

# Lab 2: Introduction to Real Time Workshop

## 1 Introduction

In this lab, you will be introduced to the experimental equipment. What you learn in this lab will be essential in each subsequent lab. Document what you learn now and the rest of your ECE147A/B laboratory experience will be much more pleasant.

### 1.1 Hardware

The *MultiQ-3 Terminal Board* (Figure 1) is a general purpose data acquisition and control board which has eight single ended analog inputs, eight analog outputs, sixteen bits of digital input, sixteen bits of digital output, three programmable timers, and eight encoder inputs decoded in quadrature. For the purposes of this lab, we will be primarily concerned with the analog outputs, the encoder inputs, and, to a lesser extent, the analog inputs.

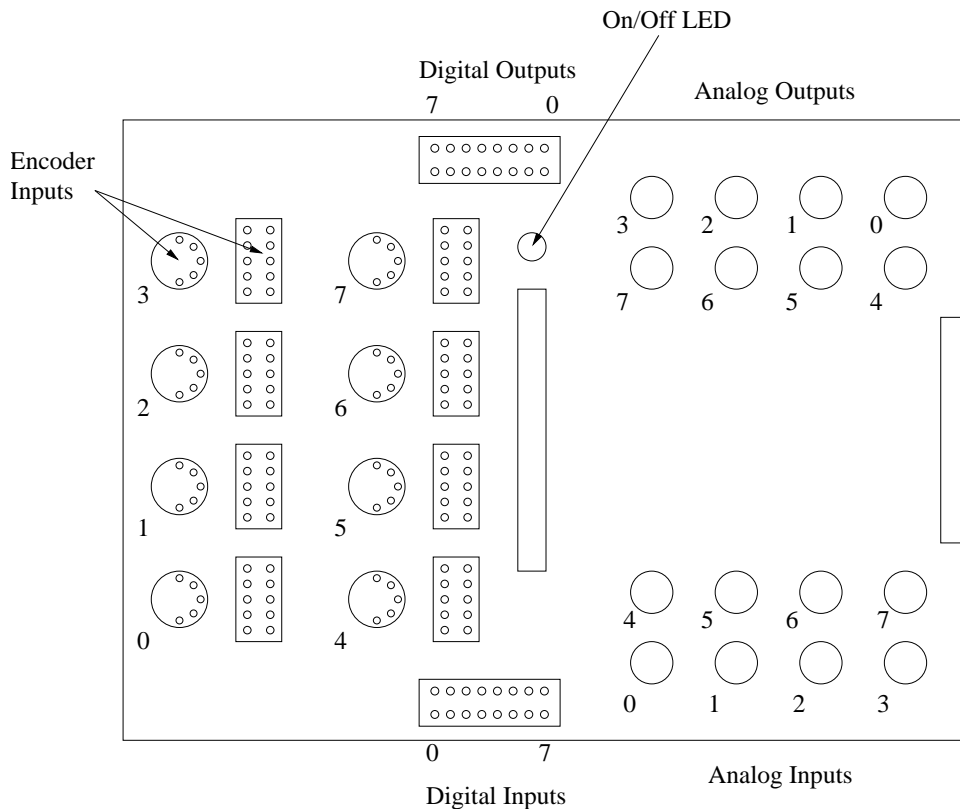


Figure 1: MultiQ-3 Terminal Board Schematic

Both the analog inputs and outputs are accessed via clearly labeled RCA jacks on the MultiQ-3 board and have a range of  $\pm 5$  volts. The encoder inputs are accessed via a 5 pin DIN socket and connect directly to the experimental equipment.

## 1.2 Software

The functionality of the MultiQ-3 board can be completely controlled via SIMULINK and Real Time Workshop (RTW). RTW is a real-time Windows application that runs SIMULINK generated code on a PC.

To use the Quanser MultiQ-3 board, go to the Simulink Library Browser and

1. Select the Real-Time Windows Target library.
2. Select Analog Input/Output or Encoder Input from the selections on the right.
3. Drag the selected block onto your Simulink Model.

