University of Utah
Electrical & Computer Engineering Department
ECE 3510 Lab 2
Introduction to dSPACE
Using a First - Order System
(Velocity Control of a DC Motor)

A. Stolp, 1/22/08
rev, 1/16/13, 1/21/17 Bhavana Mukunda, 2/1/18 Abid Hossain
Version d: 1/24/19

Note : In future labs I'll expect you to be able to set up and use the dSPACE system with Matlab without all the detailed instructions provided in this lab handout. You may want to highlight sections of this handout and keep it for reference.

Objectives
- Learn how to setup and use the dSPACE equipment and software.
- Learn how to import dSPACE data into Matlab for analysis and plotting.
- Simplify the DC motor model shown in the appendix of Lab 1 to a first-order model.
- Observe the step response of a first-order system and determine the time constant and DC gain.
- Observe discrepancies from the first-order model.

Equipment and materials from stockroom:
- DC Motor (Record the DC __ number on the motor in your lab notebook)
- Dual power amplifier (Lunch-box sized aluminum box with handle)
- dSPACE cable

If you are in a lab with more than four other groups, you will get the following items from your TA as needed. They are in limited supply.
- Two aluminum wheels with two hex wrenches

dSPACE
Select a workstation along west or north wall of MEB 2365. It should have a dSPACE I/O breakout box (shown at right).

Introduction to dSPACE
The dSPACE system is an independent processing and data acquisition system that can implement digital control models. The system includes three main components:
- The DS1104 PCI Development Board (plugged into a PCI slot inside the computer)
- I/O Breakout Box connected to that board by a huge black cable
- ControlDesk software run on the computer to interface with the development board. (Some computers also require a USB software protection dongle.) ControlDesk will accept a ".sdf" file generated by Matlab's Simulink which will tell it what you want the development board to do.

The dSPACE PCI development board installed in the lab computer has it's own "embedded" processor. Before the system can run someone must write some code that can be uploaded to the this board. This is done by building a Simulink model which defines the inputs and outputs and the operations performed by the board. When the system is running, all of the input-to-output processing is done by the board. All the PC does is command the board to run or stop running and handle the data storage and display.

The ControlDesk software loaded on the PC interfaces with the development board and creates the user interface (called the "Layout"). Our Layout was originally created in Simulink, but you won't be dealing with that today. When the Layout is running ("Online"), it collects, stores, and displays data. In 3510 labs the user interfaces and most of the software will be created for you. In this first lab you will simply be the user and will not create any software. In later labs you will make some additions to the pre-written software to implement different types of control.

The "Layout" is a user interface area which shows "Instruments". Instruments are data input, output and recording tools. In ControlDesk, these instruments can be dragged and dropped into the layout from the *Instrument selector* tab.

The DS1104 PCI dSPACE board together with it's I/O breakout box provide the following:

- 8 analog inputs with 12 or 16 bit analog to digital converters (ADCs)
- 8 analog outputs with 12 or 16 bit digital to analog converters (DACs)
- 2 position encoder inputs
- Onboard independent 64-bit floating point processor
- Onboard Slave Digital Signal Processor (DSP)
- Onboard memory
- Other I/O capabilities

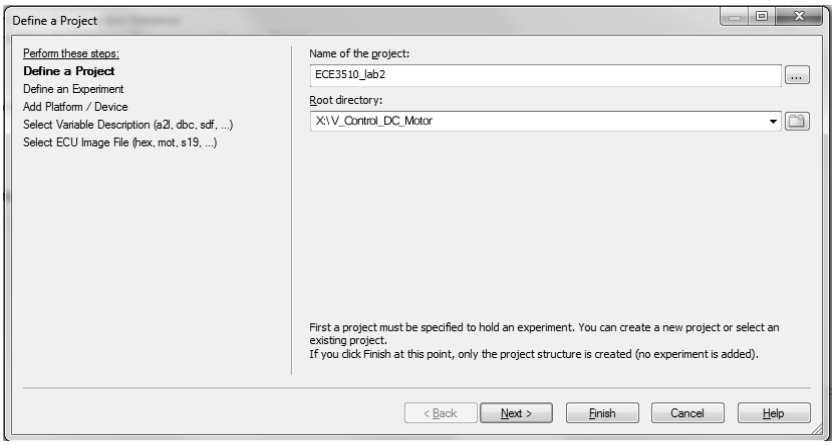
We will only use a small fraction of these in our labs. You'll make connections to the breakout box later.

