QUANSER
INNOVATE. EDUCATE.

STUDENT WORKBOOK

Ball and Beam Experiment for MATLAB®/Simulink® Users

Standardized for ABET* Evaluation Criteria

Developed by:
Jacob Apkarian, Ph.D., Quanser
Michel Lévis, M.A.Sc., Quanser
Hakan Gurocak, Ph.D., Washington State University

SRV02 educational solutions are powered by:
MATLAB SIMULINK

Course material complies with:
ABET

CAPTIVATE. MOTIVATE. GRADUATE.

*ABET Inc., is the recognized accreditor for college and university programs in applied science, computing, engineering, and technology. Among the most respected accreditation organizations in the U.S., ABET has provided leadership and quality assurance in higher education for over 75 years.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com

ACKNOWLEDGEMENTS

Quanser, Inc. would like to thank Dr. Hakan Gurocak, Washington State University Vancouver, USA, for his help to include embedded outcomes assessment.

CONTENTS

1 INTRODUCTION

The objective of this laboratory is to stabilize a ball to a desired position along a beam. Using the proportional-derivative (PD) family, a cascade control system is designed to meet a set of specifications.

Topics Covered

- Modeling dynamics of the ball from first-principles.
- Obtaining transfer function representation of the system.
- Design of a proportional-velocity (PV) compensator to control the position of the servo load shaft according to time-domain requirements.
- Assessment of how well the system meets design specifications using root locus.
- Design of a cascade control system to regulate the position of the ball on the beam.
- Simulation of the Ball and Beam control to ensure that the specifications are met without any actuator saturation.
- Implementation of the controllers on the Quanser BB01 device and evaluation of its performance.

Prerequisites
In order to successfully carry out this laboratory, the user should be familiar with the following:

- Data acquisition card (e.g., Q2-USB), the power amplifier (e.g., VoltPAQ-X1), and the main components of the SRV02 (e.g. actuator, sensors), as described in References [5], [6], and [4], respectively.
- Wiring and operating procedure of the SRV02 plant with the amplifier and DAQ devices, as discussed in Reference [4].
- Transfer function fundamentals, e.g. obtaining a transfer function from a differential equation.
- Laboratory described in Reference [2] or the *SRV02 QUARC Integration* lab in [8] to get familiar with using QUARC®with the SRV02.

2 BACKGROUND

2.1 Modeling from First Principles

As illustrated in Figure 2.1, this system is comprised of two plants: the Rotary Servo (SRV02) and the Ball and Beam (BB01).

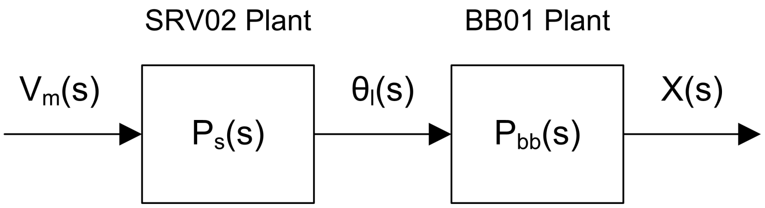


Figure 2.1: Ball and Beam open-loop block diagram.

The main objective in this section is to obtain the complete SRV02+BB01 transfer function

$$P(s) = P_{bb}(s)\, P_s(s) \tag{2.1}$$

where the BB01 transfer function is

$$P_{bb}(s) = \frac{X(s)}{\Theta_l(s)} \tag{2.2}$$

and the SRV02 transfer function is

$$P_s(s) = \frac{\Theta_l(s)}{V_m(s)}. \tag{2.3}$$

The BB01 transfer function describes the linear displacement of the ball, $x(t)$, with respect to the load angle of the servo, $\theta_l(t)$. In the next few sections, the time-based motion equations are developed and the transfer function is obtained. The SRV02 transfer function models the servo load gear position, $\theta_l(t)$, with respect to the servo input voltage, $V_m(t)$. Recall that in Modeling Laboratory ([8]), this transfer function was found to be:

$$P_s(s) = \frac{K}{s\,(\tau s + 1)} \tag{2.4}$$

The nominal model parameters, K and τ, of the SRV02 with no load and in high-gear configuration are:

$$K = 1.53 \text{ rad/(V.s)} \tag{2.5}$$

and

$$\tau = 0.0248 \text{ s} \tag{2.6}$$

Note: These parameters are different than the those found in the Modeling Laboratory because it does not include the inertial disc load.

2.1.1 Nonlinear Equation of Motion

In this section, the equation describing the motion of the ball, x, relative to the angle of the beam, α, is derived. Refering to Figure 2.2, the equation of motion, or *eom* for short, can be found starting from Newton's Law of Motion. The sum of forces acting on the ball along the beam equals:

$$m_b \left(\frac{d^2}{dt^2}\, x(t) \right) = \sum F \tag{2.7}$$

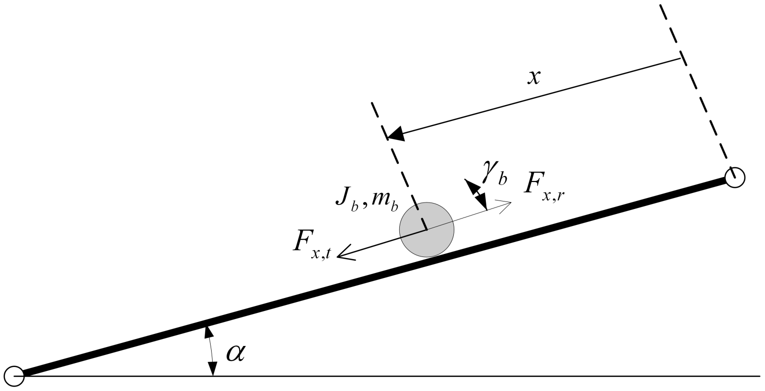


Figure 2.2: Free-body diagram of Ball and Beam.

where m_b is the mass of the ball.

Neglecting friction and viscous damping, the ball forces can be represented by

$$m_b \left(\frac{d^2}{dt^2} x(t) \right) = F_{x,t} - F_{x,r} \tag{2.8}$$

where $F_{x,r}$ is the force from the ball's inertia and $F_{x,t}$ is the translational force generated by gravity. For the ball to be stationary at a certain moment, i.e. be in equilibrium, the force from the ball's momentum must be equal to the force produced by gravity. As illustrated in Figure 2.3, the force $F_{x,t}$ in the x direction (along the beam) that is caused by gravity can be found as:

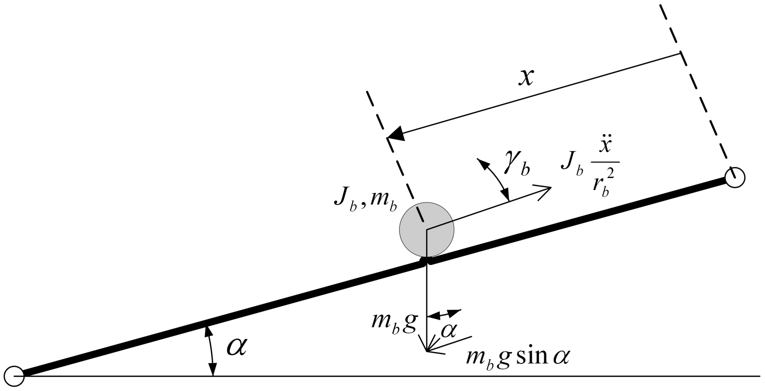


Figure 2.3: Completed Ball and Beam free-body diagram.

$$F_{x,t} = m_b\, g\, \sin \alpha(t) \tag{2.9}$$

The force caused by the rotation of the ball is

$$F_{x,r} = \frac{\tau_b}{r_b} \tag{2.10}$$

where r_b is the radius of the ball and τ_b is the torque which equals

$$\tau_b = J_b \left(\frac{d^2}{dt^2} \gamma_b(t) \right) \tag{2.11}$$

where γ_b is the ball angle. Using the sector formula, $x(t) = \gamma_b(t)\, r_b$, we can convert between linear and angular displacement. Then, the force acting on the ball in the x direction from its momentum becomes:

$$F_{x,r} = \frac{J_b\left(\frac{d^2}{dt^2}x(t)\right)}{r_b^2} \tag{2.12}$$

Now, by substituting the rotational and translational forces into Equation 2.8, we can get the nonlinear equation of motion for the ball and beam as:

$$m_b\left(\frac{d^2}{dt^2}x(t)\right) = m_b\, g\, \sin\alpha(t) - \frac{J_b\left(\frac{d^2}{dt^2}x(t)\right)}{r_b^2} \tag{2.13}$$

Solving for the linear acceleration gives:

$$\frac{d^2}{dt^2}x(t) = \frac{m_b\, g\, \sin\alpha(t)\, r_b^2}{m_b\, r_b^2 + J_b} \tag{2.14}$$

2.1.2 Adding SRV02 Dynamics

In this section, the equation of motion representing the position of the ball relative to the angle of the SRV02 load gear is derived. The obtained equation will be nonlinear (includes a trigonometric term). Therefore, it will have to be linearized to use in control design.