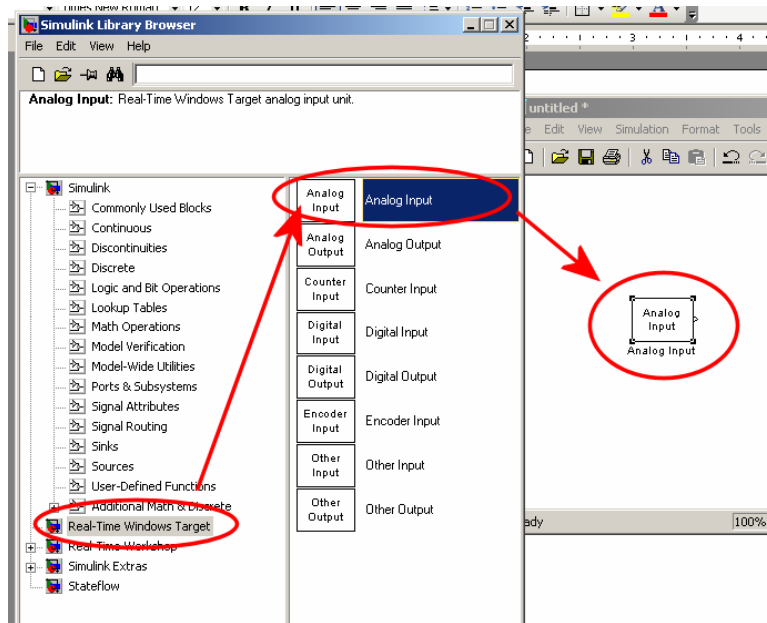


Figure 2: RTW Target

4. Double click the I/O block to bring up its properties box. Under the Data acquisition board area click the pull down menu to see a list of installed boards. If Quanser MultiQ-3 [320h] shows up, then select it. If not, click the Install new board button and select MultiQ-3 from the Quanser option in the menu that pops up. Then select the board from the pull down menu.
5. Set the Sample Time : field to the model sample time, e.g. 0.001
6. Select the ports that your signals will use. The Quanser MultiQ-3 Terminal Boards channels are labeled 0-7, while Simulinks indexes run 1-8, so Channel 1 in Simulink implies Channel 0 on the Terminal Board.
7. IMPORTANT: For any ANALOG OUTPUT blocks, set FINAL VALUE = 0. This way, when the model finishes its run, it forces the output voltage to the DC motor to be 0 V. If you have an unstable controller this is very important!

- Note: Channel 0 on the Terminal Board is used most often and is the most likely to not function. It is a good idea to try other channels if the model does not seem to be outputting the correct voltage (assuming your model is running correctly).

See Figure 3 for an example of how to do the above.

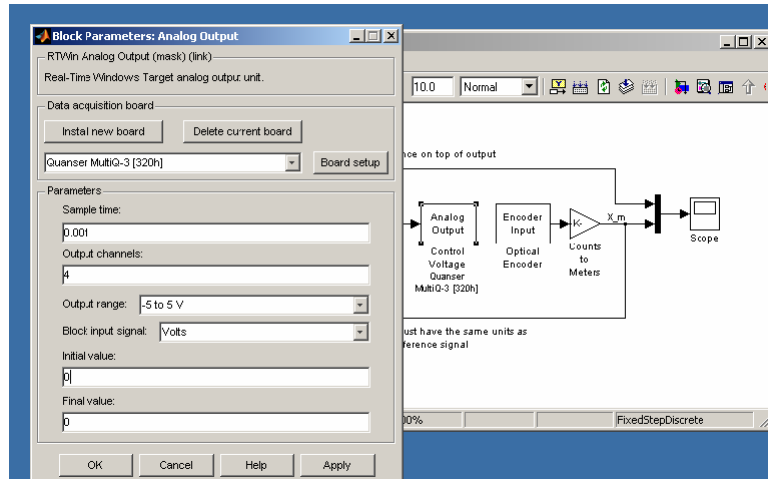


Figure 3: Configuring the I/O boards

### 1.2.1 Data Logging

In the Real Time Windows Target, data is logged by scope blocks. Like an oscilloscope, to start the data logging procedure, a trigger is needed to initiate the process. A simple approach is to define a dedicated scope block, with one trigger signal entering it. Figure 4 shows a step function set with a step time of 0.001 seconds, which corresponds to one sample time. This set-up will save all but the very first data point. To record data, follow the procedure below.

