

Rotary Motion Servo Plant: SRV02

Rotary Experiment #9: Double Inverted Pendulum Control

*Double Inverted Pendulum
using QUARC*



User and Lab Manual

Table of Contents

1. Laboratory Objectives.....	1
2. Prerequisites.....	1
3. Experiment Design Files.....	2
4. SRV02+DBIP Modeling and Control Design.....	3
4.1. System Modeling.....	3
4.2. Model Parameters.....	5
4.3. Balance Controller Design.....	6
5. In-Lab Procedure.....	8
5.1. System Setup.....	8
5.2. Wiring Procedure.....	8
5.2.1. Cable Nomenclature.....	9
5.3. Typical Connections.....	10
5.3.1. Wiring when using ETS Option.....	11
5.4. Controller Simulation.....	12
5.4.1. Objectives.....	12
5.4.2. Procedure.....	12
5.5. Controller Implementation.....	17
5.5.1. Objective.....	17
5.5.2. QUARC Controller Description.....	18
.....	18
5.5.3. Experimental Procedure.....	18
6. References.....	23

1. Laboratory Objectives

The challenge in this laboratory is to design a state-feedback control system that balances a rotary double inverted pendulum and positions the rotary arm to a commanded angular position. The system when running the balance controller is shown in Figure 1.

The plant has two main components: the Quanser SRV02 rotary motion plant and the Quanser Double Pendulum, i.e. DBPEN module. This is denoted as SRV02+DBPEN plant. As depicted in Figure 1, the double pendulum is composed of a rotary arm that attaches to the SRV02 system, a short 7-inch bottom blue rod, an encoder hinge, and the top 12-inch blue rod. The balance control computes a voltage based on the angle measurements from the encoders. This control voltage signal is amplified and applied to the SRV02 motor. The rotary arm moves accordingly to balance the two links and the process repeats itself.

Section 2 describes the requirements of the experiment and Section 3 overviews the files supplied with the system. The modeling, model parameters, and control design is explained in Section 4. In Section 5, the plant wiring and setup is described followed by procedures to run a simulation of the system and run the control on the actual SRV02+DBPEN plant in real-time. Lastly, Section 6 contains references that are cited throughout this manual.



Figure 1 SRV02+DBPEN system with balance control running.

2. Prerequisites

The prerequisites to successfully carry out this laboratory are:

- i) To be familiar with your SRV02 main components (e.g. actuator, sensors), your data acquisition card (e.g. QPID), and your power amplifier (e.g. VoltPAQ), as described in References [1], [2], and [3], respectively.
- ii) To be familiar in using QUARC to control and monitor the plant in real-time and in designing their controller through Simulink. Otherwise, review Reference [4] as needed.
- iii) To be familiar with the complete wiring and operating procedure of your SRV02 servo plant and dedicated amplifier, as discussed in Reference [1].

3. Experiment Design Files

Table 1, below, lists and describes the various computer files coming with the experiment.

<i>File Name</i>	<i>Description</i>
Rotary Double Inverted Pendulum – User and Lab Manual.pdf	The laboratory manual for the <i>SRV02 – Rotary Experiment #8: Double Inverted Pendulum (DBPEN) Control</i> laboratory. It contains information to setup and configure the laboratory, outlines the control design and its implementation, as well as the experimental procedure.
srv02_exp09_dbip.mws	Maple worksheet used to analytically derive the system's nonlinear equations of motion as well as its state-space model. Waterloo Maple 9, or a later release, is required to open, modify, and execute this file.
srv02_exp09_dbip.html	HTML presentation of the <i>HFLC DBIP Equations.mws</i> file. It allows to view the content of the Maple file without having Maple 9 installed. No modifications to the equations can be performed when in this format.
quanser_tools.mws	Executing this worksheet generates the <i>quanser</i> repository containing the <i>Quanser_Tools</i> package. The two package files are named: <i>quanser.ind</i> and <i>quanser.lib</i> . The <i>Quanser_Tools</i> module defines the generic procedures used in Lagrangian mechanics and resulting in the determination of a given system's equations of motion and state-space representation. It also contains data processing routines to save the obtained state-space matrices into a Matlab-readable file.
quanser_tools.rtf	Rich Text Format presentation of the <i>quanser_tools.mws</i> file. It allows to view the content of the Maple worksheet without having Maple 8 installed. No modifications to the Maple procedures can be performed when in this format.
setup_srv02_exp09_dbip.m	The main Matlab script that calls <i>config_srv02.m</i> , <i>config_sp.m</i> , <i>SRV02_DBIP_ABCD_eqns.m</i> to set the model, control, and configuration parameters. Run this file only to setup the laboratory.
setup_srv02_configuration.m	Returns the SRV02 model parameters Rm , kt , km , Kg , eta_g , Bm , Jm , eta_m , the encoder resolution constant K_EC , the amplifier voltage saturation.
config_sp.m	Returns the Single Pendulum (SP) model parameters m , L , l , J , and B according to the pendulum specified.

<i>File Name</i>	<i>Description</i>
SRV02_DBIP_ABCD_eqns.m	Matlab script file generated using the Maple worksheet <i>SRV02 DBPEN Equations.mws</i> . It sets the <i>A</i> , <i>B</i> , <i>C</i> , and <i>D</i> matrices for the state-space representation of the SRV02+DBPEN open-loop system used to design a control gain <i>K</i> in the <i>setup_srv02_exp8.m</i> script.
s_srv02_dbip.mdl	Simulink file that simulates the closed-loop SRV02+DBPEN system using its linear equations of motion model and a PID balance controller designed using LQR.
q_srv02_dbip.mdl	Simulink file that implements the real-time state-feedback DBPEN controller for the SRV02+DBIP system.

Table 1 Laboratory Design Files

4. SRV02+DBIP Modeling and Control Design

4.1. System Modeling

As already mentioned in Table 1, above, the Maple worksheet named *SRV02+DBPEN Equations.mws* or its HTML equivalent *SR02+DBPEN Equations.html* derives both nonlinear and linear models of the SRV02+DBPEN system. The worksheet equations can be edited and re-calculated by executing the worksheet using Maple 9. Accompanying the Maple worksheet is Figure 2 below.

