saveas(gcf, 'example_signal.jpg', 'jpg'); % good format for web
saveas(gcf, 'example_signal.eps', 'eps'); % good format for Latex
```



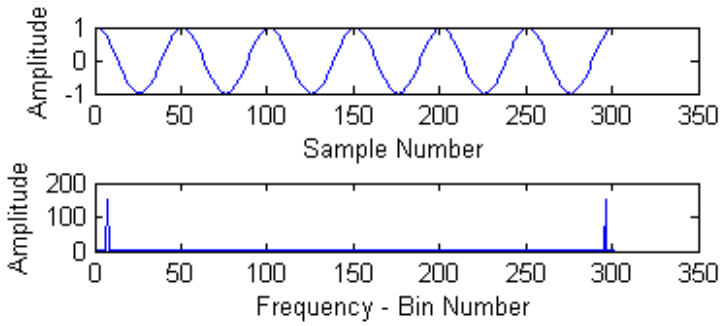
Create multiple plots on the one figure using subplot

The subplot command is very useful and allows multiple plots appear on the one figure.

The following lines of code create a signal x and its corresponding magnitude frequency content Xmags. x and Xmags are then plotted.

```
T = 0.01; %samling period
n = 0 : 300; % sample number
frequency = 2; % frequency of a sinusoid in hertz
x = cos(2*pi*frequency*n*T); % create three seconds of a cosine sinusoid
Xmags = abs(fft(x)); % get the magnitudes of the Discrete Fourier Transform
```

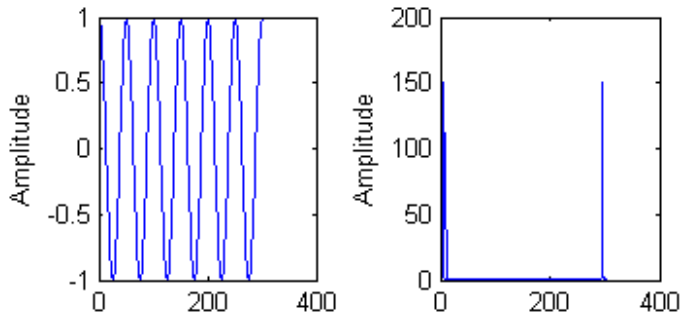
```
subplot(2, 1, 1)
plot(x);
ylabel('Amplitude')
xlabel('Sample Number');
subplot(2, 1, 2);
plot(Xmags);
ylabel('Amplitude')
xlabel('Frequency - Bin Number');
```



Second example

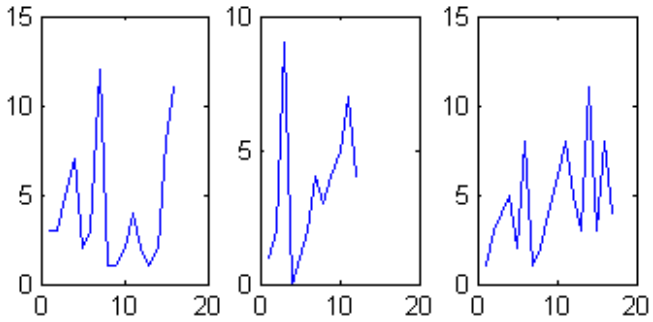
```
subplot(1, 2, 1)
plot(x);
ylabel('Amplitude')
xlabel('Sample Number');
subplot(1, 2, 2);
plot(Xmags);
ylabel('Amplitude')
```

```
xlabel('Frequency - Bin Number');
```



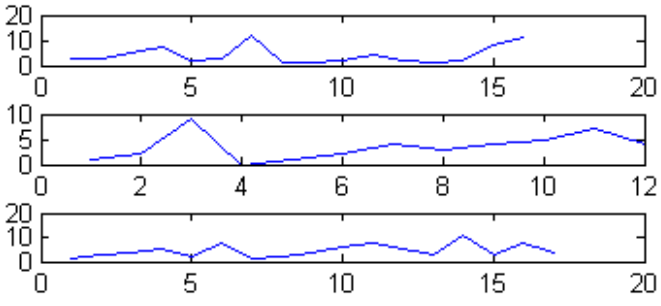
Third example

