QUANSER
INNOVATE. EDUCATE.

STUDENT WORKBOOK

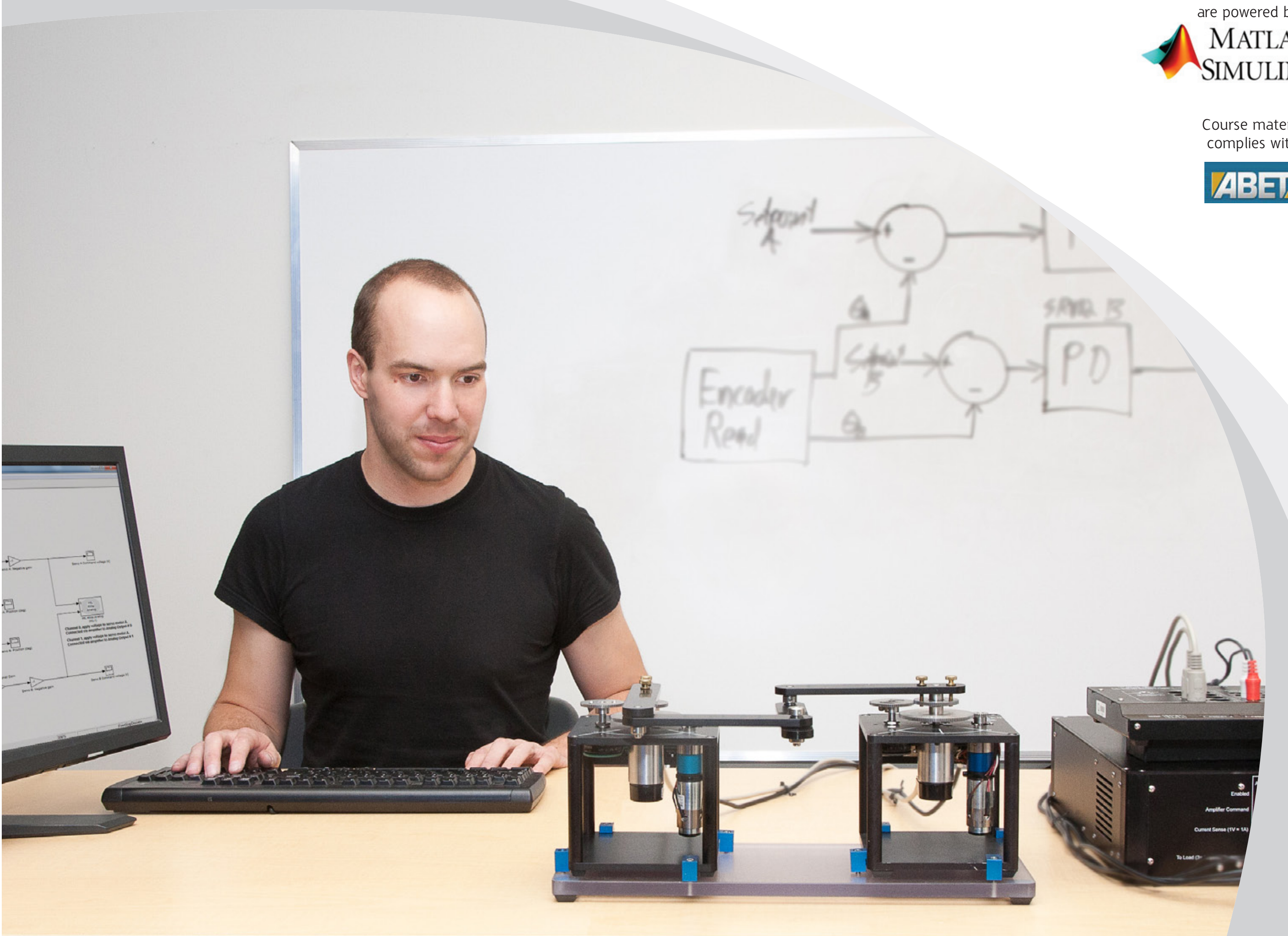
2 DOF Robot Experiment for MATLAB®/Simulink® Users

Standardized for ABET* Evaluation Criteria

Developed by:
Jacob Apkarian, Ph.D., Quanser
Hervé Lacheray, M.A.SC., Quanser
Michel Lévis, M.A.SC., Quanser

Quanser 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; and 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 challenge in this experiment is to design a controller that manipulates the tip of a two degree of freedom (2 DOF) pantograph type robot. The Quanser 2 DOF Robot is composed of two Quanser Rotary Servo Base Units (SRV02) connected together with a four-bar linkage. See the 2 DOF Robot User Manual [6] for information about the system.

In order to control the X-Y position of the end-effector (i.e., the tip of the robot links), the direct and inverse kinematics of the mechanism must be derived. This is an experiment in kinematic control rather than dynamical control. The servo or joint space control is done using PD-based control.

Topics Covered

- Design PD-based control to position the servo (i.e., robot joint) angles according to certain specifications, i.e., the *joint space* control.
- Simulate the joint space control and ensure it meets the given specifications.
- Run the joint space control on the actual 2 DOF Robot system.
- Find direct (or forward) kinematics and inverse kinematics of a 2 DOF pantograph type robot.
- Using the two PD control loops on the servos with the kinematics, simulate the closed-loop X-Y positioning of the end-effector, i.e., simulate the *workspace* control.
- Implement the workspace controller on the 2 DOF Robot system.

Prerequisites

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

1. See the system requirements in Section 5 for the required hardware and software.
2. Transfer function fundamentals, e.g., obtaining a transfer function from a differential equation.
3. Proportional-velocity control detailed in the *Position Control* lab in the SRV02 Workbook [4].
4. Basics of Simulink®.
5. QUARC Integration lab detailed in Appendix A in the SRV02 Workbook [4].

2 SYSTEM REPRESENTATION

The 2 DOF planar manipulator is shown in Figure 2.1. As documented in the 2 DOF Robot User Manual [6], the two Quanser Rotary Servo units are named SRV02 A and SRV02 B. In Figure 2.1, the servos are represented by the actuated revolute joints A and B. All four bars comprising the manipulator linkage have the same length, denoted L_b. The 2 DOF robot end-effector is depicted by joint E. The two actuated angles are denoted θ_A and θ_B and they are the output shaft angles of SRV02 A and SRV02 B, respectively.

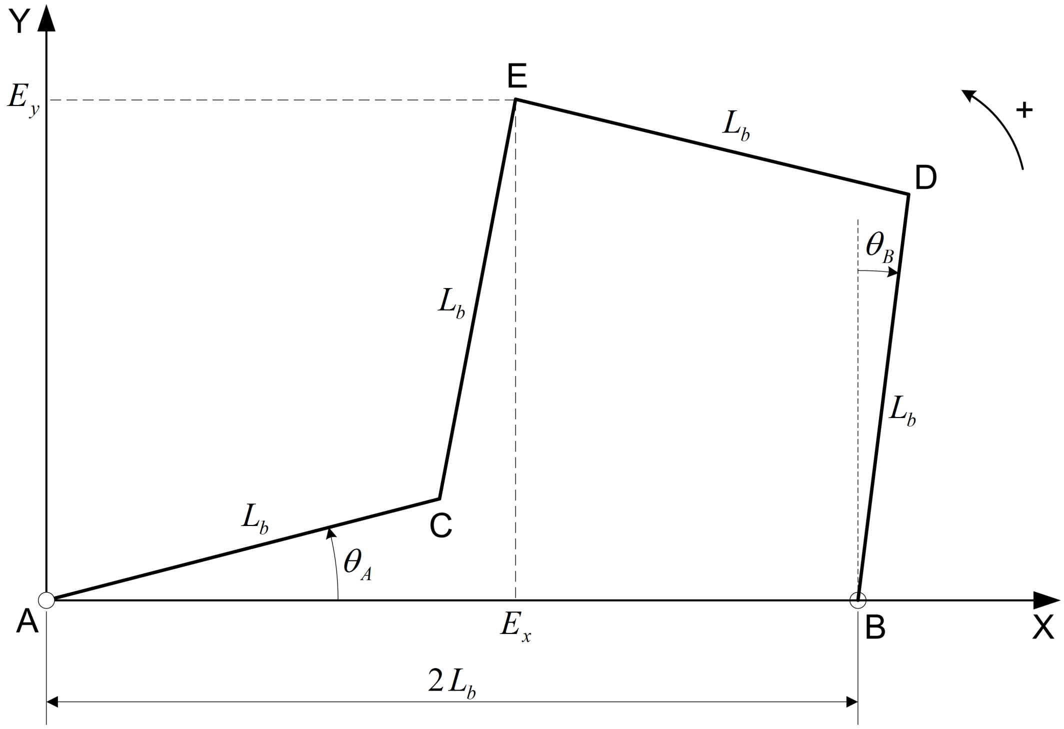


Figure 2.1: Angles and lengths in the 2 DOF Robot system

The positive direction of rotation is counter-clockwise when looking from the top of the 2 DOF planar manipulator. In Figure 2.1, the Cartesian coordinates of the actuated joints A and B are fixed. Joint A is at the origin of the reference frame with the positive X- and Y- directions defined as shown in Figure 2. The X-Y coordinates of joints A and B results in $(0,0)$ and $(B_x,0)$, respectively.

