ECE 433 Industrial Electronics and Control Systems

Lab #1 Matlab Programming

<u>*Objective*</u>:
Learn how to use MATLAB to work with transfer functions and differential equations.

<u>*Introduction*</u>*:*
MATLAB has a handy **help** command to get information on all its functions. We will make extensive use of the commands **tf, roots**, **pole**, **tzero**, **pzmap**, **iopzmap**, **conv**, **poly**, **zpk**, **zp2tf**, **tf2zp**, **step**, **ltiview**, **tf2ss**, and **lsim**. Please study the help carefully for these commands. All of these commands are relevant to linear time-invariant (LTI) differential equations. For a single-input, single-output (SISO), system, its scalar *transfer function*, *G,* is the ratio of its single output to its single input. The transfer function of a LTI differential system in the Laplace transform domain can be expressed as the ratio of polynomials

$$G(s) = \frac{b_1 s^m + b_2 s^{m-1} + \ldots + b_m s + b_{m+1}}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \ldots + a_{n-1} s + a_n}.$$

Such a transfer function is completely described by vectors of its numerator and denominator polynomial coefficients. These can be entered into the MATLAB environment by the following MATLAB statements:

num = **[b_1, b_2, …, b_m, b_{m+1}]**;
den = **[1 a_1, a_2, …, a_{n-1}, a_n]**;
G = **tf(**num, den**)**;

An alternative procedure for entering transfer functions is to define the MATLAB variable, s as the MATLAB transfer function object *s.* Do this with the command

```
s = tf('s');
```

or equivalently as

```
s = tf([1,0],[1]);
```

This method allows transfer functions to be entered directly in terms of *s* using algebraic notation, instead of as coefficient vectors. Depending on the situation, one method or the other is easier to apply.

Another way to represent the transfer function of a LTI/SISO system is the zero-pole-gain (ZPG) representation,

$$G(s) = K\frac{(s-z_1)(s-z_2)\ldots(s-z_m)}{(s-p_1)(s-p_2)\ldots(s-p_n)},$$

where K is referred to as the gain of the system, z_i (i = 1, …, m) are the zeros and p_i (i = 1,

…, n) are the poles of the system. The *poles* of the system occur when the value of the transfer function goes to infinity, as it will when the denominator equals zero. The *zeros* of the system occur when the value of the transfer function equals zero, as it will when the numerator equals zero. Note that K is **<u>NOT</u>** the zero frequency or dc gain G(0). The **zp** family of MATLAB commands work with transfer functions in this form.

Finally, a transfer function $G(s)$ can also be expressed in a partial fraction expansion as

$$G(s) = \frac{r(1)}{s-p(1)} + \frac{r(2)}{s-p(2)} + ... + \frac{r(n)}{s-p(n)} + k(s).$$

The partial fraction expansion can be carried out in Matlab by using the **residue** command. In this form, the transfer function is easy to transform back into the time domain as each partial fraction represents an exponential mode of the system.

<u>*Lab Problems:*</u>

1. <u>Poles & zeros</u>:
   (1) Use the **roots()** command to determine the poles and zeros of the following transfer functions:
   a. $G(s) \;=\; \dfrac{s^2+2}{s^3+2s^2-s+1}$
   b. $G(s) \;=\; \dfrac{s^3+1}{s^4+2s^2+1}$
   c. $G(s) \;=\; \dfrac{4s^2+8s+10}{2s^3+8s^2+18s+20}$

   (2) Repeat (1) by using **pole** and **tzero** commands to find the poles and zeros.
   (3) Use the **pzmap** command to plot the poles and zeros in part (1). Show the values of the poles and zeros on the complex plane. Which systems are stable?
   (4) Use the **iopzmap** command to show the poles and zeros of all three transfer functions in part (1) on the same plot.
   (5) Use the **conv** command to expand the numerator and denominator of the transfer function and express the function as the ratio of two polynomials.
   $$G(s) \;=\; \frac{6(s+5)}{(s^2+3s+1)^2(s+6)(s^3+6s^2+5s+3)}$$
   Show the same result by entering the factors algebraically as **tf** objects and using the * operator for multiplication to combine them. Which was easier?

2. Use the **poly** command to get the polynomial coefficients for the numerators and denominators of the following transfer functions:
   a. $G(s) \;=\; \dfrac{(s+1)(s+2)}{(s+3)(s+1+j3)(s+1-j3)}$
   b. $G(s) \;=\; \dfrac{3(s+3)(s+1)}{(s+2)(s+2-j)(s+2+j)}$

3. Repeat problem 2 by using the **zpk** command.

4. Repeat problem 2 by using **zp2tf** command. Compare the results with those from problem 3 and describe the difference(s) if there is any.

5. Use the **zpk** command to express the following transfer function in the ZPG format:

   $$G(s) = \frac{2s^2 + 4s - 16}{s^3 + 6s^2 + 11s + 6}.$$

6. Repeat problem 5 by using the **tf2zp** command. Compare the result with that from problem 5 and describe the difference(s) if there is any.

7. Use the **residue** command to find the partial fractions of the following transfer functions:

   a. $G(s) = \dfrac{-s-4}{s^2 + 3s + 2}$

   b. $G(s) = \dfrac{s^4 + 2s^3 + 5s^2 + 3s + 6}{s^3 + 12s^2 + 39s + 28}$

   c. $G(s) = \dfrac{2s-1}{s^2 + 2s + 1}$

8. Use the **step** command to solve the following differential equation and plot the solution.

   $$\mathrm{y''(t) + y'(t) + 2y(t) = 8u(t),\; y(0) = y'(0) = 0.}$$

   Use the right-click mouse menu and data cursor to annotate the location and numeric coordinates of the peak response, 10-90% rise time, and 2% settling time.

9. Repeat problem 8 by using the **ltiview** command. Use the drop down manual under **Edit** to change the rise time and settling time under **Options** in “Viewer Preferences”. Use the right-click mouse menu to (1) annotate the location and numeric coordinates of the 0-100% rise time and 5% settling time, and (2) show the impulse response of the system.

10. Consider the following differential equation with sinusoidal drive and non-zero initial states:

    $$\mathrm{y''(t) + 2y'(t) + 5y(t) = \sin(5t),\; y(0) = 2,\; y'(0) = 0.}$$

    Use the **tf2ss** command to express the above system in the state-space form,

    $$\dot{X}(t) = A \cdot X(t) + B \cdot u(t);\;\; y(t) = C \cdot X(t) + D \cdot u(t).$$

    Use the **lsim** command to simulate the following differential equation for 15 seconds and plot the solution vs. time. Be sure to fully annotate your plot.