Let's look at how we can find the relationship between the SRV02 load gear angle, θ_l, and the beam angle, α. Using the schematic given in *BB01 User Manual* ([7]), consider the beam and servo angles required to change the height of the beam by h. Taking the sine of the beam angle gives the expression

$$\sin\alpha(t) = \frac{h}{L_{beam}} \tag{2.15}$$

and taking the sine of the servo load shaft angle results in the equation

$$\sin\theta_l(t) = \frac{h}{r_{arm}} \tag{2.16}$$

From these we can obtain the following relationship between the beam and servo angle

$$\sin\alpha(t) = \frac{\sin\theta_l(t)\, r_{arm}}{L_{beam}} \tag{2.17}$$

To find the equation of motion that represent the ball's motion with respect to the SRV02 angle θ_l we need to linearize the equation of motion about the servo angle $\theta_l(t) = 0$. Insert the servo and beam angle relationship, Equation 2.17, into the nonlinear eom found in 2.14

$$\frac{d^2}{dt^2}x(t) = \frac{m_b\, g\, \sin\theta_l(t)\, r_{arm}\, r_b^2}{L_{beam}\,(m_b\, r_b^2 + J_b)} \tag{2.18}$$

About angle zero, the sine function can be approximated by

$$\sin\theta_l(t) \approx \theta_l(t) \tag{2.19}$$

Applying this to the nonlinear eom gives the *linear* equation of motion of the BB01

$$\frac{d^2}{dt^2}x(t) = \frac{m_b\, g\, \theta_l(t)\, r_{arm}\, r_b^2}{L_{beam}\,(m_b\, r_b^2 + J_b)} \tag{2.20}$$

To simplify the equation, we can lump the coefficient parameters of $\theta_l(t)$ into a single parameter K_{bb}. Here, K_{bb} is called the *model gain* of the Ball and Beam system.

2.1.3 Obtaining Transfer Function

In this section, we will derive the transfer function that describes the servo voltage to ball position displacement. The transfer function $P_{bb}(s)$ of the BB01 can be found by taking the Laplace transform of the linear equation of motion in 2.20 as:

$$P_{bb}(s) = \frac{X(s)}{\Theta_l(s)} = \frac{K_{bb}}{s^2} \tag{2.21}$$

As illustrated in Figure 2.1, both systems are in series. By inserting the BB01 plant transfer function 2.21 and the SRV02 plant transfer function 2.4 into Equation 2.1, we can derive the complete process transfer function $P(s)$ as:

$$P(s) = \frac{X(s)}{V_m(s)} = \frac{K_{bb}\, K}{s^3\, (\tau\, s + 1)} \tag{2.22}$$

This is the servo voltage to ball displacement transfer function.

Note: Finding the K_{bb} gain is left as an exercise in Section 3.

2.2 Desired Control Response

2.2.1 Time-Domain Specifications

The steady-state error, peak time, and percentage overshoot time-domain specifications for controlling the position of the SRV02 load shaft are:

$$e_{ss} = 0 \tag{2.23}$$

$$t_p = 0.15 \text{ s, and} \tag{2.24}$$

$$PO = 5.0 \text{ \%} \tag{2.25}$$

Thus, when tracking the load shaft step reference, the transient response should have a peak time less than or equal to 0.15 seconds, an percentage overshoot (PO) less than or equal to 5 %, and no steady-state error.

The steady-state error, settling time, settling percentage, and percentage overshoot specifications for controlling the position of the ball are:

$$|e_{ss}| \leq 0.005 \text{ m} \tag{2.26}$$

$$t_s = 3.5 \text{ s} \tag{2.27}$$

$$c_{ts} = 0.04, \text{ and} \tag{2.28}$$

$$PO = 10.0 \text{ \%} \tag{2.29}$$

Thus, given a step reference, the peak position of the ball (i.e., percentage overshoot) should not exceed 10%. After 3.5 seconds, the ball should settle within 4% of its steady-state value (i.e. not the reference) and the steady-state should be within 5 mm of the desired position.

2.3 Ball and Beam Cascade Control Design

The cascade control that will be used for the SRV02+BB01 system is illustrated by the block diagram in Figure 2.4. Based on the measured ball position $X(s)$, the ball and beam compensator, $C_{bb}(s)$ in the outer-loop computes the servo load shaft angle $\Theta_d(s)$ to attain the desired ball position $X_d(s)$. The inner loop is a servo position control system as described in the SRV02 Position Control Laboratory ([8]). Thus, the servo compensator $C_s(s)$ calculates the motor voltage required to track the desired load shaft angle.

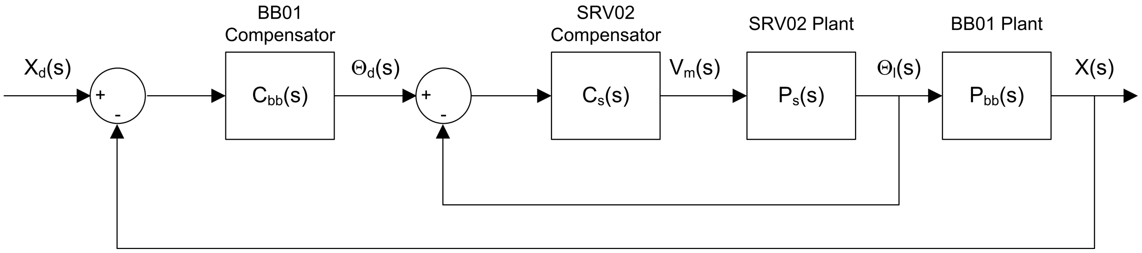


Figure 2.4: Cascade control system used to control ball position in SRV02+BB01 plant.

2.3.1 Inner Loop Controller Design: SRV02 PV Position Controller

The inner loop implements a PV controller to manage the position of the SRV02 load shaft. In Figure 2.5 this is shown as $C_s(s)$. The design of this controller was explored previously in the SRV02 Position Control Laboratory. To meet the desired performance specifications given in Section 2.2.1, we need to compute the k_p and k_v (proportional and velocity) gains of the controller when the SRV02 is in the high-gear configuration.

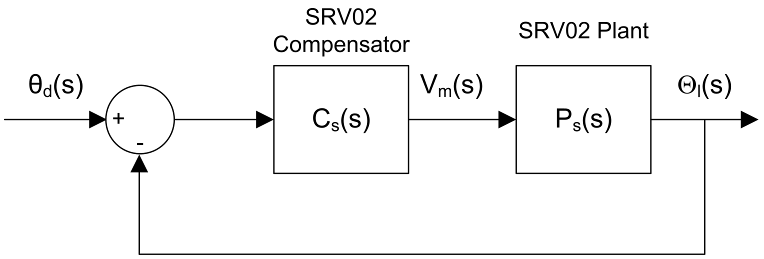


Figure 2.5: SRV02 closed-loop system.

Recall that the standard definitions for peak time, t_p, and percent overshoot, PO, are:

$$t_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \tag{2.30}$$

and

$$PO = 100 \text{ exp} \left(\frac{-\pi\,\zeta}{\sqrt{1-\zeta^2}} \right) \tag{2.31}$$

Using these formulas with the specifications in Equation 2.25, we can calcuate the damping ratio needed to obtain the desired percent overshoot:

$$\zeta = 0.690 \tag{2.32}$$

And, the minimum natural frequency required to meet the desired peak time, given in 2.24, is

$$\omega_n = 28.9 \text{ rad/s} \tag{2.33}$$

We can find the necessary control gains by using the definitions for k_p and k_v from SRV02 Position Control Laboratory. The k_p gain is found from the model parameters in 2.5 and 2.6 as well as the desired natural frequency found in 2.33:

$$k_p = 13.5 \text{ V/rad} \tag{2.34}$$

The velocity gain k_v is obtained using the model parameters and the minimum damping ratio specification in 2.32:

$$k_v = 0.078 \text{ V.s/rad} \tag{2.35}$$

2.3.2 Outer Loop Controller Design

We plan to use a dynamic compensator in the outer loop in Figure 2.6. Generally speaking, adding a zero in the forward-path increases the bandwidth of the closed-loop system. Adding a pole increases the rise time and overshoot of the system and makes it overall less stable. In our case, the bandwidth must be increased and the overshoot has to be minimized.