When the two SRV02 devices are mounted on the supplied base plate, the two servo output shafts are apart a distance of $B_x = 2L_b$.

2.1 Home Position

The home position of the 2 DOF manipulator is illustrated in Figure 2.2. The 2 DOF robot home defines the zero joint angles. Therefore, the angular position of both actuated joints results in $\theta_A = 0$ and $\theta_B = 0$. Assuming that joint A is at the origin of the reference frame, the X-Y global coordinates of the end effector joint E in the home position equals $(E_x, E_y) = (L_b, L_b)$. See the 2 DOF Robot [6] for more information on the home position.

When close to the home position, SRV02 B causes motion only in the X direction while SRV02 A causes motion only in the Y direction. Both actuated joints are thus decoupled.

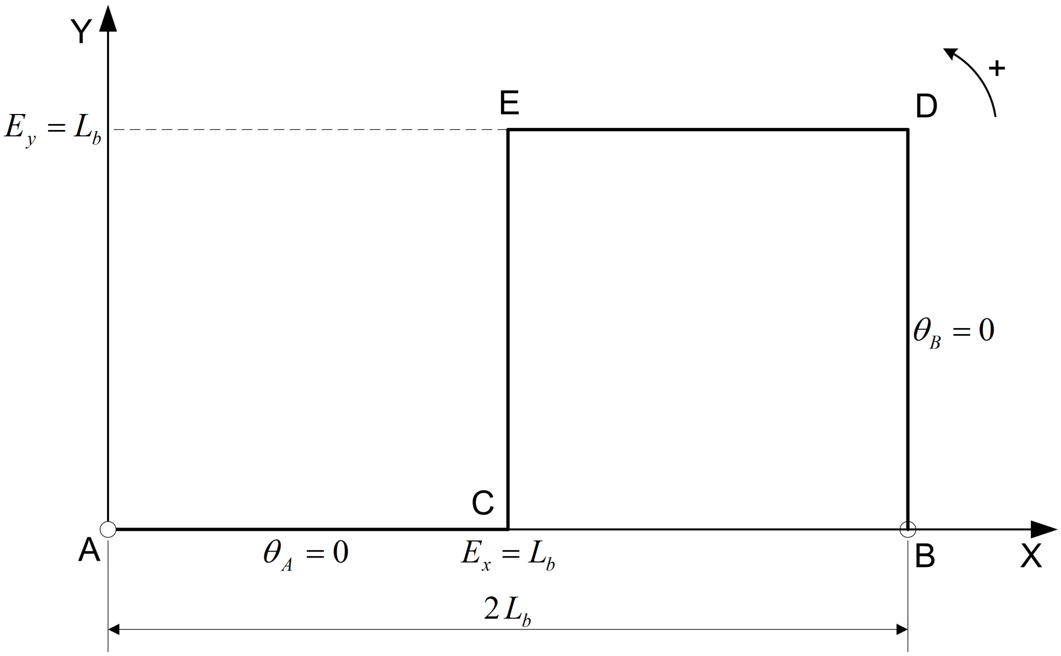


Figure 2.2: Home position of the 2 DOF Robot system

3 JOINT SPACE CONTROL

Joint space is the coordinate system describing the state of the robot based on its joints. In the 2 DOF Robot system, the joints are the two Rotary Servo units (i.e., Quanser SRV02). This section describes how to develop a PID-based control to manipulate the angular position of each robot joint, i.e., to control the position of each servo load gear.

3.1 Background

3.1.1 Model

Recall in the Modeling Laboratory in the SRV02 Workbook [4], the SRV02 voltage-to-speed transfer function was derived. To find the voltage-to-position transfer function, we can put an integrator $(1/s)$ in series with the speed transfer function (effectively integrating the speed output to get position). Then, the resulting open-loop voltage-to-load gear position transfer function becomes:

$$P(s) = \frac{K}{s(\tau s + 1)} \tag{3.1}$$

For this laboratory, you can use the following model parameters are:

$$K = 1.53 \text{ rad/(V-s)} \tag{3.2}$$

and

$$\tau = 0.0414 \text{ s} \tag{3.3}$$

These model parameters are slightly different than those derived in the Modeling Laboratory in [4] because it includes the additional inertia from the four-bar linkage attached to the load gear shaft of the servo.

3.1.2 Desired Position Control Response

The block diagram shown in Figure 3.1 is a general unity feedback system with compensator (controller) C(s) and a transfer function representing the plant, P(s). The measured output, Y(s), is supposed to track the reference signal R(s) and the tracking has to match to certain desired specifications.

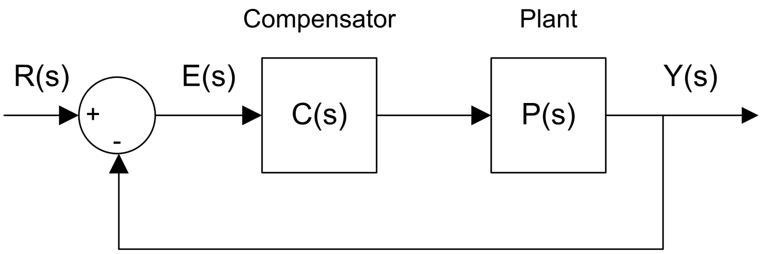


Figure 3.1: Unity feedback system.

The output of this system can be written as:

$$Y(s) = C(s)\,P(s)\,\left(R(s) - Y(s)\right) \tag{3.4}$$

By solving for $Y(s)$, we can find the closed-loop transfer function:

$$\frac{Y(s)}{R(s)} = \frac{C(s)\,P(s)}{1 + C(s)\,P(s)} \tag{3.5}$$

In fact, when a second order system is placed in series with a proportional compensator in the feedback loop as in Figure 3.1, the resulting closed-loop transfer function can be expressed as:

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\,\omega_n\,s + \omega_n^2} \tag{3.6}$$

where ω_n is the natural frequency and ζ is the damping ratio. This is called the *standard second-order* transfer function. Its response properties depend on the values of ω_n and ζ.

3.1.3 Peak Time and Overshoot

Consider a second-order system as shown in Equation 3.6 subjected to a step input given by

$$R(s) = \frac{R_0}{s} \tag{3.7}$$

with a step amplitude of $R_0 = 1.5$. The system response to this input is shown in Figure 3.2, where the red trace is the response (output), $y(t)$, and the blue trace is the step input $r(t)$.

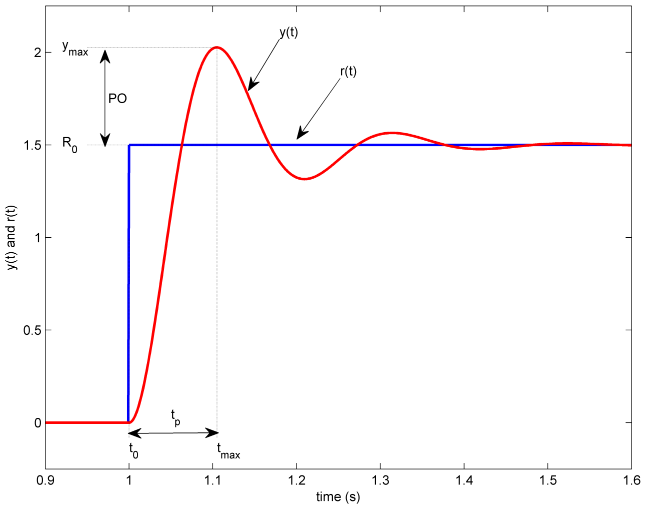


Figure 3.2: Standard second-order step response.

