Experiment 3: Ball and Beam Position Control using QuaRC

Objective: The objective of the Ball and Beam experiment is to stabilize the ball to a desired position along the beam using the remote sensor unit. Using the proportional-derivative (PD) family, a cascade control system will be designed to meet a set of specifications.

Apparatus/software required: MATLAB, Quanser SRV02 unit, Quanser Ball and Beam module, Q8-USB, UPM-2405 amplifier and remote sensor unit.

1. Introduction:

1.1 Description: The Quanser Ball and Beam module, shown in Figure 1, consist a track on which the metal ball is free to roll. The track is fitted with a linear transducer to measure the position of the ball, i.e. it outputs a voltage signal proportional to the position of the ball. One side of the beam is attached to a lever arm that can be coupled to the load gear of the Quanser SRV02 unit. By controlling the position of the servo, the beam angle can be adjusted to balance the ball to a desired position.

1.2 Remote Sensor Option:

The SRV02 Ball and Beam module can also be accompanied by a remote ball sensor called the SS01 module. This permits a master-slave configuration where the ball command is generated by the SS01 instead of through a program.

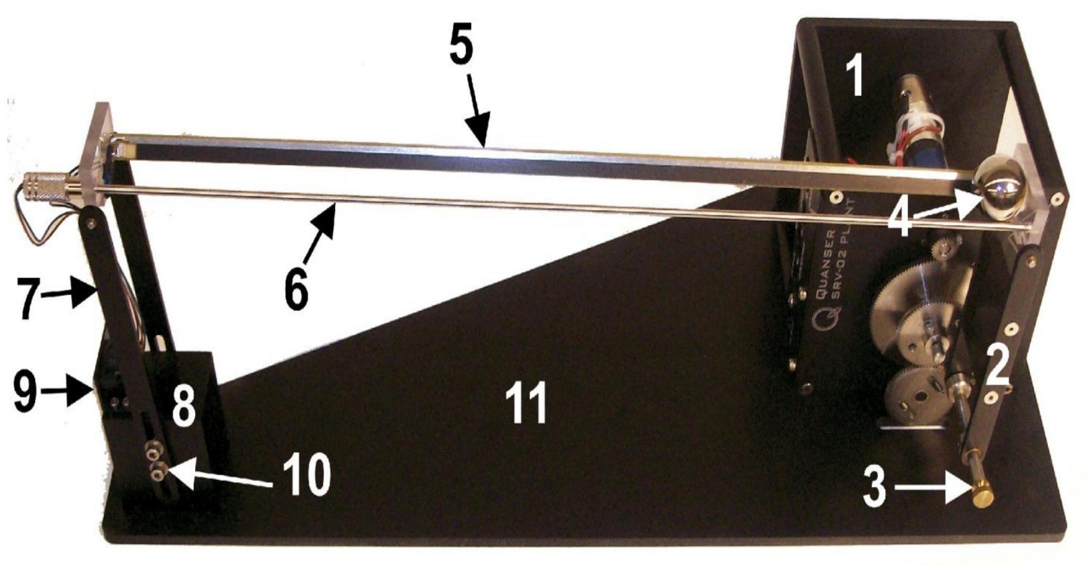


Figure 1: SRV02 Ball and Beam Module with components.

1.3 Ball and Beam Components:

The components of the Ball and Beam module, i.e. the BB01 device, and the Remote Sensor system, i.e. SS01, are listed in figure 3 below and labeled in Figure 1 and Figure 2.

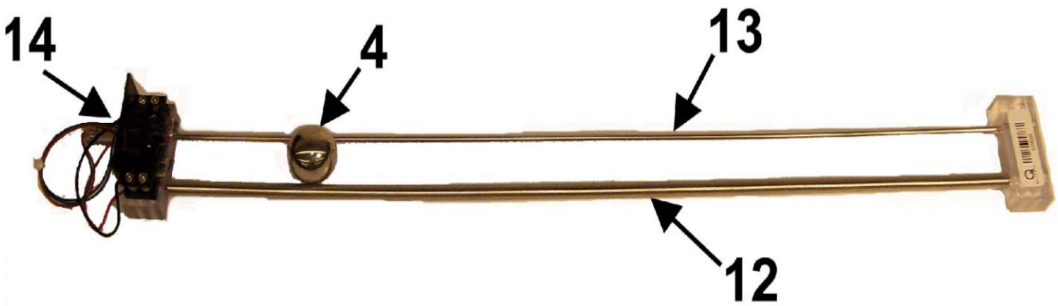


Figure 2: Remote sensor components.

ID #	*Component*	*ID #*	*Component*
1	SRV02	8	Support base
2	Lever arm	9	Support arm screws
3	Coupling screw	10	Analog ball position sensor connector
4	Steel ball	11	Calibration base
5	BB01 Potentiometer sensor	12	SS01 Potentiometer sensor
6	BB01 Steel rod	13	SS01 Steel rod
7	Support arm	14	Analog remote sensor connector

Figure 3: BB01 components.

1.3.1 Ball Position Sensor

The track of the BB01 linear transducer module on which the metal ball is free to roll consists of a steel rod in parallel with a nickel-chromium wire-wound resistor forming the track. The resistive wire is the black strip that is stuck on the plastic which is fastened onto the metal frame. The position of the ball is obtained by measuring the voltage at the steel rod. When the ball rolls along the track, it acts as a wiper similar to a potentiometer resulting in the position of the ball.

1.3.2 Remote Sensor

Similarly to the BB01, the SS01 has a wiper potentiometer sensor that detects the position of the ball.

1.4 Ball and Beam Specifications

Table 1, below, lists and characterizes the main parameters associated with the BB01. See Figure 4 for an illustration of the Ball and Beam dimensions and the variables α, θ, and *x* that are associated with the system. Some of the parameters listed in Table 1 will be used in the mathematical model of the system.