We can simplify the cascade controller shown in Figure 2.4, by replacing the inner loop with a single block $G_s(s)$ as:

For now, we will consider the dynamics of the inner loop as negligible ($G_s(s) = 1$). Therefore, it is assumed that the inner loop can control the position of the SRV02 load shaft perfectly, hence the desired load shaft angle equals the actual load shaft angle:

$$\theta_l(t) = \theta_d(t) \tag{2.36}$$

We will design a PD controller for the ``BB01 Compensator'' block in Figure 2.6. This controller is in the outer loop. Given the desired ball position, $X_d(s)$, and the actual ball position $X(s)$, the error in the system will be used by the controller to generate $\Theta_d(s)$ as a desired input for the inner loop controller $G_s(s)$. Since we assumed that the inner loop controller can perfectly track the desired shaft angle, it will manage the dynamics of the SRV02 and move the SRV02 shaft to the $\Theta_l(s)$ position.

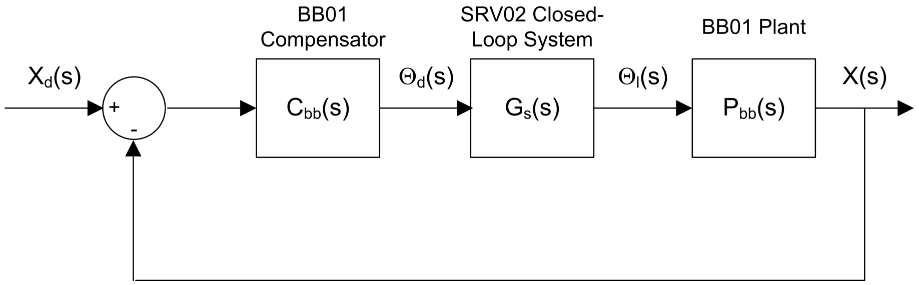


Figure 2.6: BB01 closed-loop system.

A *traditional* PD controller is given by $G_{PD}(s) = K_c \cdot (s + z)$. However, before we can proceed, we need to make a slight variation in the traditional PD controller. As we have done previously in the SRV02 Position Control Laboratory, we will replace the traditional PD controller with the one shown in Figure 2.7. Since this version does not directly feed the setpoint velocity $s\,X_d(s)$, it is more desirable in practical applications due to the noise in the signals.

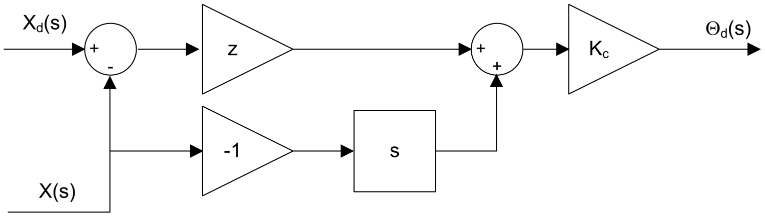


Figure 2.7: Ball and beam ideal PD controller (Also called PV controller).

Ideal PD Control Design

Using the block diagram, the outer-loop controller can be written as:

$$\Theta_d(s) = K_c\,\left(z\,\left(X_d(s) - X(s)\right) - s\,X(s)\right) \tag{2.37}$$

Recall that $\Theta_l(t) = \Theta_d(t)$ when the servo dynamics is ignored. By substituting the above transfer function into 2.21 and solving for $X(s)/X_d(s)$, we can obtain the BB01 closed-loop transfer function as:

$$\frac{X(s)}{X_d(s)} = \frac{K_{bb}\,K_c\,z}{s^2 + K_{bb}\,K_c\,s + K_{bb}\,K_c\,z} \tag{2.38}$$

In Figure 2.7, z and K_c are the zero location and the gain for the controller, respectively. We can compute the value for z to meet the settling time and overshoot specifications. Also, we can find the value of K_c to satisfy a given natural frequency and damping ratio.

A standard second order system is given by the following transfer function:

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\,\zeta\,\omega_n\,s + \omega_n^2} \tag{2.39}$$

By matching the coefficients of the denominator of this equation to those in equation 2.38, we can find the expressions for z and K_c to meet a certain ω_n and ζ:

$$z = \frac{\omega_n^2}{K_{bb}\,K_c} \tag{2.40}$$

and the gain must be

$$K_c = \frac{2\,\zeta\,\omega_n}{K_{bb}} \tag{2.41}$$

Practical PD Controller

Even though the *ideal* PD controller is an improvement over the *traditional* PD controller, it still needs one more adjustment before we can use it in practice. The position of the ball is measured using an analog sensor and it has some inherent noise. Taking the derivative of this type of signal would output results in an amplified high-frequency signal that is eventually fed back into the motor and causes a grinding noise. As illustrated by $H(s)$ in Figure 2.8, this is prevented by using a high-pass filter.

The first-order filter replaces the derivative in Figure 2.7 and has the form

$$H(s) = \frac{\omega_f\, s}{s + \omega_f} \tag{2.42}$$

For adequate filtering of the noise found in the BB01 linear transducer, the cutoff frequency, ω_f, will be set to 1 Hz, or

$$\omega_f = 6.28 \text{ rad/s} \tag{2.43}$$

Also, added to the controller is the setpoint weight parameter b_{sp}. This varies the amount of setpoint that is used to compute the error velocity. This compensator is called *practical* PD controller.

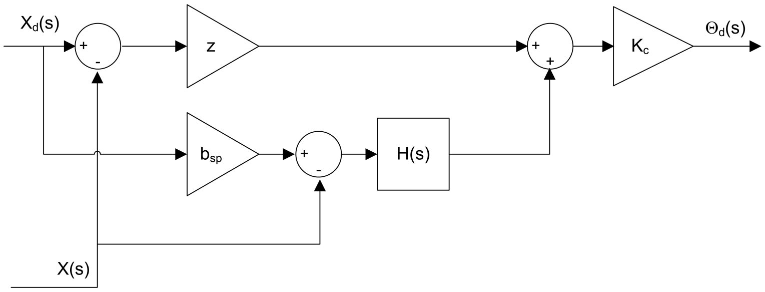


Figure 2.8: BB01 practical PD controller with filtering.

Although filtering is often necessary when controlling actual systems to make them more robust against noise, it adds dynamics to the system. Thus, the compensator gain and zero location have to be re-computed to meet the specifications listed in Section 2.2.1.

$$\Theta_d(s) = K_c\left(z\,(X_d(s) - X(s)) + \frac{\omega_f\, s\,(b_{sp}\, X_d(s) - X(s))}{s + \omega_f}\right) \tag{2.44}$$

Given that there are no servo dynamics, i.e. $\Theta_d(s) = \Theta_l(s)$, the closed-loop equation of the BB01 can be found by substituting the above controller into the BB01 open-loop transfer function in 2.21 and solving for $X(s)/X_d(s)$,

$$\frac{X(s)}{X_d(s)} = \frac{K_{bb}\, K_c\, ((z + \omega_f\, b_{sp})\, s + z\, \omega_f)}{s^3 + s^2\, \omega_f + (K_{bb}\, K_c\, \omega_f + K_{bb}\, K_c\, z)\; s + K_{bb}\, K_c\, z\, \omega_f} \tag{2.45}$$

When the setpoint weight of the *practical* PD controller is 1 ($b_{sp} = 1$), the BB01 compensator, $C_{bb}(s)$ in Figure 2.6 can be found as:

$$C_{bb}(s) = \frac{\Theta_d(s)}{X_d(s) - X(s)} \tag{2.46}$$

Setting $b_{sp} = 1$ in the controller in Equation 2.44 and solving for the above gives

$$C_{bb}(s) = \frac{((z + \omega_f)\, s + z\, \omega_f)\; K_c}{s + \omega_f} \tag{2.47}$$

The locations of the zero and the pole of the compensator are:

$$z_c = \frac{-\, z\, \omega_f}{z + \omega_f} \tag{2.48}$$

and

$$p_c = -\omega_f \tag{2.49}$$

respectively. Since $p_c < z_c$, the pole comes before the zero along the negative real-axis, this is a *lead* type compensator.

