ME C134 Lab 3:
Quanser Hardware and Proportional Control

Group: Trey Fortmuller, Charlene Shong, Jenny Wong

W 11am – 2pm Section; February 21, 2018

1 Purpose

The purpose of this lab is to demonstrate controller design methodology from start to finish. We started by deriving plant dynamics, modeling the system in state space and transfer functions, obtaining appropriate gains for a simple proportional controller through analysis of the closed loop poles in simulation, and finally hardware-in-the-loop tests using the Quanser QuaRC cart and DAQ board.

2 Pre-lab

2.1 Plant Dynamics

Initially, we are given Eq. 1-4, which describe the torque generated by the motor (Eq. 1), the current-voltage relationship (Eq. 2), the relationship between the torque and the applied force (Eq. 3), and the relationship between the angular and linear velocities of the cart (Eq. 4).

$$T_m = K_T I_m - J_m \ddot{\theta} \tag{1}$$

$$V = I_m R_m + K_m \dot{\theta} \tag{2}$$

$$K_g T_m = F_a \cdot r \tag{3}$$

$$K_g \dot{x} = \dot{\theta} \cdot r \longrightarrow K_g \ddot{x} = \ddot{\theta} \cdot r \tag{4}$$

Based on the Fig. 1, which depicts the free body diagram of the given cart system, we can derive Eq. (5) from Newton's Second Law of Motion.

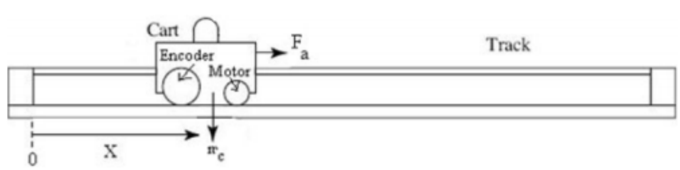


Figure 1: Free Body Diagram of the Cart (Ignoring Friction)

$$F_a = m_c \ddot{x} \tag{5}$$

We substitute Eq. 1-4 into Eq. 5 and simplify

$$F_a = \frac{k_g}{r}\left[K_t I_m - J_m\ddot{\theta}\right]$$

$$m_c\dot{x} = \frac{K_g}{r}\left[K_t\left[\frac{v - K_m K_g \dot{x}/r}{R_m}\right] - \frac{J_m K_g}{r}\ddot{x}\right]$$

$$m_c\ddot{x} = \frac{K_g K_t}{rR_m}V - \frac{K_t K_m G_g^2}{r^2 R_m}\dot{x} - \frac{J_m K_g^2}{r^2}\ddot{x}$$

Eliminating the denominators, we arrive at the full equation of motion of the cart, shown in Eq. (6). We also note that the system is linear.

$$(m_c r^2 R_m + R_m K_g^2 J_m)\ddot{x} + (K_t K_m K_g^2)\dot{x} = (rK_t K_g)V \tag{6}$$

2.2 System Models

From our linear system, we are able to derive both the Transfer Function of the cart system, shown in Eq. 7. We take the Laplace transform of Eq. 6 and rearrange.

$$\mathcal{L}\left[(m_c r^2 R_m + R_m K_g^2 J_m)\ddot{x} + (K_t K_m K_g^2)\dot{x} = (rK_t K_g)V\right]$$

$$(m_c r^2 R_m + R_m K_g^2 J_m)s^2 X(s) + (K_t K_m K_g^2)sX(s) = (rK_t K_g)V(s)$$

$$H(s) = \frac{X(s)}{V(s)} = \frac{rK_g K_t}{(m_c r^2 R_m + R_m K_g^2 J_m)s^2 + (K_t K_m K_g^2)s} \tag{7}$$

To arrive at the State Space Model (shown in Eq. 8-9) we arrange Eq. 6 in terms of $\ddot{x}$ and form the matrices.

