

EE-371 CONTROL SYSTEMS LABORATORY

Assignment 9 Ball and Beam Design Project

Purpose

The objectives of this project are:

- To obtain a mathematical model and a linearized model of the ball and beam system.
- Design two controllers to control the position of the ball along the track make it follow a commanded input.
- Build a Simulink model of the plant and simulate off line
- Build the WinCon application, implement, test and analyze the system on the real-time hardware
- To use the remote sensor for making the Ball on the beam follow the Ball on the remote sensor for operation in master / slave tracking mode.

Introduction

The purpose of the design is to control the position of the ball along the track by manipulating the angular position of the servo. The system is often used as a bench mark problem for many different control schemes such as the control model in the rocket toppling control system, where a feedback system is used to prevent rockets to topple out of balance during launch by forces and moments that could perturb the vertical motion.

The Ball and Beam system is driven by the servomotor as shown in Figure 9.1. The beam consists of a steel rod in parallel with a nickel-chromium wire-wound resistor forming a track upon which a metal ball is free to roll. One end of the beams is coupled to the servomotor through a lever arm and gears and the other end is fixed. The wire-wound resistor is biased and when the ball rolls along the track it acts as wiper similar to a potentiometer. The position of the ball along the track is obtained by measuring the voltage at the steel rod.

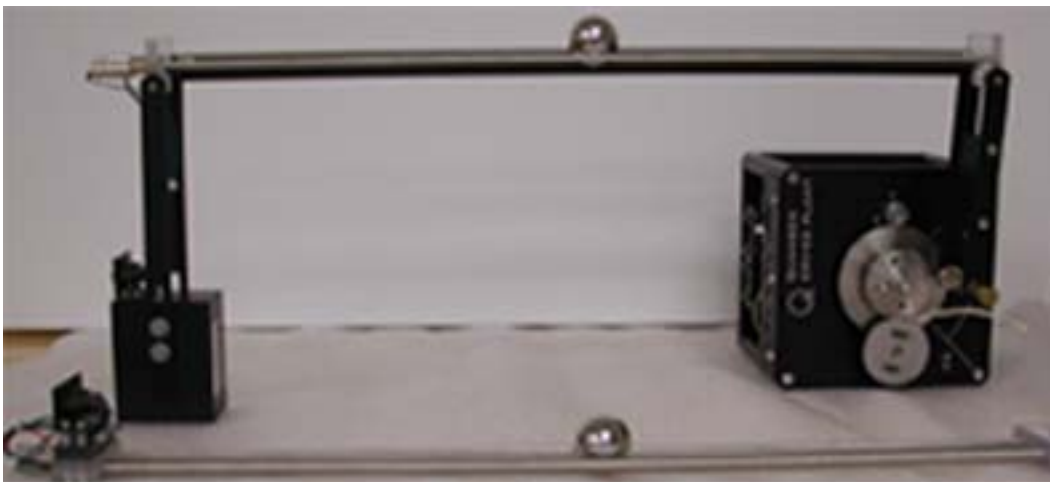


Figure 9.1: Ball and beam module coupled to the servo plant.

An optional remote sensor is also available for operation in master / slave tracking mode. This sensor is used, as the input to the Ball & Beam desired position, making the Ball on the beam follow the ball on the remote sensor.

The first step in the design of a control system is the mathematical modeling of the physical system as follows

1. Mathematical Modeling

1.1 Ball and beam Model

The parameters of the ball and beam module shown in Figure 9.2 are defined as follows:

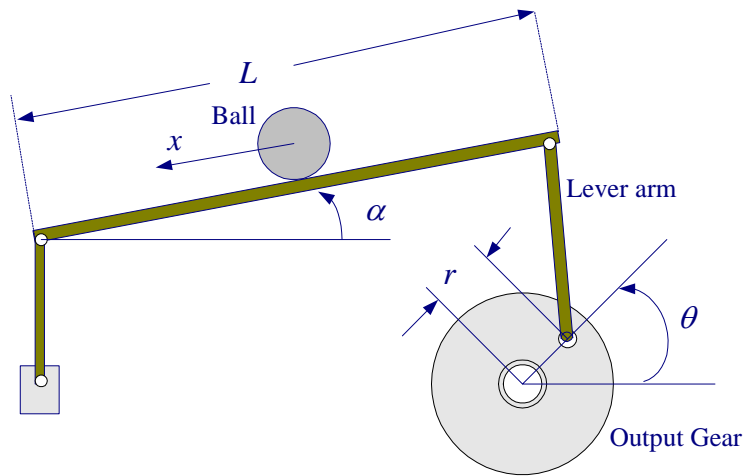


Figure 9.2 Simplified ball and beam schematic

L	Beam length ($L = 16.75$ cm)
r	Lever arm offset ($r = 1$ in)
α	Beam angle coordinate
θ	Servo gear angle
m	Mass of Ball
R	Radius of the ball
J	Ball's moment of inertia ($J = \frac{2}{5} mR^2$)
g	Gravitational acceleration ($g = 9.8$ m/s ²)

Consider the free-body diagram shown in Figure 9.3. Neglecting frictional force, two forces influencing the motion of the ball are

