MATLAB Marina: Plotting 2D Examples

Student Learning Objectives

After completing this module, one should:

1. Be able to generate and annotate 2D plots using MATLAB to present data.

Terms

figure, parametric plot

MATLAB Functions, Keywords, and Operators

figure, plot, subplot, xlabel, ylabel, title, legend, grid, axis, close, stem, semilogx

Parametric Plots

Parametric plots are plots where the x and y coordinates are functions of an independent variable or variables, often time. Kinematic equations are often expressed as parametric equations. For example, the x and y components of a projectile position can be represented as functions of time:

$$s_x = s_{x0} + v_x t + \tfrac{1}{2} a_x t^2$$

$$s_y = s_{y0} + v_y t + \tfrac{1}{2} a_y t^2$$

One could plot the x and y components of position versus time and one could also plot the projectile position with s_x along the x axis and s_y along the y axis. This second case is a paramtric plot.

The MATLAB code of Figure 2 computes the trajectory of a projectile for a specified initial velocity and launch angle. The parametric plot of the projectile motion is shown in Figure 1.

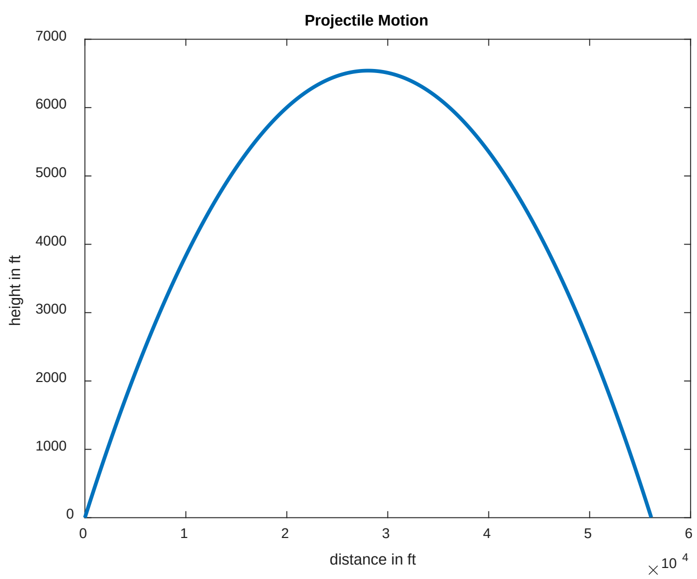

Figure 1. Two-Dimension Projectile Motion (Parametric Plot)

```
clear; clc; close all;
% 2D projectile motion
sx0 = 0.0;
sy0 = 0.0;
v0 = 1550.20;  % projectile velocity in ft/s
gx = 0.0;
gy = -32.81;   % gravity in ft/s^2
theta = 25*pi/180;  % projectile angle in rad

% x and y components of velocity
vx0 = v0*cos(theta);
vy0 = v0*sin(theta);
% range of the projectile and time to max range
range = -v0^2*sin(2*theta)/gy;
tmax = range/vx0;

% position and velocity of the projectile
t = 0:0.01:tmax;
vx = vx0 + gx*t;
vy = vy0 + gy*t;
sx = sx0 + vx0*t + 0.5*gx*t.*t;
sy = sy0 + vy0*t + 0.5*gy*t.*t;
```

Figure 2. Two-Dimensional Projectile Motion

Presenting Data using Plots

The projectile motion results can be presented in a variety of ways depending on what is being investigated. Figure 1, shown previously, is a parametric plot of the y position versus x position. The MATLAB code to create this plot is given in Figure 3.

```
% plot the projectile motion with solid blue line
% and with a line width of 2
figure(1)
plot(sx,sy,'Linewidth',2);
xlabel("distance in ft");
ylabel("height in ft");
title("Projectile Motion");
```

Figure 3. MATLAB Code to Plot sy versus sx (Parametric Plot)