The maximum value of the response is denoted by the variable y_{max} and it occurs at a time t_{max}. For a response similar to Figure 3.2, the percent overshoot is found using

$$PO = \frac{100\,(y_{max} - R_0)}{R_0} \tag{3.8}$$

From the initial step time, t_0, the time it takes for the response to reach its maximum value is

$$t_p = t_{max} - t_0 \tag{3.9}$$

This is called the *peak time* of the system.

In a second-order system, the amount of overshoot depends solely on the damping ratio parameter and it can be calculated using the equation

$$PO = 100\,e^{\left(-\frac{\pi\,\zeta}{\sqrt{1-\zeta^2}}\right)} \tag{3.10}$$

The peak time depends on both the damping ratio and natural frequency of the system and it can be derived as:

$$t_p = \frac{\pi}{\omega_n\,\sqrt{1-\zeta^2}} \tag{3.11}$$

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

3.1.4 2 DOF Robot Specifications

The desired time-domain specifications for controlling the position of the load shaft on each SRV02 system are:

$$|e_{ss}| \leq 0.5 \text{ deg} \tag{3.12}$$

$$t_p = 0.15 \text{ s} \tag{3.13}$$

and

$$PO = 5.0 \text{ \%} \tag{3.14}$$

Thus, when tracking the load shaft reference, the transient response should have a peak time less than or equal to 0.10 seconds, an overshoot less than or equal to 5 %, and the steady-state response should have less than 5% error.

3.1.5 PV Controller Design

The proportional-velocity (PV) compensator to control the position of the SRV02 has the following structure

$$V_m(t) = k_p\ (\theta_d(t) - \theta_l(t)) - k_v \left(\frac{d}{dt}\,\theta_l(t)\right) \tag{3.15}$$

where k_p is the proportional control gain, k_v is the velocity control gain, $\theta_d(t)$ is the setpoint or reference load shaft angle, $\theta_l(t)$ is the measured load shaft angle, and $V_m(t)$ is the SRV02 motor input voltage. The block diagram of the PV control is given in Figure 3.3.

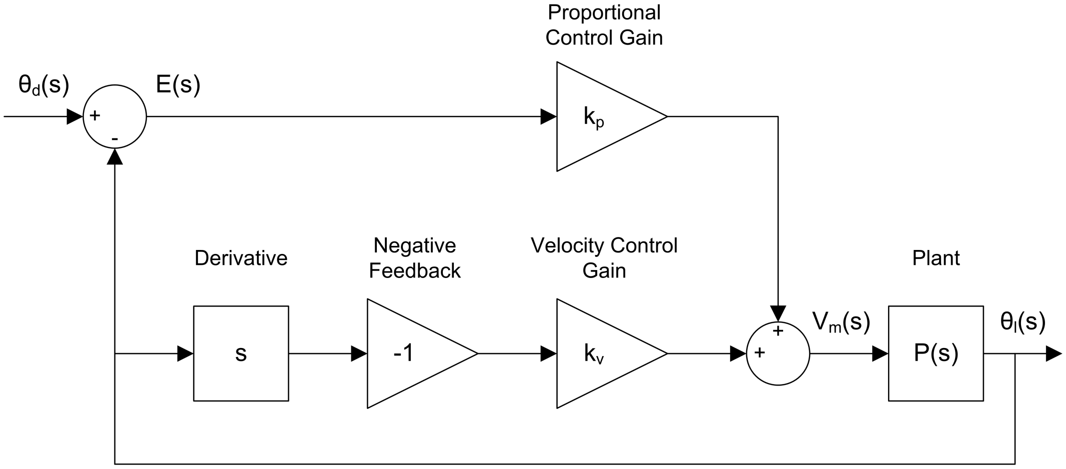


Figure 3.3: Block diagram of SRV02 PV position control.

We need to find the closed-loop transfer function $\Theta_l(s)/\Theta_d(s)$ for the closed-loop position control of the SRV02. Taking the Laplace transform of equation 3.15 gives

$$V_m(s) = k_p\ (\Theta_d(s) - \Theta_l(s)) - k_v\, s\, \Theta_l(s) \tag{3.16}$$

From the Plant block in Figure 3.3 and equation 3.1, we can write

$$\frac{\Theta_l(s)}{V_m(s)} = \frac{K}{s\,(\tau\, s + 1)} \tag{3.17}$$

Substituting equation 3.16 into 3.17 and solving for $\Theta_l(s)/\Theta_d(s)$ gives the SRV02 position closed-loop transfer function as:

$$\frac{\Theta_l(s)}{\Theta_d(s)} = \frac{K\,k_p}{\tau\,s^2 + (1 + K\,k_v)\,s + K\,k_p} \tag{3.18}$$

For more information, see the Position Control lab in the SRV02 Workbook [4].

3.2 Pre-Lab Questions

1. The SRV02 closed-loop transfer function was derived in equation 3.18 in Section 3.1. Find the control gains k_p and k_v in terms of ω_n and ζ. **Hint:** Remember the standard second order system equation.
2. Calculate the minimum damping ratio and natural frequency required to meet the specifications given in Section 3.1.4.
3. Based on the K and τ model parameters given in Section 3.1.1, calculate the control gains needed to satisfy the time-domain response requirements given in Section 3.1.4. **Note:** Alternatively, you can find the K and τ of your SRV02 systems using one of the techniques outlined in the Modeling Laboratory in the SRV02 Workbook [4].

3.3 Lab Experiments

3.3.1 Simulation

In this section you will simulate the joint space control of the Rotary 2 DOF Robot system. Each SRV02 system is modeled using the first-order transfer function and controlled using the PV controller described in Section 3.1.5. Our goals are to confirm that the desired reponse specifications are satisfied and to verify that the motor is not saturated.

Experimental Setup

The *s_2d_robot_joint_space* Simulink®diagram shown in Figure 3.4 will be used to simulate the closed-loop position control response with the PV controller. The SRV02 Model uses a *Transfer Fcn* block from the Simulink®library. The PV Control subsystem contains the PV controller detailed in Section 3.1.5.

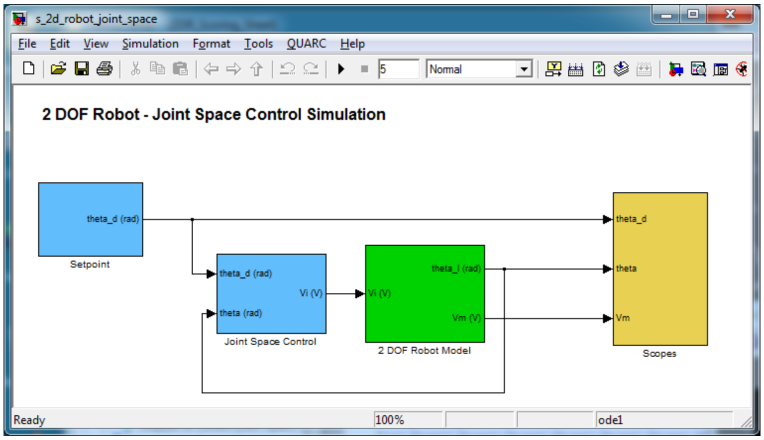
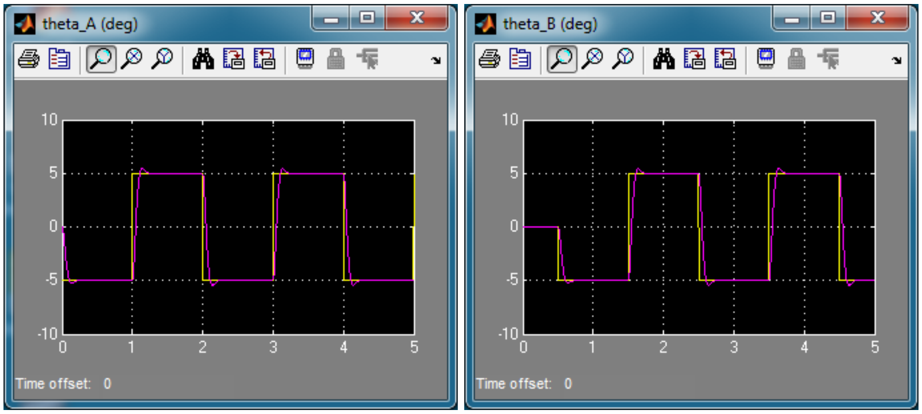


Figure 3.4: Simulink model used to simulate 2 DOF Robot joint space response.