F_{tx}	Force due to translational motion
F_{rx}	Force due to ball rotation

Denoting the acceleration $\frac{d^2x}{dt^2}$ along x as \ddot{x} , force due to translational motion is

$$F_{tx} = m\ddot{x}$$

Torque due to the ball rotation is

$$T_r = F_{rx}R = J \frac{d\omega_b}{dt} = J \frac{d(v_b/R)}{dt} = J \frac{d^2(x/R)}{dt^2} = \frac{J}{R} \ddot{x}$$

or

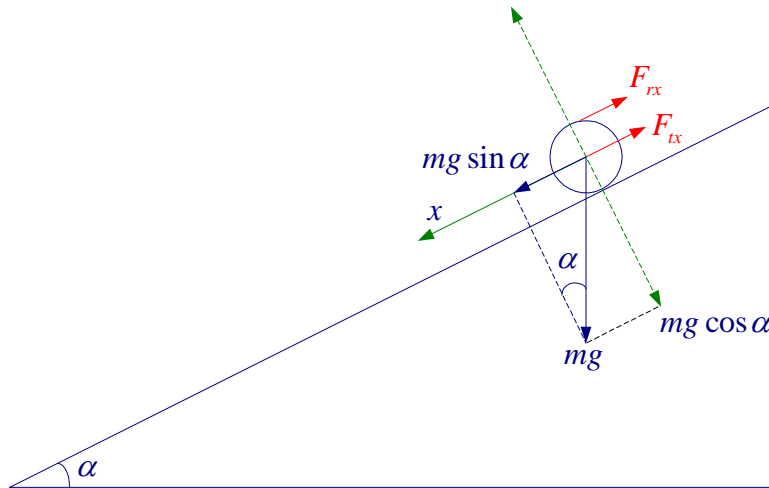


Figure 9.3 Rolling ball free-body diagram

$$F_{rx} = \frac{J}{R^2} \ddot{x}$$

Substituting for the moment of inertia of the ball (Sphere $J = \frac{2}{5}mR^2$), we get

$$F_{rx} = \frac{2}{5}m\ddot{x}$$

Applying the Newton's second law for forces along the inclination, we have

$$F_{rx} + F_{tx} = mg \sin \alpha \quad \Rightarrow \quad \frac{2}{5}m\ddot{x} + m\ddot{x} = mg \sin \alpha$$

or

$$\ddot{x} = \frac{5}{7}g \sin \alpha \tag{9.1}$$

We use the above nonlinear equation in the Simulink model. We also linearized the above equation to obtain a transfer function of the ball and beam controller design. For small angle, $\sin \alpha = \alpha$, and (9.1) becomes

$$\ddot{x} = \frac{5}{7}g\alpha \tag{9.2}$$

Taking the Laplace transform of (9.2), we find

$$\frac{X(s)}{\alpha(s)} = \frac{(5/7)g}{s^2} \quad (9.3)$$

Equating the arc distance traversed by the gear at radius r with the arc distance traversed by the beam at radius L , we have $r\theta = L\alpha$ or

$$\alpha = \frac{r}{L}\theta \quad (9.4)$$

The Simulink block diagram for the rolling ball dynamic as given by the nonlinear equation (9.1) is as shown in Figure 9.4.

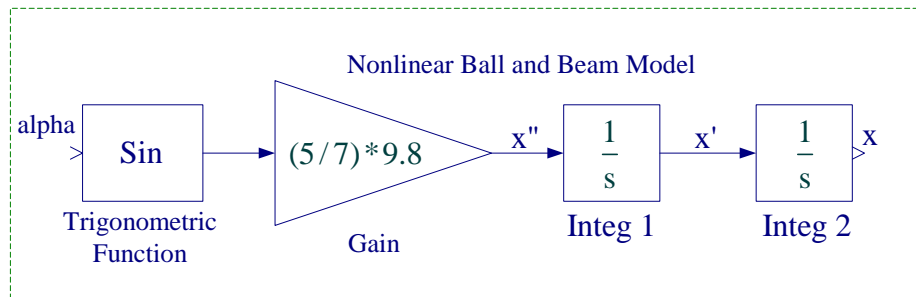


Figure 9.4 Simulink block diagram for the rolling ball model.

Create a subsystem with α as inport and x as the output in radians. To do this, enclose the blocks within a bounding box as shown in Figure 9.4. Choose Create Subsystem from the Edit menu. Edit in1 an out1 ports as alpha and x and rename the subsystem as Rolling Ball Model, and obtain the subsystem shown in Figure 9.5.

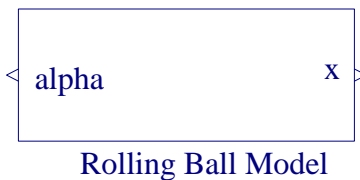


Figure 9.5 Simulink Subsystem for the rolling ball model.

1.2 Servomotor Model

In the position control experiment (Lab 5) the motor-load transfer function with position as output was found to be

$$\frac{\theta_o(s)}{V_i(s)} = \frac{\frac{\eta K_m K_g}{R_a J_{eq}}}{s \left(s + \frac{B_{eq}}{J_{eq}} + \frac{\eta K_m^2 K_g^2}{R_a J_{eq}} \right)} \quad (9.5)$$

or

$$\frac{\theta_o(s)}{V_i(s)} = \frac{a_m}{s(s+b_m)} \quad (9.6)$$

Where

$$a_m = \frac{\eta K_m K_g}{R_a J_{eq}}, \quad b_m = \frac{B_{eq}}{J_{eq}} + \frac{\eta K_m^2 K_g^2}{R_a J_{eq}} \quad (9.7)$$

In this project the servomotor is arranged for the high gear ratio as shown in Figure 9.6. For this configuration the gear ratio is $K_g = (14)(5)$.