A standard third-order characteristic equation can be written as:

$$\left(s^2 + 2\,\zeta\,\omega_n\,s + \omega_n^2\right)\,(1 + T_p\,s) \tag{2.50}$$

where T_p is the pole decay in seconds. When expanded, the third-order characteristic equation becomes

$$s^3 + \frac{(2\,\zeta\,\omega_n\,T_p + 1)\;s^2}{T_p} + \frac{\left(\omega_n^2\,T_p + 2\,\zeta\,\omega_n\right)\;s}{T_p} + \frac{\omega_n^2}{T_p} \tag{2.51}$$

The characteristic equation of the closed-loop system (equation 2.45) using the *practical* PD controller is

$$s^3 + s^2\,\omega_f + (K_{bb}\,K_c\,\omega_f + K_{bb}\,K_c\,z)\;s + K_{bb}\,K_c\,z\,\omega_f \tag{2.52}$$

The following expression is obtained after equating the coefficients of the s^2 terms in equations 2.51 and 2.52,

$$\omega_f = \frac{2\,\zeta\omega_n\,T_p + 1}{T_p} \tag{2.53}$$

Solving for T_p gives the the required pole location given the desired response specifications

$$T_p = \frac{1}{\omega_f - 2\,\zeta\,\omega_n} \tag{2.54}$$

3 PRE-LAB QUESTIONS

1. Find K_{bb} by simplifying the expression given in Equation 2.18. Then, evaluate it using the system parameters given in [7]. **Hint:** Recall that the mass moment of inertia of a solid sphere is $J = \frac{2\,m\,r^2}{5}$.
2. Find the steady-state error of the the Ball and Beam system given by the $P_{bb}(s)$ transfer function. The system is shown in Figure 3.1. The compensator is unity

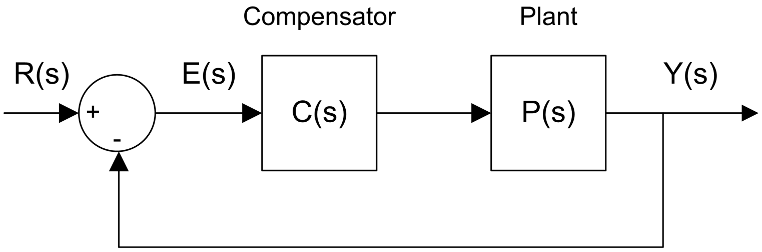


Figure 3.1: Unity feedback system.

$$C(s) = 1 \tag{3.1}$$

and the reference step is

$$R(s) = \frac{R_0}{s} \tag{3.2}$$

where R_0 is the step amplitude. Note that in this calculation the SRV02 dynamics is to be ignored and only the BB01 plant is to be considered.

3. Using Figure 2.6, find the closed-loop transfer function of the BB01 system with proportional control $C_{bb}(s) = K_c$.
4. Plot the root locus of the BB01 plant $P_{bb}(s)$. Describe how the poles behave as K_c goes to infinity.
5. Find the natural frequency and damping ratio required to achieve the time-domain specifications of the Ball and Beam plant given in Section 2.2.1.
6. After you plot the root locus of the BB01 plant $P_{bb}(s)$, describe where the poles should be to satisfy the desired response specifications.
7. Discuss the response if the poles lie beyond the radius circle along the diagonal lines, i.e. away from the imaginary axis. Also, comment on what happens if the poles of the system lie inside the diagonal lines along the radius circle, i.e. moving towards the real axis. Make references to its effects on the settling time and overshoot of the response.
8. Based on the root locus obtained in question 6 previously, can the specifications of the Ball and Beam system be satisfied using a proportional controller? Discuss.
9. Assume that a *traditional* PD controller ($C_{bb}(s) = K_c(s + z)$) is used in the system given in Figure 2.6. Find the BB01 error transfer function.
10. Find the steady-state error of the BB01 closed-loop system with the *traditional* PD controller. Can the steady-state error requirement in 2.26 be satisfied?
11. Based on the expressions found in Equations 2.40 and 2.41, evaluate numerically the zero location and gain needed to satisfy the specifications.
12. Find expressions for the zero location, z, and the compensator gain, K_c for the *practical* PD controller to satisfy ω_n and ζ in Section 2.2.1 and the desired filter cutoff frequency in 2.43. Then, evaluate numerically the pole time constant, zero location and gain needed to satisfy the specifications.

4 LAB EXPERIMENTS

The main goal of this laboratory is to explore position control of the Ball and Beam system using cascade control.

In this laboratory, you will conduct two groups of experiments:

1. Cascade control with *ideal* PD controller
   - simulation with no servo dynamics,
   - simulation with servo dynamics
2. Cascade control with *practical* PD controller and servo dynamics
   - simulation,
   - implementation

4.1 Cascade Control with Ideal PD Controller

4.1.1 Simulation with No Servo Dynamics

Experimental Setup

The *s_bb01_pos_outer_loop* Simulink diagram shown in Figure 4.1 is used to simulate the closed-loop position response of the BB01 when using the outer-loop control. Thus, the SRV02 dynamics are neglected, i.e. $\theta_d = \theta_l$. The response is simulated using the developed nonlinear model of the Ball and Beam.

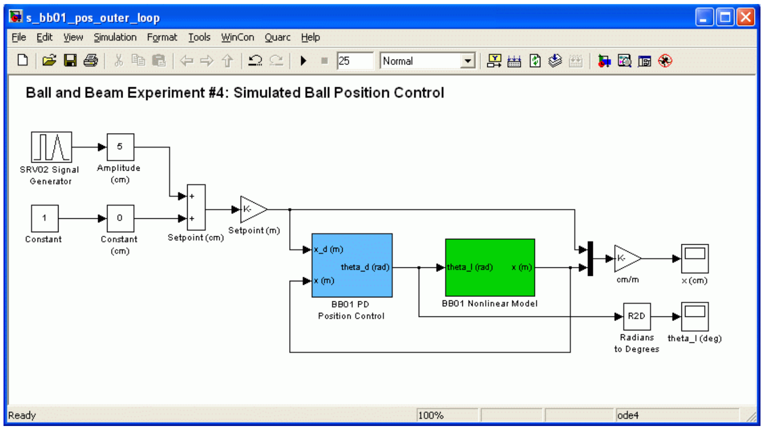


Figure 4.1: Simulink diagram used to simulate the outer closed-loop BB01 system.

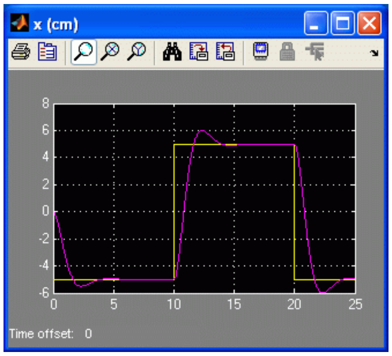
The *BB01 Nonlinear Model* subsystem includes the $P_{bb}(s)$ transfer function that was derived in Section 2.1.3. Recall that in Section 2.1.2 the model had to be linearized in order to obtain the $P_{bb}(s)$ transfer function. This nonlinearity is re-introduced in the *BB01 Nonlinear Model* subsystem in order to represent the plant more accurately and ensure the specifications can still be satisfied. The *BB01 PD Position Control* subsystem contains the *ideal PD* compensator designed in Section 2.3.2. Note that it includes a *Saturation* block that limits the SRV02 angle between ± 56 degrees.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your system setup. If they have not been configured already, then you need to go to Section 5.2 to configure the lab files first.

Simulation

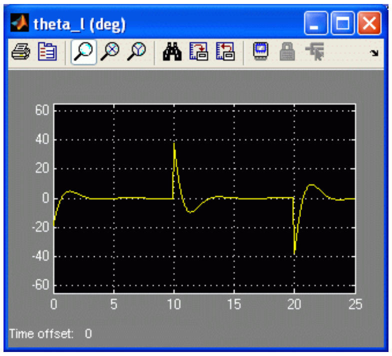
The root locus of the forward path is plotted using Matlab®and the closed-loop step position response of the BB01 is simulated to verify that the specifications are met. As previously mentioned, the simulation is performed using the nonlinear model of the Ball and Beam and the *ideal PD* controller that was designed.