dSPACE Control Desk Setup

Create a new folder in your 'X" drive or a thumb drive (let's call it "V_Control_DC_Motor"). This folder **MUST** be in some personal space of your own. Download the zip file named V_Control_DC_Motor from the lab website (currently: http://www.ece.utah.edu/~ece3510/ –> labs) and save in your new folder. In the zip file, you will find a Simulink model (ECE3510_lab2.mdl, which we will not use), several dSPACE files (named ece3510_lab2 with various extensions, Trigger Rule 1.txt, and Recorder 1.xml), and Matlab data unpacking code (Mat_Unpack.m). Extract all of these to your new folder.

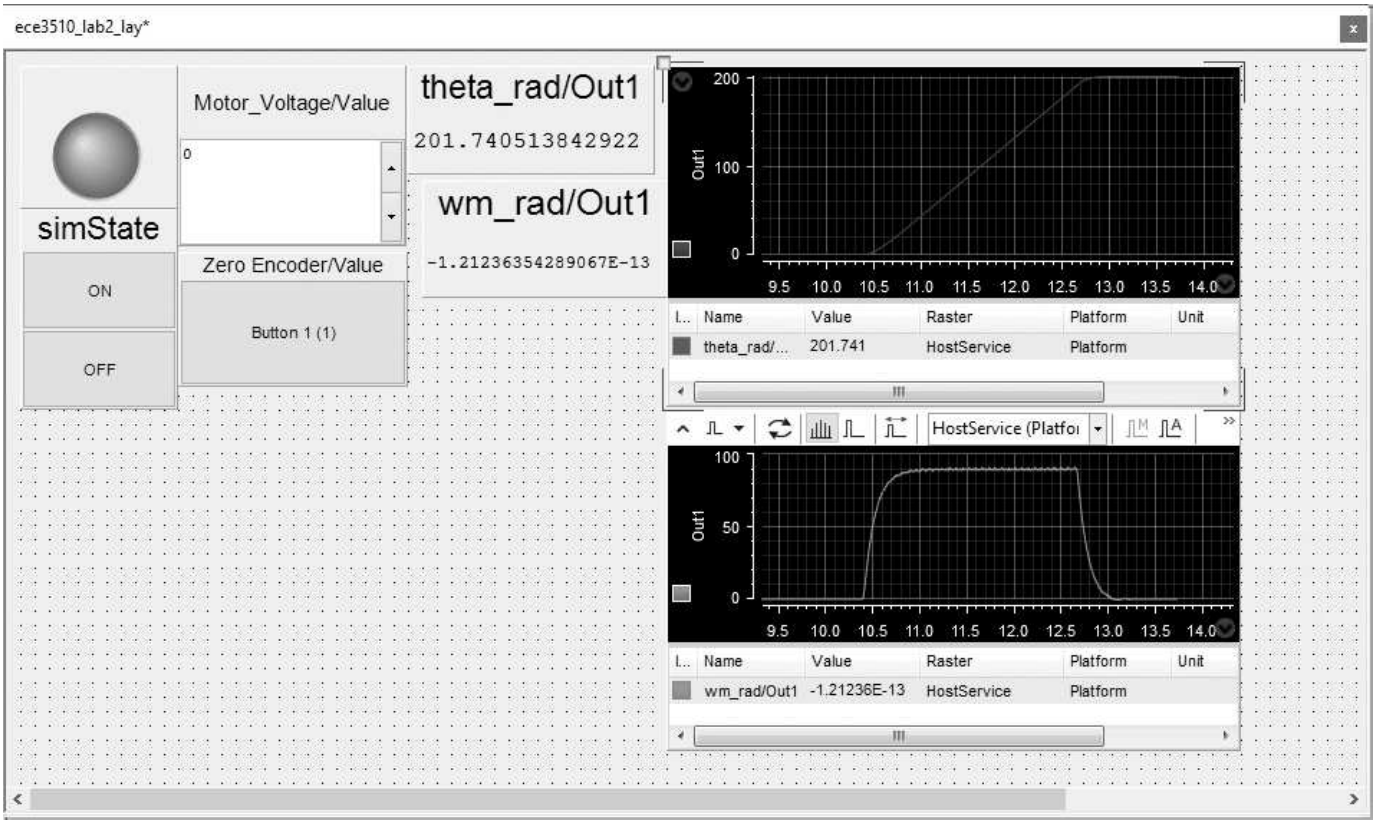
Find the dSPACE ControlDesk (6.1) program and start the application. Be patient, this takes a while. To create a new experiment, **File > New > Project + Experiment**

This will open a new window which will take you through the necessary steps to create a new project + experiment.