```
x = [3 3 5 7 2 3 12 1 1 2 4 2 1 2 8 11 ]; %signal 1
y = [ 1 2 9 0 1 2 4 3 4  5 7 4]; % signal 2
w = [ 1 3 4 5 2 8 1 2 4 6 8 5 3 11 3 8 4]; % signal 3

subplot(1, 3, 1)
plot(x)
subplot(1, 3, 2)
plot(y)
subplot(1, 3, 3)
plot(w)
```

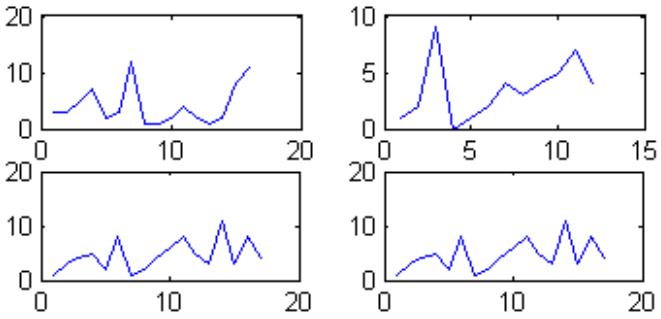


Fourth example

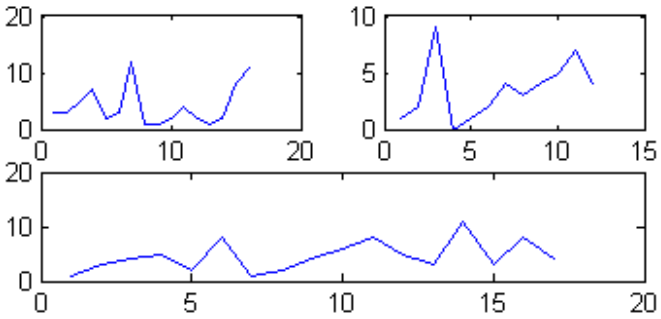
```
x = [3 3 5 7 2 3 12 1 1 2 4 2 1 2 8 11 ]; %signal 1
y = [ 1 2 9 0 1 2 4 3 4  5 7 4]; % signal 2
w = [ 1 3 4 5 2 8 1 2 4 6 8 5 3 11 3 8 4]; % signal 3

subplot(3, 1, 1)
plot(x)
subplot(3, 1, 2)
plot(y)
subplot(3, 1, 3)
plot(w)
```



```
 % Fifth Example
x = [3 3 5 7 2 3 12 1 1 2 4 2 1 2 8 11 ]; %signal 1
y = [ 1 2 9 0 1 2 4 3 4  5 7 4]; % signal 2
w = [ 1 3 4 5 2 8 1 2 4 6 8 5 3 11 3 8 4]; % signal 3
v = [ 4 8 4 3 2  5 4 3 2 10 9 2 3 5 6];
subplot(2, 2, 1)
plot(x)
subplot(2, 2, 2)
plot(y)
subplot(2, 2, 3)
plot(w)
subplot(2, 2, 4)
plot(w)
```




```
 % Sixth Example
x = [3 3 5 7 2 3 12 1 1 2 4 2 1 2 8 11 ]; %signal 1
y = [ 1 2 9 0 1 2 4 3 4  5 7 4]; % signal 2
w = [ 1 3 4 5 2 8 1 2 4 6 8 5 3 11 3 8 4]; % signal 3

subplot(2, 2, 1) % plot to top left
plot(x)
subplot(2, 2, 2) % plot to top right
plot(y)
subplot(2, 1, 2) % plot to bottom row - note subplot is now showing 2 rows and 1 column
plot(w)
```



Published with MATLAB® 7.11