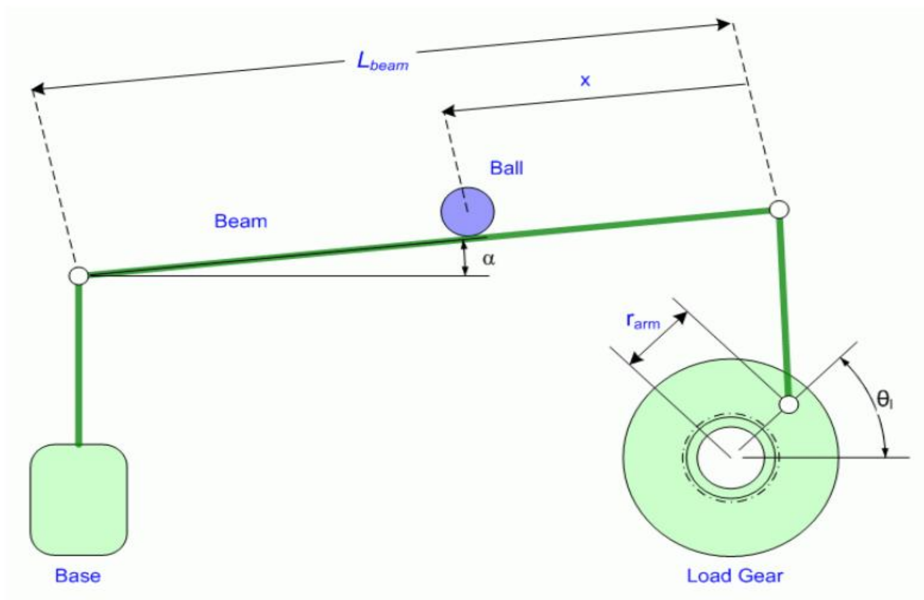


Figure 4: Ball and beam schematic.

Table 1: Ball and beam system specifications.

Symbol	Description	Matlab Variable	Value with Units
	Mass of ball and beam module		0.65 kg
	Calibration base length		50 cm
	Calibration base depth		22.5 cm
L_{beam}	Beam length	L_beam	42.55 cm
	Lever arm length		12.0 cm
r_{arm}	Distance between SRV02 output gear shaft and coupled joint	r_arm	2.54 cm
	Support arm length		16.0 cm
r_b	Radius of ball	r_ball	1.27 cm
m_b	Mass of ball	m_ball	0.064 kg

K_{bs}	Ball position sensor sensitivity	K_BS	-4.25 cm/V
V_{bias}	Ball position sensor bias power		± 12 V
V_{range}	Ball position sensor measurement range		± 5 V

The following topics will be covered in this laboratory:

I. Modelling the dynamics of the ball from first-principles.
II. Obtain a transfer function representation of the system.
III. Design a proportional-velocity (PV) compensator to control the position of the servo load shaft according to certain time-domain requirements.
IV. Design a compensator that regulates the position of the ball on the beam and meets certain specifications. This together with the servo control is the complete Ball and Beam cascade control system.
V. Simulate the Ball and Beam control using the model of the plant and ensure the specifications are met without any actuator saturation.
VI. Implement the controllers on the Quanser BB01 device and evaluate its performance with and without the remote sensor unit.

2. Assignments:

2.1 Modeling of Ball and Beam system:

This system is comprised of two plants: the SRV02 and the BB01 shown in Figure 5.

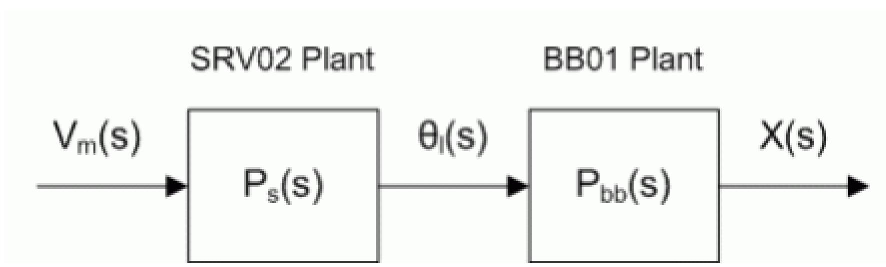


Figure 5: Ball and Beam open-loop block diagram.

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

$$P(s) = P_{bb}(s)P_s(s)$$

Where the BB01 transfer function is

$$P_{bb}(s) = \frac{X(s)}{\theta_l(s)}$$

and the SRV02 transfer function is

$$P_s(s) = \frac{\theta_l(s)}{V_m(s)}$$

The BB01 transfer function describes the displacement of the ball with respect to the load angle of the servo. The time-based motion equations are developed and, from these equations of motion, its transfer function is obtained. The SRV02 voltage-to-load angle plant transfer function was found to be

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

This can be added to the system to get the full SRV02+BB01 model.

2.2 Nonlinear Equation of motion:

The equation describing the motions of the ball, *x*, relative to the angle of the beam, α, will be derived. Thus the equation of motion will be of the form

$$\frac{d^2x(t)}{dt^2} = f(\alpha(t))$$

Where $f(\alpha(t))$ is a nonlinear function. The incomplete free-body diagram of the ball on a beam is illustrated in Figure 6 (a). Applying Newton's Law of Motion, the sum of the forces acting on the ball alongside the beam equals

$$m_b(\frac{d^2x(t)}{dt^2}) = \sum F$$

Where m_b is the mass of the ball.

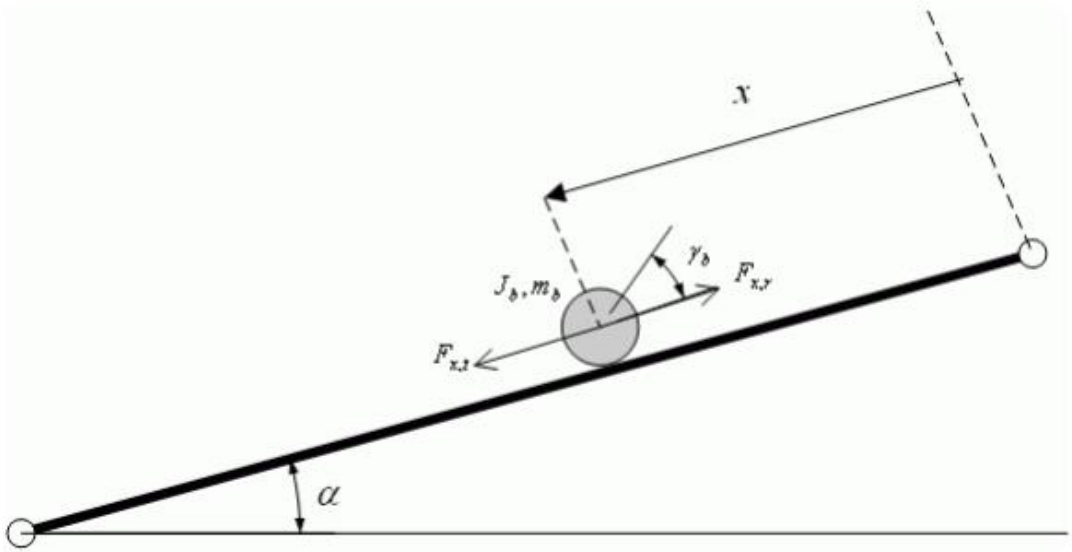


Figure 6(a): Free-body diagram of Ball and Beam.

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

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

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 equivalent to the force produced by gravity.

Assignment 1: Modeling and obtaining the complete SRV02+BB01 transfer function of the Ball and Beam systems.