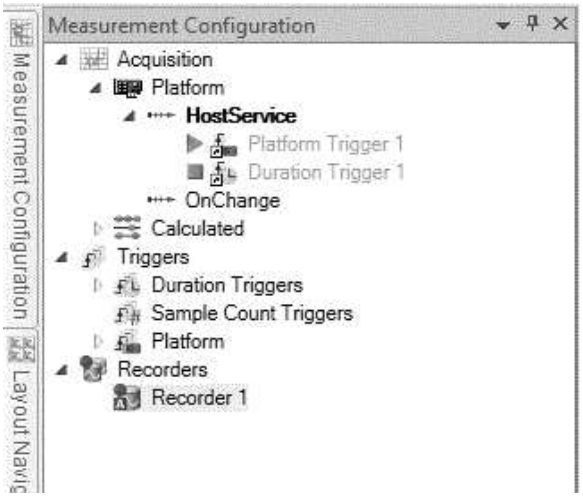


- Give the project a name, say ECE3510_lab2.
- Set the root directory as the folder you created earlier (X:\V_Control_DC_Motor in my case). **> next**
- Give the experiment a name, say ECE3510_lab2_exp. **> next**
- In the window, select the "DS1104 R&D Controller Board". **> next**
- **Import from file** Open the ECE3510_lab2.sdf **> Finish**

Once the experiment is generated, a layout will be provided. If it's blank, just close it. You will be using a layout file from the .zip file. On the toolbar (shown below), click on the **Layouting** tab and select "Import layout". Choose the ECE3510_lab2.lax. This layout will be used to control the DS1104 PCI dSPACE board in this lab.

