Lab 2: Quanser Hardware and Proportional Control

I. Objective

The goal of this lab is:

a. Familiarize students with Quanser's QuaRC tools and the Q4 data acquisition board.

b. Derive and understand a model for the dynamics of the cart (minus the pendulum).

c. Use proportional control to generate a step response on the actual hardware.

II. Equipment

– Cart system (no attachments) and power supply.

III. Theory

1. Real Time Workshop, QuaRC, and the Q4 DAQ board

Real Time Workshop is a MATLAB toolbox that enables the user to generate and execute stand-alone C code for developing and testing algorithms modeled in Simulink.

QuaRC is Quanser's rapid prototyping and production system for real-time control. QuaRC integrates seamlessly with Simulink to allow Simulink models to be run in real-time on Windows. It uses a **host** and **target** relationship that allows code generation and execution to occur on separate machines. However, we will be using "Single User Mode" or "Local Configuration," where we will be generating and executing code on the same computer, as shown below.

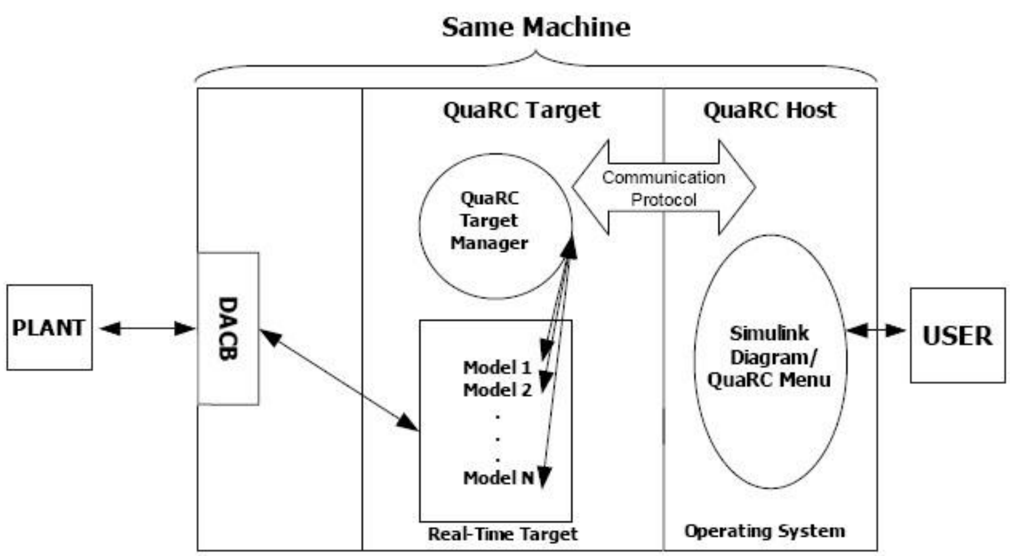

Figure 1: Local Configuration – QuaRC Host and Target on the same PC.

The **QuaRC Simulink Development Environment (SDE)** is used to generate/build code to be later run on a real-time target from MATLAB/Simulink models. The **QuaRC Windows Target** feature is required to run the generated code from MATLAB/Simulink models on a real-time Windows target (local or remote). QuaRC Windows Target needs to be open to run any QuaRC-generated code.

The functionality of QuaRC is transparent to the user, your task is to just design the controller based on either classic or state-space techniques. Then you implement the controller in Simulink via Real-Time Workshop. This is then "downloaded" to the QuaRC target.

But, how do you interface your controller with the plant? The answer is the **Q4 Data Acquisition (DAQ) board**. This board supports 4 A/D converters, 4 D/A converters, 16 Digital I/Os, 2 Realtime clocks, and up to 4 Quadrature input decoders/counters.

The Q4 board's functionality has also been abstracted from the user. The board has been set up to work with the cart and pendulum for all the station. DO NOT CHANGE ANY OF THE HARDWARE AT THE QUANSER STATIONS WITHOUT FIRST CONSULTING THE GSI!!!

2. Dynamics of the Cart

Figure 2 below shows the cart's free body diagram. For simplicity, we will ignore the effects of friction.