You can use the following steps to derive the complete transfer function of the Ball and Beam system.

Step-I: Find the force in the *x* direction (along the beam) that is caused by gravity, $F_{x,t}$.

The complete free-body diagram of the ball on the beam is shown in Figure 6 (b).

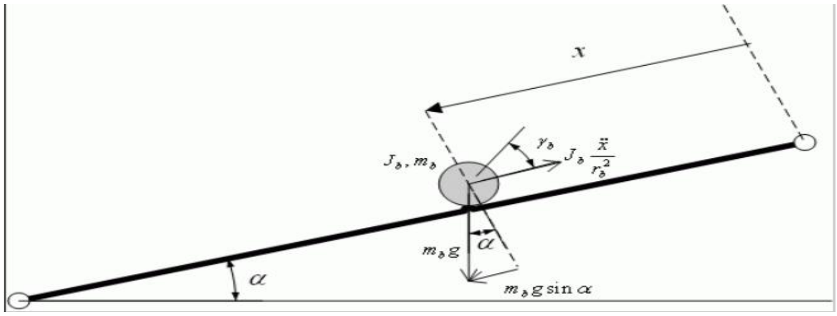


Figure 6(b): Completed Ball and Beam free-body diagram.

Step-II: Find the force that is caused by rotational inertia (momentum) of the ball in the *x* direction, $F_{x,r}$.

Hint: Use the sector formula to convert between linear and angular displacement (e.g. or velocity and acceleration)

$$x(t) = \gamma_b(t) r_b$$

Where γ_b is the angle of the ball and r_b is the ball radius.

The force caused by the rotation spin of the ball is

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

Where r_b is the radius of the sphere and τ_b is the torque. Given that the torque of the ball equals

$$\tau_b = J_b(\ddot{\gamma}_b(t))$$

Step-III: Find the nonlinear equation of motion of the ball and beam by applying Newton's Law of Motion. It should be in the form as

$$\frac{d^2 x(t)}{dt^2} = f(\alpha(t))$$

Where $f(\alpha(t))$ is a nonlinear function.

2.3 Adding SRV02 Dynamics:

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

You can use the following steps to derive the equation of motion representing the position of the ball relative the angle of the SRV02 load gear.

Step-I: Using the schematic given in Figure 4, find the relationship between the SRV02 load gear angle, θ_i , and the beam angle, α.

Step-II: Find the equation of motion that represent the ball's motion with respect to the SRV02 angle θ_1. Linearize the equation of motion about servo angle $\theta_l(t) = 0$.

Step-III: Simplify the expression by lumping the coefficient parameters of $\theta_l(t)$ into parameter K_{bb}. This is the model gain of the Ball and Beam system. Show the new simplified equation of motion. Then, evaluate the model gain numerically using the Ball and Beam parameters given in table 1.

Hint: Recall that the moment of inertia of a solid sphere is

$$J = \frac{2mr^2}{5}$$

Where m is the mass of the ball and r is its radius.

2.4 Obtaining Transfer Function

The transfer function describing the servo voltage to ball position displacement will be calculated using the following steps as

Step-I: Find the transfer function $P_{bb}(s)$ of the BB01. Assume all initial conditions are zero.

Step-II: Find the complete SRV02+BB01 process transfer function P(s).

3. Control Design for Desired Control Response:

3.1 Time-Domain Specifications:

The time-domain specifications for controlling the position of the SRV02 load shaft are: The steady-state error (e_{ss}) = 0, Peak time (t_p) = 0.15 sec and Percentage Overshoot (PO) = 5.0 %.

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 overshoot less than or equal to 5 %, and no steady-state error.

The specifications for controlling the position of the ball are: Steady-state error

(e_{ss}) ≤ 0.005 meter, Setting time (t_s) = 0.15 sec, time (c_{ts}) = 0.04 and Percentage Overshoot (PO) = 10.0 %.

Given a step reference, the peak position of the ball should not overshoot over 10%. After 3.5 seconds, the ball should settled 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.

Consider the second-order system as

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

3.2 Settling Time:

The response of a second-order system, y(t), when subjected to a unit step reference, r(t), is shown in Figure 7. This response has a 5% settling time of 0.30 seconds. Thus the response settles within 5% of its steady-state value, which is between 0.95 and 1.05, in 0.30 seconds. Settling time is defined as

$$t_s = t_1 - t_0$$

Where the initial step time is t_0 and the time it takes to settle is t_1.

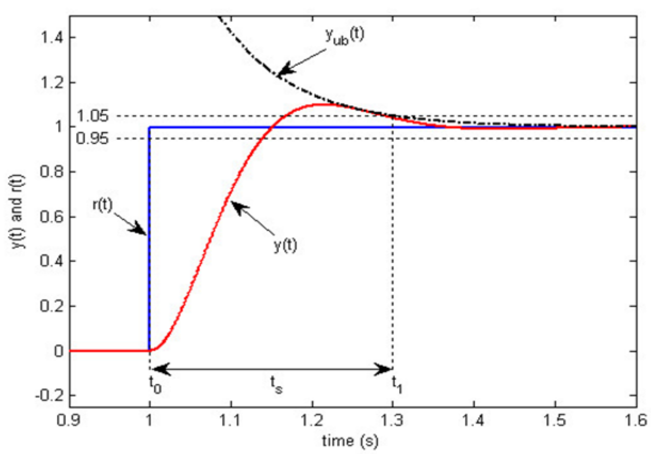


Figure 7: Settling time of unit step response.

The settling time is given as

$$t_s = -\frac{ln(c_{ts}(1-\zeta^2)^{0.5})}{\zeta\omega_n}$$

Where c_{ts} is the settling time percentage.

The peak time and percentage overshoot equations are

$$t_p = \frac{\pi}{\omega_n (1-\zeta^2)^{0.5}}$$

and

$$PO = 100 e^{\left(-\frac{\pi\zeta}{(1-\zeta^2)^{0.5}}\right)}$$

3.3 Steady-State Error: The steady-state error of the ball position is evaluated using a proportional compensator. Find the steady-state error the ball and beam, $P_{bb}(s)$, with a unity compensator $C(s)=1$ and a reference step of $R(s) = \frac{R_0}{s}$ where R_0 is the step amplitude. **In this calculation the SRV02 dynamics are ignored and only the BB01 plant is being considered**.