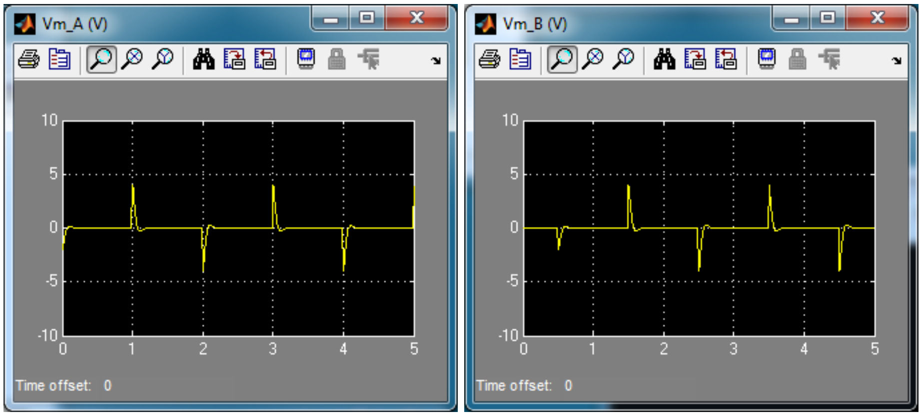
IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your SRV02 setup. If they have not been configured already, then you need to go to Section 5 to configure the lab files first.

1. Enter the proportional and velocity control gains found in Pre-Lab question 3 in Matlab®as *kp* and *kv*.
2. To generate a step reference, go to the *Setpoint* subsystem ensure the each *Signal Generator* is set to the following:
    - Signal type = *square*
    - Amplitude = 1
    - Frequency = 0.5 Hz
3. In the *Setpoint* subsystem, set both *Amplitude (deg)* gain blocks to 5 to generate a step with an amplitude of 5 degrees (i.e., square wave goes between ± 5 which results in a step amplitude of 10).
4. Open the load shaft position scope, *theta_l (rad)*, and the motor input voltage scope, *Vm (V)*.
5. Start the simulation. By default, the simulation runs for 5 seconds. The scopes should be displaying responses similar to Figure 3.5. Note that in the *theta_l* (rad) scopes, the yellow trace is the setpoint position while the purple trace is the simulated position (generated by the *SRV02 Model* block).
6. Generate a Matlab®figure showing the *Simulated Joint Space* position response and the input voltage.

    Data Saving: After each simulation run, each scope automatically saves their response to a variable in the Matlab®workspace. For instance, the *theta_A (deg)* scope saves its response to the variable called data_theta_A and the *Vm (V)* scope saves its data to the *data_Vm_A* variable.



(a) SRV02 A Angle (b) SRV02 B Angle



(c) SRV02 A Voltage (d) SRV02 B Voltage

Figure 3.5: Simulated closed-loop response in joint space.

- The *data_theta_A* variable has the following structure: *data_theta_A(:,1)* is the time vector, *data_theta_A(:,2)* is the setpoint, and *data_theta_A(:,3)* is the simulated angle.
- For the *data_Vm_A* variable, *data_Vm_A(:,1)* is the time and *data_Vm_A(:,2)* is the simulated input voltage.

7. Measure the steady-state error, the percent overshoot and the peak time of the simulated response. Does the response satisfy the specifications given in Section 3.1.4? **Hint:** Use the Matlab® *ginput* command to take measurements off the figure.

3.3.2 Implementation

The q_2d_robot_joint_space Simulink diagram shown in Figure 3.6 is used to perform the position control exercises in this laboratory. The *2 DOF Robot* subsystem contains QUARC® blocks that interface with the DC motor and sensors of the 2 DOF Robot system.

Experimental Setup

The *q_2d_robot_joint_space* Simulink® diagram shown in Figure 3.6 will be used to simulate the closed-loop position control response with the PV controller. The SRV02 Model uses a *Transfer Fcn* block from the Simulink® library. The PV Control subsystem contains the PV controller detailed in Section 3.1.5.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your SRV02 setup. If they have not been configured already, then you need to go to Section 5 to configure the lab files first.

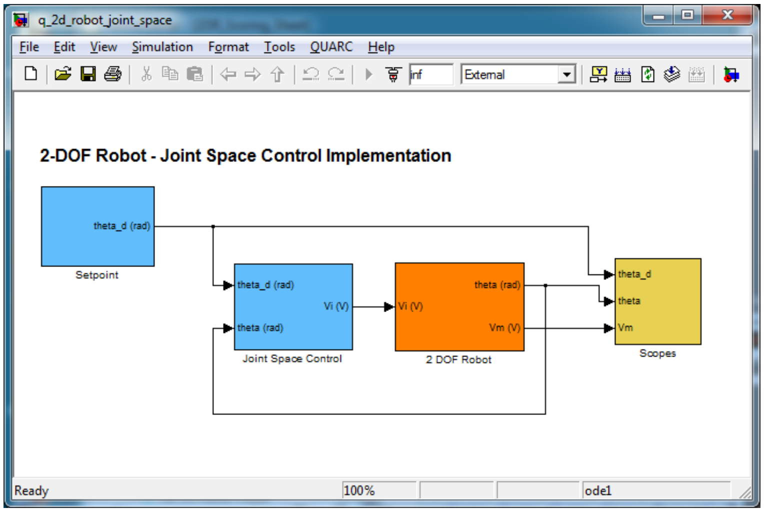
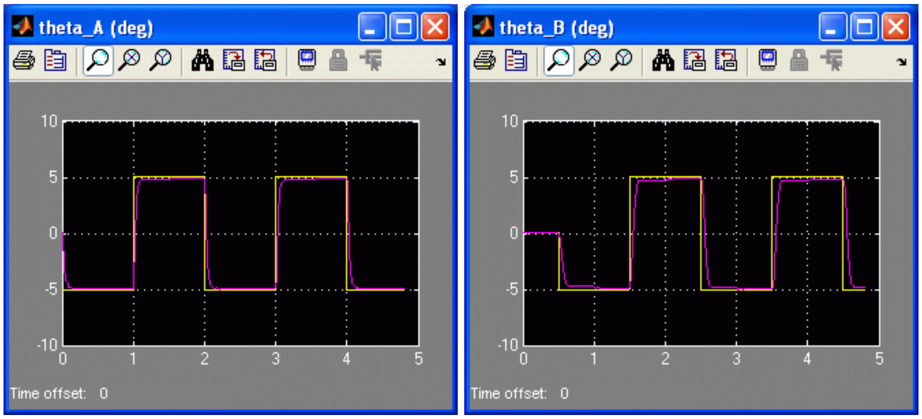


Figure 3.6: Simulink model used to joint space control on 2 DOF Robot device.

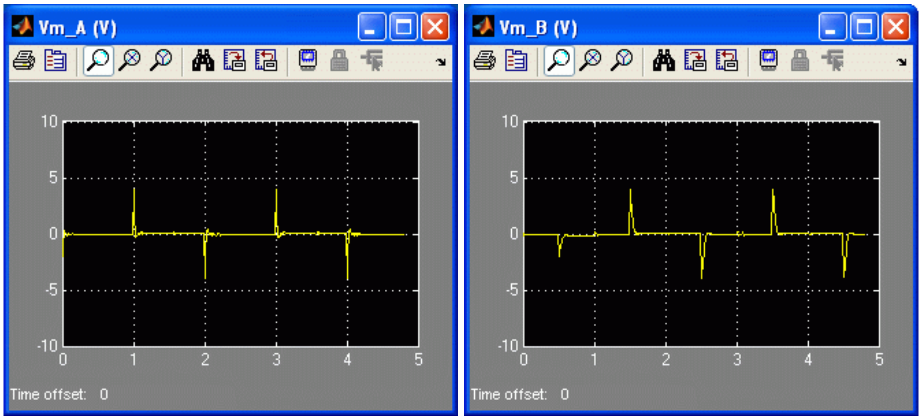
Follow this procedure:

1. Enter the proportional and velocity control gains found in Pre-Lab question 3 in Matlab® as *kp* and *kv*.
2. To generate a step reference, go to the *Setpoint* subsystem ensure the each *Signal Generator* is set to the following:
   - Signal type = *square*
   - Amplitude = 1
   - Frequency = 0.5 Hz
3. In the *Setpoint* subsystem, set both *Amplitude (deg)* gain blocks to 5 to generate a step with an amplitude of 5 degrees (i.e., square wave goes between ± 5 which results in a step amplitude of 10).
4. Open the load shaft position scope, *theta_l (rad)*, and the motor input voltage scope, *Vm (V)*.
5. In the Simulink diagram, go to QUARC | Build.
6. Click on QUARC | Start to run the controller. Each servo should beging rotation back-and-forth and the end-effector should be moving in a square-like fashhion. The scopes should be displaying responses similar to Figure 3.7. Note that in the *theta_l* (rad) scopes, the yellow trace is the setpoint position while the purple trace is the measured position.