- Place a STEP function block from the SOURCES library and a SCOPE block from the SINKS library in your model.
- Name the SCOPE block TRIGGER SCOPE so that it is easy to identify in a list.
- Set the step time of the STEP block to the time you wish to begin saving data. Set the Initial Value and Final Value to 0 and 1 respectively. See Figure 4.
- Go to the TOOLS pull down menu and select EXTERNAL MODE CONTROL PANEL.
- Click on the SIGNALS AND TRIGGERING button
- Select the TRIGGER SCOPE block from the list and click on the TRIGGER SIGNAL button. This selects the input to the scope block as the trigger signal.

- In the TRIGGER area make sure that SOURCE = SIGNAL, MODE = NORMAL, DELAY = 0, and set DURATION =  $T_f/T_s$ . The duration is the number of samples allotted in memory for each signal being logged. If we have a simulation time of  $T_f = 10$  seconds and our sampling time  $T_s = 0.001$ , the  $10/0.001 = 10,000$  samples. If you actually run the model longer than  $T_f$ , then the data in memory will be overwritten when  $T_f/T_s$  samples have passed.

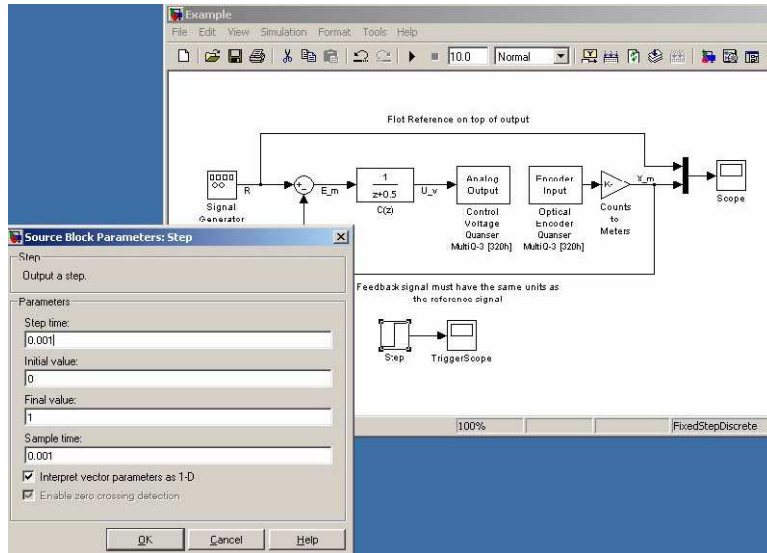


Figure 4: Step function block as a Trigger Signal

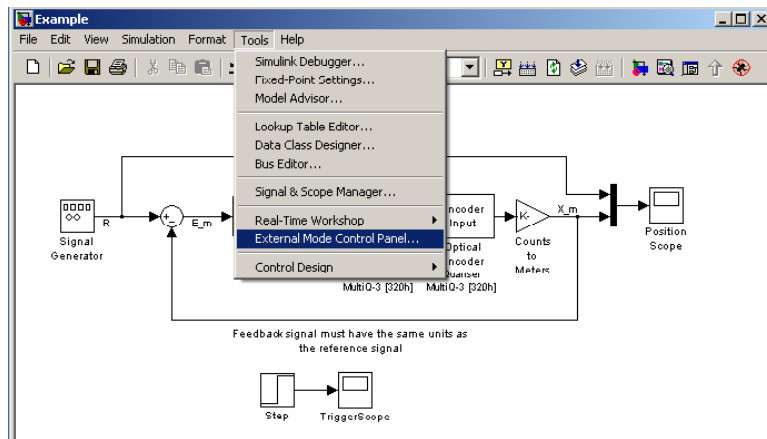


Figure 5: External Mode Control Panel

- In the TRIGGER SIGNAL area make sure that DIRECTION = EITHER or RISING, and LEVEL = 0.5. Our STEP signal will transition from 0 to 1 at 0.001 seconds, and is a rising edge. When it crosses the value 0.5 this will initiate the signal logging.
- Open all SCOPE blocks and configure them in the following way.

