Useful MATLAB Commands

You will be mainly using the MATLAB Control System Toolbox. A great way to get started with the toolbox is to run the demo. This is done by typing `demo('toolbox','control')` at the MATLAB prompt.

Presented here are the most useful MATLAB commands for this class, grouped by topic. For further details, make good use of the `help` command in MATLAB. In general, other than the necessary MATLAB lab tasks, MATLAB should be used as a tool to verify your work, not substitute for it. The exception, of course, is for computationally intensive problems, in which case make sure to note on your assignment that you used MATLAB.

General Matrices:

inv(M) — Inverse of a matrix M

conj(x) — Returns the complex conjugate of a number, vector, or matrix.

transpose(M); M.' — Transpose of a matrix. You can do the complex conjugate transpose using `A'`.

eig(M) — Will return the eigenvalues of `M`. If used in the form `[V D] = eig(M)`, it will return return a diagonal matrix `D` whose entries are the eigenvalues, and a matrix `V` whose columns are eigenvectors of `M`.

rank(M) — Returns the rank of `M`.

Transfer Functions (TFs):

H = tf(num, den) — Creates a transfer function defined by $H(s) = \frac{num(s)}{den(s)}$. The parameters `num` and `den` are row vectors of the numerator and denominator coefficients, respectively, given in decreasing order of the degree.

Ex: `H = tf([1 2],[1 0 5])` creates the transfer function $H(s) = \dfrac{s+2}{s^2+5}$.

conv(a, b) — Convolve two polynomials `a` and `b` (represented as row vectors of their coefficients). It is particularly useful for determining the expanded coefficients for factored polynomials.

Ex: enter the TF $H(s) = \dfrac{s+2}{(s+1)(s-3)}$ by typing `H = tf([1 2],conv([1 1],[1 -3]))`.

series(G, H); G*H — Combine two TFs `G` and `H` in series. This will return the transfer function $T(s) = G(s)H(s)$

feedback(G, H) — Computes the transfer function of the feedback loop, where $G(s)$ is on the forward path and $H(s)$ is on the feedback path.

step(ss); step(H) — Plots the step response of a system, given either by a state-space representation `ss`, or by a transfer function `H`.

State Space:

sys = ss(A, B, C, D) — Creates a state-space model given by the equations $\dot{x} = Ax + Bu$, $y = Cx + Du$.

[A, B, C, D] = tf2ss(num, den)
[num, den] = ss2tf(A, B, C, D) — Conversion between state-space representation and transfer function (represented as row vectors of coefficients of the numerator and denominator).

`[A, B, C, D] = tf2ss(num, den)` returns the state space representation of the system given by the transfer function $T(s) = \frac{num(s)}{den(s)}$.

`[num, den] = ss2tf(A, B, C, D)` does the inverse operation (returns the numerator and denominator as row vectors of coefficients).

ctrb(A, B) — Compute the controllability matrix $\mathcal{C} = [B \ AB \ \dots \ A^{n-1}B]$.

obsv(A, C) — Compute the observability matrix $\mathcal{O} = [C \ CA \ \dots \ CA^{n-1}]^T$.

Stability Plots:

rlocus(H); rlocus(ss) — Plot the root locus of a system, given by its transfer function `H` or its state-space representation `ss`. **Keep in mind that this command is used on the loop gain of the system** as opposed to the closed-loop transfer function. For example, consider the standard negative feedback system with forward path G and feedback path H. The loop gain would be $G(s) * H(s)$ whereas the closed-loop TF would be $\frac{G(s)}{1+G(s)H(s)}$

bode(H); bode(ss)	Plots the frequency response (magnitude and phase) of the system given by transfer function `H` or state-space representation `ss`.
[Gm, Pm, Wcg, Wcp] = margin(H)	Calculates the gain margin `Gm`, the phase margin `Pm`, and the corresponding crossover frequencies `Wcm` and `Wcp`. If used without return values, i.e. `margin(H)`, plots the gains and crossover frequencies on the open-loop Bode plot.
nyquist(H); nyquist(ss)	Plots the Nyquist frequency response of LTI models. **Does NOT display contour at infinity.** Plot can be misleading because of bad scaling. Make sure to zoom and change axes view to get a complete picture.

Plotting:

plot(x, y)	Plots vector y vs. x. You can add options such as color, line-style, and more as a third argument. See `help plot` for more info.
figure(n)	Marks `Figure n` as active, so that the next plot command plots on the window called `figure n`. Ex: To plot into the window titled 'Figure 1', use `figure(1)` before the plot command. To open a new window, type `figure` and it will open a new window with the appropriate number.
subplot(m, n, p)	Splits the current figure into an m by n matrix of sub-figures, and selects the p-th element of the matrix as the active sub-figure. Ex: To plot y_1 v.s. x and $y2$ vs. x on the same window vertically (i.e. 2 rows and 1 column), use: `subplot(2,1,1), plot(x,y1)` followed by `subplot(2,1,2), plot(x,y2)`.
clf	Clears the current figure.
hold on	keeps the plots in the current figure, i.e. subsequent plot commands will be added to the current figure instead of replacing the content of the figure. Useful for plotting several graphs on the same figure. Ex: To plot y_1, y_2 and y_3 vs. x on the same figure, use: `hold on; plot(x,y1); plot(x, y2); plot(x, y3)`.
plotyy(x1, y1, x2, y2)	Plots two lines on the same figure (`y1` vs. `x1` and `y2` vs. `x2`), with a separate y-axis for each. One will be displayed on the left of the figure, the other on the right.

axis([xmin, xmax, ymin, ymax])	Sets the ranges of the axes. Other commands allow you to controls axis scaling and appearance. There are also a number of preset modes. See `help axis` for more information.
title('Some title')	Adds a title (character vector) on top of the currently active figure.
xlabel('x'), ylabel('y')	Labels the axes of the currently active figure. These take a character vector (string) as argument and place it on the appropriate axis.
legend('legend1', 'legend2', ...)	Adds a legend to the currently active figure. It will assign the legends to the graphs in the order they were plotted.