1. Ensure the s_bb01_pos_outer_loop Simulink®diagram is open and setup as described in Section 5.2.
2. Enter the BB01 model gain found in Pre-Lab question 1 in the Matlab® Command Window as variable *K_bb* (e.g., type K_bb = 1 in the Matlab prompt).
3. In Matlab, enter the BB01 compensator gain *K_c* and the compensator zero, z, that were found in Pre-Lab question 11.
4. Using Matlab®, plot the root locus of BB01 loop transfer function when using the *ideal PD* controller. Use the *sgrid* command to include the dashed lines that show the desired locations of the poles. Ensure the poles go through the desired locations at the gain that was computed in Pre-Lab question 11.
5. Select square in the *Signal Type* field of the *SRV02 Signal Generator* in order to generate a step reference
6. Set the *Amplitude (cm)* slider gain block to 5 to generate a step with an amplitude of 5.0 centimeters
7. Open the load shaft position scope, *theta_l (deg)*, and the ball position scope, *x (m)*.
8. Start the simulation. By default, the simulation runs for 25.0 seconds. The scopes should be displaying responses similar to Figure 4.2 and Figure 4.3. In the *x (m)* scope, the yellow trace is the desired ball position and the purple trace is the simulated response.



Time offset: 0

Figure 4.2: Outer-loop ideal PD ball position response.



Time offset: 0

Figure 4.3: Outer-loop ideal PD servo angle response.

9. Generate a Matlab®figure showing the nonlinear Outer-Loop BB01 ball position response and the corresponding servo angle. After each simulation run, each scope automatically saves their response to a variable in the Matlab workspace. The *x (m)* scope saves its response to the variable called *data_x* and the *theta_l (deg)* scope saves its response to the variable *data_theta_l. The data_x* variable has the following structure: *data_x(:,1)* is the time vector, *data_x(:,2)* is the setpoint, and *data_x(:,3)* is the simulated ball position. For the *data_theta_l* variable, *data_theta_l (:,1)* is the time and *data_theta_l (:,2)* is the servo angle.
10. Measure the steady-state error, the settling time, and the percent overshoot of the simulated response.
11. Does the outer-loop ideal PD response satisfy the specifications given in Section 2.2.1 while keeping the servo angle between ± 56 degrees? Some tolerance is allowed on the settling time specification: it should not exceed 3.75 seconds (rather then 3.50 seconds). If the steady-state error and percent overshoot do no meet the desired specifications and the settling time goes over 3.75 seconds then go back to your control design. If the settling time does not satisfy the original specifications but is kept below the allowed tolerance, explain any possible source for this discrepancy.

4.1.2 Simulation with Servo Dynamics

Experimental Setup

The servo dynamics can now be added and the closed-loop position response with the cascade control system can be simulated using the Simulink®diagram *s_bb01_pos* pictured in Figure 4.4. This Simulink®model is for the block diagram shown in Figure 2.4.

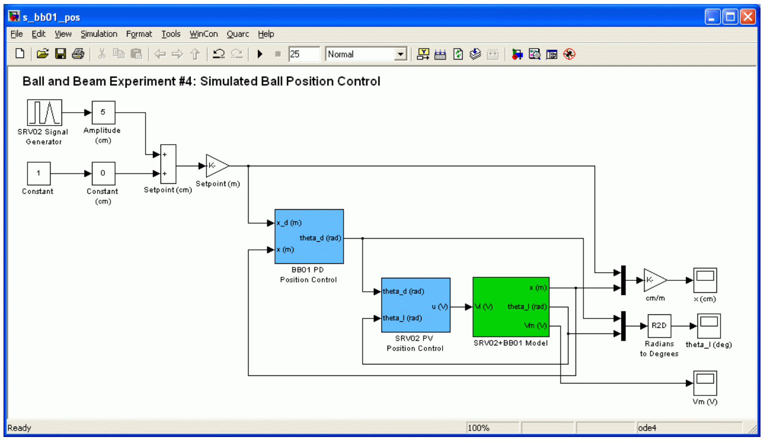


Figure 4.4: Simulink diagram used to simulate cascade control system.

The *SRV02+BB01 Model* subsystem includes the nonlinear model of the BB01 plant and the transfer function representing the SRV02 voltage-to-position relationship. The proportional-velocity position controller designed in Section 2.3.1 is implemented in the *SRV02 PV Position Control* block.

The cascade controller is the algorithm that will be implemented on the actual SRV02+BB01 device. Before deployment, we need to confirm that the specifications are still satisfied when the servo dynamics are added. In addition, the servo angle must be kept between ± 56 degrees and the servo voltage cannot exceed $\pm 10V$.

Simulation

The purpose of this simulation is to see how the settling time, overshoot, and steady-state error of the response changes when the inner-loop servo control that was designed in Section 2.3.1 is added. In this simulation, you will be using the *ideal PD* controller.

1. Ensure the s_bb01_pos Simulink®diagram is open and setup as described in Section 5.2.
2. Follow steps 1 and 2 in Section 4.1.1 to setup the Matlab® workspace.
3. In Matlab®, enter the SRV02 model gain, K, and the model time constant, *tau*, in Matlab®from Section 2.1. Note that the time constant, τ, is denoted as *tau* in Matlab®.
4. Enter the SRV02 PV gains: called variables *kp* and *kv* in Matlab®from Section 2.3.1.
5. Select square in the *Signal Type* field of the *SRV02 Signal Generator* in order to generate a step reference.
6. Set the *Amplitude (cm)* slider gain block to 5 to generate a step with an amplitude of 5.0 centimeters.
7. Open the ball position scope *x (m)*, the load shaft position scope *theta_l (deg)*, and the SRV02 motor input voltage scope *Vm(V)*.
8. Place the *Manual Switch* in the *BB01 PD Position Control* subsystem to the upward position in order to use the *ideal PD* controller when simulating.

9. Start the simulation. By default, the simulation runs for 25.0 seconds. The scopes should be displaying responses similar to figures 4.5, 4.6, and 4.7. The yellow and purples plots in the *x (m)* scope is the ball position setpoint and the its simulated response. Similarly in the *theta_l (deg)* scope, the yellow trace is the desired servo angle position, which is generated by the outer-loop control, and the the purple plot is the simulated servo response.

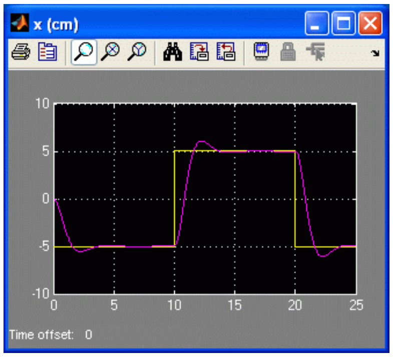


Figure 4.5: Ideal PD cascade control ball position response.

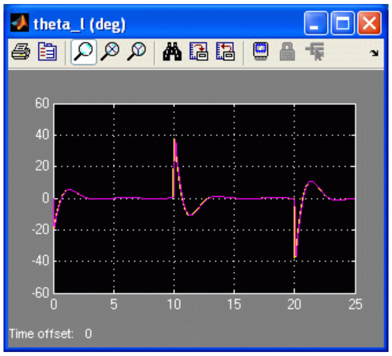


Figure 4.6: Ideal PD cascade control servo angle response.

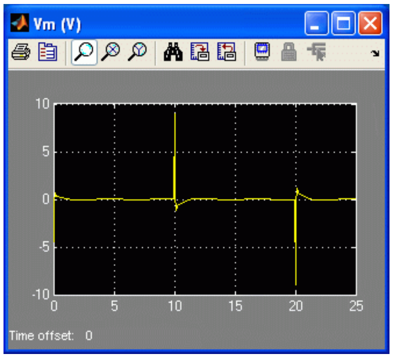


Figure 4.7: Ideal PD cascade input voltage.

10. Generate a Matlab® figure showing the *Ideal PD* cascade ball position, servo angle, and servo input voltage response. The response from each scope is saved to a Matlab® variable after each simulation run. The SRV02 motor input voltage is saved in the *data_vm* variable (*data_vm (:,1)* is the time and *data_vm (:,2)* is the voltage).

11. Measure the steady-state error, the settling time, and the percent overshoot of the ideal PD cascade control response.

12. Are the specifications in Section 2.2.1 still satisfied after adding the servo dynamics? Also, make sure the servo angle is within ± 56.0 degrees and the servo voltage is between $\pm 10.0 V$. Don't go back in the control design if some specifications are not met.

