

**School of Engineering
University of California, Merced**

ME 141 - Control Engineering

Experiment No. One

Linear Servo Base

Objectives

1. Learn how to derive a transfer function to describe the linear motions of a cart.
2. Gain hands-on experience with experiments and Quanser data acquisition systems.
3. Learn how to use Matlab for post-processing experimental data.

Background

Modeling the System

The equations of motion for the cart are given by

$$M\dot{v}_c(t) + B_{eq}v_c(t) = A_m v_m(t) \quad (1)$$

where M is the mass of the cart, v_c is the linear velocity of the cart, v_m is the motor input voltage, and B_{eq} is the equivalent damping coefficient. The equation of motion and the variables B_{eq} and A_m are derived from in the Appendix. By taking the Laplace transform, we can obtain the following first-order transfer function for the linear velocity of the IP02 cart with respect to the input motor voltage:

$$\frac{V_{cart}(s)}{V_{motor}(s)} = \frac{K}{\tau s + 1} \quad (2)$$

where V_{cart} is the Laplace transform for the cart's speed [$v_c(t)$], V_{motor} is the Laplace transform for the motor input voltage [$v_m(t)$], K is the steady-state gain, τ is the time constant and s is the Laplace operator.

(b) Position Control Response

For a position control feedback system, we can design the system as shown below, $Y(s)$ being the Laplace transform of our measured output, namely the position of the cart.

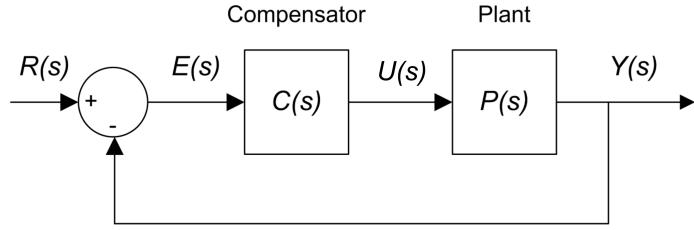


Figure 1: Unity Feedback System

Then $R(s)$, $E(s)$, and $U(s)$ be the Laplace transforms of the reference signal $r(t)$, error $e(t) = r(t) - y(t)$, and the input to the plant $u(t)$.

The output of this system can be written as

$$Y(s) = C(s)P(s)(R(s) - Y(s)) \quad (3)$$

We can rewrite this as a closed-loop transfer function by solving for $Y(s)$

$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \quad (4)$$

To get the voltage-to-position transfer function $P(s) = Y(s)/U(s)$, we introduced a integrator $\frac{1}{s}$ in series with the voltage-to-speed transfer function from equation 2 (effectively integrating the speed to get position). The result is

$$P(s) = \frac{Y(s)}{U(s)} = \frac{K}{s(\tau s + 1)} \quad (5)$$

We will use proportional-velocity (PV) control law for our control input $u(t)$. This has the form:

$$u(t) = k_p [r(t) - y(t)] - k_d \dot{y}(t) \quad (6)$$

Taking the Laplace transform allows use to represent the input signal $U(s)$ in terms of the reference $R(s)$ and output $Y(s)$:

$$U(s) = k_p [R(s) - Y(s)] - k_d s Y(s) \quad (7)$$

where k_p is the proportional gain, k_d is the derivative gain, $r = x_d(t)$ is the set-point (desired) cart position, $y = x_a(t)$ is the measured cart position, and $u = v_m(t)$ is the control input or IP02 motor input voltage. Figure 2 shows the block diagram representation for the PV control scheme. Note that the proportional term is based on the error ($e = r - y$) while the derivative term is based only on the derivative of the output (position, y). Note also that if the the reference cart position r is kept constant, then the derivative of the error is equal to the derivative of the output.