Figure 9.6 High gear ratio configurations

The system parameters are as follows:

Armature resistance, $R_a = 2.6 \Omega$

Motor voltage constant, $K_m = 0.00767 \text{ V-s/rad}$

Motor torque constant, $K_t = 0.00767 \text{ N-m/A}$

Armature inertia, $J_m = 3.87 \times 10^{-7} \text{ Kg m}^2$

Tachometer inertia, $J_{tach} = 0.7 \times 10^{-7} \text{ Kg m}^2$

High gear ratio, $K_g = (14)(5)$

Equivalent viscous friction referred to the secondary gear $B_{eq} = K_g^2 B_m + B_L \approx 4 \times 10^{-3} \text{ Nm/(rad/s)}$

Motor efficiency due to rotational loss $\eta_{mr} \approx 0.87$

Gearbox efficiency, $\eta_{gb} \approx 0.85$

$\eta = \eta_{mr} \eta_{gb} = (0.87)(0.85) = 0.7395$

Gear inertia:

$$J_{120} = 4.1835 \times 10^{-5} \text{ Kg m}^2$$

$$J_{72} = 5.4435 \times 10^{-6} \text{ Kg m}^2$$

$$J_{24} = 1.0081 \times 10^{-7} \text{ Kg m}^2$$

$$\text{Load inertia, } J_L = J_{120} + 2(J_{72}) + J_{24} = 5.2823 \times 10^{-5} \text{ Kg m}^2$$

$$J_{eq} = K_g^2 (J_m + J_{tach}) + J_L = (14 \times 5)^2 (3.87 + 0.70) \times 10^{-7} + 5.2823 \times 10^{-5} = 0.0023 \text{ Kg m}^2$$

Control system strategy and controller design

The ball and beam module provides two measurements: the encoder provides the servomotor output $\theta_o(t)$, and the beam sensor provides the ball's displacement $x(t)$. The control system design requires two feedback loops; one inner loop for the motor angle control, which actuates the beam and an outer loop for ball position control. In this project a proportional gain and a rate gain is to be designed for each feedback loops. Therefore two controllers are to be designed. The objective of the servo controller is to make the servomotor output $\theta_o(t)$ follow the input angle, $\theta_i(t)$. The objective of the ball and beam controller is to make the ball position $x_o(t)$ follow the reference position $x_i(t)$. If the servomotor controller is made to respond sufficiently faster than the ball and beam controller, these controllers can be designed independently.

2. Pre Laboratory Assignment

2.1 Servomotor controller Design

Evaluate a_m , and b_m of the plant transfer function as given by (9.7) and obtain the plant transfer function. In this project, the high gear ratio is used, i.e., $K_g = (14)(5)$, and $B_{eq} = 4 \times 10^{-3}$ Nm/(rad/s). Therefore, a_m , and b_m are different from those evaluated in Lab5.

As in position control (Lab 5) we use a rate feedback and a position feedback given by

$$V_i(s) = K_p[\theta_i(s) - \theta_o(s)] - K_D\Omega_o(s) \quad (9.8)$$

This feedback loop is shown in Figure 9.7. The purpose of this system is to have the output angle, $\theta_o(t)$ follow the input angle, $\theta_i(t)$. Design this controller to meet the following time-domain specifications:

- Step response damping ratio $\zeta = 0.8$
- Step response peak time $t_p = 0.1$ second

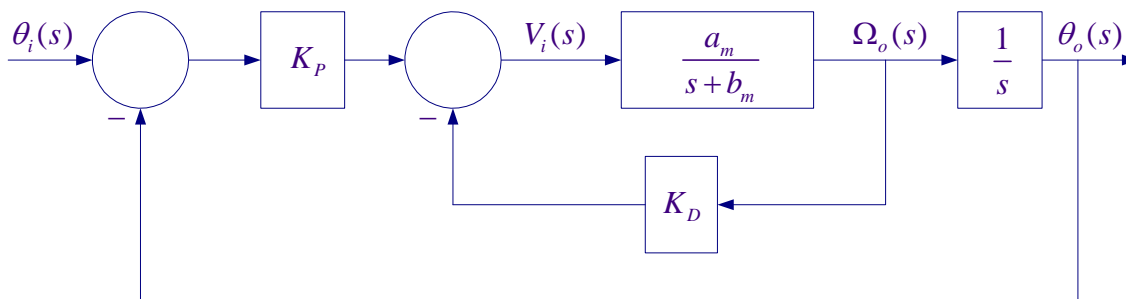


Figure 9.7 Position Control using position and rate feedback.