4.2 Cascade Control with Practical PD Controller and Servo Dynamics

4.2.1 Simulation with Practical PD Controller

The *practical* PD controller developed in Section 2.3.2 is simulated in this section. This is the compensator that will be used to control the actual BB01 device. The control gain and zero may have to be fine-tuned in order to compensate for the added dynamics of the filtering and the inner-loop servo control. Follow these steps to simulate the closed-loop *practical cascade PD* response:

1. Ensure the s_bb01_pos Simulink® diagram is open and setup as described in Section 5.2.
2. Enter the BB01 model gain found in Pre-Lab question 1 in Matlab® as variable K_{bb}.
3. Enter the *practical PD* compensator gain K_c, and zero, z, that were found in Pre-Lab question 12. The filter cutoff filter, ω_f, is already set by the script (See Section 5.2 for more details).
4. Follow steps 2-6 in Section 4.1.2 to setup the SRV02 model parameters and control gains and setup the Simulink® diagram.
5. To simulate using the *practical PD* controller, set the *Manual Switch* in the BB01 PD Position Control subsystem to the downward position.
6. Using Matlab® , plot the root locus of BB01 loop transfer function when using the *practical PD* compensator. Show the desired locations of the poles on the plot and ensure the poles go through the desired locations at the gain that was computed.
7. Open the ball position scope *x (m)*, the load shaft position scope *theta_l (deg)*, and the SRV02 motor input voltage scope *Vm(V)*.
8. Start the simulation. By default, the simulation runs for 25.0 seconds. The scopes should be displaying responses similar to figures Figure 4.8, Figure 4.9 and Figure 4.10.

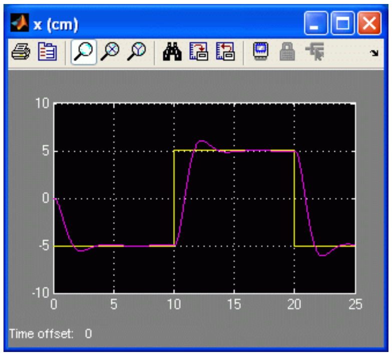


Figure 4.8: Practical cascade control ball position response.

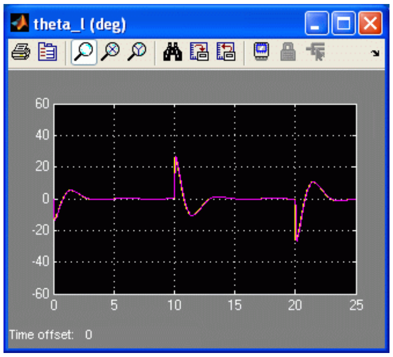


Figure 4.9: Practical cascade control servo angle response.

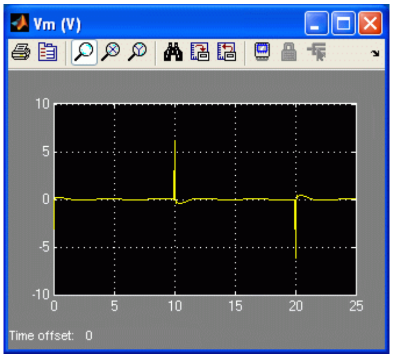


Figure 4.10: Practical cascade control input voltage.

9. Generate a Matlab® figure showing the *practical cascade* ball position, servo angle, and servo input voltage response.

10. Measure the steady-state error, the settling time, and the percent overshoot of the simulated *practical cascade PD control* response.

11. Does the simulated response satisfy the specifications given in Section 2.2.1 while keeping the servo angle between ± 56.0 degrees and the servo voltage between $\pm 10.0V$?

12. If a specification is not satisfied, then the control parameters need to be fine-tuned. One method is to redesign the compensator gain, zero location, and pole time constant, according to more stringent restrictions. For instance, try simulating the system for a K_c, z, and T_p, generated according to a percent overshoot of 8% instead of 10%. To do this, write a short Matlab® script that computes the gains automatically according to a given set of percent overshoot, settling time, and filter cutoff frequency specifications. Then, simulate the system and see if the specifications are satisfied. Note that the cutoff filter frequency should remain as specified in Equation 2.43, or 1 Hz.

13. **Tuned Practical PD #1 (Simulation)**: Record the gain and zero that have been fine-tuned for the response to meet the specifications along with the new specifications used to generate those control parameters.

14. Plot the simulated response in a Matlab®figure.

15. Give the resulting steady-state error, settling time, and percent overshoot of the response.

4.2.2 Implementation with Practical PD Controller

In this experiment, the position of the ball on the BB01 device will be controlled using the developed *practical* PD controller. Measurements will then be taken to ensure that the specifications are satisfied.

Experimental Setup

The *q_bb01_pos* Simulink® diagram shown in Figure 4.11 is used to perform the position control exercises in this laboratory. The *SRV02-ET+BB01* subsystem contains QUARC® blocks that interface with the DC motor and sensors of the Ball and Beam system. The *BB01 PD Position Control* subsystem implements the *practical PD* control detailed in Section 2.3.2.

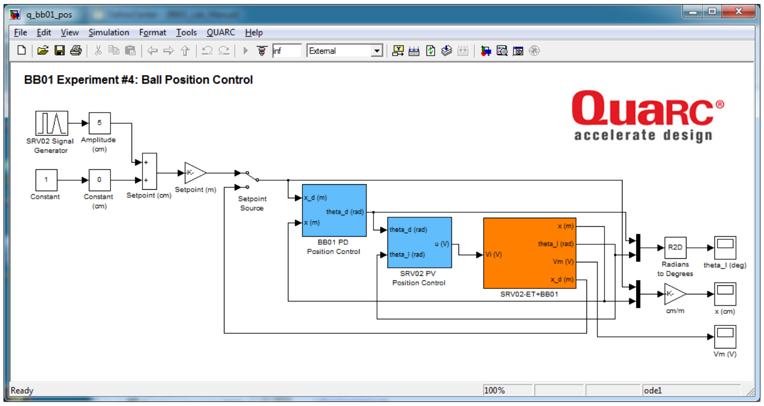


Figure 4.11: Simulink model used with QUARC to run the practical PD controller on the Ball and Beam system.

IMPORTANT: Before you can conduct this experiment, you need to make sure that the lab files are configured according to your system setup. If they have not been configured already, then you need to go to Section 5.3 to configure the lab files first.

1. Ensure the q_bb01_pos Simulink®diagram is open and setup as described in Section 5.3.

2. Enter the BB01 model gain found in Pre-Lab question 1 in Matlab® as variable K_{bb}.

3. Enter the *Tuned Practical PD #1 compensator* gain K_c, and zero, z, that were found in Section 4.2.1 (or the original gains if tuning was not required).

4. Follow steps 2-6 in Section 4.1.2 to setup the SRV02 model parameters and control gains and setup the Simulink diagram.

5. Turn ON the power amplifier.

6. Click on QUARC | Build to compile the Simulink® diagram.

7. Select QUARC | Start to run the controller. The scopes should be displaying responses similar to figures 4.12, 4.13 and 4.14.

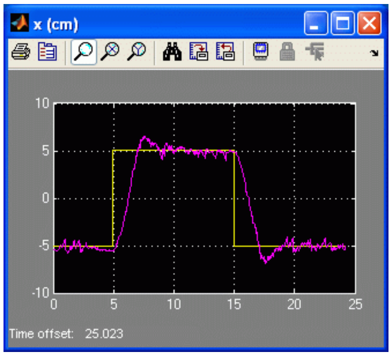


Figure 4.12: BB01 ball position response.

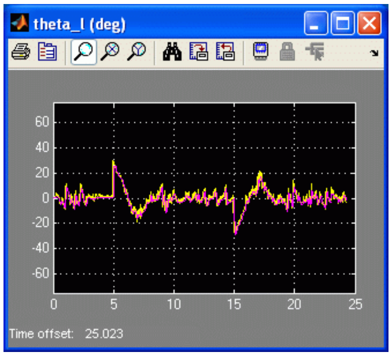


Figure 4.13: BB01 servo angle response.

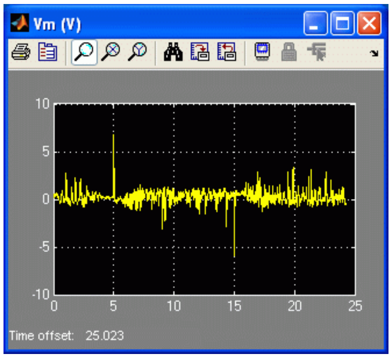


Figure 4.14: BB01 servo input voltage.

8. When a suitable response is obtained, click on the *Stop* button in the Simulink® diagram tool bar (or select QUARC® | Stop from the menu) to stop running the code. Generate a Matlab® figure showing the ball position and servo angle response as well as the input voltage. As in the *s_bb01_pos* Simulink® diagram, each scope automatically saves their response to a variable in the Matlab® workspace when the controller is stopped.

9. **Tuned Practical PD #1 (Implementation)**: Measure the steady-state error, the settling time, and the percent overshoot. Does the response satisfy the specifications given in Section 2.2.1? Give a reason why the designed gain and zero could fail to give a successful closed-loop response on the actual system?