Figure 2 depicts the coordinate systems used to find the kinematics of the system. Thus the arm tip, O_1 , the center of mass of link 1, O_2 , the encoder hinge, O_h , and the center of mass of link 2, O_2 , are all expressed as *XYZ* Cartesian coordinates in the O_0 system with respect to the arm and pendulum angles. More precisely the different coordinate systems are expressed in terms of the rotary arm angle, $\theta(t)$, the lower pendulum angle about the perfect upright vertical position, $\alpha(t)$, and the upper pendulum angle relative to the lower pendulum, $\gamma(t)$. Figure 2 illustrates how these angles are defined. As detailed in the Maple worksheet, the positions attained are used to calculate the potential energy and their respective velocities are used to calculate kinetic energy. The potential and kinetic energy are then used in the Euler-Lagrange equation to find the nonlinear equations of motion. These equations are linearized and the state-space representation of the system that is used in the control design is found.

The linear state-space model is

$$\frac{\partial}{\partial t} x = A x + B u \quad [1]$$

and

$$\frac{\partial}{\partial t} y = C y + D u \quad [2]$$

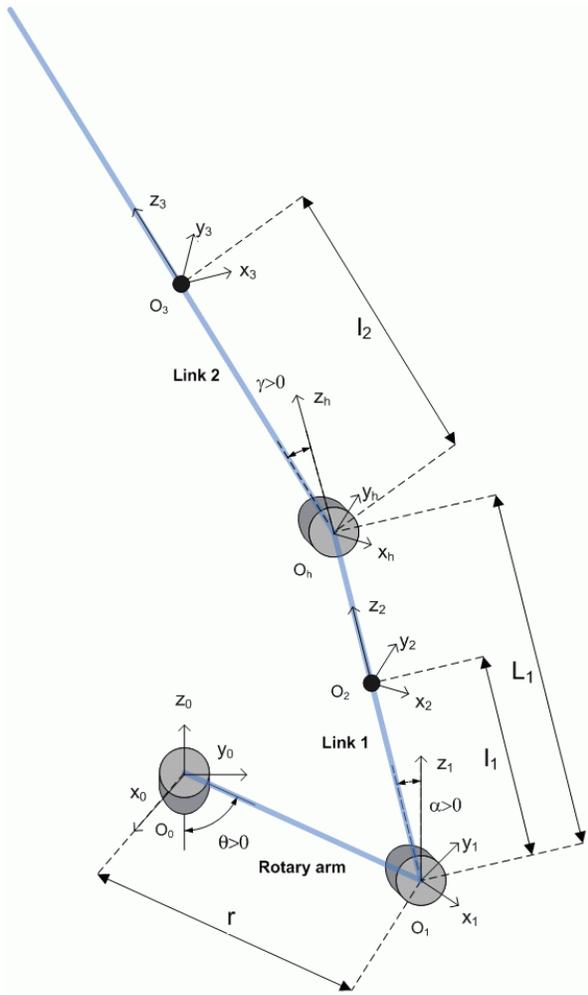


Figure 2 Kinematics of rotary double-pendulum system.

The A and B state-space matrices of the SRV02+DBPEN system are too large to be printed in this document but they are included in the Maple worksheet and its HTML equivalent. For the control design all the states are considered measured and the sensor noise is deemed negligible, thus

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad [3]$$

and

$$D^T = [0, 0, 0, 0, 0, 0] \quad [4]$$

The state used in [1] is defined as

$$x^T = \left[\theta(t), \alpha(t), \gamma(t), \frac{d}{dt}\theta(t), \frac{d}{dt}\alpha(t), \frac{d}{dt}\gamma(t) \right] \quad [5]$$

where $\theta(t)$ is the rotary arm angle, $\alpha(t)$ is the lower pendulum angle, and $\gamma(t)$ is the upper pendulum angle as described in Figure 2. All the angle are defined as counter-clockwise (CCW) and the motor turns CCW when the control input is positive, $u > 0$.

4.2. Model Parameters

Table 2, below, lists and characterizes the model parameters of the rotary double pendulum. See Reference [1] for the parameters associated with the SRV02 unit: R_m , k_t , k_m , K_g , η_{a_g} , B_m , J_m , η_{a_m} .

When the double pendulum is in the inverted configuration, the *lower* pendulum is the short pendulum that is attached to the SRV02 rotary arm and the *upper* pendulum is the longer pendulum that is attached to the lower pendulum. As described in Figure 2 the lower and upper pendulums are also referred to as link 1 and link 2, respectively.

<i>Symbol</i>	<i>Matlab Notation</i>	<i>Description</i>	<i>Value</i>	<i>Unit</i>
r	r	Rotary arm: length from pivot to tip.	0.2159	m
m_{arm}	m_arm	Rotary arm: mass.	0.2570	kg
J_m	Jm	Equivalent moment of inertia of SRV02 motor and the rotary arm.	0.0041	kg.m ²
m_1	m1	Lower pendulum: mass	0.0530	kg
B_1	B1	Lower pendulum: viscous damping.	0.0024	N.m.s/rad
L_1	L1	Lower pendulum: length from pivot to tip.	0.1778	m
l_1	l1	Lower pendulum: length from pivot to center-of-mass.	0.1635	m
m_2	m2	Upper pendulum: mass.	0.1060	kg
B_2	B2	Upper pendulum: viscous damping.	0.0024	N.m.s/rad

<i>Symbol</i>	<i>Matlab Notation</i>	<i>Description</i>	<i>Value</i>	<i>Unit</i>
l_2	l2	Upper pendulum: length from pivot to its center-of-mass.	0.1556	m
m_h	m_h	Mass of encoder hinge located between the lower and upper pendulum.	0.1410	kg