Applying the Mason's gain formula, the overall transfer function becomes

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{K_p a_m}{s^2 + (K_D a_m + b_m)s + K_p a_m} \quad (9.9)$$

Substitute for a_m , and b_m , and obtain the closed loop transfer function in terms of the numerical values.

$$\frac{\theta_o(s)}{\theta_i(s)} =$$

The second-order response peak time t_p , is given by

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad (9.10)$$

The servo motor transfer function has the same form as the standard second-order transfer function

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (9.11)$$

Comparing the plant characteristic equation given in (9.9) with the standard second-order characteristic equation in (9.11), find two equations for K_p and K_D in terms of a_m, b_m, ζ , and ω_n . Substitute for the parameters and obtain the values of K_p , and K_D for the above design specifications.

$$K_p = \underline{\hspace{2cm}}, \quad K_D = \underline{\hspace{2cm}} \quad (9.12)$$

The SIMULINK simulation diagram named “Lab9_Sim.mdl” is constructed as shown in Figure 9.8. In the Simulink block diagram you can replace am, bm, KP , and KD with their values. Alternatively, you can place equations (9.7), and (9.12) in a script m-file to compute am, bm, KP , and KD which is sent to the MATLAB Workspace, then simulate the Simulink diagram. From the Simulink/Simulation Parameters select the Solver page and for Solver option Type select Fixed-step and ode4 (Runge-Kutta) and set the fixed step size to 0.001.

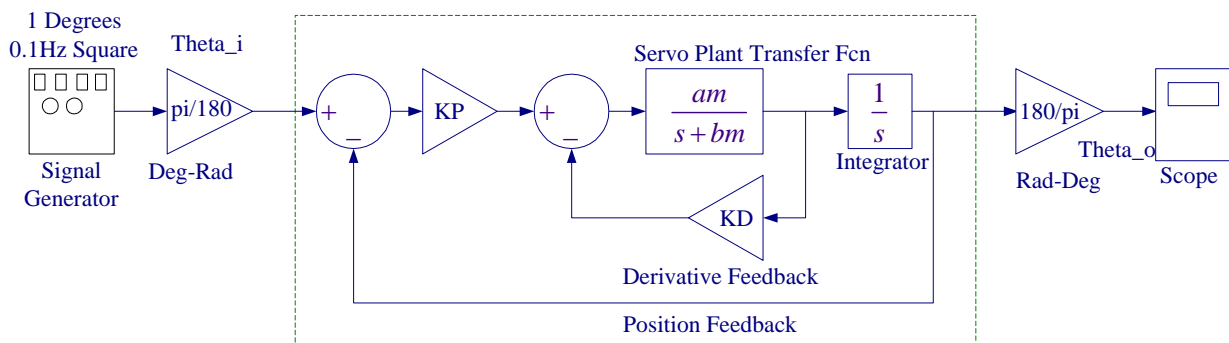


Figure 9.8 Simulink diagram for the servo plant position control. Obtain the response to a square input of amplitude 1 and frequency 0.1 Hz.

Create a subsystem with θ_i and θ_o in radians. To do this, enclose the blocks within a bounding box as shown. Choose Create Subsystem from the Edit menu. Edit in an out port as Theta_o and Theta_o and rename the subsystem as Compensated Servomotor as shown in Figure 9.9.

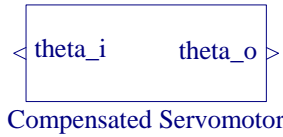


Figure 9.9 Compensated Servo plant position subsystem

2.2 Ball and beam controller Design

The outer loop controls the ball position using the same feedback law.

$$\alpha = K_{bp}(x_i - x_o) - K_{bd}\dot{x}_o \quad (9.13)$$

The purpose of this system is to have the ball position $x_o(t)$ follow the reference position $x_i(t)$.

Design this controller to meet the following time-domain specifications:

- Step response damping ratio $\zeta = 0.707$
- Step response peak time $t_p = 1.8$ second

Taking the Laplace transform of (9.13), we have

$$\alpha(s) = K_{bp}[X_i(s) - X_o(s)] - K_{bd}sX_o(s) \quad (9.14)$$

From (9.3)

$$\alpha(s) = \frac{7s^2}{5(9.8)} X_o(s) = \frac{s^2}{7} X_o(s) \quad (9.15)$$

Substituting for $\alpha(s)$ from (9.15) into (9.14) results in the following closed loop transfer function

$$\frac{X_o(s)}{X_i(s)} = \frac{7K_{bp}}{s^2 + 7K_{bd}s + 7K_{bp}} \quad (9.16)$$

The second-order response peak time t_p , is given by

$$t_p = \frac{\pi}{\omega_{bn}\sqrt{1-\zeta_b^2}} \quad (9.17)$$

The servo motor transfer function has the same form as the standard second-order transfer function given by

$$\frac{X_o(s)}{X_i(s)} = \frac{\omega_{bn}^2}{s^2 + 2\zeta\omega_{bn}s + \omega_{bn}^2} \quad (9.18)$$

From (9.16) and (9.18), we have

$$K_{bp} = \frac{\omega_{bn}^2}{7} \text{ rad/m} , \quad K_{bd} = \frac{2\zeta\omega_{bn}}{7} \text{ rad-s/m} \quad (9.19)$$

Comparing the plant characteristic equation given in (9.16) with the standard second-order characteristic equation in (9.18), find two equations for K_{bP} and K_{bD} in terms of a_m, b_m, ζ_b , and ω_{bn} . Substitute for the parameters and obtain the values of K_{bP} in deg/cm, and K_{bD} in deg-sec/cm for the above design specifications

$$K_{bP} = \underline{\hspace{10em}} \qquad K_{bD} = \underline{\hspace{10em}}$$

2.3 Digital Simulation

Utilizing the ball placement feedback equation (9.14), equation (9.4), compensated servo plant subsystem (Figure 9.9), equation (9.4), and rolling ball model (Figure 9.5) construct a Simulink diagram similar to the one shown in Figure 9.10. Apply a unit square wave of 0.1Hz and a gain of say 5 cm for the ball amplitude displacement. The movement of the servo gear is constraint to a maximum value of $\pm 65^\circ$. Therefore make sure that you place a limiter and set the upper and lower limits to no more than $\pm 50^\circ$.

