Graphing Transfer Functions:

Graphic presentations of transfer function phases and magnitudes provide insight into the general behavior of the circuit or system for a broad range of inputs.

Three ways to generate a transfer function plot:

- **Enter the transfer function expression in a math program: For these programs you must specify plot range and axis details. The most convenient programs automatically handle complex numbers and have built-in graphics.**
- **Sketch a Bode plot: This can be done by hand and provides a general shape of the transfer function, but does not provide details.**
- **Use a spread sheet program: You must enter the formula for frequency axis values, the magnitude, and phase as a function of ω (or *f*). These programs typically cannot handle complex numbers.**

To graph a transfer function you must know the range of frequencies, the number of evaluation points, and the nature (scale) of the axes. For the initial range use a log scale on the frequency axis (decade scale). Begin the plot from a decade before the smallest magnitude non-zero pole or zero and end the plot a decade after the largest magnitude pole or zero. You can later change the range if you want to examine a particular part of the graph.

Graphing with Matlab:

"Matlab" is short for matrix laboratory. This program is primarily designed to handle matrices and vectors (both complex and real) and provided a wide selection of graphics. It comes with large set of functions to easily perform most tasks for engineering with a single command. It is also easy to create your own functions and scripts (programs).

Example:
Graph the magnitude and phase of the transfer function. Create both a linear scaled plot and a log (base 10) scaled plot.

$$\hat{H}(p) = \frac{\hat{I}_0}{\hat{V}_s} = \frac{\frac{4}{5}p^2}{p^3 + 6p^2 + \frac{14}{5}p + 4}$$

```
% Define range for w
% Find poles and zeros:
ps = roots([1, 6, (14/5), 4]);  %  Vector containing polynomial coefficients
                                %  ps will be a vector containing the roots
zs = roots([(4/5), 0, 0]);      %  Vector containing polynomial coefficients
                                %  zs will be a vector containing the roots

%  Find maximum magnitude pole and zero
fend1 = max(abs(ps))    %  fend1 will be the maximum of the magnitudes of ps.
```

```
fend1 =
    5.6288
fend2 =  max(abs(zs))%  fend2 will be the maximum of the magnitudes of zs.
fend2 =
     0
%  Pick out the maximum value between the two (5.6 in this case) round up to
%  the next decade (which is 10) and increment to next decade (100 in this
%  case).
%  Find minimum magnitude pole and zero
fbeg1 = min(abs(ps))    %  fbeg1 will be the minimum of the magnitudes of ps.
fbeg1 =
    0.8430
fbeg2 =  min(abs(zs))%  fbeg2 will be the minimum of the magnitudes of zs.
fbeg2 =
     0
%  Pick out the minimum non-zero value between the two (.84 in this case)
%  round down round down to the next decade (which .1) and decrement to
%  next decade (.01 in this case).
%  Now create the w-axis vector with 201 equally spaced point on a log
%  (base 10) scale:
w = logspace(-2, 2, 201);  % w  is now a vector of points from 10^-2 to 10^2
% Note that the transfer function is computed for p=j*w, therefore assign:
p=j*w;    %  now p is a vector of imaginary numbers (j=sqrt(-1) by default)
%  Now evaluate the transfer function at all points defined by p:
h = (4/5)*p.^2 ./ (p.^3 + 6*p.^2 + (14/5)*p + 4);

%  Plot magnitude on semilog axis in decibels:
```