10. **Tuned Practical PD #2 (Implementation)**: If the specification have not been satisfied, tune the gains as described in Section 4.2.1 and run the experiment again until a satisfactory response is obtained. In the case where the steady-state error is not satisfied, integral action can be introduced in the outer-loop controller. To do this, go into the *BB01 Position Control* subsystem and increase the *Integral Gain* block in small increments until the error is minimized enough. Briefly explain the procedure to get those new control parameters (including the integral gain, if necessary) and give the gain and zero used to obtain the response.

11. Plot the response with the *Tuned Practical PD #2* control using Matlab®.
12. Give the measured steady-state error, the settling time, and the percent overshoot of the response. Does the response satisfy the specifications given in Section 2.2.1?
13. Make sure QUARC® is stopped.
14. Shut off the power of the amplifier if no more experiments will be performed on the SRV02 in this session.

4.2.3 Controller using the Remote Sensor (Optional)

In this experiment, the position of the ball on the BB01 device will be controlled using the developed *practical PD control* but the setpoint is given with the remote sensor. This experiment can be conducted only if the remote sensor SS01 module detailed in [7] is available.

1. Follow steps 1-4 in Section 4.2.2 to setup the Matlab® workspace and build the q_bb01_pos model.
2. Place the *Setpoint Source* switch to the DOWN position in order to generate the setpoint using the SS01 module.
3. Select QUARC® | Start to run the controller.
4. Move the ball back and forth on the remote sensor and observe the response obtained in the scopes. Figures 4.15, 4.16 and 4.17 show a sample response.

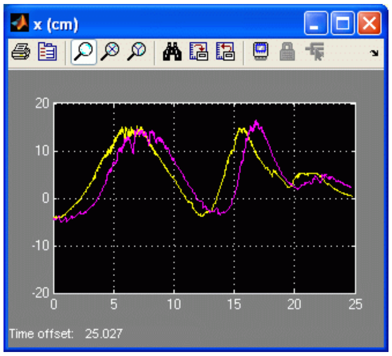


Figure 4.15: BB01 ball position response with SS01.

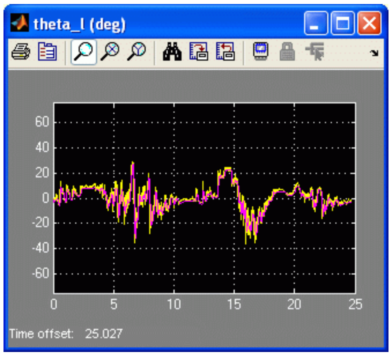


Figure 4.16: BB01 servo angle response with SS01.

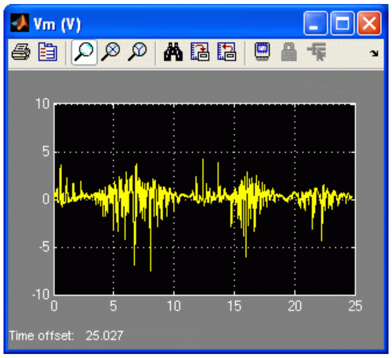


Figure 4.17: BB01 servo input voltage with SS01.

5. When done, click on the *Stop* button in the Simulink® diagram tool bar (or select QUARC® | Stop from the menu) to stop running the code.
6. Shut off the power of the amplifier if no more experiments will be performed on the SRV02 in this session.

4.3 Results

B-6 Fill out Table 4.1 below with your answers to the Pre-Lab questions and your results from the lab experiments.

Section / Question	**Description**	**Symbol**	**Value**	**Unit**
Question 11	**Pre-Lab: Ideal PD Control Design**			
	Compensator Gain	K_c		
	Compensator Zero	z		
Question 12	**Pre-Lab: Practical PD Control Design**			
	Compensator Gain	K_c		
	Compensator Zero	z		
	Compensator Pole Time Constant	T_p		
4.1.1	**In-Lab Simulation: Cascade ideal PD with no servo dyn.**			
	Steady-state error	e_{ss}		
	Settling time	t_s		
	Percentage overshoot	PO		
4.1.2	**In-Lab Simulation: Cascade ideal PD with servo dyn.**			
	Steady-state error	e_{ss}		
	Settling time	t_s		
	Percentage overshoot	PO		
4.2.1	**In-Lab Simulation: Cascade Practical PD**			
	Steady-state error	e_{ss}		
	Settling time	t_s		
	Percentage overshoot	PO		
Section 4.2.1, step 13	**In-Lab Simulation: Cascade Tuned Practical PD #1**			
	Compensator Gain	K_c		
	Compensator Zero	z		
	Steady-state error	e_{ss}		
	Settling time	t_s		
	Percentage overshoot	PO		
Section 4.2.2, step 9	**In-Lab Implementation: Tuned Practical PD #1**			
	Steady-state error	e_{ss}		
	Settling time	t_s		
	Percentage overshoot	PO		
Section 4.2.2, step 10	**In-Lab Implementation: Tuned Practical PD #2**			
	Compensator Gain	K_c		
	Compensator Zero	z		
	Integral Gain	k_i		
	Steady-state error	e_{ss}		
	Settling time	t_s		
	Percentage overshoot	PO		

Table 4.1: Summary of results for the Ball and Beam position laboratory.

5 SYSTEM REQUIREMENTS

Before you begin this laboratory make sure:

- QUARC® is installed on your PC, as described in Reference [3].
- You have a QUARC compatible data-aquisition (DAQ) card installed in your PC. For a listing of compliant DAQ cards, see Reference [1].
- SRV02 and amplifier are connected to your DAQ board as described Reference [4].

5.1 Overview of Files

Table 5.1: Files supplied with the SRV02 Speed Control laboratory.

File Name	Description
Ball and Beam User Manual.pdf	This manual describes the hardware of the Ball and Beam system and explains how to setup and wire the system for the experiments.
Ball and Beam Position Control - Student Manual.pdf	This laboratory guide contains pre-lab questions and lab experiments demonstrating how to design and implement a position controller on the Quanser SRV02 Ball and Beam plant using QUARC® .
setup_srv02_exp04_bb01.m	The main Matlab script that sets the SRV02 motor and sensor parameters, the SRV02 configuration-dependent model parameters, and the BB01 sensor parameters. Run this file only to setup the laboratory.
config_srv02.m	Returns the configuration-based SRV02 model specifications Rm, kt, km, Kg, eta_g, Beq, Jeq, and eta_m, the sensor calibration constants K_POT, K_ENC, and K_TACH, and the amplifier limits VMAX_AMP and IMAX_AMP.

(Continued on the next page)

File Name	Description
config_bb01.m	Returns the configuration-based BB01 model specifications L_beam, r_arm, r_b, m_b, J_b, and g, the servo offset THETA_OFF, the min/max servo limits THETA_MIN and THETA_MAX, and the sensor calibration constant K_BS.
d_model_param.m	Calculates the SRV02 model parameters *K* and *tau* based on the device specifications Rm, kt, km, Kg, eta_g, Beq, Jeq, and eta_m.
calc_conversion_constants.m	Returns various conversions factors.
s_bb01_pos_outer_loop.mdl	Simulink file that simulates the closed-loop system when using only the outer-loop ball position controller with the BB01 system, i.e. no inner loop control of the servo position.
s_bb01_pos.mdl	Simulink file that simulates the cascade ball position controller. Both the outer-loop ball position control and the inner-loop servo position control are used in this file.
q_bb01_pos.mdl	Simulink file that implements a closed-loop cascade position controller on the actual BB01 system using QUARC® .

5.2 Setup for Position Control Simulation

Follow these steps to configure the lab properly:

1. Load the Matlab®software.
2. Browse through the Current Directory window in Matlab and find the folder that contains the BB01 controller files.
3. Double-click on the s_bb01_pos_outer_loop.mdl file to open the Simulink diagram shown in Figure 4.1.
4. Double-click on the setup_srv02_exp04_bb01.m file to open the setup script for the BB01 Simulink models.
5. Configure setup script: When used with the Ball and Beam, the SRV02 must be in the high-gear configuration and no load is to be specified. Make sure the script is setup to match this configuration, i.e. the EXT_GEAR_CONFIG should be set to 'HIGH' and the LOAD_TYPE should be set to 'NONE'. Also, ensure the ENCODER_TYPE, TACH_OPTION, K_AMP, AMP_TYPE, and VMAX_DAC parameters are set according to the SRV02 system that is to be used in the laboratory. Next, set CONTROL_TYPE to 'MANUAL'.
6. Run the script. The following output should be see in the Matlab®prompt:

```
SRV02 model parameters:
   K   = 1.53 rad/s/V
   tau = 0.0248 s
SRV02 Specifications:
   tp = 0.15 s
   PO = 5%
BB01 model parameter:
   K_bb = 0 m/s^2/rad
BB01 Specifications:
   ts = 3.5 s
   PO = 10 %
Calculated SRV02 PV control gains:
   kp = 0 V/rad
   kv = 0 V.s/rad
Natural frequency and damping ratio:
   wn   = 0 rad/s
   zeta = 0
BB01 PD compensator:
   Kc = 0 rad/m
   z  = 1 rad/s
   wf = 6.28 rad/s
```

5.3 Setup for Position Control Implementation

Before beginning the in-lab exercises on the Ball and Beam device, the q_bb01_pos Simulink® diagram and the setup_srv02_exp04_bb01.m script must be configured.