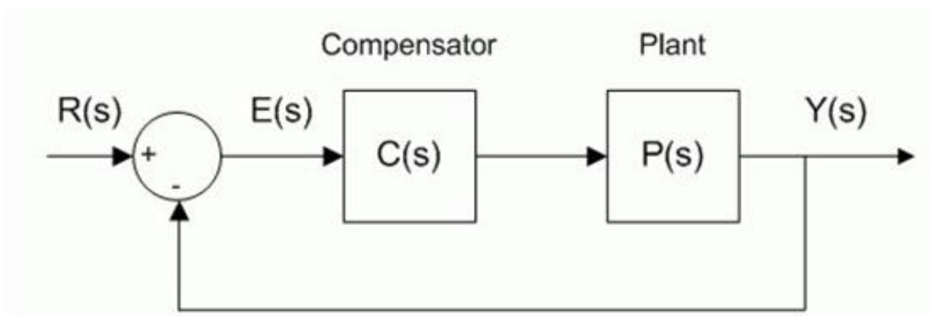


Figure 8: Unity feedback system.

4. Ball and Beam Cascade Control Design

The cascade control that will used for the SRV02+BB01 system is illustrated by the block diagram given in Figure 9. Based on the measured ball position X(s), the ball and beam compensator, $C_{bb}(s)$, in the outer-loop computes the servo load angle needed, $\Theta_d(s)$, to attain the desired ball position, $X_d(s)$.

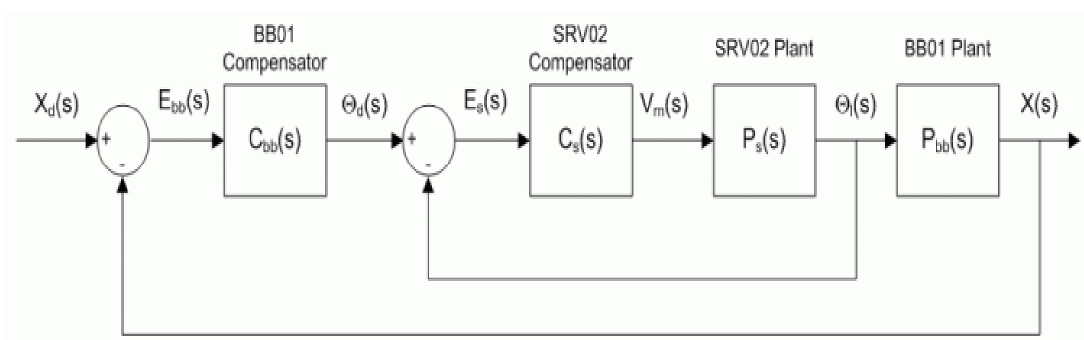


Figure 9: Cascade control system used to control ball position in SRV02+BB01 plant.

The inner loop is a servo position control system as described in next section. Thus the compensator Cs(s) calculates the motor voltage required for the load angle to track the given desired load angle.

In next Section, the position controller for the SRV02 is designed, and a compensator is introduced and it is assessed using root locus whether it can be used to meet the desired specifications. Two different variations of a compensator are designed in Section.

4.1 Inner Loop Design: SRV02 PV Position Controller

In this section, the proportional-velocity (PV) controller gains are computed for the SRV02 when it is in the high-gear configuration and based on the specifications given in Section 2. The internal control loop is depicted in the block diagram shown in Figure 10.

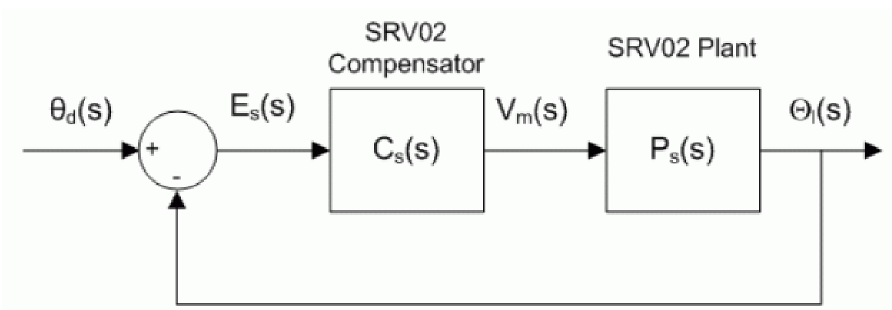


Figure 10: SRV02 closed-loop system.

Assignment 2:. The nominal model parameters, K and τ, when the SRV02 is in high-gear configuration are K = 1.76 rad/sec V and τ = 0.0285 sec.

Given these parameters, **calculate the minimum damping ratio and natural frequency required to meet the SRV02 specifications given in Section 3.1.**

SRV02 Compensator: Proportional Velocity (PV) Compensator:

The PV compensator used to control the position of the SRV02 has the structure

$$V_m(t) = k_p(\theta_d(t) - \theta_l(t)) - k_v(\dot{\theta}_d(t))$$

Where k_p is the proportional control gain, k_v is the velocity control gain, $\theta_d(t)$ is the setpoint or a reference load angle, and $\theta_l(t)$ is the measured load shaft angle, and $V_m(t)$ is the SRV02 input voltage. The block diagram of PV control is illustrated in Figure 10 (a).

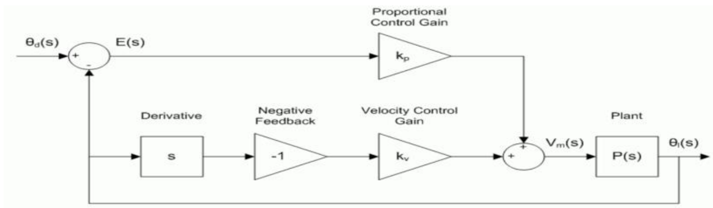


Figure 10 (a): Block diagram of SRV02 PV position control.

Assignment 3: Using Time-Domain Specifications given in section 3.1 and Fig. 10 (a), find the closed-loop SRV02 position control transfer function, $\frac{\theta_l(s)}{\theta_d(s)}$, and **compare the charatristics equations of the transfer function of second order system in general and ball and beam transfer function**, **also calculate the control gains needed to satisfy the time-domain response requirements.**

4.2 Outer Loop Design

The inner loop that controls the position of the SRV02 load shaft is complete and the servo dynamics are now considered negligible. Thus, it is assumed that the desired load angle equals the actual load angle

$$\theta_l(t) = \theta_d(t)$$

The outer-loop shown in Fig. 11 will be used to control the position of the ball on the beam.

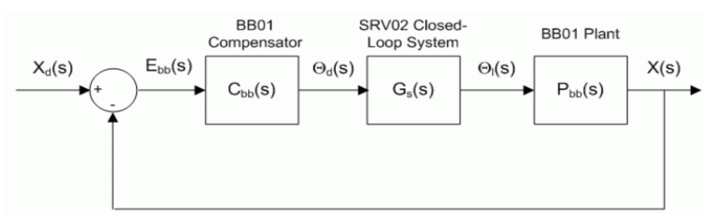


Figure 11: BB01 closed-loop system.

