# ECE 3300
# MATLAB Assignment 5

As defined in the course notes, a Bode magnitude plot is a straight-line approximation of $|H(j\omega)|$ in decibles versus $\omega$ on a logarithmic scale. Similarly, a Bode phase plot is a straight-line approximation of $\angle H(j\omega)$ versus $\omega$ on a logarithmic scale. The straight-line approximations are useful because they can be hand-drawn easily and help indicate the effects of individual poles and zeroes. Often, however, precise plots of $|H(j\omega)|$ in decibels and $\angle H(j\omega)$ versus $\omega$ on a logarithmic scale are desired; these are often simply called Bode plots (*not* approximations). This assignment shows how to plot "exact" Bode plots in MATLAB.

### Creating Logarithmic Frequency Spacing

We have learned that the command `w=a:c:b;` creates a list of equally spaced values from `a` to `b` with adjacent spacing of `c`. In a plot in which the horizontal axis is on a logarithmic scale, the values will not be equally spaced and result in a poor-looking plot. To create a list of $n$ logarithmically spaced values from $10^a$ to $10^b$ use the command `w=logspace(a,b,n)`. For example, to create 100 values per decade from $\omega = 1$ to $\omega = 10,000$ use `w=logspace(0,4,400)`.

### Creating dB Magnitude and Phase Values

Suppose `h` is a list of Fourier transform values based on `w`. Letting `hmag` denote the magnitude values in dB and `hphase` denote the phase values, use `hmag=20*log10(abs(h));` and `hphase=unwrap(angle(h));`, respectively.

### Bode Plots

The following steps are used to plot a bode magnitude plot; for a bode phase plot, simply replace `hmag` with `hphase` and adjust the vertical axis label. These steps assume `w` has been defined with `logspace`.

1. `figure();`
2. `semilogx(w,hmag,'LineStyle','-','color',[0,0,0.8]);`
3. Use `axis` if needed.
4. Add a descriptive title using `title`.
5. Add a horizontal axis label using `xlabel('\omega');`
6. Add a vertical axis label using `ylabel('Magnitude of X(j\omega) in dB');` (Note that only the magnitude is in dB, not the phase.)

### Taking Derivatives and Integrals over Logarithmic Frequencies

The following approach works when any non-uniformly spaced $x$-axis values are involved. Because slope is rise over run, if `a` contains a list of your function values (the $y$-axis values) and `b` contains the corresponding (same-length) list of the $x$-axis values, the derivative can be computed via `diff(a)./diff(b)`. Remember that the `diff` operator produces lists that have one less element than the list they start off with. Thus the list of $x$-axis values for the purposes of plotting must also be shortened by one via `b(2:end)`.

Integrals are numerically computed as sums of areas of rectangles, where the widths are the differences of $x$-axis values, and the heights are the values of the function. If `a` again contains the function values and `b` contains the corresponding $x$-axis values, the integral can be found by computing `sum(a(2:end).*diff(b)`. Plotting must again be done versus `b(2:end)`.

Note that `b` should be `log10(w)` if you are using `logspace` to define `w`.

**PROBLEM STATEMENT**

Consider a linear time-invariant system with

$$H(j\omega) = \frac{80(j\omega)(60 + j\omega)(400 + j\omega)}{(80 + j\omega)(100 + j\omega)(150 + j\omega)(250 + j\omega)}.$$

1. Plot the Bode magnitude plot (in dB) from $\omega = 0.5$ to $\omega = 50,000$ using 100 points per decade. If you have done this correctly, you should see a bandpass filter.

2. Have Matlab determine and print out the maximum value of the Bode magnitude plot in dB. Then have it print out whether the filter is active or passive, based on this value.

3. Have Matlab determine the 20 dB bandwidth of this filter. The 20 dB bandwidth is defined as the difference between the upper and lower frequencies where the magnitude drops by 20 dB from the maximum value. Have Matlab also print out the values of these upper and lower frequencies (denoted by $\omega_U$ and $\omega_L$, respectively).

4. Have Matlab plot the SLOPE in dB/decade of the magnitude plot over the full frequency range of the plot.

5. Plot the Bode phase plot from $\omega = 0.5$ to $\omega = 50,000$ using 100 points per decade. Do not use phase wrapping.

6. Using the results of the previous part, plot the group delay from $\omega = 0.5$ to $\omega = 50,000$ using 100 points per decade. *Hint:* In contrast to the methods in class, the group delay in this case is determined and plotted versus frequency using a log scale for frequency.

7. We wish to obtain a measure of how linear the phase is in the region where the bandpass filter is "on". (We don't really care if a filter introduces phase-based distortion in the region where the filter blocks frequency content.) To do this, first have Matlab determine and print out the average value of the group delay (still using the log scale for frequency) over the range of frequencies within the 20 dB bandwidth. That is, use the log-frequency scale approximation of $D_{\text{avg}} = \frac{1}{\omega_U - \omega_L} \int_{\omega_L}^{\omega_U} D(\omega) d\omega$. Then have Matlab calculate the mean square error over this same range of frequencies using the log-frequency scale approximation of $\text{MSE} = \int_{\omega_L}^{\omega_U} (D(\omega) - D_{\text{avg}})^2 d\omega$. Finally, compute the normalized quantity $\sqrt{\frac{\text{MSE}}{\omega_U - \omega_L}} / D_{\text{avg}} \times 100\%$ to get a sense of how much the group delay differs from a constant value over the frequencies in which the filter is "on".