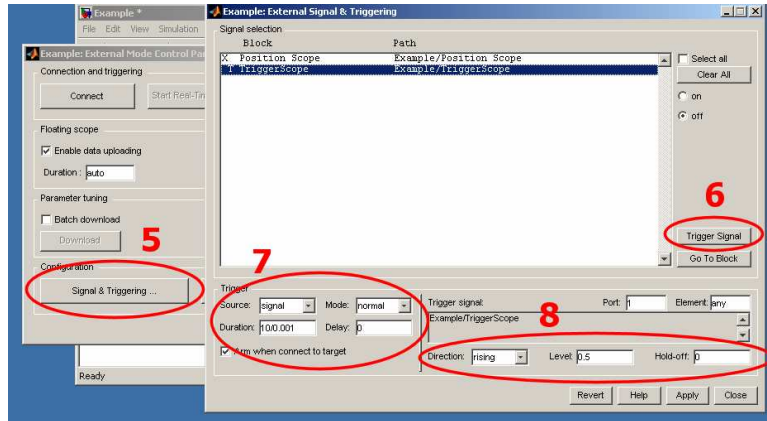


Figure 6: Configuring the Trigger Signal

10. Click the SCOPE PROPERTIES Icon, and select the DATA HISTORY tab
11. Make sure Limit data points to last is UNCHECKED
12. If you wish to save the data gathered by the scope, CHECK the Save data to workspace box, and give the Variable name a unique and meaningful name for easy reference later. In the FORMAT menu choose the ARRAY option.
13. Hint : Once you have set these parameters as you like, you can RIGHT CLICK on a scope block, drag and drop, and a copy of that block will be placed where you let go of the mouse. The new block will have all of the parameters of the previous, except the Variable name, that will be different and should be changed to something meaningful.

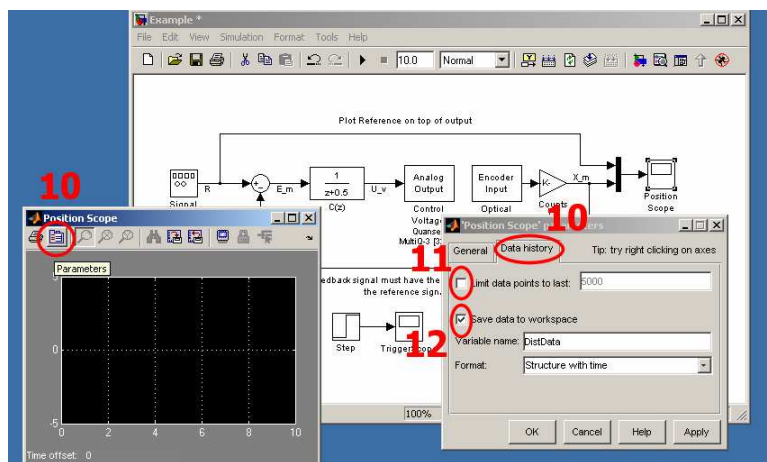


Figure 7: Configuring Scope blocks for data logging

## 1.2.2 Real Time Windows Target Simulation Parameters

You must configure the way that the RTW builds its real time executable C-code. Real time Simulink models can run only with a Fixed Step Size. The Target Language Compiler must also be set to Real Time Windows Target so that the generated code is made for running on a PC which is running Windows and Simulink on the same machine. To do this,