Related items can be plotted on the same axes. Figure 4a shows code that will plot the horizontal and vertical position with respect to time on the same set of axes. Figure 4b shows the resulting plot. Note that the plot function takes multiple pairs of arguments for plotting multiple things on the same axes and has optional arguments for specifying the point style, line style and color for the plot. Legends are generally used when one has multiple plots on the same axes. The `legend` function has an optional argument to specify the location of the legend.

```
% plot the horizontal and vertical position on same axes
% sx in blue and sy in green
% using a legend instead of ylabel
figure(2)
plot(t,sx,'b-',t,sy,'g-');
xlabel("t (s)");
legend("s_x(t)","s_y(t)",'location','east');
title("Horizontal and Vertical Position");
```

Figure 4a. MATLAB Code to Plot Horizontal and Vertical Position

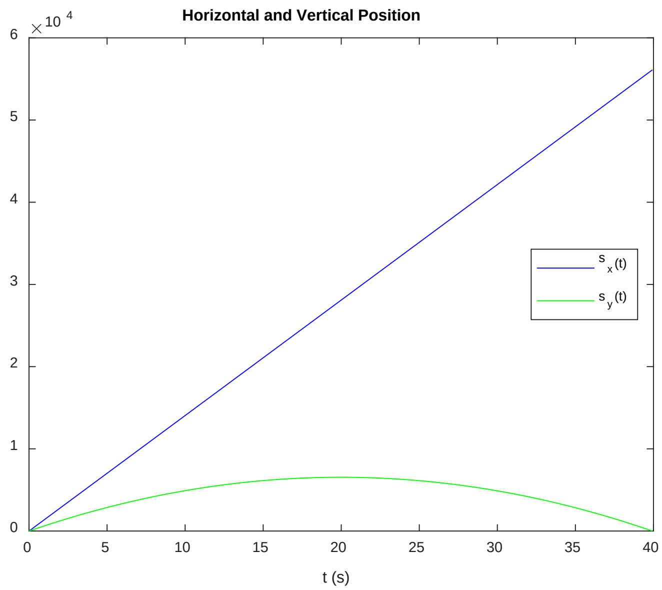

Figure 4b. Horizontal and Vertical Position for Projectile Motion

Related items can also be plotted on separate axes in the same figure. This may be preferable when the relative magnitudes or units of the items differ. Figure 5a shows code that will plot the horizontal and vertical position with respect to time on one set of axes and the horizontal and vertical velocity on a separate axis in the same window. The plots are tiled vertically. For widescreen monitors, tiling plots vertically generally looks better than tiling the plots horizontally.

```
% plot the horizontal and vertical position and velocity on
% separate axes in same figure window tiled vertically
figure(3)
subplot(2,1,1),plot(t,sx,'b-',t,sy,'g-'), grid on;
xlabel("t (s)"), ylabel("(ft)"), legend("s_x","s_y");
subplot(2,1,2),plot(t,vx,'b--',t,vy,'g--'), grid;
axis([min(t),max(t),-1000,2000]);
xlabel("t (s)"), ylabel("(ft/s)"), legend("v_x","v_y");
```

Figure 5a. MATLAB Code to Plot Position and Velocity on Separate Axes

The `subplot` function is used to create separate axes in the same figure window; `subplot(m,n,p)` breaks the figure window up into a m by n grid and specifies the pth axes for the plot. Here the figure is broken up into a 2 by 1 grid. The axes are numbered from left to right along each row starting with the top row, then the second row, etc.

The `axis` function is used to specify the x and y axis limits. It takes a 1 by 4 array of the x and y axis limits. Without this, the y axis limit made the plot of the x component of velocity difficult to see. The `grid on` statement gives the background grid. The `grid off` statement will turn off the background grid.