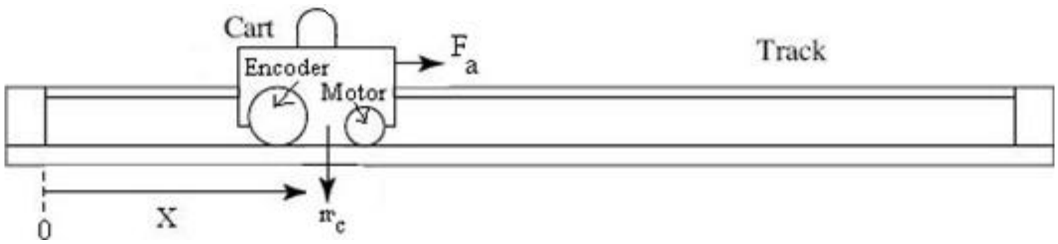

Figure 2: Pendulum system free body diagram (ignoring friction).

Parameters in Figure 2:

$\boldsymbol{F_a}$ is the input force exerted on the cart by the voltage applied to the motor.
$\boldsymbol{m_c}$ is the mass of the cart.
Encoder is used to keep track of the position of the cart on the track.

Using Figure 2 and basic Newtonian dynamics you can derive the equations governing the system.

3. Motor Dynamics

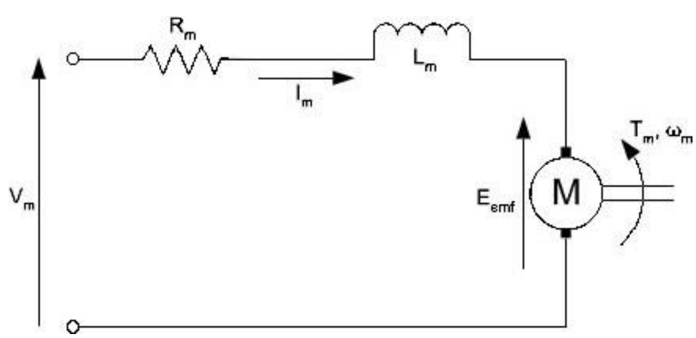

Figure 3: Classic armature circuit of a standard DC motor.

The input to your system is actually a voltage to the drive motor. Thus, you need to derive the dynamics of the system that converts the input voltage to force. These are the dynamics of the drive motor.

For this derivation we assume the following:

- $L_m \ll R_m$, so we disregard the motor inductance (treat as wire)
- Perfect efficiency of the motor and gearbox ($\eta_m = \eta_g = 1$)

The torque generated by the motor is proportional to the current flowing through the motor windings but is lessened due to the moment of inertia:

$$\tau = K_t I_m - J_m \ddot{\theta} \qquad (1)$$

$\boldsymbol{K_t}$ is the motor torque constant
$\boldsymbol{I_m}$ is the current flowing through the coil
$\boldsymbol{J_m}$ is the moment of inertia of the motor
$\boldsymbol{\ddot{\theta}}$ is the angular acceleration of the motor

Now, the current flowing through the motor can be related to the motor voltage input by:

$$V = I_m R_m + E_{emf} = I_m R_m + K_m \dot{\theta} \qquad (2)$$

$\boldsymbol{\dot{\theta}}$ is the angular velocity of the motor
$\boldsymbol{R_m}$ is the resistance of the motor windings
$\boldsymbol{K_m}$ is the back EMF constant $\left(\frac{V}{rad\ /sec}\right)$

Angular velocity is related to linear velocity and torque is also related to applied force.

$$K_g \tau = F_a \cdot r \qquad (3)$$
$$K_g \dot{x} = \dot{\theta} \cdot r \implies K_g \ddot{x} = \ddot{\theta} \cdot r \qquad (4)$$

$\boldsymbol{r}$ is the radius of the motor gear
$\boldsymbol{K_g}$ is the gearbox gear ratio

4. Step Response of a System

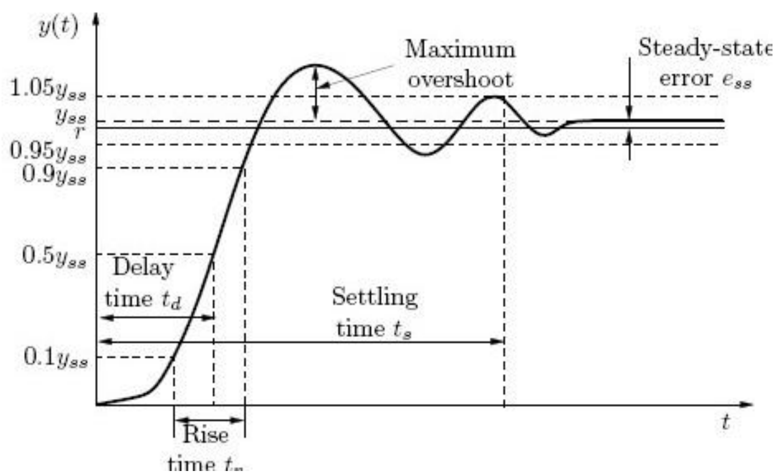

Figure 4: Typical step response of a control system.