Table 2 Double Pendulum System Parameters

The equivalent moment of inertia of the seen at the SRV02 motor load shaft is

$$J_m = K_g^2 \mu_g J_{rotor} + J_{ext_gears} + J_{tach} + J_{load}, \quad [6]$$

where J_{rotor} is the moment of inertia from the motor rotor, J_{ext_gears} is the inertia contributed by the external gears on the SRV02, J_{tach} is the moment of inertia added by having a tachometer (i.e. on the SRV02-T or SRV02-ET units), and J_{load} is the moment of inertia of the load. For the SRV02 in the high-gear configuration the external gear moment of inertia is

$$J_{ext_gears} = J_{24} + 2J_{72} + J_{120} \quad [7]$$

where J_{24} , J_{72} , and J_{120} are the moment of inertias of the 24-tooth motor shaft gear, 72-tooth potentiometer gear, and 120-tooth load shaft gear on the SRV02. The gear inertias as well as J_{tach} are defined in the *set_srv02_configuration.m* script according to the SRV02 options specified. Finally, the rotary arm load attached to the motor adds considerable inertia. The moment of inertia of the rotary arm about its center of mass, modeled as a uniform rod, is

$$J_{load} = \frac{1}{12} m_{arm} r^2, \quad [8]$$

where m_{arm} and r are specified in Table 2. The rotary arm inertia is calculated numerically in the *setup_sp_configuration.m* script.

4.3. Balance Controller Design

The objective is to balance the inverted double-pendulum that is mounted on the rotary arm while tracking a desired arm angle. For this a PID controller will be designed using the Linear-Quadratic Regular technique, or LQR.

Consider the following feedback balance control

$$u = k (\zeta_d - \zeta) \quad [9]$$

where

$$\zeta^T = \left[\theta(t), \alpha(t), \gamma(t), \frac{d}{dt} \theta(t), \frac{d}{dt} \alpha(t), \frac{d}{dt} \gamma(t), \int \theta(t) dt \right] \quad [10]$$

and the desired state is

$$\zeta_d^T = \left[\theta_d(t), 0, 0, 0, 0, 0, \int \theta_d(t) dt \right] \quad [11]$$

The linear state-space model of the SRV02+DBPEN in Equation [1] is augmented by introducing an integrator, resulting in the system

$$\frac{d}{dt} \zeta(t) = \begin{bmatrix} A & 0 \\ 1 & 0 \end{bmatrix} \zeta + \begin{bmatrix} B \\ 0 \end{bmatrix} u \quad [12]$$

where A and B are the state-space matrices of the defined in [1]. The integration term $\zeta_7(t) = \theta(t)$ improves the steady-state error of the arm position and the increases the overall robustness of the controller. Thus as the error between the measured state and the desired state, $\zeta - \zeta_d$, converges to zero the arm angle tracks the desired position, $\theta(t) \rightarrow \theta_d(t)$, and the pendulum angles stabilize about zero: $\alpha(t) \rightarrow 0$ and $\gamma(t) \rightarrow 0$.

The feedback control gain in [9] is computed to be

$$k = [1.28, -42.49, -103.15, 1.88, -11.94, -11.72, 0.32] \quad [13]$$

using the Linear Quadratic Regulator (LQR) technique with the weighting matrices

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad R = 10 \quad [14]$$

The voltage applied to the SRV02 motor is the control input, u , saturated by the amplifier limits.

Not all the states are being measured in the rotary double-inverted system. There are three encoders measuring each angle but the respected velocities are not measured. The velocities of each angle are calculated by taking a derivative and filtering the result using a second-order low-pass filter. For example the filtered derivative of the arm angle, denoted by $\omega_f(t)$, is

$$\Omega_f(s) = \frac{s \omega_c^2 \Theta(s)}{s^2 + 2 \zeta \omega_c s + \omega_c^2} \quad [15]$$

where ζ is the damping ratio, ω_c is the cutoff frequency, and $\Theta(s)$ is the Laplace transform of the arm angle $\theta(t)$. However, in order to design a control gain using the linear-quadratic regular (LQR) technique it had to be assumed that all the states were measurable.

5. In-Lab Procedure

5.1. System Setup

To setup the double pendulum on the SRV02 go through the following:

1. As illustrated in Figure 1, clamp down the base plate of the SRV02 on a table that is both sturdy and level. If the table is not level you may achieve poor control performance.
2. As described in Reference [1], setup the SRV02 device in the high-gear configuration.
3. Slide the rotary arm onto the load shaft of the SRV02. Insert and tighten the two thumb screws to fasten the arm on the external gears of the servo.
4. Attach the T-fitting pivot on link 1 of the double pendulum to the tip of the rotary arm encoder shaft. Gently tighten the axis screw as needed to ensure the pendulum is properly fastened to the metal shaft.
5. **If using the VoltPAQ, make sure the gain is set to 1.**



5.2. Wiring Procedure

To setup this experiment, the following hardware and software are required:

- **Power Module:** Quanser VoltPAQ, or equivalent
- **Data Acquisition Board:** Quanser Q8-USB, QPID, or equivalent
- **Rotary Motion Servo Plant:** Quanser SRV02
- **Double Pendulum:** Quanser DBIP module
- **Real-Time Control Software:** The QUARC-Simulink configuration, as detailed in Reference [4], or equivalent

For a complete and detailed description of the main components comprising this setup, please refer to the corresponding manuals.

5.2.1. Cable Nomenclature

Table 3, below, provides a description of the standard cables used in the wiring of the SRV02 Rotary Pendulum system.

<i>Cable</i>	<i>Designation</i>	<i>Description</i>
	2xRCA to 2xRCA	This cable connects an analog output of the data acquisition terminal board to the power module for proper power amplification.
	4-pin-DIN to 6-pin-DIN	This cable connects the output of the power module, after amplification, to the desired DC motor on the servo.
	5-pin-ste- reo-DIN to 5-pin-ste- reo-DIN	This cable carries the encoder signals between an encoder connector and the data acquisition board (to the encoder counter). Namely, these signals are: +5VDC power supply, ground, channel A, and channel B.

Figure 3: "2xRCA to 2xRCA" cable.

Figure 4 "To Load" Cable

Figure 5 "Encoder" Cable

Table 3 Cable Nomenclature

5.3. Typical Connections

This section explains how to connect the SRV02+DBPEN plant to the amplifier and the data acquisition board. See Reference [5] for a more detailed explanation of the wiring required on the SRV02 module and Reference [1] for the specifications and a description of the main components composing the SRV02 system. The connections that are explained in the following steps are summarized in Table 4 and illustrated in Figure 6.