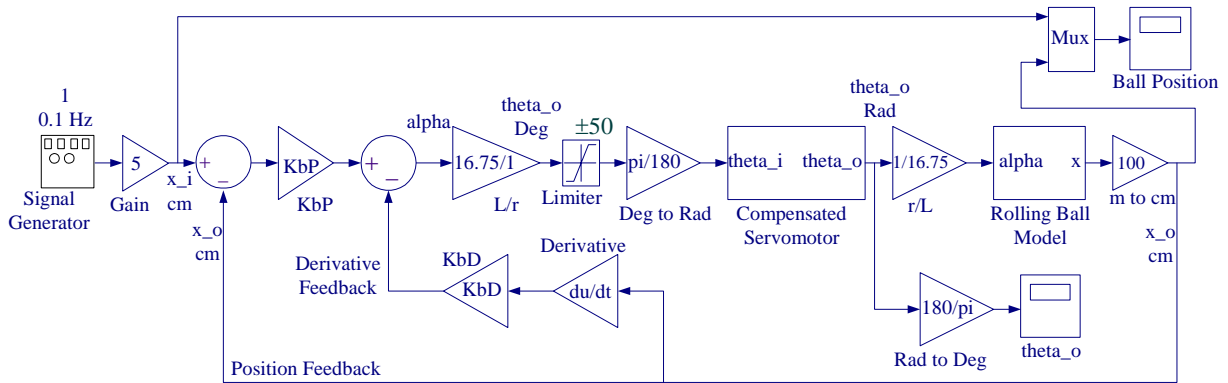


Figure 9.10 Ball and beam Simulink diagram with controllers for servo plant and ball and beam

In Simulation parameters dialog box set the Stop time to 12, for the solver select Fixed-step, ode4 (Runge-Kutta), and 0.001 for step size. Obtain the response for the servomotor angular position $\theta_o(t)$ and label as Figure 9.11. Also, obtain the response for the ball position and label as Figure 9.12. Comment on the ball position response and state if the ball position response meets the design specifications within the accuracy of the numerical solution.

All the prelab calculations, design and simulation must be completed prior to the laboratory session. The plants transfer function and the controller values must be checked and verified by your instructor.

3. Laboratory Procedure

When you have finished testing your model in SIMULINK, it has to be prepared for implementation on the real-time hardware. This means the plant model has to be replaced by the I/O components that form the interfaces to the real plant.

3.1 Wiring diagram

Using the set of leads, universal power module (UPM), SRV-02 DC-motor, and the connecting board of the MultiQ3 data acquisition board, complete the wiring diagram shown in Figure 9.13 as follows:

From	To	Cable
Ball & beam sensor	S1,S2 on UPM	6 pin mini Din to 6 pin mini Din
Remote sensor	S3 on UPM	6 pin mini Din to 6 pin mini Din
Encoder on SRV02	MultiQ/Encoder 0	5 pin Din to 5 pin Din
Motor on SRV02	UPM/To Load	6Pin to 4 Pin Din, Gain 1 Cable
D/A #0 on MultiQ	UPM – From D/A	RCA to 5 Pin Din
A/D # 0, 1, 2, 3, on MultiQ	UPM- TO A/D	5 Din to 4xRCA

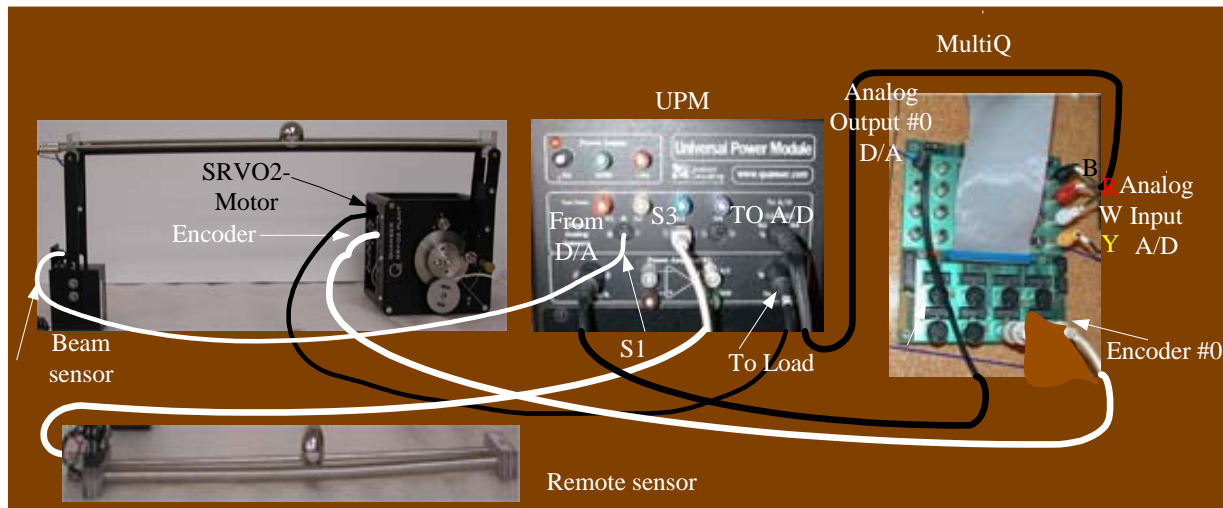
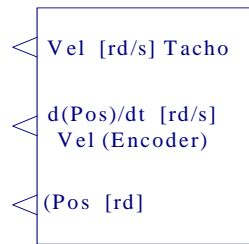


Figure 9.13 Wiring diagram for servo motor position control system.