Steady-state error: The *steady-state value* of the response $y(t)$ is defined as $\boldsymbol{y_{ss} \triangleq \lim_{t\to\infty} y(t)}$.

For a control system, we want the output, $y(t)$, to follow a desired *reference* signal, $r(t)$. Thus we can define the error as $\boldsymbol{e(t) \triangleq r(t) - y(t)}$.

and consequently, the *steady-state error* is given by $\boldsymbol{e_{ss} \triangleq \lim_{t\to\infty} e(t)}$.

Maximum overshoot: Let y_{max} denote the maximum value of $y(t)$. The *maximum overshoot* of the step response $y(t)$ is defined as **maximum overshoot** $\boldsymbol{\triangleq y_{max} - y_{ss}}$.

The maximum overshoot is often represented as a percentage of the steady-state value:

$$\text{percent maximum overshoot} = \frac{y_{max} - y_{ss}}{y_{ss}} \times 100\%$$

The maximum overshoot is often used to measure the relative stability of a system. A system with a large overshoot is usually undesirable.

Delay time: The *delay time* t_d is defined as the time required for the step response to reach 50% of its steady-state value.

Rise time: The *rise time* t_r is defined as the time required for the step response to rise from 10% to 90% of its steady-state value.

Settling time: The *settling time* t_s is defined as the time required for the step response to stay within 5% of its steady-state value.

IV. Prelab

1. Equations governing the cart dynamics

Derive the following equation of motion for the cart system shown in Figure 2.

$$\left(m_c r^2 R_m + R_m K_g^2 J_m\right)\ddot{x} + \left(K_t K_m K_g^2\right)\dot{x} = \left(r K_t K_g\right)V \qquad (5)$$

In the equation above:

$\boldsymbol{V}$ is the input voltage (volts)
$\boldsymbol{m_c}$ is the mass of the car (kilograms)
$\boldsymbol{r}$ is the radius of the motor gear (meters)
$\boldsymbol{R_m}$ is resistance of the motor windings (ohms)
$\boldsymbol{K_t}$ is the motor torque constant (N*m/A)
$\boldsymbol{K_m}$ is the back EMF constant (V*s/rad)
$\boldsymbol{K_g}$ is the gearbox gear ratio (no units)
$\boldsymbol{J_m}$ is the moment of inertia of the motor (kg*m^2)

In order to derive the equation (5), use the steps below:

Step 1. Applying Newton's second law to cart gives:

$$\sum F = m_c \ddot{x} \qquad (6)$$

Using the free body diagram of the cart from Figure 2 as a guide, replace F with the horizontal forces acting on the cart (Hint: Don't over think it).

Step 2. Substitute the motor dynamics:

Combine all the motor dynamics equations, equations 1-4, to obtain the relationship between the input voltage V and the applied force F_a. Substitute this relationship into your equation from Step 1. This is the final model of your plant.

Step 3. Is this system linear?

If not, linearize the system. If so, leave as is.

2. Derive system models

Transfer Function. Apply the Laplace transform to your linear system and solve for the transfer function $X(s)/V(s)$.

State Space. Given that the system output is the cart position, derive the state space matrices A, B, C, and D for your linear system.

SS to TF. Using the following equations, derive a transfer function from your state space matrices and verify that it matches the transfer function you got directly from taking the Laplace transform of the equation of motion.

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \tag{7}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{8}$$

3. MATLAB Step Response

Use the following values to create a Simulink block diagram of the cart system in a simple negative feedback loop with a gain K as the controller. It is your choice whether you want to use the state space or transfer function representation of the system.

$\boldsymbol{m_c}$ = 0.57 + 0.37 = 0.94 kg
$\boldsymbol{r}$ = 0.00635 m
$\boldsymbol{R_m}$ = 2.6 Ω
$\boldsymbol{K_t}$ = 0.00767 N*m/A
$\boldsymbol{K_m}$ = 0.00767 V*s/rad
$\boldsymbol{K_g}$ = 3.71
$\boldsymbol{J_m}$ = 3.9 x 10^{-7} kg*m^2

Vary the value of K until you achieve a **percent maximum overshoot < 4.5% and $\boldsymbol{t_r}$ < 0.5s**. You only need to find a single value that works, NOT a range of values. Also note that the constraints are NON-inclusive! Include your block diagram (and any code you used) as well as your final value of K and plots verifying these design conditions are met.