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

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

$$\ddot{x} = \frac{rK_t K_g}{m_c r^2 R_m + R_m K_g^2 J_m}V - \frac{K_t K_m K_g^2}{m_c r^2 R_m + R_m K_g^2 J_m}\dot{x}$$

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & \frac{-K_t K_m K_g^2}{J_m K_g^2 R_m - r^2 R_m m_c}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ \frac{rK_t K_g}{J_m K_g^2 R_m - r^2 R_m m_c}\end{bmatrix}V \tag{8}$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{9}$$

2.3 SS to TF

Now we check to make sure we arrive at the original transfer function when we convert our state space model to a transfer function through Eq. 10.

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \tag{10}$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \left[\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & \frac{-K_t K_m K_g^2}{J_m K_g^2 R_m - r^2 R_m m_c} \end{bmatrix} \right]^{-1} \begin{bmatrix} 0 \\ \frac{r K_t K_g}{J_m K_g^2 R_m - r^2 R_m m_c} \end{bmatrix} \tag{11}$$

We employ MATLAB for this matrix multiplication (see Appendix for the full code listing), and arrive back at our original transfer function in Eq. 7. which we derived directly from the plant dynamics.

$$H(s) = \frac{Y(s)}{U(s)} = \frac{r K_g K_t}{(m_c r^2 R_m + R_m K_g^2 J_m)s^2 + (K_t K_m K_g^2)s}$$

2.4 Simulink Model

Then, using given values for our system parameters, we created a Simulink block diagram of the cart system in a simple negative feedback loop with a gain K as the (proportional) controller in order to achieve a percent percent maximum overshoot $< 4.0\%$ and rise time $t_r < 1.5s$. Our block diagram, shown in Fig. 2, utilized the transfer function representation of the system. After testing several K values, we noted that $K = 12$ was able to achieve the desired step response, so Fig. 3 illustrates the Step Response of the cart system at $K = 12$.

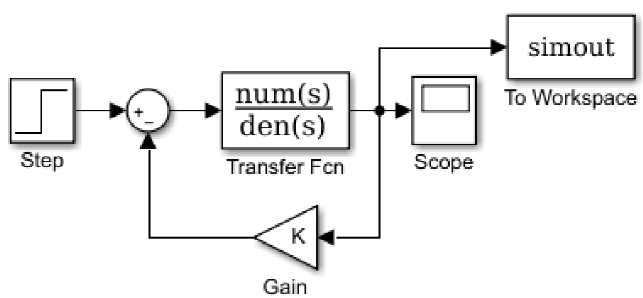


Figure 2: Simulink Model of proportional controller for the cart system

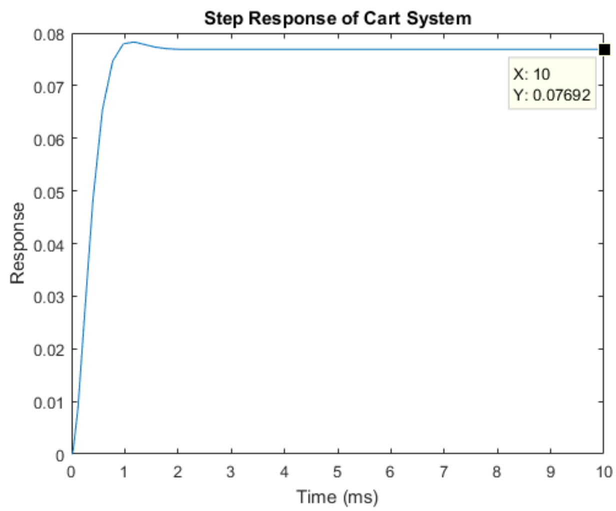


Figure 3: Plot of Step Response of the cart for K = 12

Figure 4 includes the MATLAB `stepinfo` of the simulation, which shows that the Overshoot is $1.7894\% < 4.0\%$ and RiseTime is $0.5501s < 1.5s$ for K = 12.

```
>> prelab3code

ans =

  struct with fields:

        RiseTime: 0.5501
    SettlingTime: 0.8158
     SettlingMin: 0.0747
     SettlingMax: 0.0783
       Overshoot: 1.7894
      Undershoot: 0
            Peak: 0.0783
        PeakTime: 1.1722
```

Figure 4: Step info

3 Lab

3.1 Encoder-Distance Conversion

First, we had to determine the relationship between encoder counts (referred to as "ticks" in our code and plots) and meters, so that the physical position of the cart could be accurately represented in our Simuink model. To do this, we created a Simulink model based on given instructions from the GSI which allowed us to connect to the QuaRC board and utilize its Data Aquisition functionality; the Simulink model was fairly simple (see Fig. 5), only requiring a Clock block and a "HIL Read Encoder" block.