Assignment 4: **Using Figure 11 and 11 (a), find the closed-loop transfer function of the ball and beam,** $\frac{X(s)}{X_d(s)}$ **, and gain (Kc) of the BB01 system.**

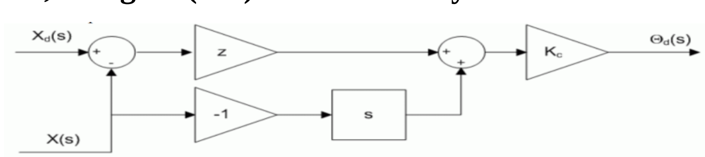


Figure 11 (a): BB01 ideal PV compensator.

Where Kc is the proportional gain and z is the location of the compensator zero given as.

$$K_c = \frac{2\zeta\omega_n}{K_{bb}} \quad \text{and} \quad z = \frac{\omega_n^2}{K_{bb}K_c}$$

5. Lab Procedures:

Assignment 5: Simulate the closed-loop response of the ball and beam using its full cascade control system.

Assignment 6: Implement the closed-loop response of the actual BB01 device using its full cascade control system.

Assignment 7: Control the position of the ball on the BB01 using the developed practical PD control with the set-point is given by the remote sensor.

5.1 Position Control Simulation

To simulate the Ball and Beam system using the designed compensator and ensure it meets the specifications listed in Section 3.1. The ball and beam is simulated using its full cascade control system includes both the outer ball position control loop and the inner servo position control feedback loop.

5.1.1 Cascade Control Simulation

The servo dynamics can now be added and the closed-loop position response when using the cascade control system will be simulated using the Simulink diagram pictured in Figure 12. This Simulink model simulates the block diagram shown in Figure 9.

The ***s_bb01_pos_.mdl*** Simulink diagram shown in Figure 12 is used to simulate the closed-loop position response of the BB01 when using the cascade system. The response is simulated using the developed nonlinear model of the Ball and Beam.

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 will be used to implemented in the *SRV02 PV Position Control* block.

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 ±10 V.

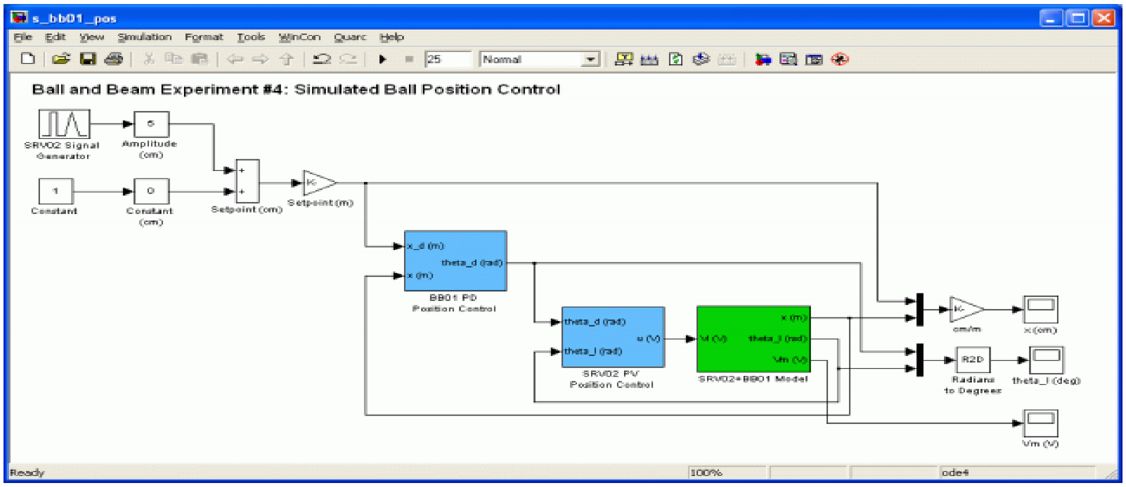


Figure 12: Simulink diagram used to simulate cascade control system

The *BB01 Nonlinear Model* subsystem includes the $P_{bb}(s)$ transfer function that was derived in Section 2. Recall previous Section that 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 previous Section. **It includes a *Saturation* block that limits the SRV02 angle between ±56 degrees.**

5.1.2 Setup for Position Control Simulation

Follow these steps to configure the lab properly:

Step-1 Load the Matlab software.

Step-2 Browse through the *Current Directory* window in Matlab and find the folder that contains the BB01 controller files.

Step-3 Double-click on the "s *_bb01_pos.mdl*" file to open the Simulink diagram shown in Figure 10.

Step-4 Double-click on the " *setup_srv02_exp04_bb01.m*" file to open the setup script for the BB01 Simulink models.

Step-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_CABLE, UPM_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'.

Step-6 Run the script by selecting the Debug | Run item from the menu bar or clicking on the *Run* button in the tool bar. The messages shown in Text 1, below, should be generated in the Matlab Command Window. The model parameters and specifications are loaded but the SRV02 PV gains and compensator gain are all set to zero and the compensator pole and zero are set to 1 – they need to be changed.

```
SRV02 model parameters:
   K    = 0 rad/s/V
   tau = 0 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
```

Text 1: Display message shown in Matlab Command Window after running setup_srv02_exp04_bb01.m.

Figure 13: Display message shown in the Matlab Command Window.

5.1.3 Practical PD Cascade Simulation

The *practical PD* controller is used to simulate the practical cascade system. 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:

Step-1 Enter the BB01 model gain given in table-2 in Matlab as variable *Kbb*.

Step-2 Enter the *practical PD* compensator gain *Kc*, and zero, *z*, which are given in table-2. The filter cutoff filter, *wf*, is already set by the script.

Step-3 Follow steps 2-6 in Section 5.1.2 to setup the SRV02 model parameters and control gains and setup the Simulink diagram.

Step-4 To simulate using the *practical PD* controller, set the *Manual Switch* in the *BB01 PD Position Control* subsystem to the downward position.

Step-5 Using Matlab, plot the root locus of BB01 loop transfer function when using the *practical PD* compensator and attach it to your report. As in Figure 10, show the desired locations of the poles on the plots and ensure the poles go through the desired locations at the gain that was computed.

Step-6 Open the ball position scope *x (m)*, the load shaft position scope *theta_l (deg)*, and the SRV02 motor input voltage scope *Vm(V)*.

Step-7 Start the simulation. By default, the simulation runs for 25.0 seconds. The scopes should be displaying responses similar to figures 14, 15, and 16.

Step-8 Generate a Matlab figure showing the *practical cascade* ball position, servo angle, and servo input voltage response and attach it to your report.

Step-9 Measure the steady-state error, the settling time, and the percentage overshoot of the simulated *practical cascade PD* control response.