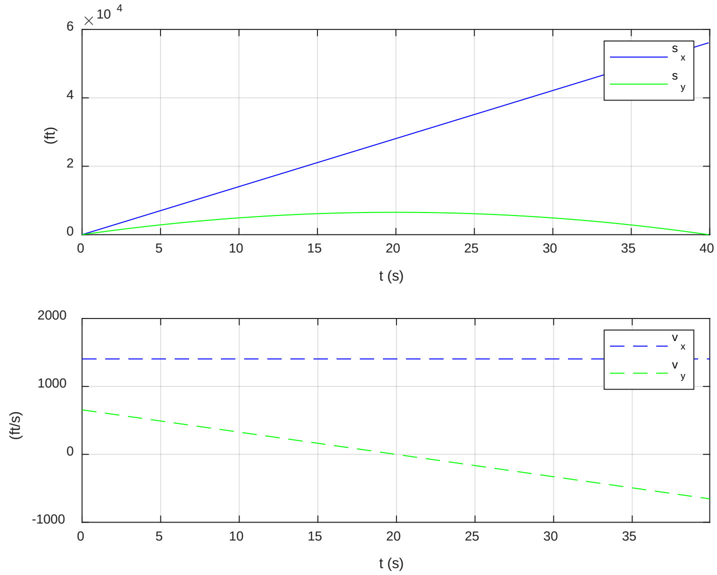

Figure 5b. Horizontal and Vertical Position for Projectile Motion

Log Plots

Some functions are better presented using log scales for the x and y axes. Figure 6b shows the magnitude portion of a system frequency response created using the MATLAB code of Figure 6a. The `semilogx` function plots using a base 10 logarithmic scale for the x axis. There are also plot functions for logarithmic scale for the y axis, `semilogy`, and logarithmic scale for both axes, `loglog`.

```
% magnitude response example
% semilog plot, axis limits specified
clear; clc; close all;
w = (0.0:10.0:4000)*2*pi;
H = 20000./((1j*w + 200*pi).*(1j*w + 800*pi));
HdB = 20*log10(abs(H));
figure(1)
semilogx(w,HdB), grid;
axis([min(w),max(w),min(HdB),max(HdB)+5]);
xlabel("w (rad/s)"), ylabel("|H(jw)| (db)");
```

Figure 6a. MATLAB Code to Plot Magnitude Portion of Frequency Response

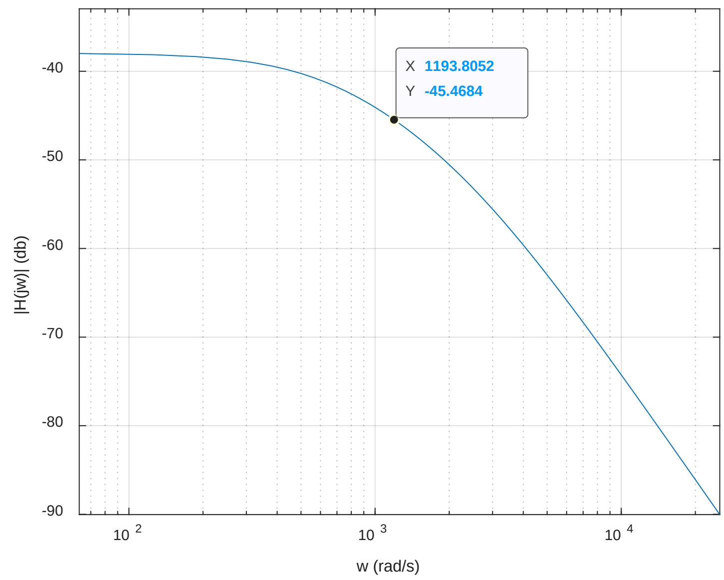

Figure 6b. Magnitude Portion of Frequency Response

The plot of Figure 6b is a log-log plot. The `semilogx` function gives a logarithmic x axis scale and the magnitude is in db, $20\log_{10}(\ )$. The plot of Figure 6b also shows the results of using the data curser. The data curser gives a curser that can be moved along the plot and display the function values at the points.

Stem Plots

Discrete-time signals and functions are displayed using stem plots. Discrete-time signals or functions are sequences of numbers and is only defined at the sample points. It is not appropriate to show an interpolated curve connecting the points.