(a) SRV02 A Angle

(b) SRV02 B Angle



(c) SRV02 A Voltage

(d) SRV02 B Voltage

Figure 3.7: Typical response when running joint space control on 2 DOF Robot system

7. Generate a Matlab®figure showing the *Implemented Joint Space* position response and the input voltage.

   Data Saving: As in the s_2d_robot_joint_space.mdl, after each run each scope automatically saves their response to a variable in the Matlab®workspace.

8. Measure the steady-state error, the percent overshoot and the peak time of the simulated response. Does the response satisfy the specifications given in Section 3.1.4? **Hint:** Use the Matlab®*ginput* command to take measurements off the figure.

3.4 Results

Fill out Table 3.1 with your answers from your control lab results - both simulation and implementation.

Description	Symbol	Value	Units
Pre Lab Questions			
Proportional Control Gain	k_p		V/rad
Velocity Control Gain	k_v		V-s/rad
Simulation			
Steady-state error	e_{ss}		deg
Peak time	t_p		s
Percent overshoot	PO		%
Implementation			
Steady-state error	e_{ss}		deg
Peak time	t_p		s
Percent overshoot	PO		%

Table 3.1: Joint Space Results

4 WORKSPACE CONTROL

Workspace is the coordinate system describing the state of the end-effector. Using kinematics and joint space control, the planar position of the 2 DOF Robot end-effector can be controlled. This is called the *workspace control*.

4.1 Background

4.1.1 Direct Kinematics

The forward, or direct, kinematics computes the global Cartesian coordinates of the robot end-effector from the joint angles. The direct kinematics of the 2 DOF Robot is shown in Figure 4.1.

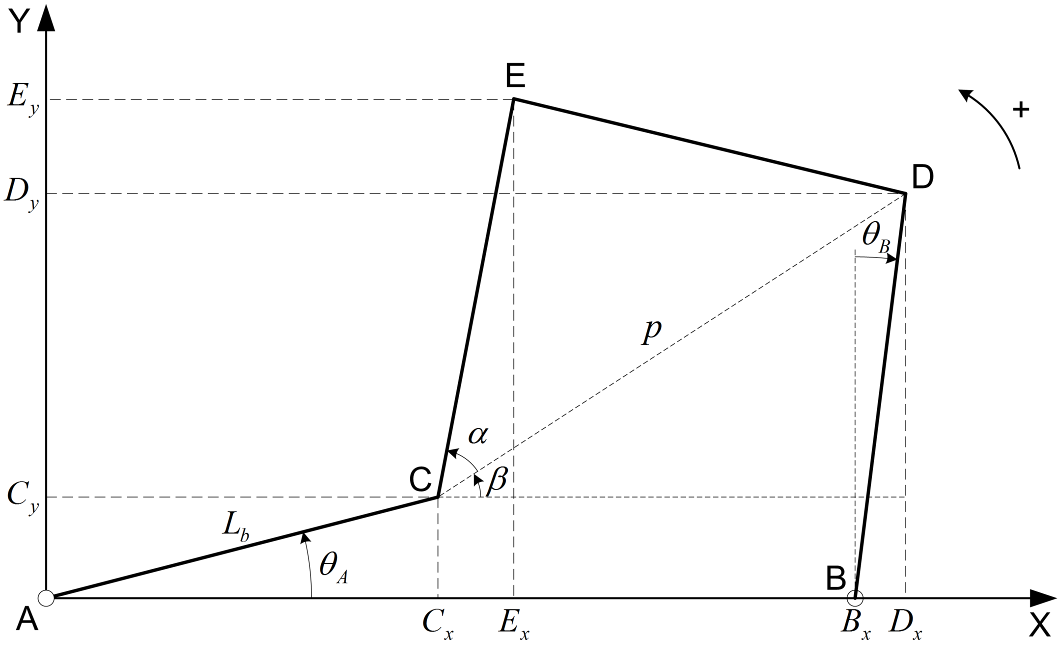


Figure 4.1: Direct kinematics of 2 DOF Robot system

Given the angles of the joints, θ_A and θ_B, located at points A and B the Cartesian coordinates of joint C are $C_x = L_b \cos\theta_A$ and $C_y = L_b \sin\theta_A$. Similarly, the Cartesian coordinate of D, are $D_x = B_x - L_b \cos\theta_B$ and $D_y = L_b \cos\theta_B$.

The distance p between points C and D, shown in Figure 4.1, can be found using the Pythagorean theorem. The x side of the triangle is $D_x - C_x$ long and the y length is $D_y - C_y$. The resulting length of the hypotenuse, i.e., the distance between C and D, is

$$p = \sqrt{(D_x - C_x)^2 + (D_y - C_y)^2}.$$

Because each manipulator link is of the same length, L_b, triangle ΔCDE is isosceles. You can express angle α in terms of L_b and p. In Figure 4.1, if you draw a line going from point E down to the middle of length p such that it is perpendicular to p, you have a right angle triangle (with hypotenuse L_b). Taking the cosine of angle α will then give the expression

$$\cos\alpha = \frac{p/2}{L_b}$$

which gives

$$\alpha = \arccos\frac{p}{2L_b}.$$

To obtain angle β at the C vertex, take the *tangent* of β to get

$$\tan\beta = \frac{D_y - C_y}{D_x - C_x}$$

and solve for β for

$$\beta = \arctan\frac{D_y - C_y}{D_x - C_x}.$$

The sum of angles α and β form the angle between link CE and the Cartesian coordinate X-axis. This leads directly to the X-Y position of E. The forward kinematics of the end-effector in the x-axis and y-axis is therefore

$$E_x = C_x + L_b\cos(\alpha+\beta)$$

and

$$E_y = C_y + L_b\sin(\alpha+\beta).$$

4.1.2 Inverse Kinematics

The backward, or inverse, kinematics calculates the joint angles from the X-Y Cartesian coordinates of the end-effector. The inverse kinematics of the 2 DOF Robot system is shown in Figure 4.2. The known quantities in the inverse kinematics are the X and Y coordinates of the end-effector location E, i.e., E_x and E_y. From this, we can obtain the angles of the motor joints A and B, i.e., θ_A and θ_B.

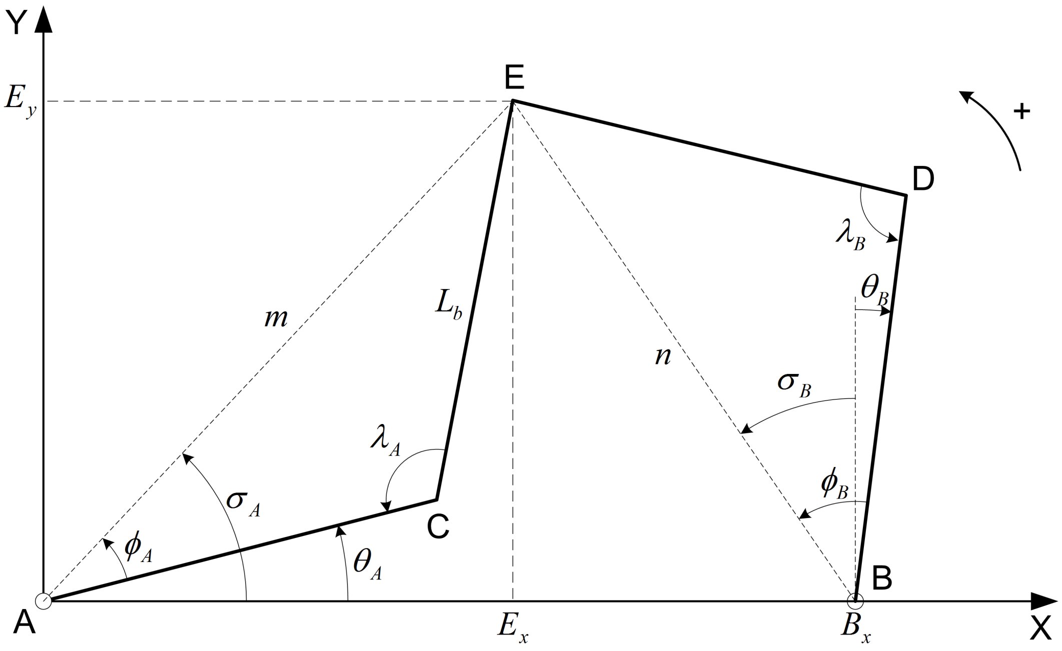


Figure 4.2: Inverse kinematics of 2 DOF Robot system

Consider ΔABE in Figure 4.2, i.e., the triangle with vertices A, B, and E. Its sides are denoted by the variables m and n. Assuming that vertex A is at the origin, the length of elements m and n in terms of the end-effector location and B_x is