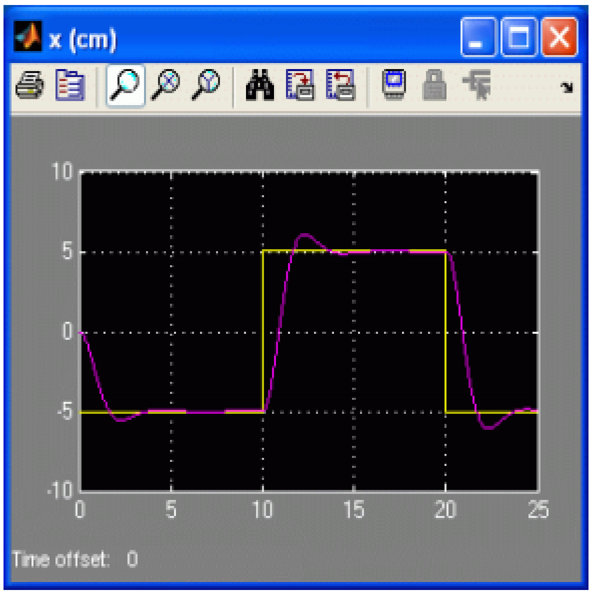
Step-10 Does the simulated response satisfy the specifications given in Section 3 while keeping the servo angle between ±56.0 degrees and the servo voltage between ±10.0 V?

Step-11 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 *Kc*, *z*, and *Tp*, generated according to a percentage 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 percentage overshoot, settling time, and filter cutoff frequency specifications. Then simulate the system and see if the specifications are satisfied.

Step-12 Record the gain and zero have been fine-tuned for the response to meet the specifications along with the new specifications used to generate those control parameters. These control parameters will be called the *Tuned Practical PD #1*.

Step-13 Plot the simulated response in a Matlab figure and attach it in your report.

Step-14 Give the resulting steady-state error, settling time, and percentage overshoot of the response.



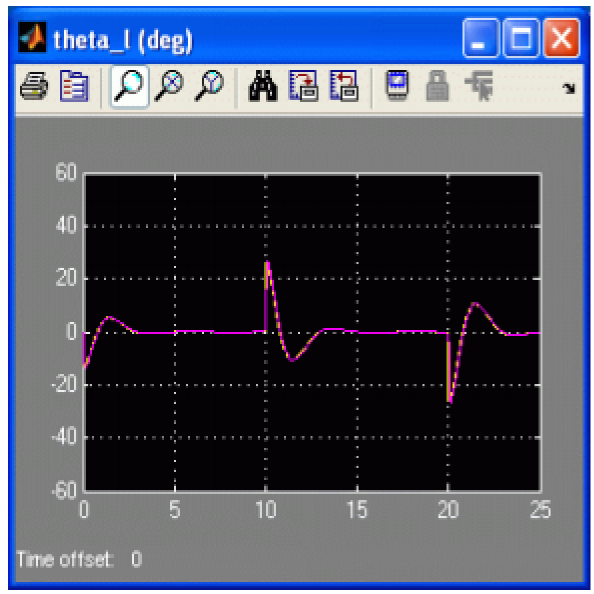


Figure 14: Practical cascade control ball position Figure 15: Practical cascade control servo angle

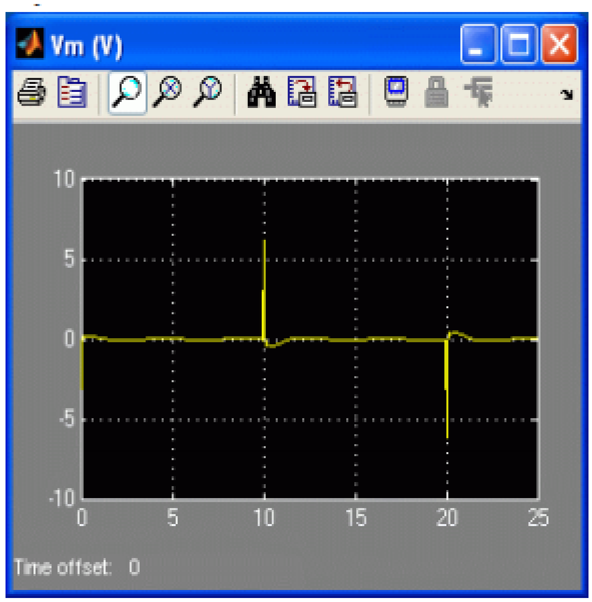


Figure 16: Practical cascade control input voltage

5.2 Position Control Experimental

The ***q_bb01_pos.mdl*** Simulink diagram shown in Figure 17 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 by using the *practical PD* control.

Go through the steps in Section 5.1 to setup the Matlab workspace. The procedure to run the developed *practical PD* controller is outlined in Section 5.2.

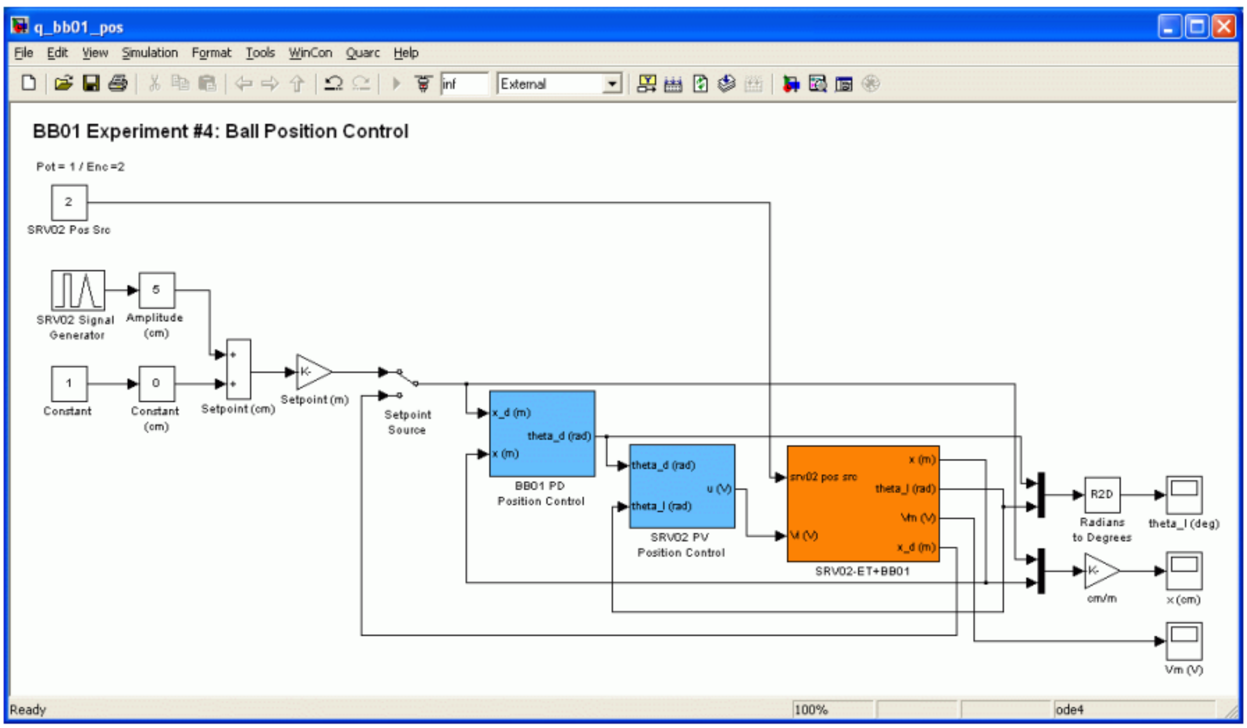


Figure 17: Simulink model used with QuaRC to run the practical PD controller on the Ball and Beam system