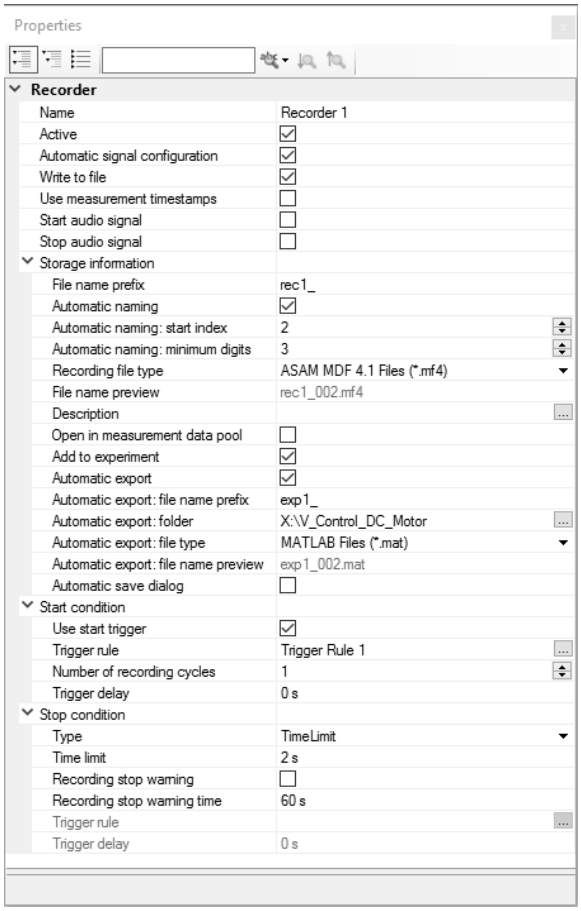


Setting up data recording



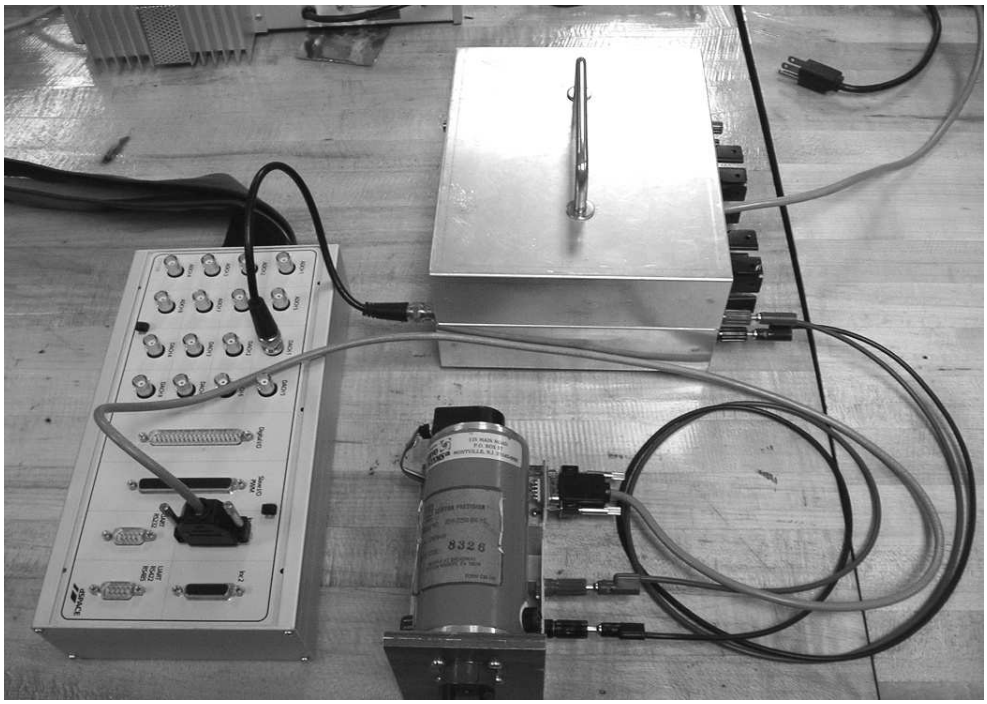
- On the left hand side of the big screen, click on the Measurement Configurations tab.
- Under Acquisition, expand Platform, right click on HostService, select "Measure Continuously (Disables Triggers)" and ensure that "Auto Repeat" is unchecked. On choosing to measure continuously, Platform Trigger 1 and Duration Trigger 1 will be disabled.
- Back to the Measurement Configurations tab, In "Recorders" section right click on "Recorder 1" and remove it. Right Click on "Recorders" then click on "Import Recorder". Choose "Recorder 1.xml" from the browser section > Open. Right click on Recorder 1 and select Properties. Find the "Storage information". Under this section, check the "Automatic export" box. (You may edit the Automatic export: file name and data will be saved with the name you give and three digit count as suffix. We'll just leave it as "exp1".)
- In the Automatic export: folder line, use the browse button change the directory to your folder. (X:\V_Control_DC_Motor in my case)
- In the Automatic export: file type line, make sure to change the file type to .mat.

- Check the “Automatic save dialog” box.
- Under the Start Condition in the recorder properties, check the “Use start trigger” box. This will allow you to set a trigger rule for which the recording would commence.
- To set a trigger rule, click on the browse button to open the “Edit trigger rules” window. On the bottom left hand side, click on “Import”. Find the Trigger Rule 1.txt file (you may have to move to a higher folder) > Open. This sets the recorder trigger to start recording data whenever the output voltage increases.
- You are not likely to need a Trigger delay, so let it remain zero. But, if on examining your data, you realize that the data should have started to record a little earlier, entering “-0.2” for the trigger delay should be sufficient. This will ensure that the data will include data from 0.2 seconds before the trigger event.
- Under the Stop Condition in the recorder properties, set Type to TimeLimit and set the Time Limit to 1 or 2 seconds.
- Your data capture settings are now complete.



Hardware Setup

- Use the the dSPACE cable you checked out, a BNC-to-BNC cable and two banana-to-banana wires (hanging in the lab) to compete the connections shown at right. Remove the banana wires from the motor and temporarily hook a voltmeter in place of the motor.



- On the computer screen, you will first engage the layout (set it to control the DS1104 PCI dSPACE board) by clicking on the **Home >** “Go Online” button and then engage the plotters by clicking on “Start Measuring”. Now you can turn the experiment ON by clicking on the ON button on the simState instrument. You will notice that the simState “light” will change from RED to GREEN.

- Check the operation of the encoder by manually spinning the motor and noting a change in

encoder position in the layout window (computer screen). Note that you can only see these changes when the layout is online, Measuring is on, and the experiment is ON (simState).