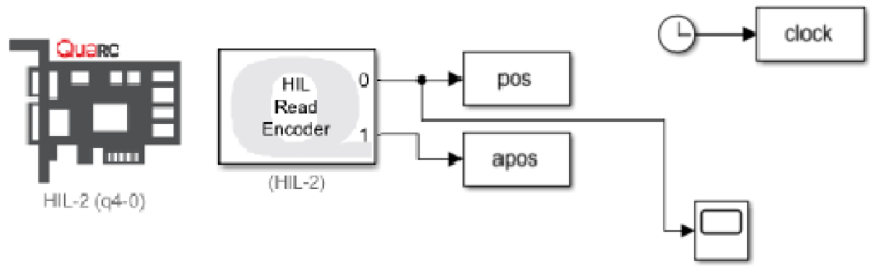
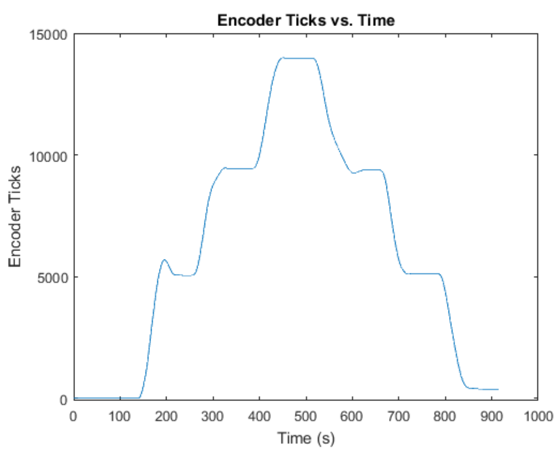
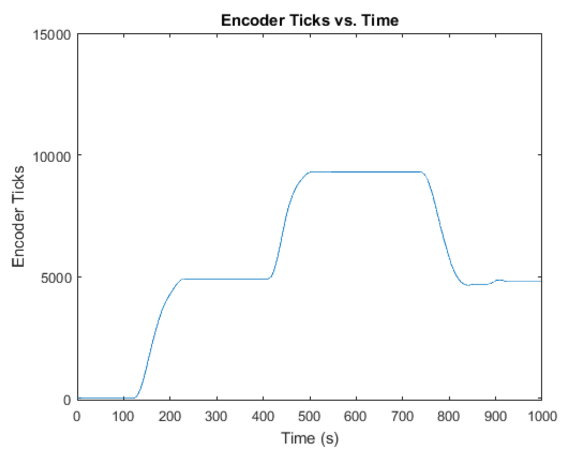


Figure 5: Simulink Model for Reading from Linear Encoder with QuaRC

Once the program was running, we manually moved the cart to specific distances (we chose to do ”steps” of 10 cm or 0.1 m), allow it to sit for a while so we can tell what value it is reading, then repeat at different values. We were able to generate two separate plots of moving the cart 10 cm successively, as shown in Fig. 6. From these, we determined that (at least for our encoder) the ”actual” encoder resolution was 44,000 counts/m (which is only a 0.093627% error from the theoretical value of 4096 counts/revolution or 43,958 counts/m).