5.2.1 Setup for Position Control Implementation

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

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

Setp-1 Setup the SRV02 with the BB01 module as detailed as discussed earlier.

Step-2 Load the Matlab software.

Step-3 Browse through the *Current Directory* window in Matlab and find the folder that contains the QuaRC BB01 control file "**q_bb01_pos.mdl**".

Step-4 Double-click on the "***q_bb01_pos.mdl***" file to open the Ball and Beam Position Control Simulink diagram shown in Figure 17.

Step-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.

Step-6 **Configure Sensor**: The position of the load shaft can be measured using various sensors. Set the *Pos Src* Source block in ***q_bb01_pos.mdl***, as shown in Figure 17, as follows:

- 1 to use the potentiometer
- 2 to use to the encoder

Note that when using the potentiometer, there will be a discontinuity.

Step-7 **Configure Setpoint**: The set-point can be generated through the SRV02 Signal Generator Simulink block or via the SS01 device. Place the *Setpoint Source* switch to the UP position in order to generate the set-point through the Simulink model.

Step-8 **Configure setup script**: Set the parameters in the ***setup_srv02_exp04_bb01.m*** script according to your system setup.

5.2.2 Running the Practical PD Controller

In this section, the position of the ball on the BB01 device will be controlled using the developed control. Measurements will then be taken to ensure that the specifications are satisfied.

Follow the steps below:

Setp-1 Enter the BB01 model gain given in table-2 in Matlab as variable *Kbb*.

Step-2 Enter the *Tuned Practical PD #1* compensator gain *Kc*, and zero, *z*, which are given in table-2.

Step-3 Follow steps 2-6 in Section 5.1 to setup the SRV02 model parameters and control gains and setup the Simulink diagram.

Step-4 Click on QuaRC | Build to compile the Simulink diagram.

Step-5 Select QuaRC | Start to begin running the controller. The scopes should be displaying responses similar to figures 18, 19, and 20.

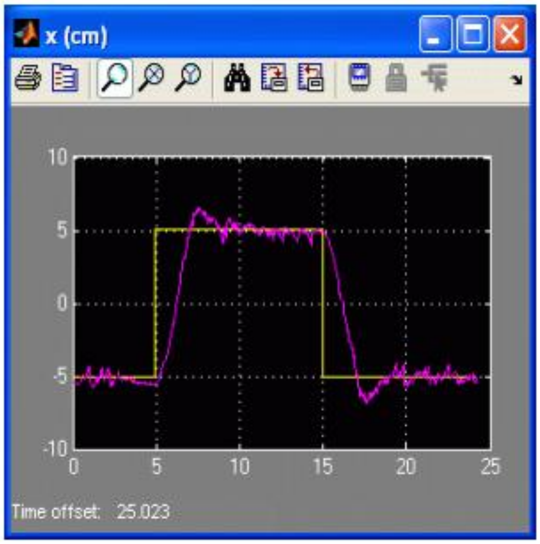


Figure 18: BB01 ball position response.

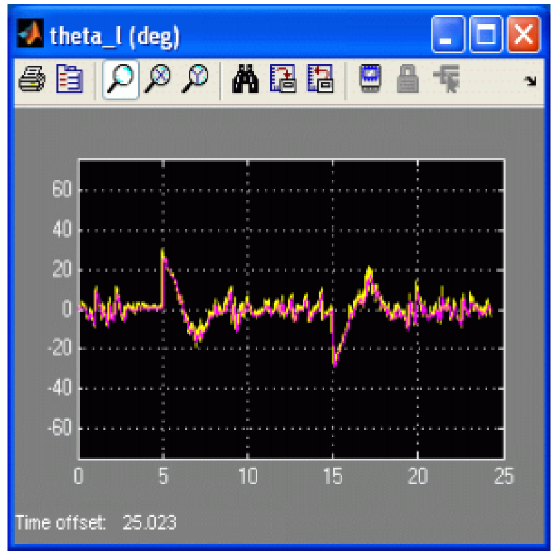


Figure 19: BB01 servo angle response.

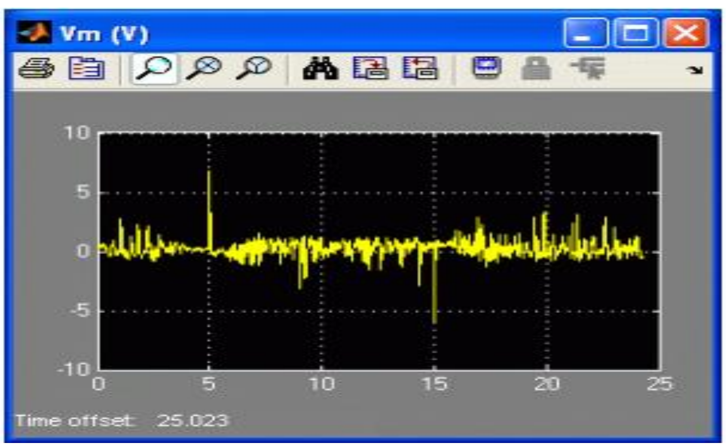


Figure 20: BB01 servo input voltage.

Step-6 When a suitable response is obtained, click on the *Stop* button in the Simulink diagram toolbar (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. Attach it to your report. As in the ***s_bb01_pos.mdl*** Simulink diagram, each scope automatically saves their response to a variable in the Matlab workspace when the controller is stopped.

Step-7 Measure the steady-state error, the settling time, and the percentage overshoot. Does the response satisfy the specifications given in Section 3.

Step-8 If the specification have not been satisfied, the control parameters need to be fine tuned 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 until the error is minimized. 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. This is called the *Tuned Practical PD #2* control.

Step-9 Plot the response using the *Tuned Practical PD #2* control in a Matlab figure and attach it to your report.