1. Click on the SIMULATION pull down menu, and select CONFIGURATION PARAMETERS.

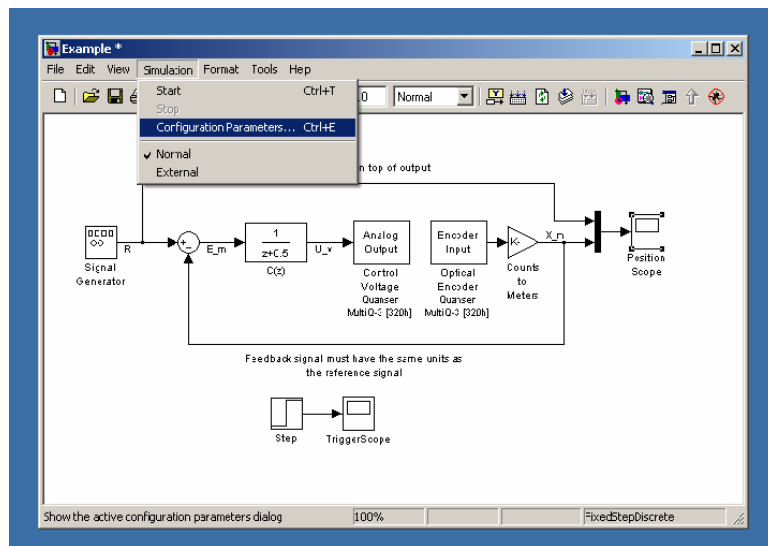


Figure 8: Configuration Parameters

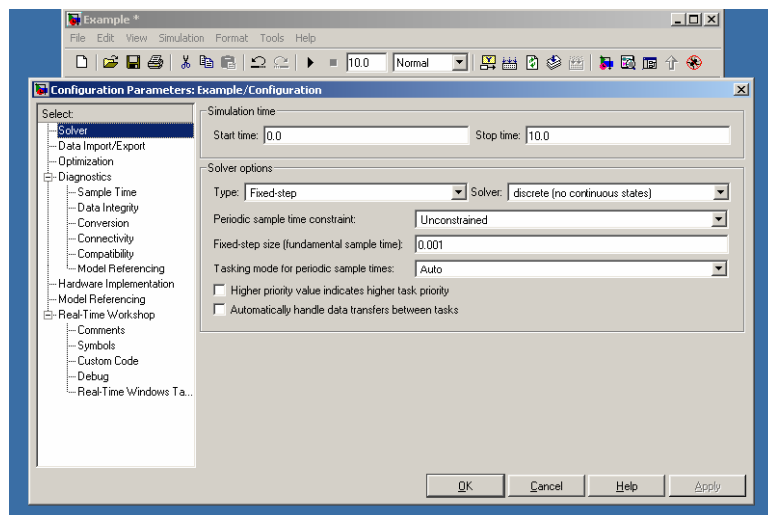


Figure 9: Solver settings

2. On the left side panel select SOLVER, and in the SOLVER OPTIONS area, select FIXED STEP for the solver TYPE. Choose an appropriate SOLVER for your model.

If all blocks use discrete time blocks, then the discrete (no continuous states) is a good choice. If you are implementing a continuous time controller for example with a Laplace transform  $C(s)$ , then you need to choose a differential equation solver like Ode4.

3. Set your fixed step size for the model, we usually use 0.001.
4. Next, select the REAL-TIME WORKSHOP on the left.
5. In the TARGET SELECTION area, click BROWSE, and scroll down to the bottom, and select rtwin.tlc from the menu. Verify that the other settings in the window match the picture in Figure 10.