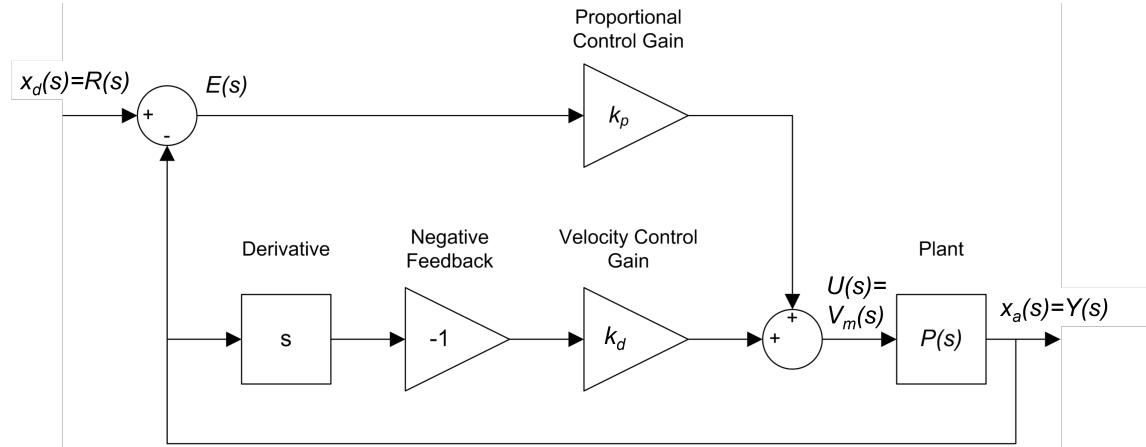


Figure 2: Block diagram of PV control

We can use equations 5 and 7 to solve for the closed-loop transfer function, which gives:

$$\frac{Y(s)}{R(s)} = \frac{Kk_p}{\tau s^2 + (1 + Kk_d)s + Kk_p} \quad (8)$$

Since the plant is a second-order system, introducing the integrator to the feedback system can help us to rewrite the closed-loop transfer function in a general form as:

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (9)$$

where ω_n is the natural frequency and ζ is the damping ratio. We can calculate the values of ω_n and ζ necessary for us to achieve our desired time-domain specifications. The desired time-domain specifications for controlling the position will be:

- peak time, $t_p = 0.15s$
- percent overshoot, $PO = 10\%$
- steady-state error, $e_{ss} = 0$

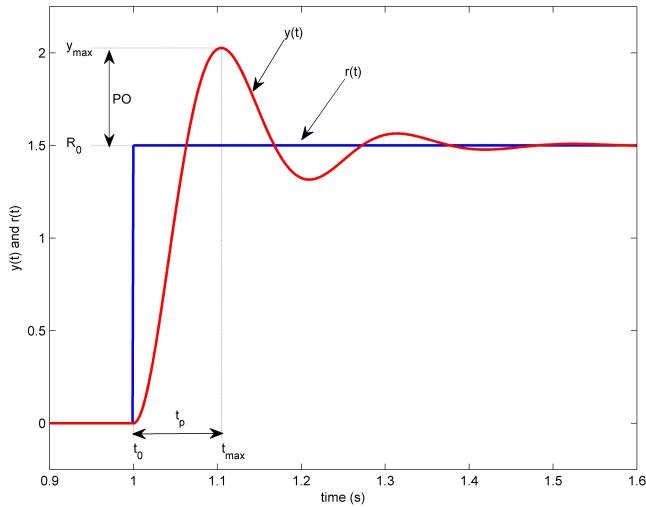


Figure 3: Standard Second-Order Step Response

Percent overshoot (PO in figure 3) is defined as:

$$PO = \frac{100(y_{max} - R_0)}{R_0} \quad (10)$$

where y_{max} is maximum output value and R_0 is the magnitude of the reference step input. Percent overshoot is dependent on the damping ratio, ζ , and can be expressed as:

$$PO = 100 \exp \left\{ \left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}} \right) \right\} \quad (11)$$

Another quantity we want to control is the peak time, t_p , which is defined as:

$$t_p = t_{max} - t_0 \quad (12)$$

where t_{max} is the time at which the maximum output is reached and t_0 is the initial time at which the step input is applied. While the percent overshoot only depends on the damping ratio; the peak time (t_p in Figure 2) needs to include both the damping ratio and the natural frequency:

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (13)$$

Generally speaking, the damping ratio affects the shape of the response while the natural frequency affects the speed of the response.