(a) Run 1



(b) Run 2

Figure 6: Plots of Linaer Encoder Ticks vs. Time for Two Separate Runs

3.2 Cart Step Response

3.2.1 Simulated Step Response

First, we changed the step function height of our pre-lab Simulink model to be 0.15 (which would correspond to the cart moving 0.15m) and found that for a gain K value of 15, the percent maximum overshoot $< 4.0\%$ and rise time $t_r < 1.5s$.

```
ans =

  struct with fields:

       RiseTime: 0.4854
   SettlingTime: 1.1976
    SettlingMin: 0.0098
    SettlingMax: 0.0103
      Overshoot: 3.1600
     Undershoot: 0
           Peak: 0.0103
       PeakTime: 0.8635
```

(a) Step Height = 0.15

```
ans =

  struct with fields:

       RiseTime: 0.4836
   SettlingTime: 1.1959
    SettlingMin: 0.0644
    SettlingMax: 0.0688
      Overshoot: 3.1360
     Undershoot: 0
           Peak: 0.0688
       PeakTime: 1.0362
```

(b) Step Height = 1

Figure 7: Plots of Linear Encoder Ticks vs. Time for Two Separate Runs

Comparing the `stepinfo` of the system for the different step heights (see Fig. 7) to our data from the pre-lab for K = 15, we note that though the settling time is lower for a larger step height, the rest of the values (percent overshoot, rise time, etc.) are actually higher for a smaller step height of 0.15 than for 1.

3.2.2 Actual Hardware

In order to use the cart, we had to adjust our Simulink model to include the system block (either the State Space or Transfer Function) in the feedback loop with the hardware subsystem (the QuaRC blocks) and the counts/m conversion. Furthermore, including a saturation block to limit the input voltage to the motor to ±6V was necessary, as shown in Fig. 8.