The MATLAB code of Figure 7a plots a discrete-time sinusoidal signal. The resulting stem plot is shown in Figure 7b.

```
% stem plot example
clear; clc; close all;
n = 0:1:49;
xn = 1.0 + 1.2*cos(0.4*pi*n) + 1.8*cos(0.6*pi*n);
figure(1)
stem(n,xn);
xlabel("n"), ylabel("x[n]");
figure(2)
stem(n,xn,'LineStyle','none','Color','black','Marker','o');
xlabel("n"), ylabel("x[n]");
```

Figure 7a. Stem Plot

The `stem` function supports optional arguments specifying point and line style similar to the `plot` function. To suppress the stem lines, the LineSpec name pair arguments must be used.

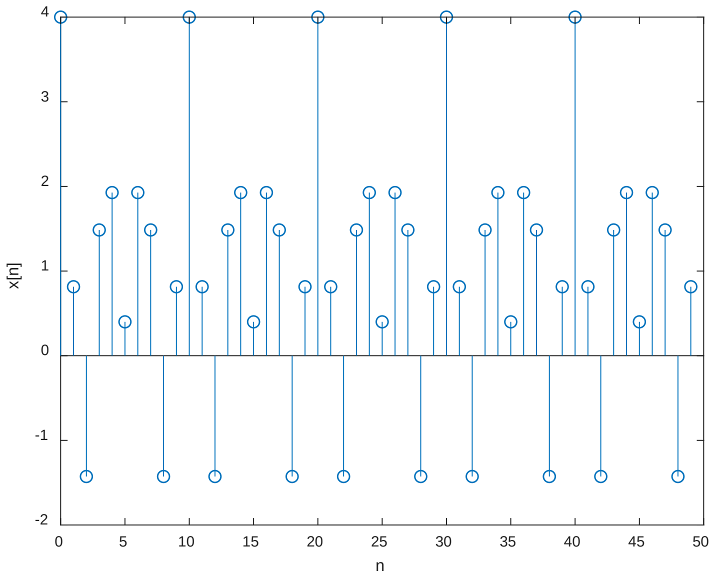

Figure 7b. Discrete-time Sinusoidal Signal

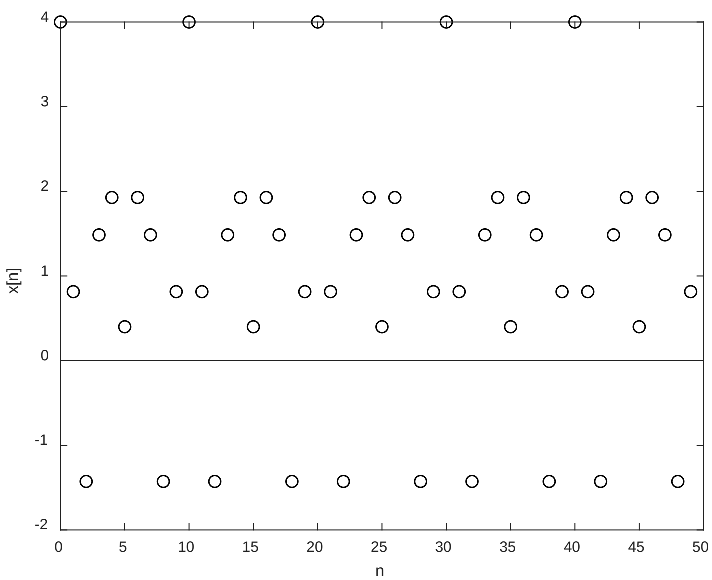

Figure 7c. Discrete-time Sinusoidal Signal (Stem Lines Suppressed)

Scatter Plots

Scatter plots are used to identify relationships between two variables. If one of the variables is the independent (control) variable it is plotted along the horizontal axis. Scatter plots use markers to represent the values; there are no connecting lines/curves between the values. The scatter function creates a scatter plot. The plot function with the correct LineSpec options can be used create scatter plots for single color and single marker size.