$$m = \sqrt{E_x^2 + E_y^2}$$

and

$$n = \sqrt{(E_x - B_x)^2 + E_y^2}.$$

By using the Cosine Law on ΔACE and ΔBDE, we find

$$m^2 = 2L_b^2(1 - \cos\lambda_A)$$

and

$$n^2 = 2L_b^2(1 - \cos\lambda_B).$$

Solving for λ_A and λ_B

$$\lambda_A = \arccos\frac{2L_b^2 - m^2}{2L_b^2}$$

and

$$\lambda_B = \arccos\frac{2L_b^2 - n^2}{2L_b^2}.$$

The expressions that calculate the joint angles required for a certain X-Y end-effector position is left as an exercise in the *Pre-Lab Questions*.

4.2 Pre-Lab Questions

1. Recalling that ΔACE and ΔBDE are both isosceles, find the vertex angles ϕ_A and ϕ_B.
2. Find angles σ_A between the X-coordinate axis and AE, and σ_B between the Y-coordinate axis and BE relative to E_x, E_y, and B_x.
3. Give the inverse kinematics of the end effector - the joint angles θ_A and θ_B.

4.3 Lab Experiments

4.3.1 Simulation

In this section you will simulate the workspace control of the Rotary 2 DOF Robot system.

Experimental Setup

The *s_2d_robot_workspace* Simulink®diagram shown in Figure 4.3 will be used to simulate the workspace control for the 2 DOF Robot. **This file is not complete.** The direct and inverse kinematics must be completed.

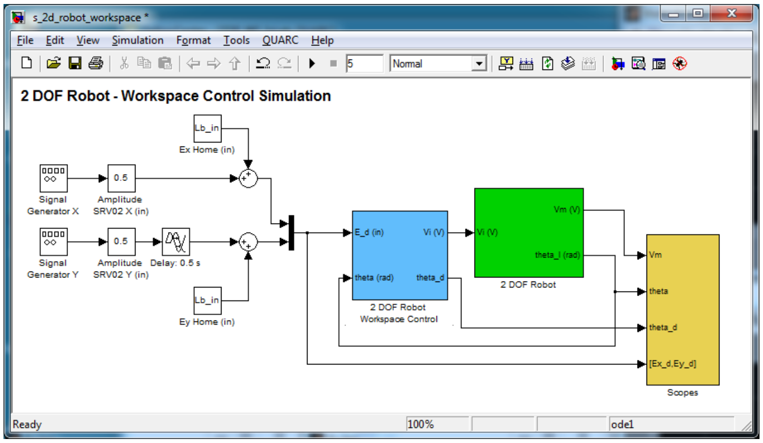


Figure 4.3: Simulink model used to simulate 2 DOF Robot workspace response.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your SRV02 setup. If they have not been configured already, then you need to go to Section 5 to configure the lab files first.

The workspace controller is implemented in the Simulink subsystem shown in Figure 4.4. As illustrated, it includes the same servo-based controller used in Section 3 but in this case the servo commands are generated by Embedded Matlab Function called *Inverse Kinematics* □ *Student*. The servo angles required for the end-effector to be at a desired Cartesian X-Y location is computed by this block.

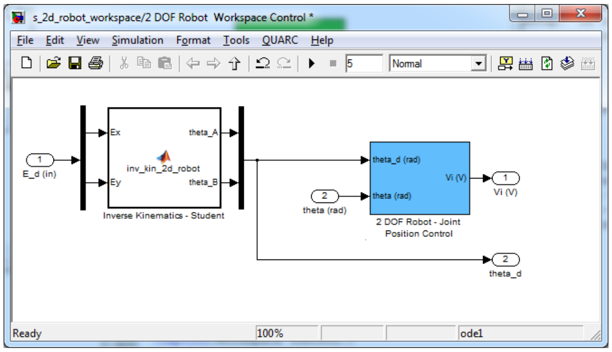


Figure 4.4: Workspace Control Simulink subsystem

The forward kinematics, on the other hand, are computed from the measured or simulated servo angles to give the resulting planar end-effector position. This is implemented in another Embedded Matlab Function called *Forward Kinematics - Student* and is located in the the *Scopes* subsystem shown in Figure 4.5.

1. Enter the proportional and velocity control gains found in Pre-Lab question 3 in Matlab®as *kp* and *kv*.
2. To generate a step reference, ensure each *Signal Generator* is set to the following:

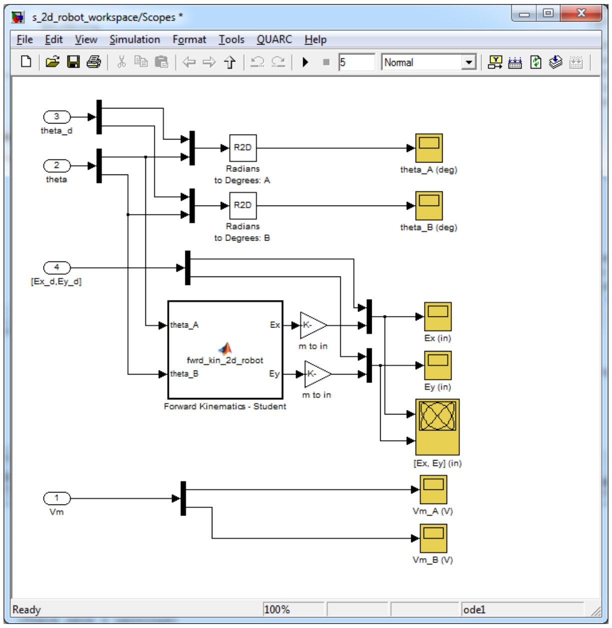


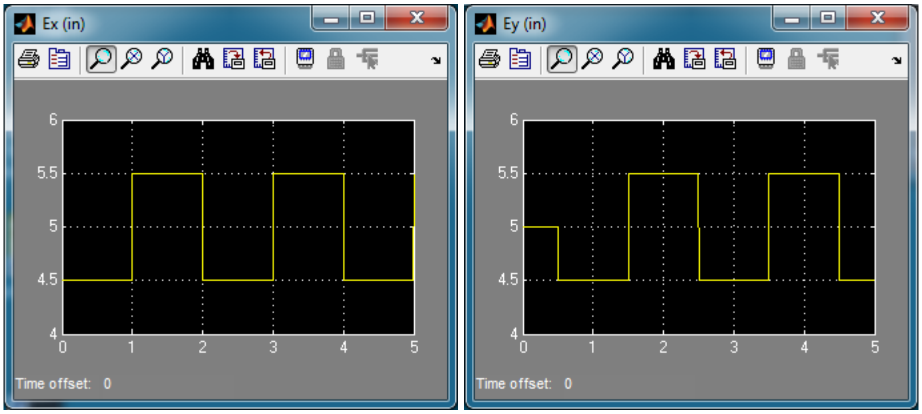
Figure 4.5: Scopes Simulink subsystem

- Signal type = *square*
- Amplitude = 1
- Frequency = 0.5 Hz

3. Set both *Amplitude (in)* gain blocks to 0.5 to generate a step with an amplitude of 0.5 inches (i.e., square wave goes between ± 0.5 which results in a step amplitude of 1 inch).

4. Open x and y end-effector position scopes *E_x (in)* and *E_y (in)*, as well as the motor input voltage scope, *Vm (V)*.

5. Start the simulation. By default, the simulation runs for 5 seconds. The scopes should be displaying responses similar to Figure 4.6. In the *E_x (in)* and *E_y (in)* scopes, the yellow trace is the setpoint position while the purple trace is the simulated position. However, in this case **the simulated position is not shown because the kinematics blocks have not been completed**.