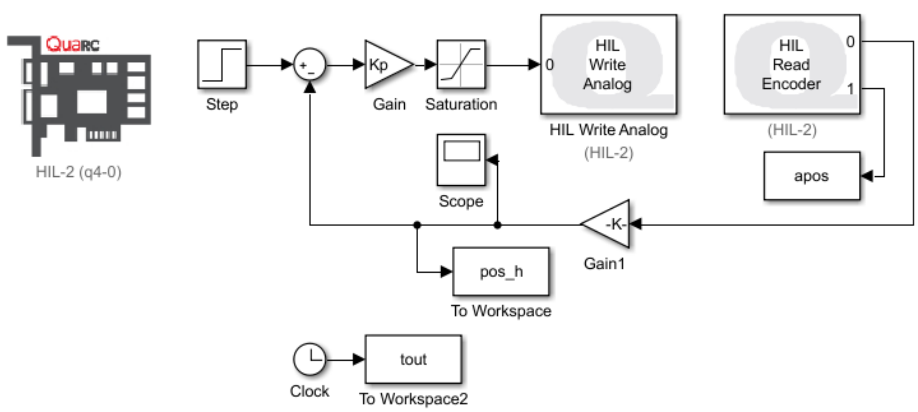
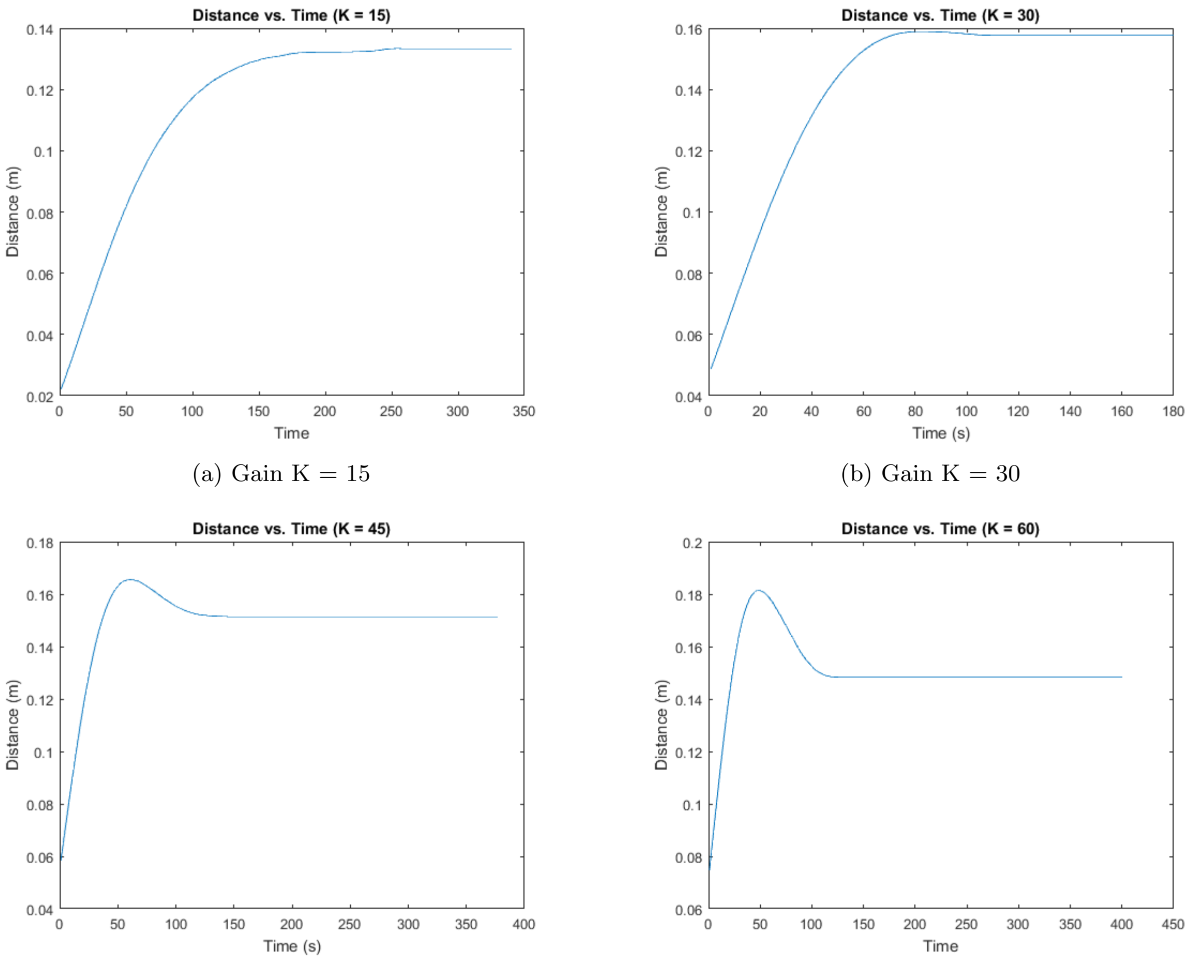


Figure 8: Simulink Model for Proportional Control of the Actual Cart System with Encoder Conversion

Now we adjust K and analyze the `stepinfo` output of the tests. We find that a gain of 30 satisfies the design constraints with a substantial margin but a gain of 15 isn't satisfactory. This discrepancy likely arises due to the fact that we didn't model friction or motor inductance and capacitance, so a higher gain is required for the test rig because all these effects slow the response compared to our simple dynamics model.



(a) Gain K = 15

(b) Gain K = 30

(c) Gain K = 45

(d) Gain K = 60

Figure 9: Plots of Cart System Step Response for Variable K of Proportional Controller

With a gain of 45 we are able to satisfy the faster design requirement of $t_r < 0.3$s and $\%OS < 12\%$. This was not possible in your Simulink model from the pre-lab, since efforts to tune our K would see lower t_r but $\%OS$ would shoot up. The closest we were able to get to barely meeting the system requirements was somewhere between K = 25 and K = 26 (as shown in Fig. 11). Our Simulink model from the pre-lab doesn't have as accurate of a model of the physical system as the hardware provides and also operates without the extra encoder conversion and saturation blocks so these could be reasons why the results are different.

```
ans =

  struct with fields:

        RiseTime: 0.5178
    SettlingTime: 0.9914
     SettlingMin: 0.1223
     SettlingMax: 0.1334
       Overshoot: 0.0170
      Undershoot: 0
            Peak: 0.1334
        PeakTime: 1.4150
```