Note: You will find the MATLAB function **`find(cond,N)`** to be very useful for this. This returns at most the first N *indices* that match the condition `cond`. Type "`help find`" to read about the other various uses for this function if you wish. For example, for an array of output values `out` and time values `time`, you can use the following code to find the time of the first value of **`out`** that exceeds the value of 0.1:

```
>> find(out >= 0.1,1)
ans =
   110
>> time(110)
ans =
    1.0900
```

To get more precise time and output values, it is suggested that you set Simulink to a small, fixed-step interval. You are also welcome, but not required, to use linear interpolation between points.

V. Lab

1. Cart Dynamics

Confer with your group to agree on a system representation (either state space or transfer function) to use in this lab.

2. The Quanser Hardware

The GSI will give a demonstration of how to properly use the Quanser system. Make sure you understand the system functionality so you can implement your controller easily.

MAKE SURE TO OBSERVE SAFETY PRECAUTIONS AT ALL TIMES. Any group that leaves the amplifier on after they are finished using Quanser hardware will automatically lose 5 points from their lab report. This penalty will stack for every offense.

We have a limited number of working systems, so expect delays if you have to wait for a system to open up. Some parts of this lab do NOT require the hardware. Please work on those sections while waiting a station. Be respectful of the sharing situation. Any group that leaves one of the computers at the designated stations (204-02, 204-06, and 204-12) LOCKED, will automatically lose 5 points from their lab report. This penalty will stack for every offense.

3. Using the Actual Hardware – Find Encoder-Distance Conversion

The GSI will cover how to interface with the actual cart hardware by building a Simulink subsystem. Encoder values for the position of the cart will be read in encoder counts. Our input will be in inches. In feedback, the two values you compare MUST be in the same units, so **we need a conversion factor**.

- Build a simple Simulink file that does nothing but read the position encoder count. You can set the final time in Simulink to "inf" for infinity to run the program indefinitely. You can use QuaRC => Stop to stop the program. If you use a Scope, it will update in real time.
- Attached at the end of this lab is a paper ruler that you can print and cut out. Feel free to bring an actual ruler to lab if you so desire.
- Once you start running the QuaRC program, manually move the cart along the track and watch the encoder values update. Using a ruler, move the cart manually an inch and let it sit for a while. Then move the cart another inch and repeat this a couple of times.
- Using the plateaus in your plot (from letting the cart sit at a position for a while), calculate the change in encoder count from moving the cart an inch. Average your values over all the inch-length moves for better accuracy.
- Include the plot of the cart encoder values (not converted) vs. time in your lab report and document your calculation of the encoder counts/inch.

The Quanser manual gives the position encoder resolution to be 4096 counts/revolution. Given that the radius of the position pinion is r_{pp} = 0.01482975 m, what is the "actual" encoder resolution in count/inch? How close was your estimated encoder resolution?

4. Using the Actual Hardware – Cart Step Response

- Go back to your Simulink model of the cart system from the prelab. Now change the step function to be of height 6, corresponding to the cart moving 6 inches (watching it move 1 inch isn't very exciting). Again, try to find a value of K so that percent maximum overshoot < 4.5% and t_r < 0. 5s.

  Report this value in your lab report. How different is the value you found here from the value of K you found in the prelab for a step size of 1?

- Once the simulated step response looks fine, you can move over to the actual hardware. Replace your system block (ss or tf) in the feedback loop with the hardware subsystem, *with the counts/inch conversion included*. Run the QuaRC program.

  Plot the initial hardware response and compare with the plot from the Simulink model. How close was the actual to the predicted? What might have caused any discrepancies?

- Now change your value of K until you achieve a percent maximum overshoot of < 4.5%. Report your new K value and plot the hardware response.

  Show your modified hardware step response to the GSI before the end of the lab session or when you turn in your lab report.

VI. Revision History

Semester and Revision	Author(s)	Comments
Fall 2010 Rev. 2.3	Wenjie Chen, Jansen Sheng	Modified some questions and solutions from Fall 2009 lab
Fall 2009 Rev. 2.2	Justin Hsia	Re-did EOM derivation for cart to include moment of inertia (more accurate).
Winter 2008 Rev. 2.1	Justin Hsia	Reformatted, made corrections based on Fall 2008 student reactions.
Fall 2008 Rev. 2.0	Justin Hsia	Completely revamped lab. Kept parts of theory section, but updated for the new hardware and software.
Summer 2008 Rev. 1.0	Bharathwaj Muthuswamy	1. Formatted writeup into different sections. 2. Typed up solutions