The error of the system is given by the difference between the reference (desired cart position) and the output (actual measured cart position):

$$e(t) = r(t) - y(t) \quad (14)$$

Taking the Laplace transform gives:

$$E(s) = R(s) - Y(s) \quad (15)$$

The steady-state error, e_{ss} , is the error as time goes to infinity. Using the Final-Value Theorem, this is given by:

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) \quad (16)$$

Procedure

For the performed experiments in the instructions below, we will need to use Matlab 2019b (other versions will not connect to the QUARC interface!).

(a) Simulation

After opening Matlab, change the workspace directory to the folder on the Desktop labeled "ME141 Lab #1" > "Position Control". Open the Simulink file named *s_ip02_position*

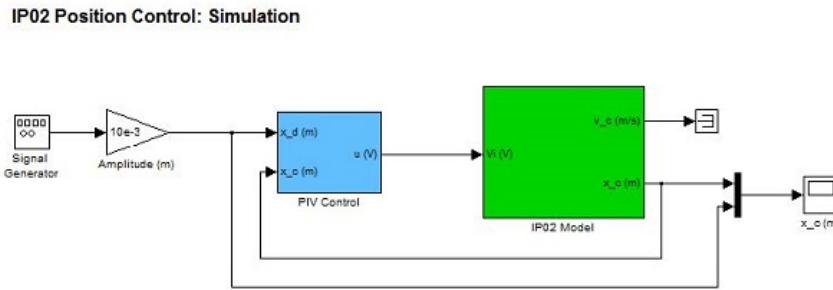


Figure 4: Simulink Block Diagram for Position Control Simulation

1. Run the *setup_ip02_position* script
2. To generate a step reference, verify the *Signal Generator* is set to the following:
 - Signal Type = *square*
 - Amplitude = 1
 - Frequency = 0.66 Hz
3. In the Simulink diagram, set the *Amplitude (m)* gain block to 0.01 to generate a step of 10 mm
4. Inside the *PV Control* subsystem, set the *Manual Switch* to the upward position so the *Derivative block* is used
5. Open the cart position scope, $x_c (m)$, and the motor input voltage scope, $V_m (V)$
6. Start the simulation and observe the response with the parameters assigned; take note of the colored lines in the $x_c (m)$ graph
 - Yellow Trace - set-point position

- Purple Trace - simulated position
7. Change the values for k_p and k_v and observe how the changes affect the response. Run the simulation with the k_p and k_v values that have the best response within the desired time-domain specifications.
 8. Export the data from the Matlab workspace variables to calculate the steady-state error, percent overshoot, and peak time:
 - $x_c (m)$: data_pos
 - * data_pos(:,1) = time
 - * data_pos(:,2) = simulated position
 - * data_pos(:,3) = set-point position
 - $data_{vm} (V)$: data_vm
 - * data_vm(:,1) = time
 - * data_vm(:,2) = simulated input voltage

(b) Position Control

Open the Simulink file named *q_ip02_position* to view the block diagram as seen below:

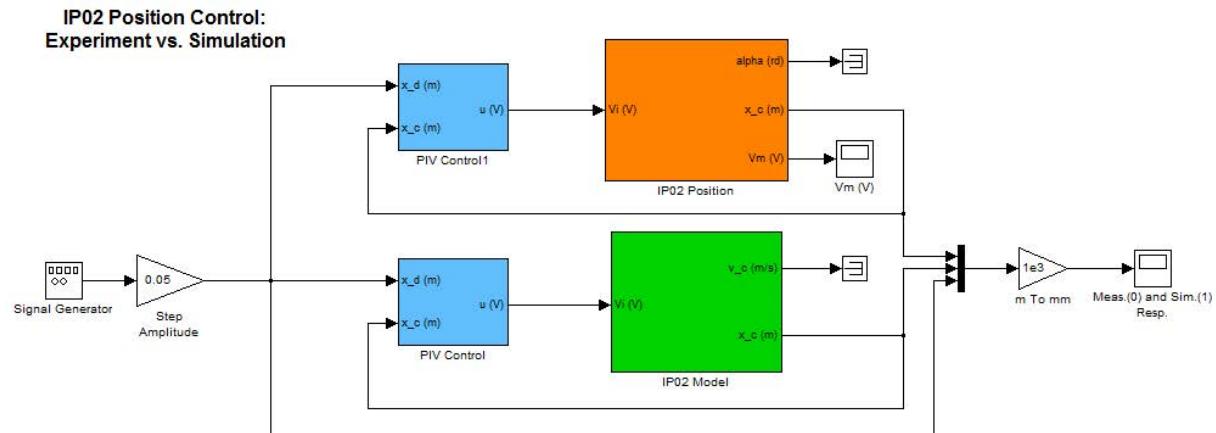


Figure 5: Simulink Block Diagram for Position Control

Closed-Loop Response with the PV Controller

1. Run the *setup_ip02_position* script
2. Enter the calculated values for k_p and k_v in the Matlab command line
3. From the Simulink diagram, adjust the *Signal Generator* to a *square* step reference
 - Amplitude = 1
 - Frequency = 0.66
4. Set the *Amplitude (m)* gain block to 0.01 to generate a step of 10 mm
5. Open the cart position scope (*Meas.(0)andSim.(1)Resp.*) and the motor input voltage scope (V_m (V)) to view the graphical data in real-time
6. At the top, open the "Monitor & Tune" drop-down menu and click on "Build for Monitoring"
 - Verify that the physical amplifier unit is turned on with the gain set to 1x
 - Position the cart so that it is aligned in the center of the railing
7. Once the file has compiled, click on *Connect* then *Start*; click "Stop" when you see 3 steps on the graphs
 - Keep clear of the rail the cart travels on until the connection has been stopped
8. Export the data from *data_xc* and *data_vm*

Action Items

In your lab report, provide the following:

1. Take the Laplace transform of equation 1 to derive the voltage-to-velocity transfer function in equation 2. Express K and τ in terms of the system variables A_m , B_{eq} and M .
2. Derive the closed-loop transfer function for PV control given by equation 8. Express the gains k_p and k_d in terms of ω_n and ζ .
3. Use equations 11 and 12, calculate the minimum damping ratio ζ and natural frequency ω_n to meet the design specifications for percent overshoot and peak time. Use these to calculate the required values of k_p and k_d for the position control.

4. Using the error transfer function (equation 15) and the PV control equation (equation 8), find an expression for the transfer function from the reference $R(s)$ to the error $E(s)$. Then use this to find steady-state error of the system, e_{ss} , for the step input $R(s) = R_0/s$, where R_0 is the desired cart position. Using the Final-Value Theorem, show that e_{ss} is zero for any value of R_0 .
5. Provide a table with the parameter values used for the position control (k_p and k_v).
6. Provide graphs of the collected data.
7. Calculate the steady-state error, the percent overshoot, and the peak time for the Position and Speed data.
8. Briefly discuss any sources of error, and how they affect your final results.

Relevant files

<i>FileName</i>	<i>Description</i>
setup_ip02_position.m	Run this file only to set up the experiment's position control parameters.
config_ip02.m	Returns the configuration-based IP02 model specifications Rm , Jm , Kt , Eff_m , Km , Kg , Eff_g , M , r_{mp} , and Beq
d_ip02_position.m	Determines the response specifications ζ and ω_n , and the control gains k_p and k_v
d_model_param.m	Determines the model parameters K , and τ
s_ip02_position.mdl	Simulink file that simulates the closed-loop IP02 position control step response
q_ip02_position.mdl	Simulink file that implements the closed-loop IP02 position controller using QUAD

Table 1: Matlab files needed for the Position Control Experiment

Appendix

In this section, we derive the equations of motion for the cart system. By Newton's Second Law, we can represent the force between the cart's DC motor and the motion as follows:

$$M\dot{v}_c(t) = F_c(t) - B_c v_c(t) \quad (17)$$

where M is the mass of the cart, v_c is the linear velocity of the cart, and B_c is the equivalent viscous damping coefficient as seen at the motor pinion. The driving force generated by the motor on the cart can be expressed as:

$$F_c = \frac{\eta_g K_g \tau_m}{r_{mp}} \quad (18)$$

<i>FileName</i>	<i>Description</i>
setup_ip02_speed.m	Run this file only to set up the experiment's speed control parameters.
config_ip02.m	Returns the configuration-based IP02 model specifications R_m , J_m , K_t , Eff_m , Km , Kg , Eff_g , M , r_mp , and Beq
d_ip02_speed_lead.m	Determines the control parameters K_c and a , and T .
d_ip02_speed_pi.m	Determines the control parameters K_p and k_i .
d_model_param.m	Determines the PI control parameters K , and τ
s_ip02_speed_lead.mdl	Simulink file that simulates the closed-loop IP02 lead speed control step response.
q_ip02_speed_lead.mdl	Simulink file that implements the closed-loop IP02 lead speed controller using QUINTUS.
q_ip02_speed_pi	Simulink file that implements the closed-loop IP02 PI speed controller.