- Plug in the amplifier box and turn it on (one switch by the AC power input and one by the BNC signal input). On the layout, find the numeric input labeled "Motor_Voltage/Value". Increase this to 10 V, either by clicking on the arrows or by simply typing in the number into the instrument and hitting enter. You should be able to measure a DC voltage of about 10 V at the red and black terminals of the amplifier. (If you measure the input voltage to the amplifier you would only see about 2V. The amplifier has a voltage gain of 5, but we don't much care since the layout has been calibrated to account for that.

- Reduce the Motor_Voltage to 0 and click the OFF button on simState to stop the experiment. Also click on "Go Offline" button on the toolbar to disengage the layout. Disconnect the voltmeter and return the leads to the motor.

- Go Online, Start Measuring and click on the ON button on simState and change the motor voltage up and down to get a feel for how the system drives the motor. Also, note the values and graphs of the interface. Reduce the voltage to 0 and go Offline when done.

Draw a schematic or block diagram of the system in your notebook and writeup a little about the hookup and function.

Experiment

Introduction

In the appendix of Lab 1 the transfer function of a DC permanent-magnet motor was found as:

$$\frac{\theta(s)}{V(s)} = \frac{K_T}{JL_a s^3 + (JR_a + B_m L_a)s^2 + (B_m R_a + K_T K_V)s}$$

We are going to consider the output to be angular velocity ($\omega(s) = s\theta(s)$) rather than angular position. Furthermore, we are going to neglect L_a , that is, consider it to be 0. In your notebook, show that leads to a transfer function of this form:

$$P(s) = \frac{\omega(s)}{V(s)} = \frac{k}{s+a}$$

Express k and a in terms of J, R_a , B_m , K_T , and K_V . This is a first-order transfer function. The pole of this system is at $-a$ and the time constant of the system is:

$$\tau = \frac{1}{a}$$

The DC gain of the system is: $P(0) = \frac{k}{a}$

Our input will be a step voltage: $V(s) = \frac{v_0}{s}$

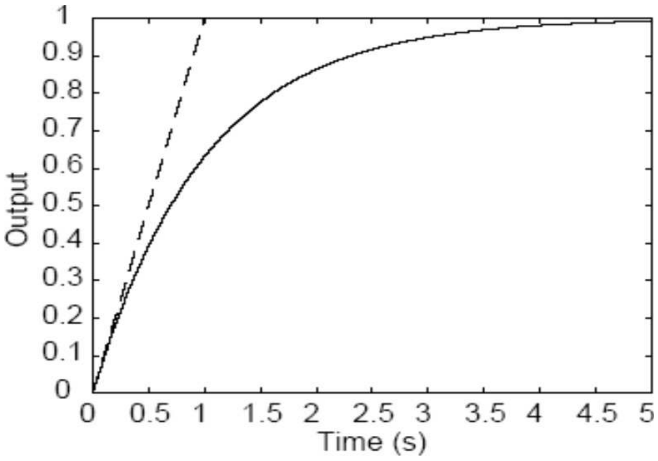
The step response will be the input times the transfer function. Show (by partial fraction expansion) that the step response in the time domain is:

$$\omega(t) = \frac{k}{a} v_0 (1 - e^{-t/\tau})$$

Notice that the steady-state output is: $\omega(\infty) = \frac{k}{a} v_0$

That is, the DC gain (k/a) times the input voltage (v_0). In other words, the DC gain relates the eventual output shaft speed to a given the input voltage. Notice that since the DC gain relates a speed to a voltage, it will have units, like (rad/sec)/V. It is not the V/V type gain factor that you are used to other classes.

A typical first-order response is shown at right. Notice that the initial slope is $1/\tau$. Also notice that the curve reaches 63% of it's final value in one time constant.