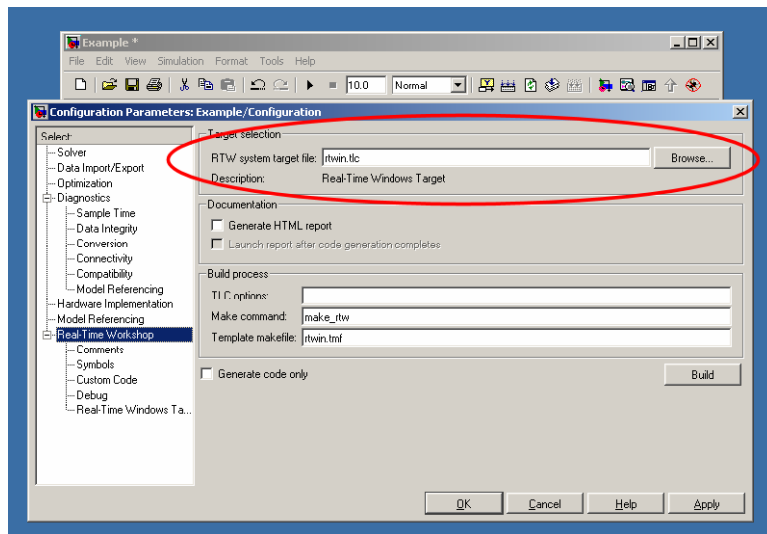


Figure 10: Real-Time Workshop Target selections

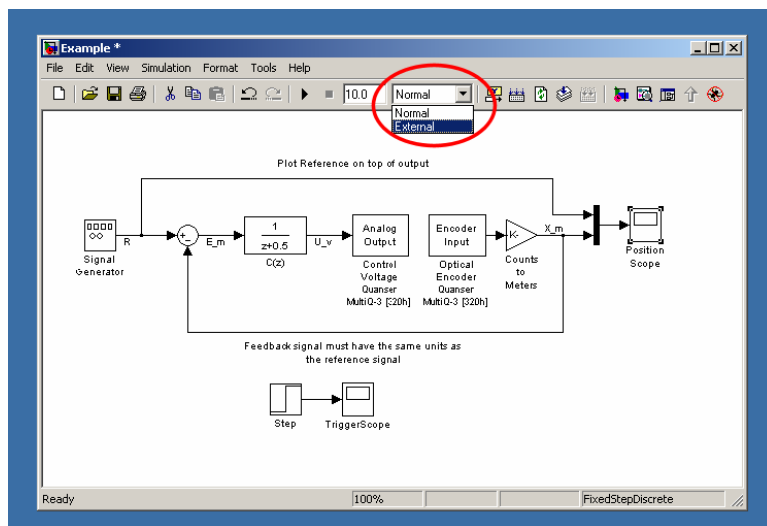


Figure 11: Choosing External Mode to run real time code and log data

6. This sets the windows machine as the target for the real time code.
7. Click APPLY and OK
8. Lastly, the model must be set to EXTERNAL mode. In the model toolbar to the right of the final simulation time and the play button, is a pull down menu that usually says NORMAL, click the menu and select EXTERNAL. See Figure 12.
9. The model must be BUILT first before it can be run. Building involves generating and compiling the real time code. To do this click on the TOOLS menu and under Real-Time Workshop select BUILD MODEL, or just hit CTRL+B.
10. The Matlab command window will show all of the steps of the build process and indicate if it completes successfully or not.
11. To run the model, you must connect to it by clicking the CONNECT TO TARGET icon to the right of the PLAY button in the models toolbar.

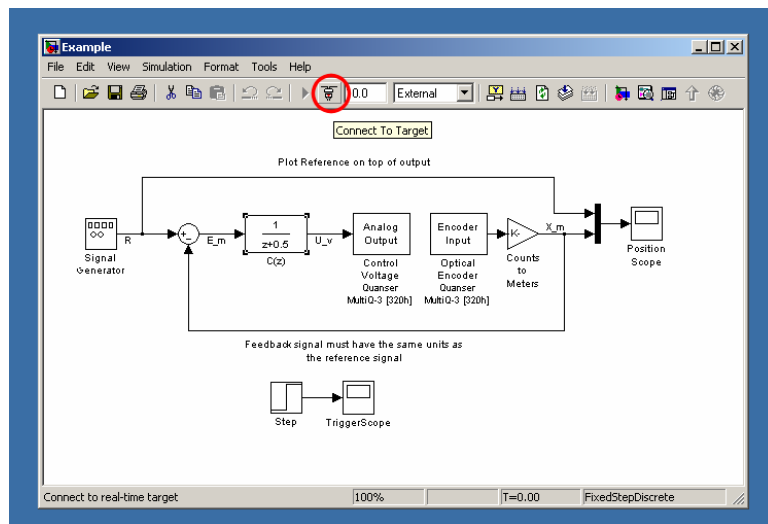


Figure 12: Connect to RT Windows Target

12. If everything is working correctly the model will start the Real Time Kernel which supersedes Windows and guarantees real time execution. Clicking on the PLAY button will run the model.
13. You will often get an error CHECKSUM MISMATCH. This means that something important in the model has changed since the last real time build, and the model must be built again. Hit CTRL+B. You will find yourself building these models often as you tune your Simulink diagrams.