Table 2: Matlab files needed for the Speed Control Experiment

where η_g is the gearbox efficiency, K_g is the gear ratio, τ_m is the torque generated by the motor, and r_{mp} is the motor pinion radius.

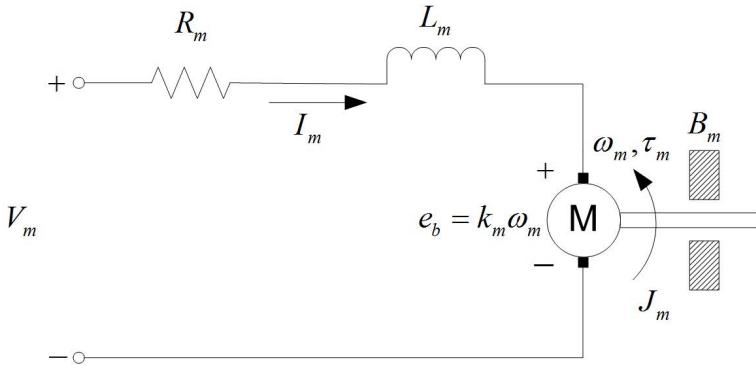


Figure 6: IP02 DC Motor Armature Circuit

Figure 6 displays the circuit for the driving motor where R_m is the motor resistance, L_m

is the inductance, B_m is the motor damping, and k_m is the back-emf constant. Using Kirchoff's Voltage Law, we can write the voltage equation to solve for the inductance (given that L_m represents a much smaller value, it can be neglected):

$$V_m(t) - R_m I_m(t) - L_m I_m(t) - k_m \omega_m(t) = 0 \rightarrow I_m(t) = \frac{V_m(t) - k_m \omega_m(t)}{R_m} \quad (19)$$

By introducing the motor torque constant, k_t , we can expand equation (3) by substituting all the components:

$$F_c = \frac{\eta_g K_g \eta_m k_t (V_m(t) - k_m \omega_m(t))}{R_m r_{mp}} \quad \text{where } \tau_m(t) = \eta_m k_t I_m(t) \quad (20)$$

The translation between angular to linear velocity from the motor to the cart is as follows

$$F_c = \frac{\eta_g K_g \eta_m k_t (V_m(t) r_{mp} - k_m K_g v_c(t))}{r_{mp}^2 R_m} \quad \text{where } \omega_m(t) = \frac{K_g v_c(t)}{r_{mp}} \quad (21)$$

We can now substitute the force function into equation (2):

$$M \dot{v}_c(t) + B_c v_c(t) = \frac{\eta_g K_g \eta_m k_t (V_m(t) r_{mp} - k_m K_g v_c(t))}{r_{mp}^2 R_m} \quad (22)$$

We can rearrange the equation to combine like-terms:

$$M \dot{v}_c(t) + \left(\frac{k_m \eta_g K_g^2 \eta_m k_t}{r_{mp}^2 R_m} + B_c \right) v_c(t) = \frac{\eta_g K_g \eta_m k_t V_m(t)}{r_{mp} R_m} \quad (23)$$

This equation can be simplified as follows:

$$M \dot{v}_c(t) + B_{eq} v_c(t) = A_m V_m(t) \quad (24)$$

where the equivalent damping term is:

$$B_{eq} = \frac{\eta_g K_g^2 \eta_m k_t k_m + B_c r_{mp}^2 R_m}{r_{mp}^2 R_m} \quad (25)$$

and the actuator gain is:

$$A_m = \frac{\eta_g K_g \eta_m k_t}{r_{mp} R_m} \quad (26)$$