The MATLAB code of Figure 8a creates scatter plots of the spot price for LPG [1] using the plot function and the scatter function.

```
%% scatter plot of commodity data
figure(1)
plot(date,pr,'bo');
xlabel("date"), ylabel("Price per Gallon (USD)");
title("Spot Price LPG Mont Belieu");

figure(2)
% plot to see if price related to month of year
sz = 36;     % default marker size
scatter(month(end-48:end), pr(end-48:end), sz, 'o'), axis tight;
xlabel("month"),ylabel("Price per Gallon (USD)");
title("Spot Price LPG Mont Belieu by Month");
```

Figure 8a. MATLAB Code for Scatter Plots

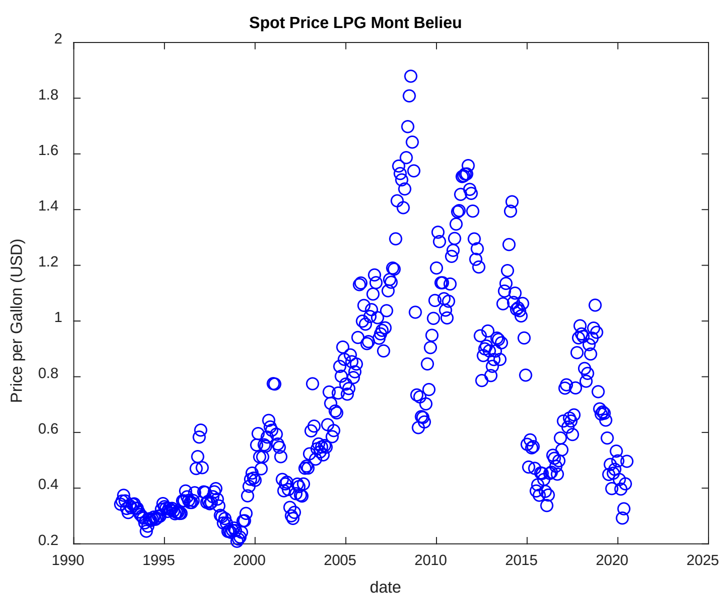

Figure 8b. Scatter Plot of Spot Price LPG Mont Belieu

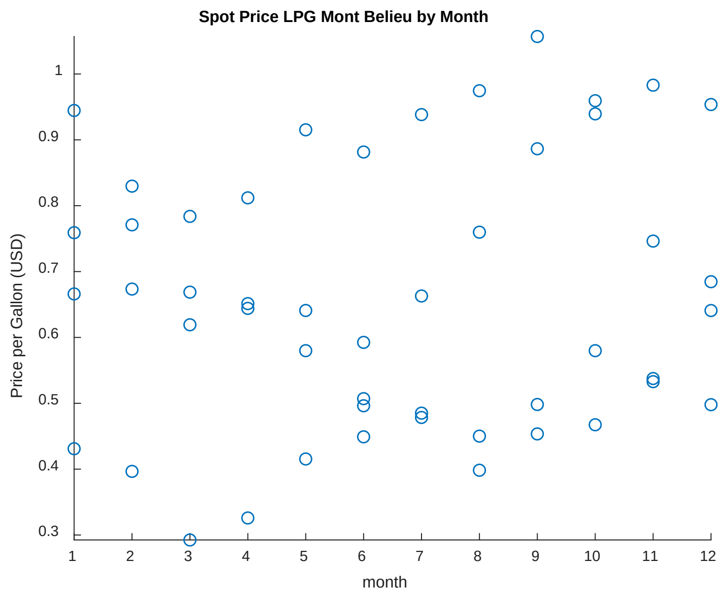

Figure 8c. Scatter Plot of Spot Price LPG Mont Belieu by Month

References

[1] IMF Primary Commodity Prices, Retrieved from https://www.imf.org/en/Research/commodity-prices
Retrieved Friday, July 24, 2020, International Monetary Fund, 2020

Last modified Friday, September 18, 2020