Make sure all the connections are performed when both the amplifier and PC are powered off.

1. Connect the 5-pin-DIN to RCA cable from the Analog Output Channel #0 on the DAQ terminal board to the "**Amplifier Command**" connector on the amplifier. This supplies the voltage signal computed by the controller running on the PC to the power amplifier.
2. Connect the 4-pin-DIN to 6-pin-DIN motor cable from the "**To Load**" connector on the amplifier to the "**Motor**" connector on the SRV02. This amplifies the control signal and drives the motor on the SRV02 unit.
3. Connect the encoder cable from the SRV02 "**Encoder**" connector to the encoder input #0 on the DAQ terminal board.
4. Wire the encoder cable from the pendulum 1 (i.e. short) encoder connector to encoder input #1 of the DAQ terminal board.
5. Wire the encoder cable from the pendulum 2 (i.e. medium) encoder connector to encoder input #2 of the DAQ terminal board.

<i>Cable #</i>	<i>From</i>	<i>To</i>	<i>Function</i>
1	Terminal Board: Analog Output #0	Amplifier: "Amplifier Command" Connector	Control signal generated by controller running on PC.
2	Amplifier: "To Load" Connector	SRV02: Motor Connector	Amplified control signal applied to SRV02 motor, V_m .
3	SRV02: Encoder Connector	Terminal Board: Encoder Input #0	Measures the SRV02 rotary arm angle, θ .
4	DBPEN: Short Link Pendulum Encoder Connector	Terminal Board: Encoder Input #1	Measures the lower pendulum angle, α .
5	DBPEN: Medium Link Pendulum Encoder Connector	Terminal Board: Encoder Input #2	Measures the upper pendulum angle, γ .

Table 4 SRV02+DBPEN Wiring Summary

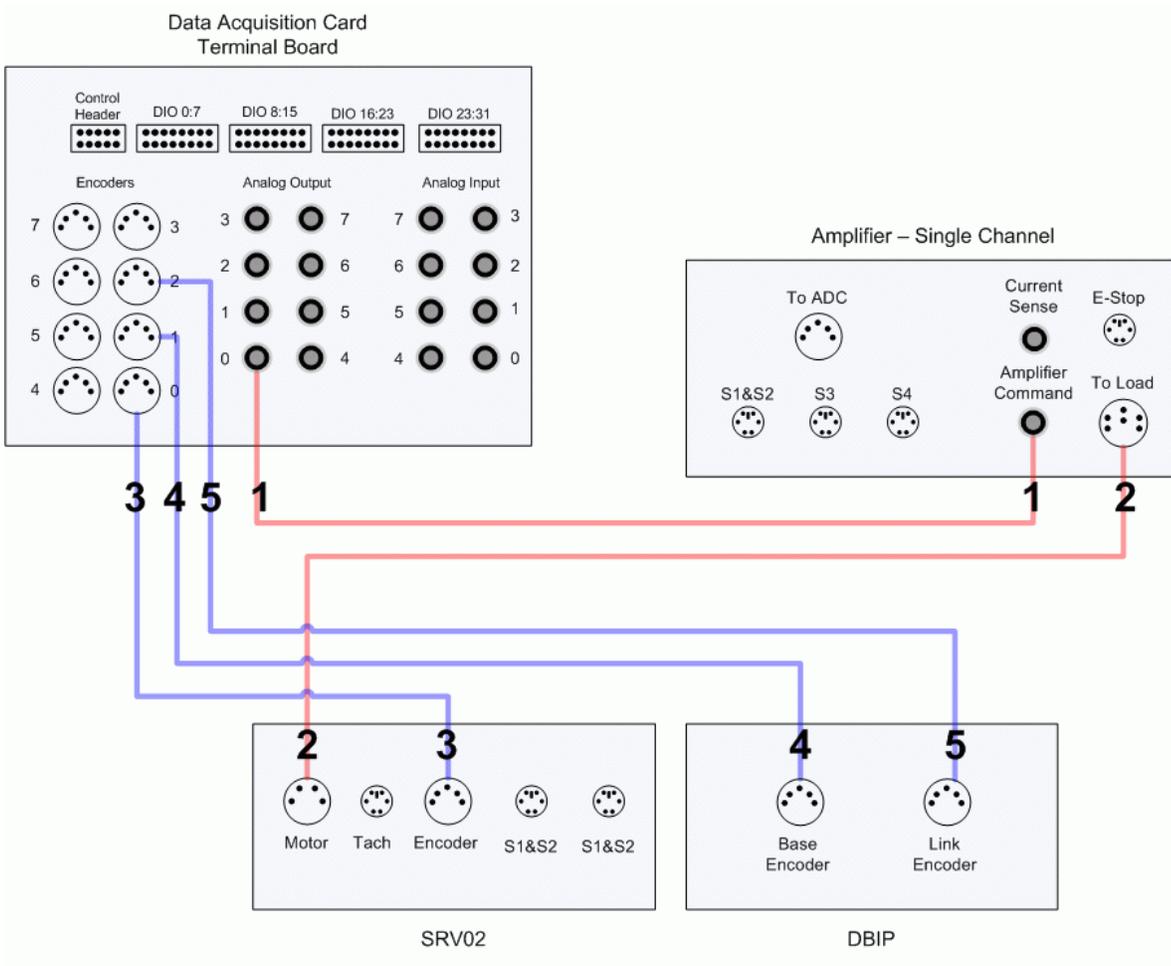


Figure 6: Connecting the double-inverted pendulum

5.3.1. Wiring when using ETS Option

If you are using the SRV02-ETS system then the encoder signals are passed through the slip ring. In this configuration, see the wiring summary in Table 5.