(a) End-Effector X Position

(b) End-Effector Y Position

Figure 4.6: Initial simulated response

6. As shown below, the *Inverse Kinematics - Student* block outputs 0 for both the SRV02 A and B angles. Complete the function using the equations found in Section 4.2.

```
function [theta_A,theta_B] = inv_kin_2d_robot(Ex, Ey, Lb_in)
% link length (in)
Lb = Lb_in;
%
theta_A = 0;
theta_B = 0;
```

7. The *Forward Kinematics* □ *Student* block script shown below. It outputs 0 for the *Ex* and *Ey* end-effector positions. Using the exercises in Section 4.2, fill-in this Matab Embedded Function so the end-effector position can be measured from the servo angles.

   ```
   function [Ex,Ey] = fwrd_kin_2d_robot(theta_A, theta_B, Lb_in)
   % Convert link length to meters.
   Lb = 0.0254*Lb_in;
   %
   Ex = 0;
   Ey = 0;
   ```

8. Run the simulation with the completed kinematic blocks. Attach a plot of the *[Ex, Ey] (in)* Matlab figure to your report showing the end-effector setpoint and simulated response in the X-Y plane.
9. Produce a Matlab figure showing the time-based E_x and E_y response. Show both the setpoint and simulated positions in each response. Also, try to have the E_x and E_y responses in two separate plots.
10. In terms of steady-state error, percentage overshoot, and peak time, the E_x and E_y responses should be similar to the servo angle responses, θ_A and θ_B. Do you notice any additional effects in the end-effector responses?

4.3.2 Implementation

Experimental Setup

The q_2d_robot_workspace Simulink diagram shown in Figure 4.7 is used with QUARC® to implement the workspace control on the 2 DOF Robot system. The *2 DOF Robot* subsystem contains QUARC blocks that interface with the DC motor and sensors of the system.

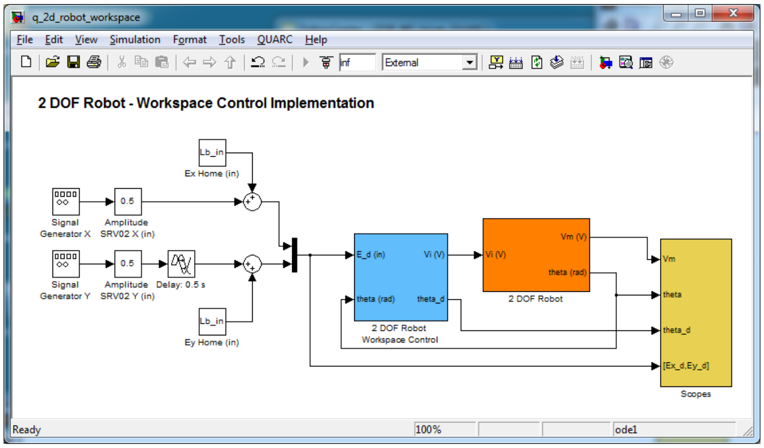
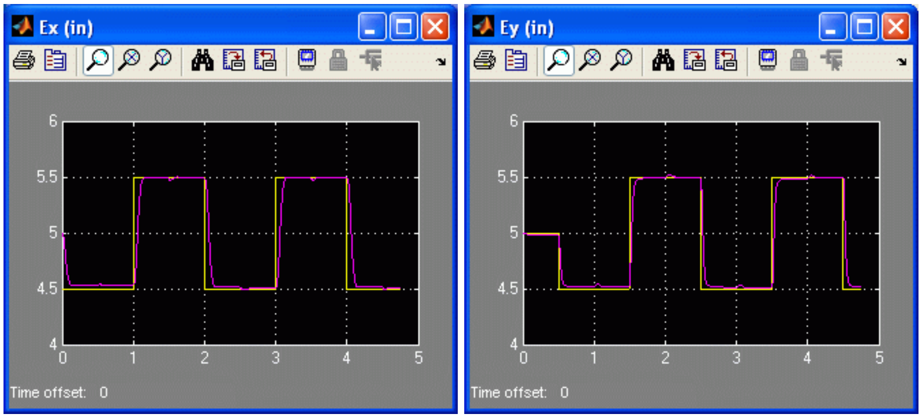


Figure 4.7: Simulink model used to workspace control on 2 DOF Robot device.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your SRV02 setup. If they have not been configured already, then you need to go to Section 5 to configure the lab files first.

Follow this procedure:

1. Enter the proportional and velocity control gains found in Pre-Lab question 3 in Matlab®as *kp* and *kv*.
2. To generate a step reference, go to the *Setpoint* subsystem ensure the each *Signal Generator* is set to the following:
    - Signal type = *square*
    - Amplitude = 1
    - Frequency = 0.5 Hz
3. Set both *Amplitude (in)* gain blocks to 0.5 to generate a step with an amplitude of 0.5 inches (i.e., square wave goes between ± 0.5 which results in a step amplitude of 1 inch). With these setting a 0.5 by 0.5 inch square reference trajectory will be generated.
4. Open x and y end-effector position scopes *E_x (in)* and *E_y (in)*, as well as the motor input voltage scope, *Vm (V)*.
5. In the Simulink diagram, go to QUARC | Build.
6. Copy the the kinematic blocks that were designed in s_2d_robot_workspace Simulink diagram in Section 4.3.1. That is, drag-and-drop your completed versions of the Inverse Kinematics□Student and Forward Kinematics□Student blocks.
7. Click on QUARC | Build to compile the Simulink diagram.
8. Select QUARC | Start to begin running the controller. The scopes should be displaying responses similar to Figure 4.8. The yellow and purple traces in the *Ex (in)* and *Ey (in)* scopes are the setpoint position and the simulated position, respectively.



(a) End-Effector X Position

(b) End-Effector Y Position

Figure 4.8: Typical workspace control response on 2 DOF Robot

9. Attach a plot of the *[Ex, Ey] (in)* Matlab figure to your report showing the end-effector setpoint and measured response in the X-Y plane.
10. Produce a Matlab figure showing the time-based E_x and E_y response. Show both the setpoint and measured positions in each response. Also, try to have the E_x and E_y responses in two separate plots.
11. Is the X-Y end-effector response comparable to the specifications (i.e., steady-state error, percentage overshoot, and peak time) given for the servo angle responses, θ_A and θ_B. Besides this, do you witness the same effect in the measured signal as seen in the simulation? Explain.
12. Click on the *Stop* button in the tool bar to stop QUARC.
13. Shut off the power of the amplifier if no more experiments will be performed on the SRV02 in this session.

5 SYSTEM REQUIREMENTS

Required Software

- Microsoft Visual Studio (MS VS)
- Matlab® with Simulink®, Real-Time Workshop, and the Control System Toolbox
- QUARC®

See the QUARC® software compatibility chart in [3] to see what versions of MS VS and Matlab are compatible with your version of QUARC and for what OS.

Required Hardware

- Data acquisition (DAQ) device that is compatible with QUARC®. This includes Quanser DAQ boards such as Q2-USB, Q8-USB, QPID, and QPIDe and some National Instruments DAQ devices. For a full listing of compliant DAQ cards, see Reference [1].
- Quanser SRV02-ET rotary servo.
- Quanser SRV02 2 DOF Robot (attached to SRV02).
- Quanser VoltPAQ-X1 power amplifier, or equivalent.

Before Starting Lab

Before you begin this laboratory make sure:

- QUARC® is installed on your PC, as described in [2].
- DAQ device has been successfully tested (e.g., using the test software in the Quick Start Guide or the *QUARC Analog Loopback Demo*).
- SRV02 2 DOF Robot and amplifier are connected to your DAQ board as described Reference [5].

5.1 Overview of Files

File Name	Description
2 DOF Robot User Manual.pdf	This manual describes the hardware of the 2 DOF Robot system and explains how to setup and wire the system for the experiments.
2 DOF Robot Workbook (Student).pdf	This laboratory guide contains pre-lab questions and lab experiments demonstrating how to design and implement controllers for both the joint space and work space on the 2 DOF Robot plant using QUARC®.
setup_2d_robot.m	The main Matlab script that sets the SRV02 motor and sensor parameters, the SRV02 configuration-dependent model parameters, and the 2 DOF Robot 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.
config_2d_robot.m	Returns the 2 DOF Robotrelated model parameters.
calc_conversion_constants.m	Returns various conversions factors.
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.*
s_2d_robot_joint_space.mdl	Simulink file that simulates a joint space, servo-based closed-loop PV controller for the 2 DOF Robotsystem.
s_2d_robot_workspace.mdl	Simulink file that simulates controlling the X-Y Cartesian position of the end-effector, i.e., the *workspace* control. **To be completed by student.**
q_2d_robot_joint_space.mdl	Simulink file that implements a local, servo-based closed-loop PV controller on the 2 DOF Robot system using QUARC®.
q_2d_robot_workspace.mdl	Simulink file that controls the X-Y position of the end-effector on the actual 2 DOF Robot system using QUARC®. **To be completed by student.**

Table 5.1: Files supplied with the 2 DOF Robot

5.2 Setup for Joint Space Simulation

Before beginning the in-lab procedure outlined in Section 4.3.1, the s_2d_robot_joint_space Simulink diagram and the setup_2d_robot.m script must be configured.