A word of warning: RTW gives you the capability to change gain parameters in real-time. In other words, changing a gain value while the controller is running will *immediately* change that value. This should be done with great care as the equipment may be damaged by allowing the cart to run against the end of the track or by feeding high frequency signals to the motor.

### 1.3 Motor Cart Model

Once we understand how to interface MATLAB to “the real world” using RTW and the MultiQ-3 board, the main task during this week’s lab will be to control the motor cart system shown in Figure 13. Many future laboratory exercises will build upon this motor cart system.

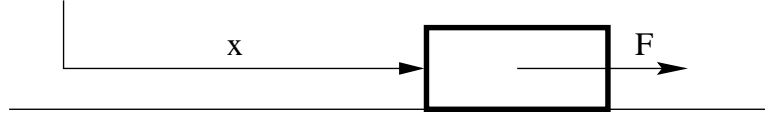


Figure 13: Free Body Diagram of Motor Cart System

The DC motor equations are given by

$$\begin{aligned} V &= I_m R_m + K_m K_g \omega_g \\ &= I_m R_m + K_m K_g \frac{\dot{x}}{r}, \end{aligned}$$

where  $V$ (volts) is the voltage applied to the motor,  $I_m$ (Amp) is the motor current,  $K_m$ ( $\frac{\text{volts}}{\text{rad/sec}}$ ) is the back EMF constant,  $K_g$  is the gear ratio in the motor gearbox,  $\omega_g$ ( $\frac{\text{rad}}{\text{sec}}$ ) is the motor output angular velocity,  $\dot{x}$ ( $\frac{\text{m}}{\text{sec}}$ ) is the cart velocity, and  $r$ (m) is the radius of the motor pinion that meshes with the track. The torque  $\tau$  generated by the motor is

$$\tau = K_m K_g I_m,$$

which is transmitted to the cart via the pinion by

$$F = \frac{\tau}{r}.$$

This force results in an acceleration of the cart governed by Newton’s second law,

$$F = m_c \ddot{x},$$

where  $m_c$  is the mass of the cart. Eliminating  $F$ ,  $\tau$  and  $I_m$  by combining the above equations and taking the Laplace transform yields

$$\frac{X(s)}{V(s)} = \frac{K_m K_g r}{s(m_c R_m r^2 s + (K_m K_g)^2)}.$$

If we define our plant as  $P(s) = \frac{X(s)}{V(s)}$ , and substitute numerical values for the constants in the above equation, we conclude

$$P(s) = \frac{3.78}{s(s + 16.9)}. \quad (1)$$

Equation 1 describes the linear position servo which we will control in this lab.

Parameter		Value	
Back EMF Constant	$K_m$	0.00767	volts/(rad/sec)
Motor Resistance	$R_m$	2.6	$\Omega$
Gear Ratio	$K_g$	3.7	
Pinion Radius	$r$	0.00635	m
Mass of Cart	$m_c$	0.455	kg

## 2 Prelab Exercises

1. Work through the derivation of the model for the motor cart system. Show all intermediate steps.
2. Using SIMULINK, construct the block diagram shown in Figure 14 and use it to answer the following.

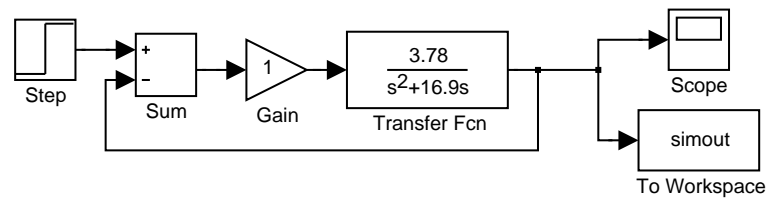


Figure 14: Proportional Control of Motor Cart

- (a) Find the gain  $K$  which has approximately 10% overshoot given a step input. The percentage overshoot  $M_p$  is defined as the maximum amount the system overshoots its final value divided by its final value (expressed as a percentage).
- (b) Plot (on the same well-labeled graph) the expected step response given the controller gain  $G = \{\frac{K}{2}, \frac{K}{4}, K, \frac{3K}{2}, 2K\}$ .