3.2 Creating the Implementation model

The Encoder_Tach.mdl subsystem, which was created in the position control experiment shown in Figure 9.14(a).



Encoder and Tacho input

Figure 9.14(a) Interface to the SRV02 Feedback Signals.

If you double-click on the above subsystem it would display the underlying system as shown in Figure 9.14(b).

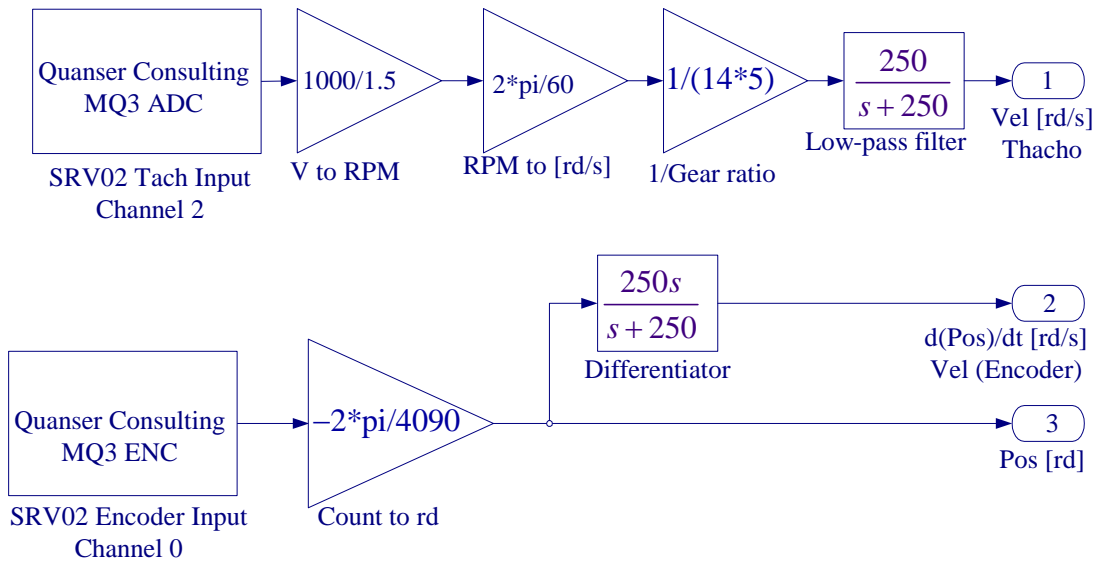


Figure 9.14(b) Content of the subsystem Encoder-Tach.

Get a new model in SIMULINK and begin to construct the implementation model with the required I/O components that form the interfaces to the real plant. Load the above Encoder_Tach.mdl subsystem (this was constructed in Lab5). Get the Analog Output block from the Quanser MultiQ3 and set the Channel Use to 0 as is in the wiring diagram. Add the position and velocity feedback gains to the signal coming from the Motor Encoder, complete the feedback loops and connect the resulting signal to the Quanser Analog output. Set the gains K_P and K_D to the values found in part 2.1. Get an analog input for getting the ball position signal from the ball and beam sensor and the remote sensor. Double-click to open its dialog box and in the Channel to Use box enter **[1,2]** in MATLAB array form. Get a Demux block from the Signal & System library to separate the signals coming from Channel 1 and 2 of the sensor analog input. The beam length is $(16.75)(2.54)$ cm, and the sensor voltage range is 10 V. Thus you need two Gain blocks to multiply the incoming signals by the factor $(16.75)(2.54)/10$ to measure the ball position in cm. Get a Manual Switch from the Nonlinear Library to connect the Signal generator and the Remote sensor to the Summing point block as shown. Add the position and velocity feedback gains to the signal coming from beam sensor, complete the feedback loop. Set the ball and beam controller gains K_{bP} and K_{bD} to the values found in part 2.2. Complete the remaining

blocks as in the SIMULINK simulation diagram. The movement of the servo gear is constraint to approximately $\pm 65^\circ$. Therefore make sure that you place a limiter and set the upper and lower limits to no more than $\pm 50^\circ$. When the servo gear is in the resting position, the servo angle is about -56 degrees. We start the controller from the resting position as shown in Figure 9.16(a). Thus, In order to ensure that the servo is at the zero position when the command is zero degrees, we command the servo with a -56 degree offset from the starting position. Use a Signal Generator with Amplitude 1, and Frequency 0.1 Hz and set the gain block value to 5. This gives a displacement range of 10 cm. Place as many Scopes as you like to monitor the ball position and the servomotor angle. Your completed model should be the same as shown in Figure 9.15. You may save the model under the file name say, “Lab9_Imp.mdl”.