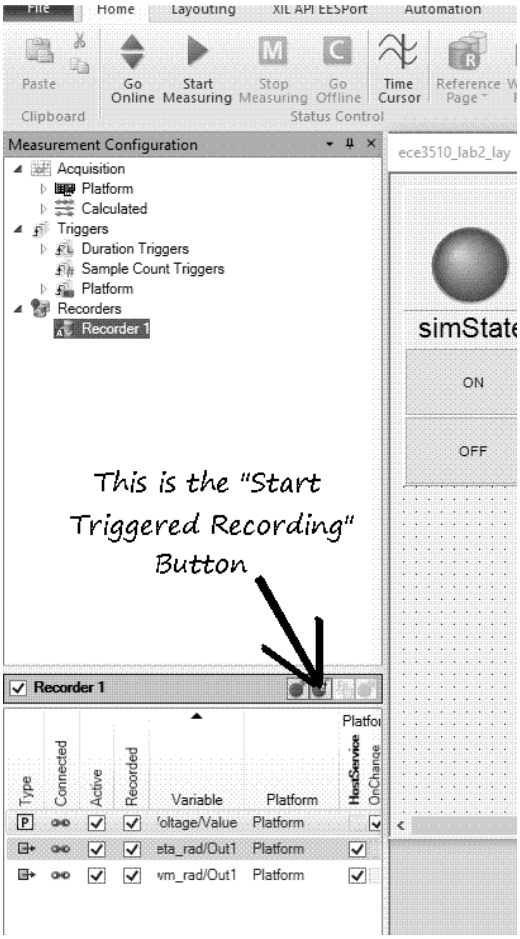
Step Resonse

Now you're all set up to apply voltages to the motor and record data in a format that can later be imported into Matlab. The "Motor_Voltage" input in the layout window will set he voltage level and data will be captured at 500 samples/sec or one sample every 2 msec.

At the risk of stating the patently obvious... **Caution**, each time you run the following experiment the **motor will run**. Keep fingers, hair, and loose watches, jewelry, etc. away from the shaft, coupler, and/or wheels.

Do the following:

- To capture the data, Go Online, Start Measuring and click on the "Start Triggered Recording" (second button from the left on the recorder). Click on the ON button on simState. Type "10" in the "Motor_Voltage/Value" and hit enter Recorder 1 should run for the time you set earlier (1 or 2 seconds) and then a "Save Measurement Data" window will pop up. Don't mind the "rec1_00x.mf4" file, just note the name of the Auto export file "exp1_00x.mat". OK. Reduce the voltage to 0 and go Offline when done. Note: just hitting the "Start Triggered Recording" button will automatically go Online and Start Measuring too.



- If you want to see this data and check it before proceeding, jump forward to the "Data Analysis" section of this handout. If you are short for time, continue here and take the rest of the data first.

- To be extra safe, temporally disconnect one of the wires to the motor. Remove the coupler from the motor shaft and replace it with one of the aluminum wheels. (The set screws are different sizes and the wheels have two.) You may need to get the hex wrenches and

a wheel from your TA. Note: It is always a good idea to check that the experiment is OFF and offline before making any adjustments to the hardware or software so that the motor won't start unexpectedly. (Of course it can't with the wire disconnected.) Reconnect the wire.

- Capture the 10-V step response with the greater inertia of this wheel and record the name of the data file.

- Replace the wheel with the other size wheel and note which is which so that you can later separate your data for the coupler-only, small wheel, and large wheel.

- Capture the 10-V step response with the inertia of this new wheel and record the name of the data file.

- Describe the procedures on your lab notebook.

- Reinstall the coupler.

Data Analysis

Using Mat_Unpack to import the data into Matlab

dSPACE generates .mat file when it saves data to a file. The structure uses a "dot" convention to differentiate elements in the structure (ie.. name.field.data). This can be somewhat tedious to work with when you want to perform analysis and manipulation in Matlab®, so a custom Matlab® script called Mat_Unpack.m is provided to unpack the .mat data structure. Mat_Unpack is designed to list the contents of the structure and unpack the data to variables for use in the workspace.