Step-10 Make sure QuaRC is stopped.

Step-11 Shut off the power of the UPM if no more experiments will be performed on the SRV02 in this session.

5.3 Controlling using the Remote Sensor (Optional)

In this section, the position of the ball on the BB01 device will be controlled using the developed practical PD control, but the set-point is given with the remote sensor.

Follow the steps below:

1. Follow steps 1-4 in Section 5.2.2to 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 set-point using the SS01 module.
3. Select QuaRC | Start to begin running the controller.
4. Move the ball back and forth on the remote sensor and observe the response obtained in the scopes. Figures 21, 22, and 23 show a sample response.
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.

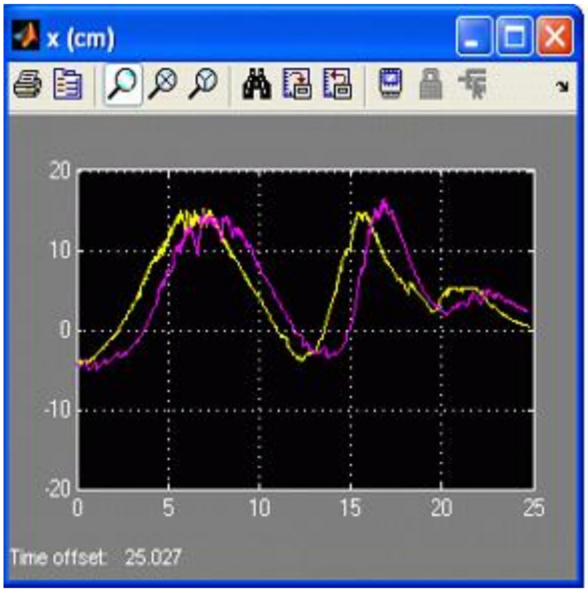


Figure 21: BB01 ball position response with SS01

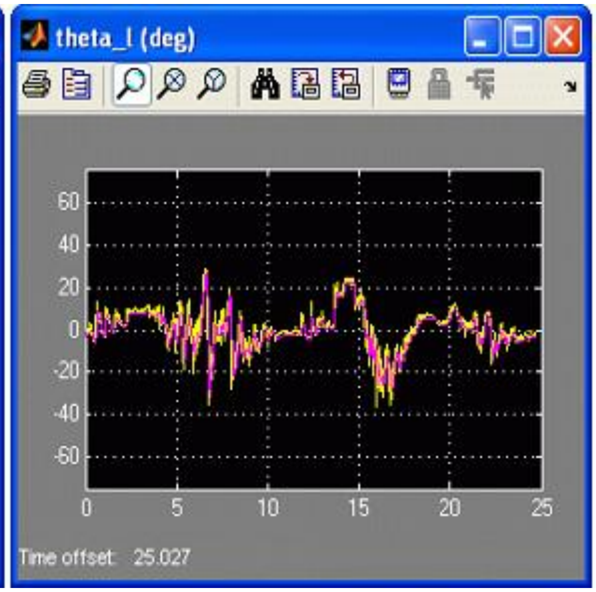


Figure 22: BB01 servo angle response with SS01

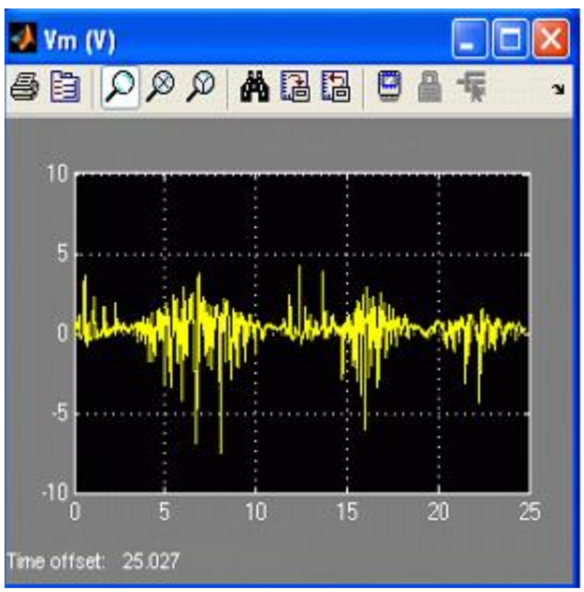


Figure 23: BB01 servo input voltage with SS01

6. Results Summary:

Table 2, below, summarizes the parameters which will be used in this experiment.

Table 2: Ball and Beam control parameters

Description	Symbol	Value (Units)
Pre-Lab: Model Parameters		
Open-Loop Steady-State Gain	K	1.76 rad/(V.s)
Open-Loop Time Constant	τ	0.0285 s
Pre-Lab: PV Gain Design		
Proportional gain	K_p	13.5 V/rad
Velocity gain	k_v	0.078 V.s/rad
Pre-Lab: Ideal PD Control Design		
Compensator Gain	K_c	4.68 rad/m
Compensator Zero	Z	1.41 rad/s
Pre-Lab: Practical PD Control Design		
Compensator Gain	K_c	3.56 rad/m
Compensator Zero	Z	1.28 rad/s
Compensator Pole Time Constant	T_p	0.231 s
In-Lab Simulation: outer-loop Ideal PD		
Steady-state error	e_{ss}	0.00 cm
Settling time	T_s	3.59 s
Percentage overshoot	PO	10.0 %
In-Lab Simulation: Cascade Ideal PD		
Steady-state error	e_{ss}	0.00 cm
Settling time	T_s	3.45 s
Percentage overshoot	PO	10.9 %
In-Lab Simulation: Cascade Practical PD		
Steady-state error	e_{ss}	0.00 cm
Settling time	T_s	3.47 s
Percentage overshoot	PO	11.0 %

In-Lab Simulation: Cascade Tuned Practical PD		
Compensator Gain	K_c	3.96 rad/m
Compensator Zero	Z	1.26 rad/s
Steady-state error	e_{ss}	0.00 cm
Settling time	T_s	3.2 s
Percentage overshoot	PO	7.99 %
In-Lab Implementation: Tuned Practical PD		
Steady-state error	e_{ss}	-0.225 cm
Settling time	T_s	3.72 s
Percentage overshoot	PO	12.9 %
In-Lab Implementation: Tuned#2 Practical PD		
Compensator Gain	K_c	3.95 rad/m
Compensator Zero	Z	1.20 rad/s
Steady-state error	e_{ss}	-0.396 cm
Settling time	T_s	3.09 s
Percentage overshoot	PO	9.16 %
Integral Gain	K_i	1.50 rad/m/s

In Lab-Report

1. **Objective**
2. **Apparatus/software required**
3. **Brief Theory**
4. **All Assignments**
5. **Results and Discussions**
6. **Conclusions**