## 3 Experimental Procedure

### 3.1 Loopback Walk-through

This example will illustrate how SIMULINK, RTW, and the MultiQ-3 board interact. The example is a simple loopback system where a signal is generated from an analog output channel and then read in through an analog input channel. You need to connect analog output channel zero to analog input channel zero.

1. Change directories to your own directory.
2. Generate the SIMULINK diagram shown in Figure 15. We will generate a sinusoidal output and measure the voltage using an analog input channel.

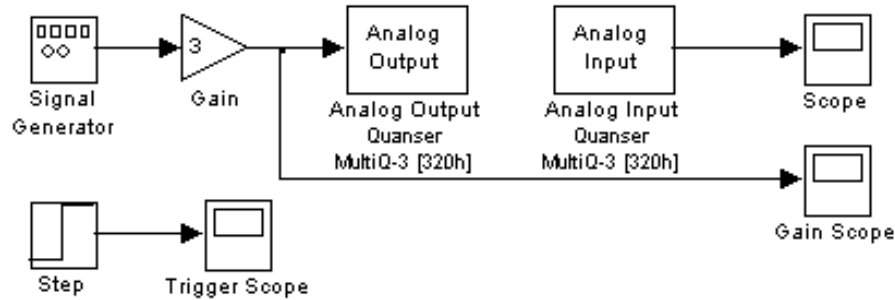


Figure 15: SIMULINK Loopback System

3. After creating the system diagram (and saving it as *lpbk.mdl*), build the model following steps 9-13 of 1.2.2. This generates the real-time code for the diagram. Wait until the compilation is complete. The MATLAB window displays the progress of the code generation task and, when it is complete, the following message appears:

```
### Successful completion of RTW build procedure for model : lpbk
```

4. To be able to run the model click **CONNECT TO TARGET**. Once the target is connected you should be able to run the model by clicking **PLAY**.
5. Select the data you wish to plot by following steps 9-13 in 1.2.1. In this case, we would like select *Gain Scope* and *Scope*. The output of *Gain Scope* is the output of analog output channel zero. The output of *Scope* is the voltage measured on analog input channel zero. This should display a clipped sine wave and a full sine wave. The clipped sine wave is the voltage measured at the analog input channel. Note that the clipping occurs since the analog input and output channels have a signal range of  $\pm 5$  volts.
6. Allow the plot to continue drawing and return to the SIMULINK diagram. Click on the *Gain* block and change the gain value to 3. Note that the measured sine wave is no longer clipped and that the input exactly matches the output. **Note also that you were able to change a gain in real-time!** As previously mentioned, when implementing controllers on the experimental equipment, this should be done carefully and you should be prepared to shut the controller off if the unexpected occurs.

## 3.2 Motor Cart Experiment

Implement constant gain control for the motor cart system by:

1. Locate the MultiQ-3 Board, the Universal Power Module (UPM) and the motor cart. **Ensure that the Universal Power Module is off (no red light)**. Do not turn on the UPM until the TA has checked your design and has given approval.
2. Attaching the necessary wires.

- (a) Connect an analog output of the MultiQ-3 board (RCA) to the UPM “From D/A” (5 pin DIN).
  - (b) Connect the UPM “To Load” (6 pin DIN) to the motor cart’s motor (4 pin DIN).
  - (c) Connect the optical encoder which measures cart position (5 pin DIN) to one of the MultiQ-3 Encoder Inputs (5 pin DIN).
3. Build the simulink diagram. Start with constant gain  $G = 0$ .
  4. Build and start your code (with the UPM off and  $G = 0$ ). Create a plot to look at the measured cart position. Move the cart with your hand. The output is an integer corresponding to counts. There are a certain number of counts/revolution of the optical encoder’s shaft. Find a method to convert counts to meters. You may need to change the simulink diagram and rebuild.
  5. Prepare your diagram so that the motor cart will follow a **0.1 meter** step input. Start with gain  $G = \frac{K}{2}$  from the prelab exercises. Build your diagram but do not start it.
  6. **Obtain permission from the TA to continue.**
  7. Turn on the UPM and place the motor cart in the center of the track.
  8. Start the model and watch your controller work. Be prepared to stop if something unexpected occurs.
  9. Compare the actual and expected response of the system.