Lab Week 7: Process Identification and Empirical Modeling

0. Learning objectives

To practice the procedures for 1st and 2nd order time response process identification using data from a 1st order sensor response and a 2nd order RLC circuit.

1. Collection of 1st Order System IO Data (Step Input Test Demo)

The device to be modeled is a temperature IC chip. The chip outputs voltage in mV. The output temperature in °F can be found by dividing the millivolts by 10. We wish to identify the empirically based transfer function (relating the input temperature and output temperature) and first order time constant for the temperature chip. The data is available on Blackboard.

- Total time: 72 seconds
- Sampled every: 0.12 seconds
- The first sample occurs at time = 0 seconds
- System input: step input at time = 0 seconds from y_o to y_{ss}

2. Collection of 2nd Order System IO Data (Step Input Test Demo)

The device to be modeled is a standard R-L-C circuit (see Figure 2.15, page 51 in the textbook), whose input and output are in mV. We wish to identify the empirically based transfer function and time constant for the component. The data is available on Blackboard.

- Total time: 140 μsec (sec)
- Sampled every: 0.04 μsec (sec)
- The first sample occurs at time = 0 seconds
- C = 1 μF
- System input: step input at time = 0 seconds from y_o to y_{ss}

3. Assignment

3.1 Analyze the data from the temperature sensor to fit a 1st-order time model to the system (refer to lecture slide 96 and 126).

a) Develop the *time function* and *transfer function* of the 1st order system.
   a. Remember: for a first order system, $y(t) = y_{ss} + (y_o - y_{ss})e^{-t/\tau}$
   b. When developing your *transfer function*, you do not need to perform an inverse Laplace transform on the time function.
   c. Start by determining the time constant (τ) of the temperature sensor

b) Identify the *output variable* (remember that a transfer function is output divided by input)

c) Using a Matlab script, *plot* the output response from the experiment and the output response calculated using the `step()` command (both traces on the same graph using Matlab).
   a. Use the `step()` command to plot the output response of your model (i.e., output response of your transfer function)
   b. See suggestions in Section 4 for tips on plotting with the `step()` command

d) *Locate* online the specifications for the National Instruments LM34 TO-92 IC and *compare* the published time constant with your estimate.
   a. Look for the relevant graph from the PDF spec sheet found online. Assume air velocity is 0 fpm

e) Draw the block diagram for this 1st order system (as seen in Figure 1)
   a. Include name and units for the input and output arrows
   b. Include the transfer function equation within the block

Figure 1: Block diagram

3.2 Analyze the data from RLC circuit and fit a 2nd-order model to the system (refer to lecture slide 103 and 126)

a) Let the input variable be V_{in} and your output variable be V_C and calculate the *differential equation* using the methods developed in class (textbook Fig. 2.15, pg. 51). Show your work.

b) Develop the *time function* (the **normalized** equation from textbook pg. 77 and lecture slide 101 is shown below) and *transfer function* of the output.

$$c(t) = 1 - \frac{e^{-\zeta \cdot \omega_n \cdot t}}{\sqrt{1-\zeta^2}} \cdot \sin\left(\omega_n \sqrt{1-\zeta^2} \cdot t + \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}\right)$$

c) Plot using Matlab
   a. On one graph plot:
      i. Step response of time function (using `ezplot()` command)
      ii. Step response of transfer function (using `step()` command)
      iii. Experimental output (V_c) vs. time

d) Given the capacitor value, calculate the *resistance* and *inductance* as determined by the coefficients in your transfer function model (be careful with units).

f) Draw the block diagram for this 2nd order system (as seen in Figure 1)
   a. Include name and units for the input and output arrows
   b. Include the transfer function equation within the block

4. Lab Suggestions

- Make new columns of your data and (a) start at time = 0 and (b) scale your data to physical system units. The scaling factors are provided above. You can use Matlab or Excel to scale the data, but you must use Matlab for plots.
  - See instructional PDF on Blackboard for adding Excel data to Matlab
- Your transfer functions should represent the dynamic characteristics of the system (sensor or circuit), independent of the input and initial conditions. However, your time domain equations (time functions) will be for the specific response seen in the respective data set.
- For plotting time-domain functions, you can use:

```
ezplot('your equation in terms of x',[0 60])
```

  - For example, if you want to plot the equation $x^2 + x - 10$ from 0 to 60, you would enter to command:

```
ezplot('x^2 + x - 10',[0 60])
```

- For plotting s-domain functions, you can use the `step()` command:
  - See the following code for an example of how to plot the step response of a first order transfer function $\frac{1}{s+1}$ using a unit step input with no initial conditions

```
numerator = [1]
denominator = [1 1]
sys = tf(numerator,denominator)
step(sys)
```

- The initial conditions of the `step()` command can be set using the `stepDataOptions()` command. For example, if you want to set initial conditions and apply them to your `step()` command, you would enter these two lines of code.

```
stepOpts = stepDataOptions('InputOffset',y_o,'StepAmplitude',y_f - y_o);
step(sys, t_final, stepOpts);
```

5. Deliverables

Answer the questions from Problem 3.1 and 3.2. For items c and d in parts 3.1 and 3.2, also *comment* on the quality of the fit, why there are or are not differences between the experimental and model plots. For 3.1.d, *compare* the measured and published values. Also include the *Matlab* code. You do not need an executive summary.