<i>Cable #</i>	<i>From</i>	<i>To</i>	<i>Function</i>
1	Terminal Board: Analog Output #0	Amplifier: “Amplifier Command” Connector	Control signal generated by controller running on PC.
2	Amplifier: “To Load” Connector	SRV02: Motor Connector	Amplified control signal applied to SRV02 motor, V_m .
3	SRV02: Encoder Connector	Terminal Board: Encoder #0	Measures the SRV02 rotary arm angle, θ .
4	DBPEN: Pendulum 1 Encoder Connector	SRV02: Slip-Ring Left Encoder Connector	Channels lower pendulum angle measurement to SRV02 Left connector via slip ring.
5	DBPEN: Pendulum 2 Encoder Connector	SRV02: Slip-Ring Right Encoder Connector	Channels upper pendulum angle measurement to SRV02 Right connector via slip ring.
6	Left Encoder Connector on SRV02	Terminal Board: Encoder #1	Measures the lower pendulum angle, α .
7	Right Encoder Connector on SRV02	Terminal Board: Encoder #2	Measures the upper pendulum angle, γ .

Table 5: Wiring summary when using SRV02-ETS with DBPEN.

5.4. Controller Simulation

5.4.1. Objectives

- Investigate the closed-loop performance of the balance control using the linear model of the SRV02+DBPEN system.
- Ensure the balance controller does not saturate the actuator.

5.4.2. Procedure

Follow these steps to simulate the response of the SRV02+DBPEN using the balance controller:

Step 1. Load Matlab and go to the DBIP lab files.

Step 2. Open Simulink model *s_dbip.mdl* as shown in Figure 7.

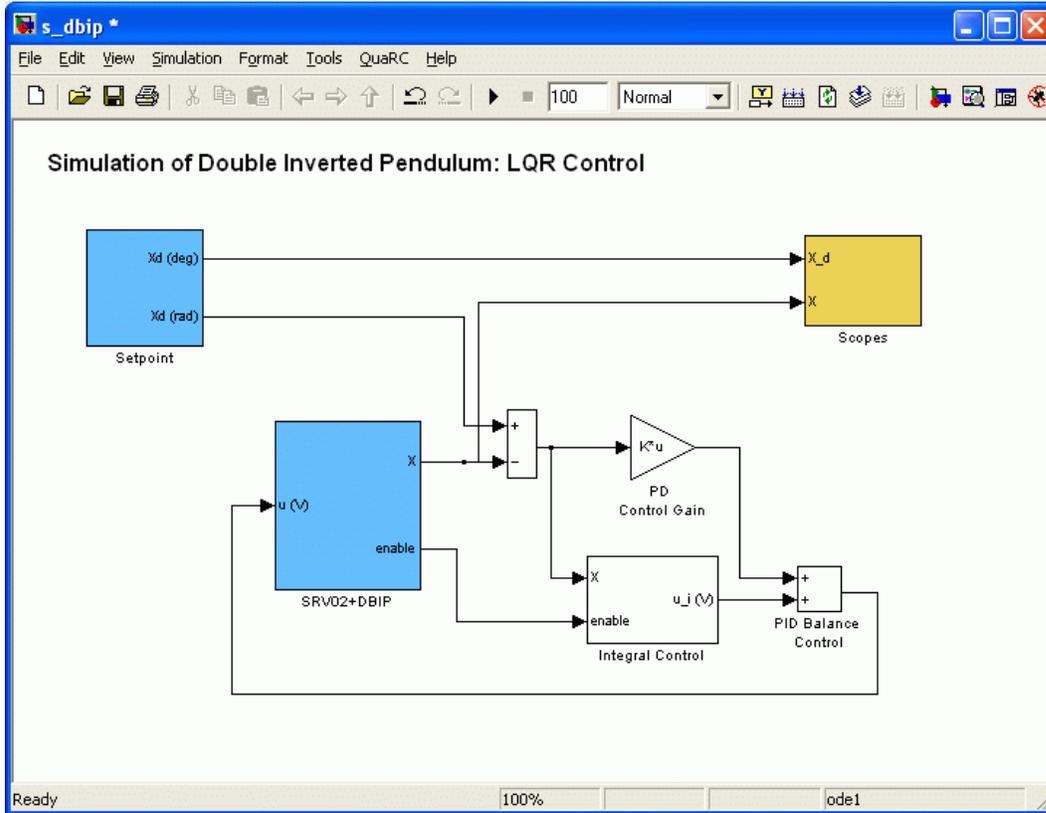


Figure 7 Simulink model used to simulate SRV02+DBIP closed-loop system.

- Step 3. Run the Matlab script `setup_srv02_exp09_dbip.m` to set the state-space matrices A , B , C , and D , the control gain k , the initial state X_0 , the maximum output voltage of the power amplifier V_{MAX_AMP} , and the maximum output voltage of the data acquisition board V_{MAX_DAC} .
- Step 4. In the *Setpoint* subsystem shown in Figure 8, below, set the *Amplitude (deg)* gain and the *Frequency* in the *Signal Generator* block to generate the desired position of the rotary arm. By default the setpoint is a square wave with an amplitude of 30 degrees at a frequency of 0.02 Hertz.