Follow these steps:

1. Load the Matlab software.
2. Browse through the *Current Directory* window in Matlab and find the folder that contains the file *setup_2d_robot.m*.
3. Open the *setup_2d_robot.m* script.
4. **Configure setup_2d_robot.m script**: When used with the 2 DOF Robot, the SRV02 has no load (i.e., no disc or bar) and has to be in the high-gear configuration. Make sure the script is setup to match this setup:
    - EXT_GEAR_CONFIG to 'HIGH'
    - LOAD_TYPE to 'NONE'

- K_AMP to 1 (unless your amplifier gain is different)
- AMP_TYPE to your amplifier type (e.g., VoltPAQ).
- Ensure other parameters such as ENCODER_TYPE, TACH_OPTION, and VMAX_DAC match your system configuration.
- CONTROL_TYPE to 'MANUAL'.

5. Run setup_2d_robot.m to setup the Matlab workspace.
6. Open the *s_2d_robot_joint_space.mdl* Simulink diagram, shown in Figure 3.6.

5.3 Setup for Implementing Joint Space Control

Before performing the in-lab exercises in Section 3.3.2, the q_2d_robot_joint_space Simulink diagram and the setup_2d_robot.m script must be configured.

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

1. Setup the SRV02 with the 2 DOF Robot module as detailed in the 2 DOF Robot User Manual ([6]).
2. Make sure the 2 DOF Robot is in the HOME position.
3. Configure and run setup_2d_robot.m as explained in Section 5.2.
4. Open the *q_2d_robot_joint_space.mdl* Simulink diagram, shown in Figure 3.6.
5. **Configure DAQ:** Ensure the HIL Initialize block in the *2 DOF Robot Joint* subsystem is configured for the DAQ device that is installed in your system. See Reference [1] for more information on configuring the HIL Initialize block.

5.4 Setup for Work Space Simulation

Before beginning the in-lab procedure outlined in Section 4.3.1, the s_2d_robot_workspace Simulink diagram and the setup_2d_robot.m script must be configured.

Follow these steps:

1. Configure and run setup_2d_robot.m as explained in Section 5.2.
2. Open the *s_2d_robot_workspace.mdl* Simulink diagram, shown in Figure 4.3.
3. This model is not complete. Go through the exercises in Section 4.3.1 to complete the kinematic blocks.
4. Open the setup_2d_robot.m file. This is the setup script used for the ROTFLEX Simulink models.

5.5 Setup for Implementing Work Space Control

Before performing the in-lab exercises in Section 4.3.2, the q_2d_robot_workspace Simulink diagram and the setup_2d_robot.m script must be configured.

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

1. Setup the SRV02 with the 2 DOF Robot module as detailed in the 2 DOF Robot User Manual ([6]).

2. Make sure the 2 DOF Robot is in the HOME position.
3. Configure and run setup_2d_robot.m as explained in Section 5.2.
4. Open the *q_2d_robot_workspace.mdl* Simulink diagram, shown in Figure 4.7.
5. This model is not complete. Go through the exercises in Section 4.3.2 to complete the kinematic blocks.
6. **Configure DAQ:** Ensure the HIL Initialize block in the *2 DOF Robot Joint* subsystem is configured for the DAQ device that is installed in your system. See Reference [1] for more information on configuring the HIL Initialize block.

6 LAB REPORT

This laboratory contains two groups of experiments, namely,

1. Joint space control, and
2. Workspace control.

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 Joint Space Control Report

I. PROCEDURE

1. *Simulation*
    - Briefly describe the main goal of the simulation.
    - Briefly describe the simulation procedure in Step 6 in Section 3.3.1.
2. *Implementation*
    - Briefly describe the main goal of this experiment.
    - Briefly describe the experimental procedure in Step 7 in Section 3.3.2.

II. RESULTS

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

1. Response plot from step 6 in Section 3.3.1, *Joint space simulation*.
2. Response plot from step 6 in Section 3.3.2, *Joint space implementation*.
3. Provide applicable data collected in this laboratory (from Table 3.1).

III. ANALYSIS

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

1. Peak time, percent overshoot, steady-state error, and input voltage in Step 7 in Section 3.3.1.
2. Peak time, percent overshoot, steady-state error, and input voltage in Step 8 in Section 3.3.2.

IV. CONCLUSIONS

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

1. Whether the controller meets the specifications in Step 7 in Section 3.3.1, *Joint space controller simulation*.
2. Whether the controller meets the specifications in Step 8 in Section 3.3.2, *Joint space controller implementation*.

6.2 Template for Workpace Control Report

I. PROCEDURE

1. *Simulation*
   - Briefly describe the main goal of the simulation.
   - Briefly describe the simulation procedure in Step 8 in Section 4.3.1.
2. *Implementation*
   - Briefly describe the main goal of this experiment.
   - Briefly describe the experimental procedure in Step 9 in Section 4.3.2.

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

1. Response plot from step 9 in Section 4.3.1, *Workspace simulation*.
2. Completed inverse kinematics code, in Step 6 in Section 4.3.1.
3. Completed direct kinematics code, in Step 7 in Section 4.3.1.
4. Response plot from step 10 in Section 4.3.2, *Workspace implementation*.

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

1. Examine the simulated workspace response in Step 10 in Section 4.3.1.
2. Examine the actual workspace response running on the 2 DOF Robotin Step 11 in Section 4.3.1.

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 4.3.2, *Workspace controller implementation*.

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. *QUARC Installation Guide*, 2009.

[3] Quanser Inc. *QUARC Compatibility Table*, 2010.

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

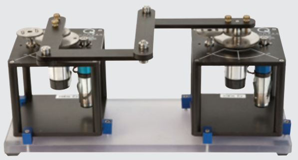
[5] Quanser Inc. *SRV02 Rotary Flexible Link User Manual*, 2011.

[6] Quanser Inc. *2 DOF Robot User Manual*, 2012.

Over ten rotary experiments for teaching fundamental and advanced controls concepts

Rotary Servo Base Unit

2 DOF Robot



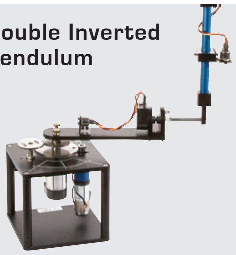
Inverted Pendulum

Flexible Link

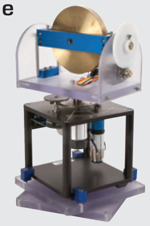
Flexible Joint

Ball and Beam

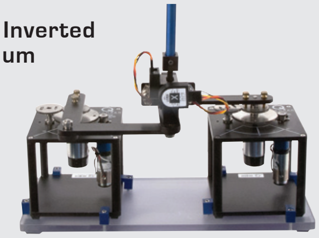
Double Inverted Pendulum



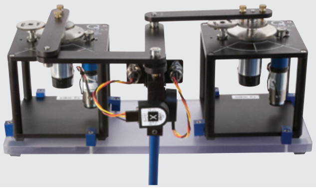
Gyro/Stable Platform



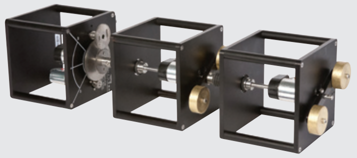
2 DOF Inverted Pendulum



2 DOF Gantry



Multi-DOF Torsion



Quanser's rotary collection allows you to create experiments of varying complexity – from basic to advanced. Your lab starts with the Rotary Servo Base Unit and is designed to help engineering educators reach a new level of efficiency and effectiveness in teaching controls in virtually every engineering discipline including electrical, computer, mechanical, aerospace, civil, robotics and mechatronics. For more information please contact info@quanser.com

QUANSER
INNOVATE. EDUCATE.

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

Solutions for teaching and research. Made in Canada.