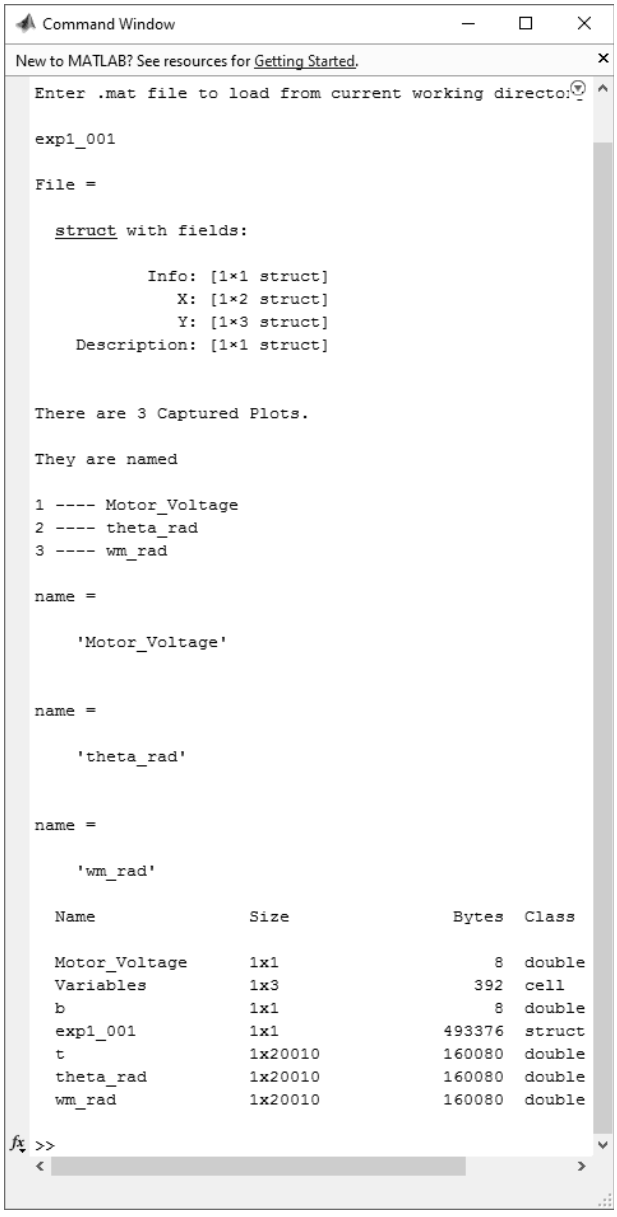
To use Mat_Unpack:

- Make sure the file you're working with is in the current working directory for Matlab®.

- Make sure Mat_Unpack.m is in the same directory as the file or in the /work directory on the machine you're using.

- Type Mat_Unpack to initialize the script in the command window of Matlab®

- Type the name of the file but do not include the .mat extension. You will be presented with a prompt that gives information about the data in the structure. **NOTE: if you changed the filename after saving it, the script will fail.** The variables are named just like they were named in dSPACE and simulink file. Time variable in named as "t".



- Finally all the variables in the workspace are listed. You will notice that along with the variable you created is a struct with the name of your file. That struct is the unpacked data.
- Now you can plot the data by using standard Matlab methods (ie.. plot(t,wm_rad))

Computations to be done using the data sets

For each of the data sets, coupler only, small wheel, and big wheel.

- Plot the velocity data verses time. Plot the response over short time periods to clearly show the curves and compare it to an exponential curve. The shape should be comparable.

- Obtain an estimate of the time constant based on the time it takes for the output to reach 63% of its steady-state value and again by extending the original slope to the final value. See the representative exponential curve a couple of pages back.

- Determine the steady-state velocity from the flat part of the curve, where it is approximately constant. You can do this on the plot, or mathematically in Matlab.

- From the results, calculate the values of the DC gain in both (rad/sec)/V and rpm/V. Calculate the parameters k and a of the system. (Refer back to the calculations and equations on page 5.)

- Discuss the effects of added inertia on the model parameters.

Inaccuracy of the first-order model

Look back at your plots and find where they are not a simple exponential curve. How can the effects of the motor inductance, which we neglected, account for the discrepancy? Nevertheless, the first-order model is not too bad.

A second-order model, rather than first-order, would result and would more accurately reflect the behavior of the motor. We will find the parameters of such a model in a later lab. In practice however, any model, no matter how high-order or how detailed it is, is only an approximation of the real system.

Nonlinear Inaccuracy

The Amplifier used in this lab cannot supply an infinite amount of current. This is a nonlinearity. If the current is limited, then the torque available to accelerate the motor will also be limited. If the limit is a constant value, then the acceleration will be constant. This could show up as a constant-slope section in the velocity “curve”. Check your curves for this effect.

Conclusion

Check off before tearing apart your system or leaving the lab. Make sure the experiment and layout are OFF and offline respectively, and save your progress. Turn off the computer before unplugging the IO breakout box. If you’ve done a good job above comparing the plots, discussing the effects of inertia and the limitations of the first-order model, your conclusions here can be pretty slim. You should at least write about how you did or did not meet the objectives of the lab.