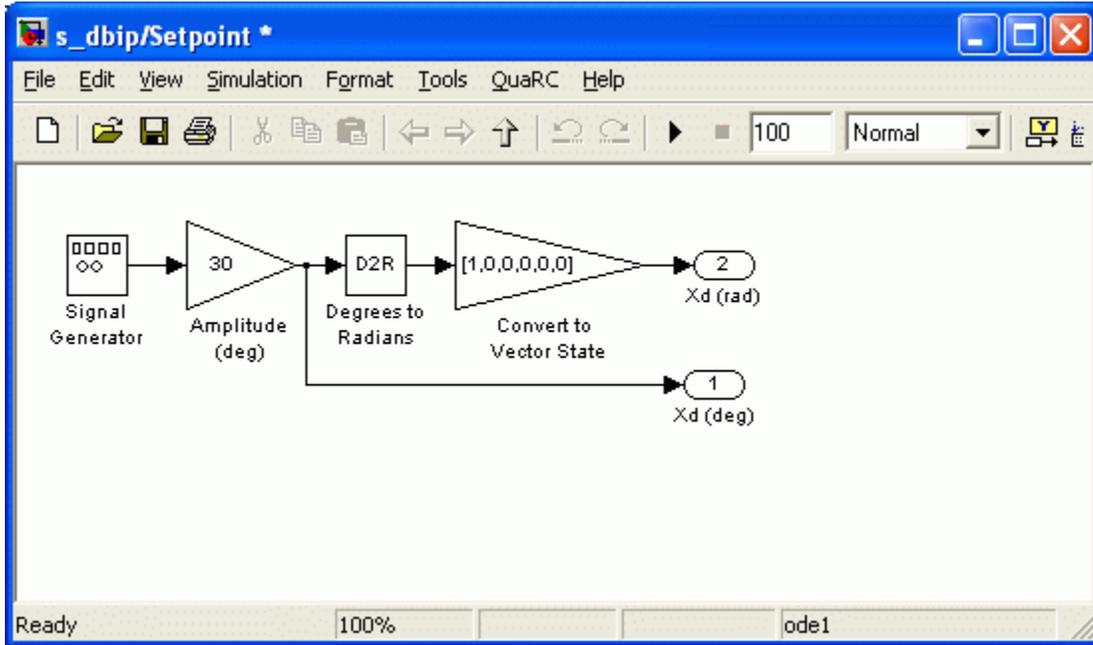


Figure 8: Setpoint subsystem in s_dbip Simulink diagram.

Step 5. The blue SRV02+DBIP contains the linear state-space model of the SRV02 and double inverted pendulum, as shown in Figure 9.

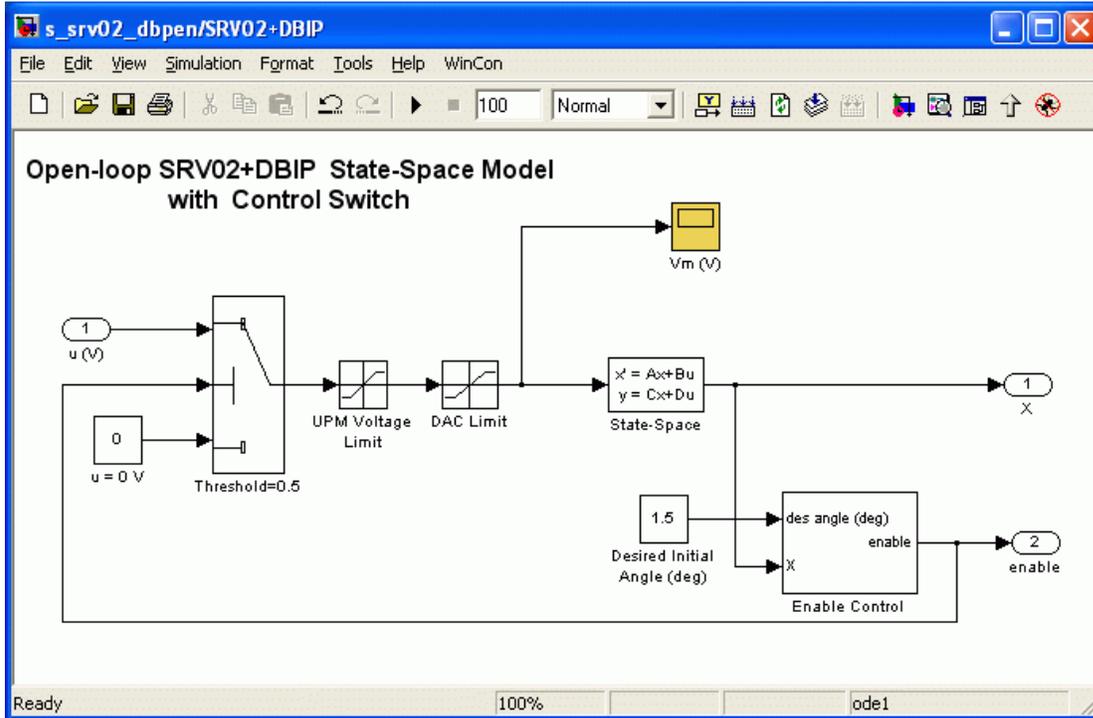


Figure 9 Interior of SRV02+DBIP subsystem.

- Step 6. The *Enable Control* subsystem outputs 1 when the pendulum angles α and γ are within the value in the *Desired Initial Angle (deg)* source block. By default, the controller is enabled when they are within 1.5 degrees.
- Step 7. Back to Figure 7, the controller has the proportional-derivative (PD) action and the integral (I) action. The *PD Control Gain* block multiplies the position and velocity states of the model by the first six elements of the vector gain k computed in earlier in [13], i.e. $k(1:6) = [k_1, k_2, k_3, k_4, k_5, k_6]$. The *Integral Control* subsystem is illustrated in Figure 10 and it implements the integral control. The integrator is reset when the controller is enable and, in addition, the output of the subsystem is zero until the controller is reset.

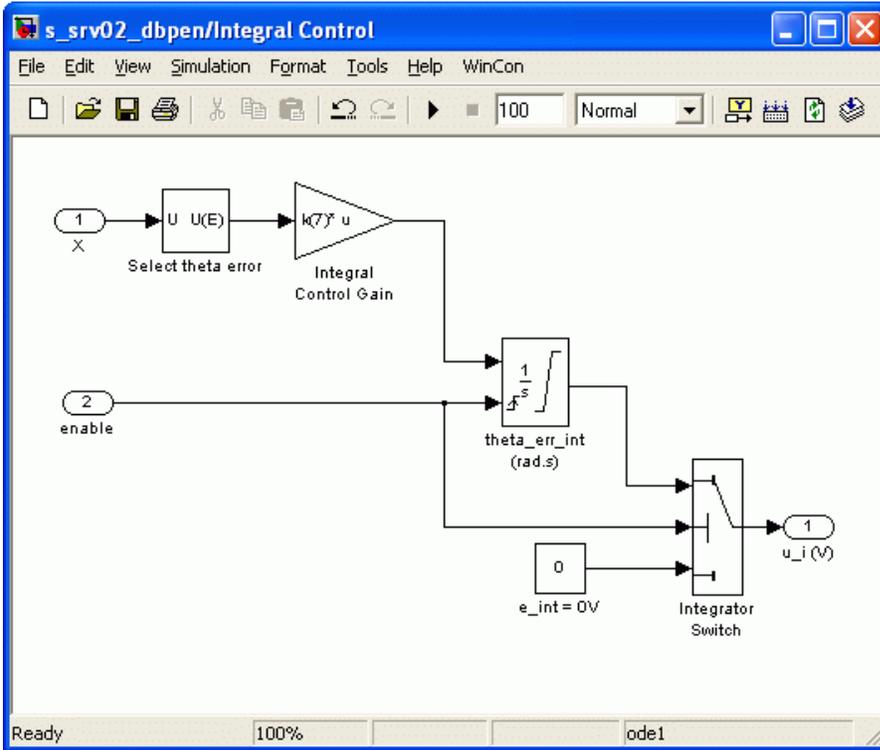


Figure 10 Integral Control subsystem.