Follow these steps to get the system ready for this lab:

1. Setup the SRV02 with the BB01 module as detailed in [7] .
2. Load the Matlab®software.
3. Browse through the *Current Directory* window in Matlab® and find the folder that contains the QUARC BB01 control file *q_bb01_pos.mdl*.
4. Double-click on the *q_bb01_pos.mdl* file to open the Ball and Beam Position Control Simulink diagram shown in Figure 4.11.
5. **Configure DAQ:** Ensure the HIL Initialize block in the *SRV02-ET+BB01* subsystem is configured for the DAQ device that is installed in your system. By default, the block is setup for the Quanser Q8 hardware-in-the-loop board. See Reference [2] for more information on configuring the HIL Initialize block.
6. **Configure Setpoint:** The setpoint can be generated through the SRV02 Signal Generator Simulink block or via the SS01 remote sensor (see Reference [7]). Place the *Setpoint Source* switch to the UP position in order to generate the setpoint through the Simulink model.
7. **Configure setup script:** Set the parameters in the *setup_srv02_exp04_bb01.m* script according to your system setup. See Section 5.2 for more details.
8. Run the script. **Note:** If you have just performed the Position Control Simulation lab, then your Matlab®workspace is already setup and you do not need to re-run this script.

6 LAB REPORT

This laboratory contains two groups of experiments, namely,

1. Cascade control with *ideal* PD controller, and
2. Cascade control with *practical* PD controller and servo dynamics.

For each experiment, follow the outline corresponding to that experiment to build the *content* of your report. Also, in Section 6.3 you can find some basic tips for the *format* of your report.

6.1 Template for Content (Cascade Control with Ideal PD Experiments)

I. PROCEDURE

1. *Simulation with no servo dynamics*
   - Briefly describe the main goal of the simulation.
   - Briefly describe the simulation procedure (Section 4.1.1)
2. *Simulation with servo dynamics*
   - Briefly describe the main goal of this simulation.
   - Briefly describe the simulation procedure (Section 4.1.2)

II. RESULTS
Do not interpret or analyze the data in this section. Just provide the results.

1. Response plot from step 9 in Section 4.1.1, *Simulation with no servo dynamics*
2. Response plot from step 10 in Section 4.1.2, *Simulation with servo dynamics*
3. Provide applicable data collected in this laboratory (from Table 4.1).

III. ANALYSIS
Provide details of your calculations (methods used) for analysis for each of the following:

1. Root locus plot from step 4 in Section 4.1.1, *Simulation with no servo dynamics*
2. Steady-state error, settling time and percent overshoot in step 10 in Section 4.1.1, *Simulation with no servo dynamics*
3. Steady-state error, settling time and percent overshoot in step 11 in Section 4.1.2, *Simulation with servo dynamics*

IV. CONCLUSIONS
Interpret your results to arrive at logical conclusions for the following:

1. Step 11 in Section 4.1.1, *Simulation with no servo dynamics*
2. Step 12 in Section 4.1.2, *Simulation with servo dynamics*

6.2 Template for Content (Cascade Control with Practical PD Experiments)

I. PROCEDURE

1. *Simulation*
   - Briefly describe the main goal of the simulation.
   - Briefly describe the simulation procedure (Section 4.2.1)
2. *Implementation*
   - Briefly describe the main goal of this experiment.
   - Briefly describe the experimental procedure (Section 4.2.2)

II. RESULTS
Do not interpret or analyze the data in this section. Just provide the results.

1. Response plot from step 9 in Section 4.2.1, *Simulation with practical PD controller*
2. Response plot from step 14 in Section 4.2.1, for *Tuned Practical PD #1 (Simulation)*
3. Response plot from step 8 in Section 4.2.2, *Simulation with practical PD controller*
4. Response plot from step 11 in Section 4.2.2, for *Tuned practical PD #2 (Implementation)*
5. Provide applicable data collected in this laboratory (from Table 4.1).

III. ANALYSIS
Provide details of your calculations (methods used) for analysis for each of the following:

1. Steady state error, the settling time and percent overshoot in step 10 in Section 4.2.1, *Simulation with practical PD controller*
2. Steady state error, the settling time and percent overshoot in step 15 in Section 4.2.1, *Simulation with practical PD controller*
3. Steady state error, the settling time and percent overshoot in step 9 in Section 4.2.2, *Implementation with practical PD controller*
4. Steady state error, the settling time and percent overshoot of the re-tuned controller in step 10 in Section 4.2.2, *Implementation with practical PD controller*
5. Steady state error, the settling time and percent overshoot in step 12 in Section 4.2.2, *Implementation with practical PD controller*

IV. CONCLUSIONS
Interpret your results to arrive at logical conclusions for the following:

1. Whether the controller meets the specifications in step 11 in Section 4.2.1, *Simulation with practical PD controller*
2. Whether the controller meets the specifications in step 9 in Section 4.2.2, *Implementation with practical PD controller*
3. Whether the controller meets the specifications in step 12 in Section 4.2.2, *Implementation with practical PD controller*

6.3 Tips for Report Format

PROFESSIONAL APPEARANCE

- Has cover page with all necessary details (title, course, student name(s), etc.)
- Each of the required sections is completed (Procedure, Results, Analysis and Conclusions).
- Typed.
- All grammar/spelling correct.
- Report layout is neat.
- Does not exceed specified maximum page limit, if any.
- Pages are numbered.
- Equations are consecutively numbered.
- Figures are numbered, axes have labels, each figure has a descriptive caption.
- Tables are numbered, they include labels, each table has a descriptive caption.
- Data are presented in a useful format (graphs, numerical, table, charts, diagrams).
- No hand drawn sketches/diagrams.
- References are cited using correct format.

REFERENCES

[1] Quanser Inc. QUARC User Manual.

[2] Quanser Inc. Srv02 quarc integration. 2008.

[3] Quanser Inc. QUARC Installation Guide. 2009.

[4] Quanser Inc. Srv02 user manual. 2009.

[5] Quanser Inc. Q2-usb User's Manual. 2010.

[6] Quanser Inc. Voltpaq User's Manual. 2010.

[7] Quanser Inc. Ball and beam user manual. 2011.

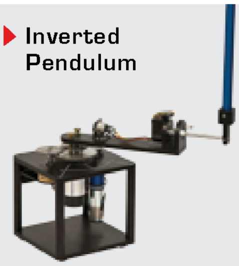
[8] Quanser Inc. SRV02 lab manual. 2011.

Ten modules to teach controls from the basic to advanced level

SRV02 Base Unit

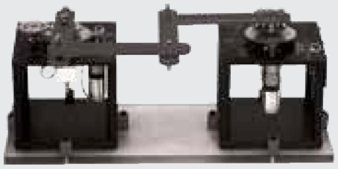
Flexible Link

Inverted Pendulum



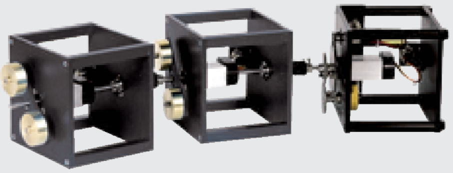
Ball and Beam

2 DOF Robot



2 DOF Inverted Pendulum

Multi-DOF Torsion



Flexible Joint

Gyro/Stable Platform

Double Inverted Pendulum

2 DOF Ball Balancer

With the SRV02 Base Unit, you can select from 10 add-on modules to create experiments of varying complexity across a wide range of topics, disciplines and courses. All of the experiments/workstations are compatible with LabVIEW™ and MATLAB®/Simulink®.

To request a demonstration or a quote, please email info@quanser.com.

QUANSER

INNOVATE. EDUCATE.

INFO@QUANSER.COM +1-905-940-3575 QUANSER.COM

Solutions for teaching and research. Made in Canada.