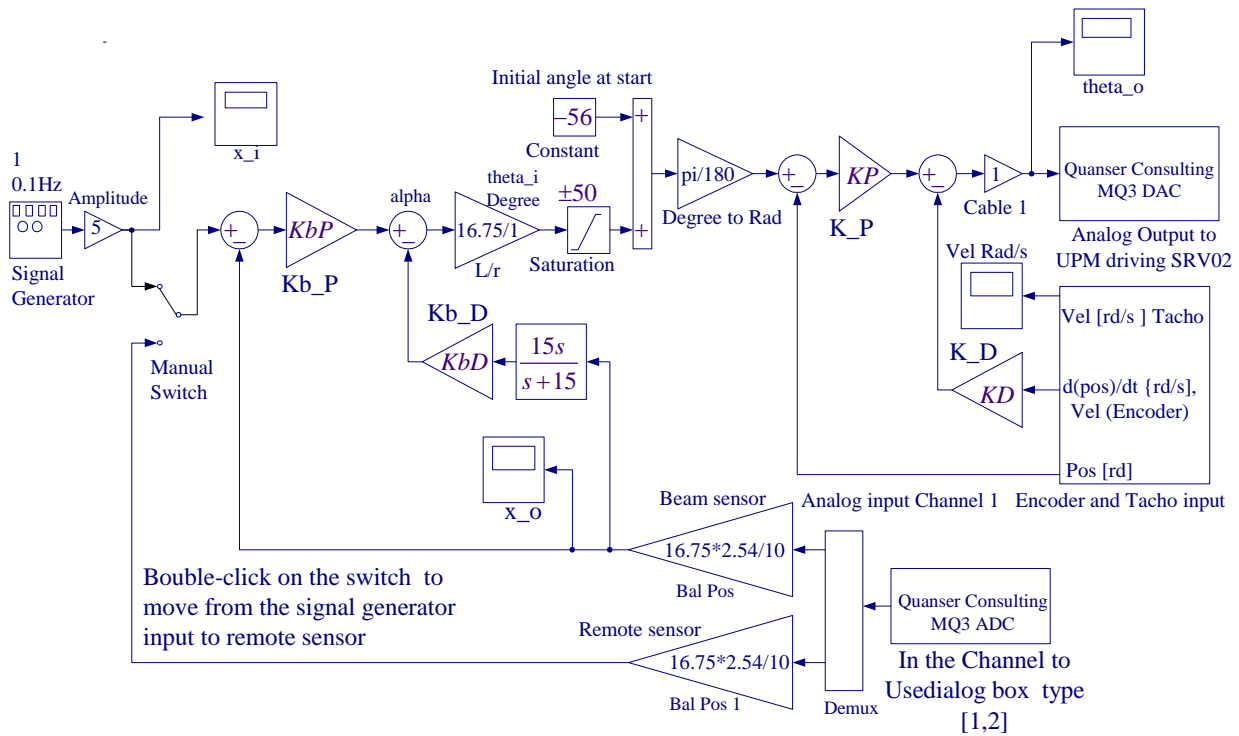


Figure 9.15 SIMULINK Implementation diagram for Ball and Beam Project

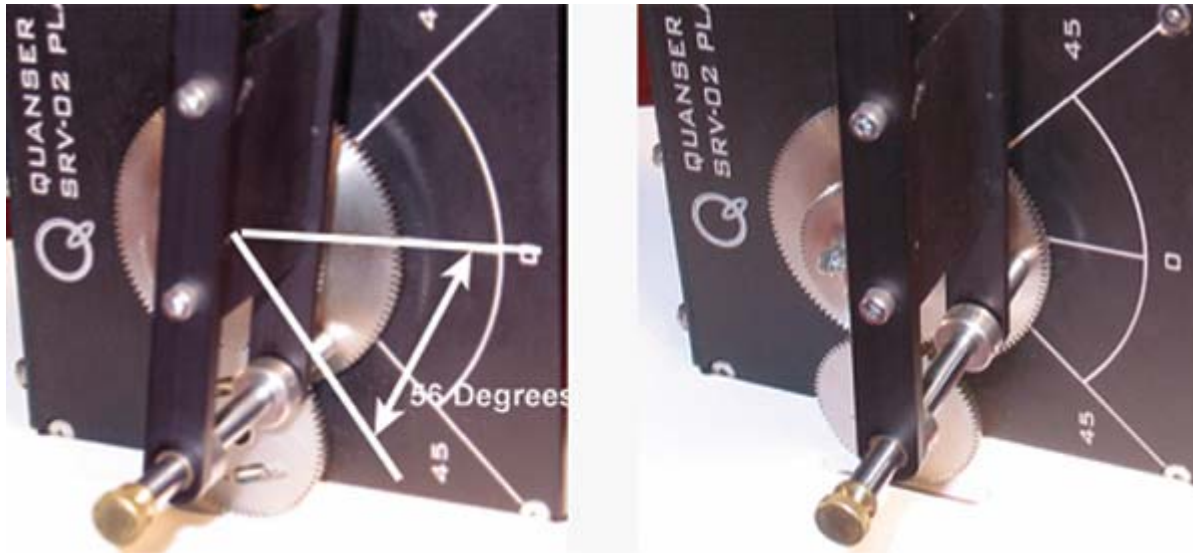


Figure 9.16(a) Starting position

(b) Zero position

At this point turn the servo gear by hand holding it at zero position, then the lever arms should be vertical and the beam should be perfectly horizontal so as if you place the ball at midpoint of the track it should almost stay without rolling to the sides. You may use proper tool to adjust the height of the support arm at the left hand side of the track, or simply raise end that would level the beam by placing few sheets of paper under it. If the beam is not perfectly horizontal when the servo angle is at zero position, there will be a steady state error that is a positive or negative offset depending on the beam tilting direction