Step 8. Set the simulation time (by default the time is 100 seconds) and click on start simulation button on the toolbar, or press Ctrl-T. As depicted in the scopes in Figure 11, the position of the arm should track the desired square position signal as the double pendulum is balanced about the upright vertical, that is α and γ are about 0 degrees.

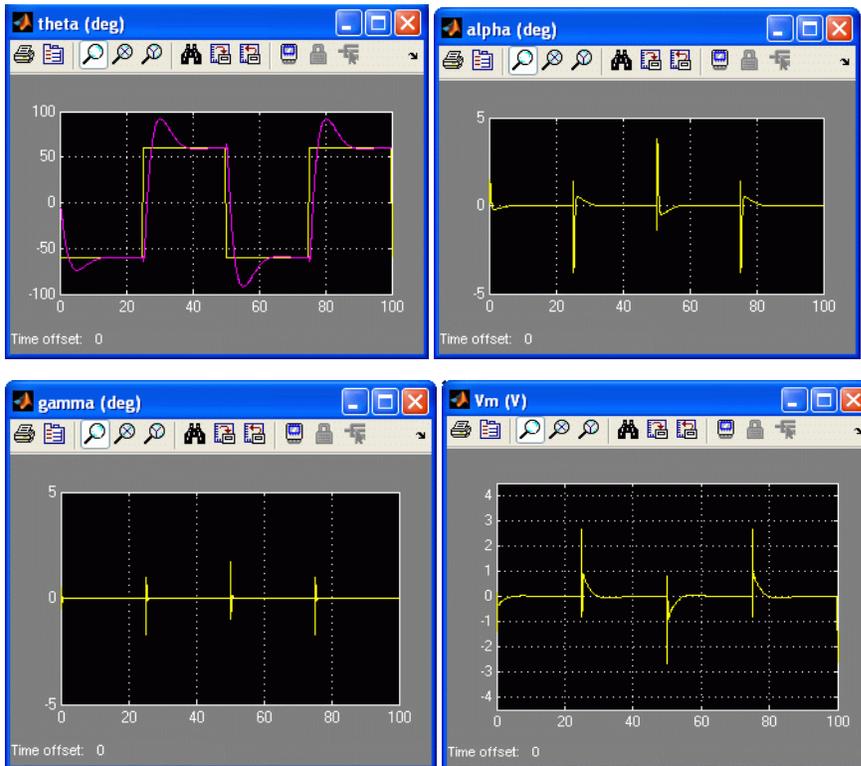


Figure 11 Scopes showing the simulated SRV02+DBPEN response.

Step 9. The motor input voltage, V_m , is shown in the $V_m (V)$ scope in Figure 11. The voltage is limited by $\pm 10V$ and as shown in Figure 11 the controller does not surpass this.

Step 10. The user is encouraged to change the initial condition $X0$ in the script and/or the desired arm position and observe the effect on the simulated closed-loop response.

5.5. Controller Implementation

5.5.1. Objective

- Implement the previously designed balance controller with QUARC to balance the double inverted pendulum system on the SRV02 rotary arm

5.5.2. QUARC Controller Description

The q_dbip Simulink diagram shown in Figure 12, below, is used to implement the $SRV02+DBIP$ subsystem. It contains QUARC blocks that interface with the DC motor and sensors of the $SRV02+DBIP$ system. The other blocks are the same as in the s_dbip Simulink diagram, which was used to simulate the closed-loop control of the rotary double-inverted pendulum.