(a) Gain K = 15

```
ans =

  struct with fields:

        RiseTime: 0.2350
    SettlingTime: 0.4801
     SettlingMin: 0.1471
     SettlingMax: 0.1589
       Overshoot: 0.8073
      Undershoot: 0
            Peak: 0.1589
        PeakTime: 0.5600
```

(b) Gain K = 30

```
ans =

  struct with fields:

        RiseTime: 0.1383
    SettlingTime: 0.6945
     SettlingMin: 0.1428
     SettlingMax: 0.1655
       Overshoot: 9.3079
      Undershoot: 0
            Peak: 0.1655
        PeakTime: 0.4400
```

(c) Gain K = 45

```
ans =

  struct with fields:

        RiseTime: 0.0850
    SettlingTime: 0.7008
     SettlingMin: 0.1412
     SettlingMax: 0.1814
       Overshoot: 22.1678
      Undershoot: 0
            Peak: 0.1814
        PeakTime: 0.4050
```

(d) Gain K = 60

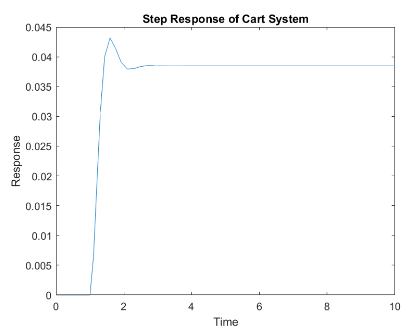
Figure 10: MATLAB `stepinfo` of Cart System Step Response for Variable K of Proportional Controller

```
ans =

  struct with fields:

        RiseTime: 0.2970
    SettlingTime: 1.9088
     SettlingMin: 0.0379
     SettlingMax: 0.0432
       Overshoot: 12.2454
      Undershoot: 0
            Peak: 0.0432
        PeakTime: 1.5908
```

(a) Step Info for K = 26



(b) Step Height = 0.15 and K = 26

Figure 11: Plots of Pre-lab Simulink Model for Step Height = 0.15 and K = 26

4 Appendices

4.1 Appendix A: Pre-lab MATLAB Code

```
%% Pre-lab MATLAB Step Input
%Charlene Shong

% System Parameters
mc = 0.94; %kg
r = 6.36e-3; %m
Rm = 2.6; %ohm
Kt = 7.67e-3; %Nm/A
Km = 7.67e-3; %Vs/rad
Kg = 3.71;
Jm = 3.9e-7; %kg*m^2

% Set Gain
K = 12;

%% Simulink Model
sim('prelab3.mdl');
open('prelab3.mdl');
plot(simout);xlabel('Time (ms)');ylabel('Response');
title('Step Response of Cart System');
stepinfo(simout.data,simout.time)
```

4.2 Appendix B: Lab MATLAB Code

```
%% Lab 3

%% Pre-lab MATLAB Step Input
%Charlene Shong

% System Parameters
mc = 0.94; %kg
r = 6.36e-3; %m
Rm = 2.6; %ohm
Kt = 7.67e-3; %Nm/A
Km = 7.67e-3; %Vs/rad
Kg = 3.71;
Jm = 3.9e-7; %kg*m^2

% Set Gain
K = 15;

% Simulink Model
sim('prelab3.mdl');
open('prelab3.mdl');
plot(simout);xlabel('Time (ms)');ylabel('Response');
title('Step Response of Cart System');
stepinfo(simout.data,simout.time)
close('prelab3.mdl');

%% 5.3
r = 0.01482975; % pinion radius
res = 4096; % counts/rev
res_t = res/(2*pi*r); % theoretical encoder resolution
res_a = 44000; % actual encoder resolution from calculations
err = abs(res_t - res_a)/res_t*100; % percent error
```

```
disp(['actual encoder resolution:',num2str(res_a)])
disp(['percent error from theoretical:',num2str(err)])

%% 5.4
Kp = 60;
```