[Before proceeding to the next part request the instructor to check your electrical connections and the implementation diagram.](#)

3.3 Compiling the model

In order to run the implementation model in real-time, you must first build the code for it. Turn on the UPM. Start WinCon, Click on the MATLAB icon in WinCon server. This launches MATLAB. In the Command menu set the Current Directory to the path where your model Lab9_Imp.mdl is. Before building the model, you must set the simulation parameters. Pull down the Simulation dialog box and select Parameters. Set the Start time to 0, the Stop time to 16, for Solver Option use Fixed-step and ode4 (Runge-Kutta) method, set the Fixed-step size, i.e., the sampling rate to 0.001. In the Simulation drop down menu set the model to **External**. Make sure all the controller gains are set. Start the WinCon **Server** on your laptop and then use **Client Connect**, in the dialog box type the proper Client workstation IP address. Generate the real-time code corresponding to your diagram by selecting the “**Build**” option of the WinCon menu from the Simulink window. The MATLAB window displays the progress of the code generation task. Wait until the compilation is complete. The following message then appears: “### Successful completion of Real-Time Workshop builds procedure for model: Lab9_Imp”.

3.4 Running the code

Following the code generation, WinCon Server and WinCon Client are automatically started. The generated code is automatically downloaded to the Client and the system is ready to run. Make sure the Manual Switch in Simulink implementation diagram is connected to the upper terminal coming from the Signal generator. If not double-click on the switch. To start the controller to run in real-time, click on the **Start** icon from the WinCon Server window shown in Figure 9.17. It will turn red and display STOP. Clicking on the **Stop** icon will stop the real-time code and return to the green button.

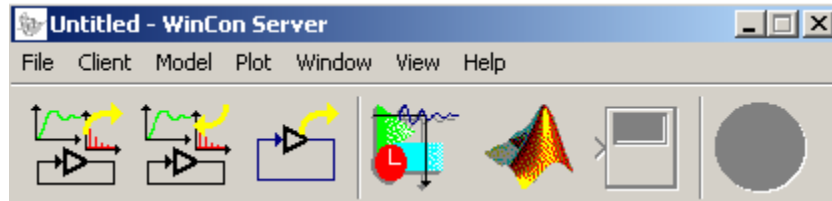


Figure 9.17 WinCon Server

If you hear a whining or buzzing in the motor you are feeding high frequency noise to the motor or motor is subjected to excessive voltage, immediately stop the motor. Ask the instructor to check the implementation diagram and the compensator gains before proceeding again.

3.5 Plotting Data

You can now plot in real-time any variables (e.g. angles, ball position) of your diagram by clicking on the "Plot/New/Scope" button (with the scope picture) in the WinCon Server window and selecting the variable you wish to visualize. Select "x_o" and click OK. This opens one real-time plot. To plot more variables in that same window, click on "File/Variables..." from the Scope window menu. The names of all blocks in the Simulink model diagram appear in a Multiple. Select Variable Tree. You can then select the variable(s) you want to plot. In this case, select, for example, "x_o" and "x_I". In the Scope pull-down menu, select Buffer and set the Buffer Size to 12. Using Freeze you can freeze the plot. From the File menu you can Save and Print the graph. If you choose Save As M-File ... you save the plot as M-file. Now at the MATLAB prompt if you type the file name you can obtain the MATLAB Figure plot. You can type grid to place a grid on the graph or edit the Figure as you wish.

You can also change the controller gains on the fly (i.e. while the controller is running in real-time). To do so, double click on the Simulink Implementation gain block, change to the desired new value, and select Apply, or OK. Note the changes in the real-time plots.

If the response is not close to the expected value the beam is not perfectly level when motor angle is at zero position. Raising or lowering the left support to level the beam will improve the response.

In the next part the remote sensor is used for operation in master/slave tracking mode. This sensor is used, as the input to the Ball & Beam desired position for making the Ball on the beam follow the ball on the remote sensor. Double-click on the Manual Switch in the Simulink

implementation diagram to connect to the Remote Sensor. Move the Ball on the remote sensor and observe the Ball on the beam tracking your movement and obtain the response.

In WinCon Server, use File Save, this saves the compiled controller including all plots as a **.wpc** (WinCon project) file. In case you want to run the experiment again, from WinCon Server use File/Open to reload this **.wcp** file, and run the project in real time independent of MATLAB/Simulink.

To prevent excessive wear to the motor and gearbox run the experiment for a short time.

4. Project Report

Your report must include the detailed servo plant and the ball and beam block diagram. Obtain formulas for the servomotor controller gains and the ball and beam controller gains, and the design values to meet the design specifications. Include your final K_P , K_D , K_{bP} , and K_{bD} after fine-tuning, if it was required. Plot the final actual system response.

Discuss the assumption and approximations made in modeling the servomotor. Explain why the control system design was feasible through two independent controller loops. Comment on your results; how does experimental response compare to simulated responses. Discuss the reason for any deviation in the actual transient response and the simulated response.