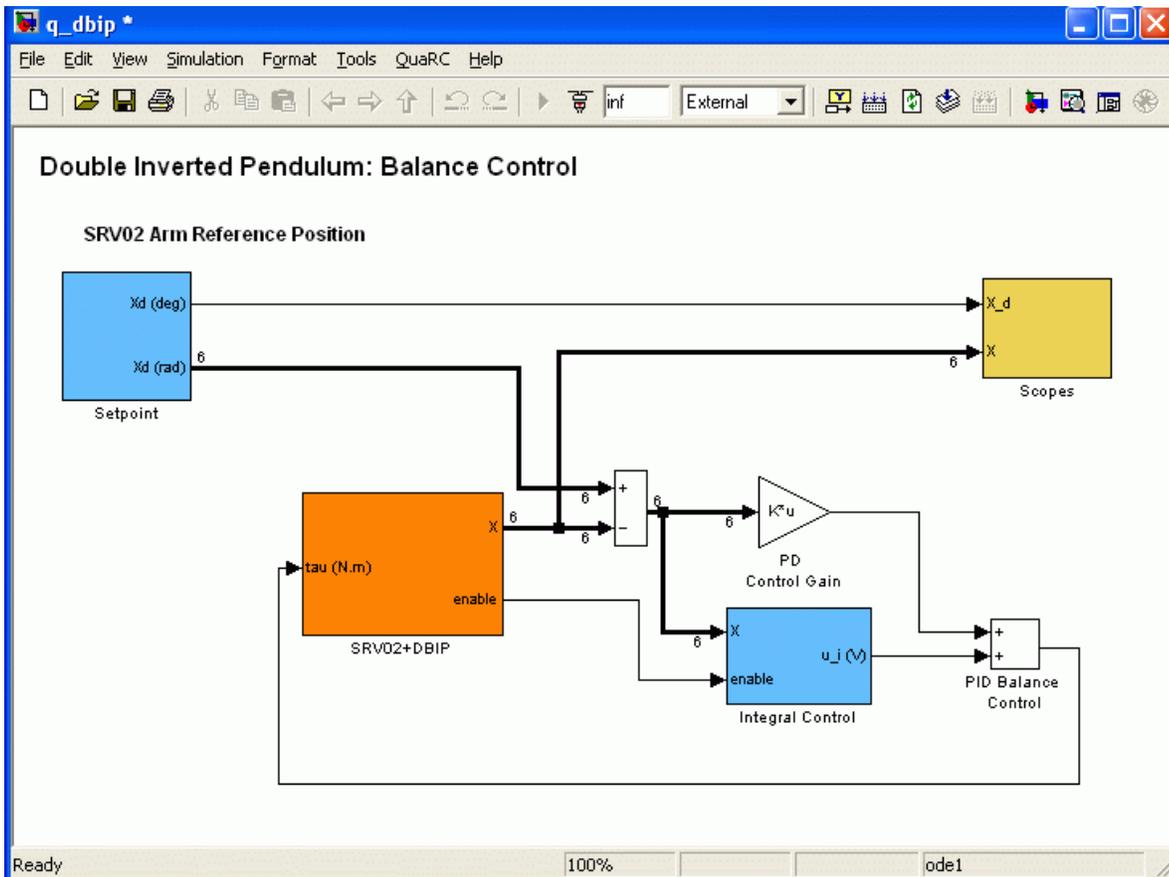


Figure 12: Simulink diagram used with QUARC to balance DBIP on SRV02.

5.5.3. Experimental Procedure

Follow the steps described below to implement the designed controller and observe its effect on the actual $SRV02+DBIP$ plant:

Step 1. Open Simulink model $q_srv02_dbip.mdl$ shown in Figure 12, above, that implements the

LQR-based PID balance controller. Similarly to the *s_dbip.mdl* Simulink model there is the *Setpoint* subsystem, the *PD Control Gain* block, and *Integral Control* subsystem. See Section 5.4.2 for a description of these elements.

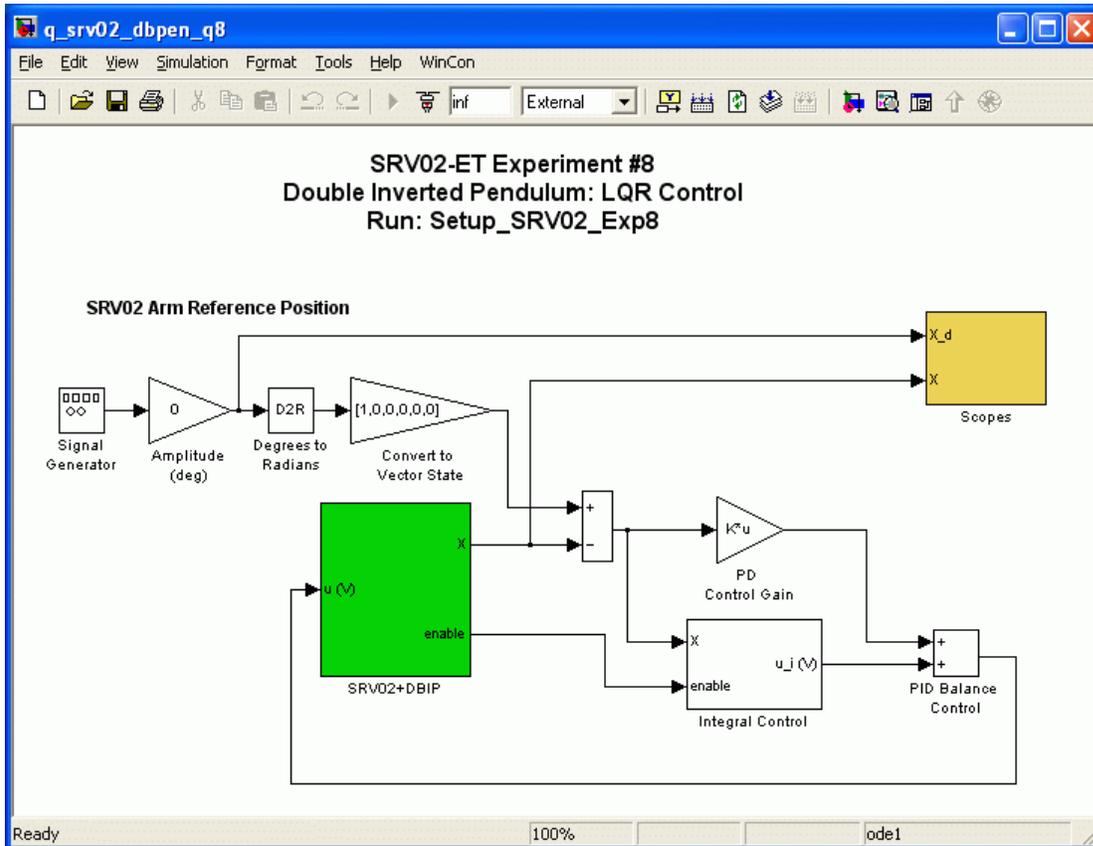


Figure 13 SRV02+DBPEN balance control implementation.

Step 2. The interior of the SRV02+DBIP block is shown in Figure 14. The *Enable Control* subsystem outputs 1 when the pendulum angles α and γ are within the value in the *Desired Initial Angle (deg)* source block. By default, the controller is enabled when they are within 1.5 degrees. The *Angle Watchdog* subsystem monitors the SRV02 rotary arm and pendulum angles. If arm angle θ goes beyond THETA_MAX or THETA_MIN, given that THETA_LIM_ENABLE = 1, then QUARC stops running and an error message is outputted. The THETA_MIN and THETA_MAX constants are specified in the *setup_srv02_exp09_dbip.m* script and by default are set to +/- 90 degrees. Similarly if the lower pendulum goes beyond ALPHA_MIN or ALPHA_MAX and the upper pendulum goes beyond GAMMA_MIN or GAMMA_MAX (given that ALPHA_LIM_ENABLE = 1 and GAMMA_LIM_ENABLE = 1) then QUARC is stopped and a descriptive error message is given. By default the pendulum limits are set to +/- 25 degrees.

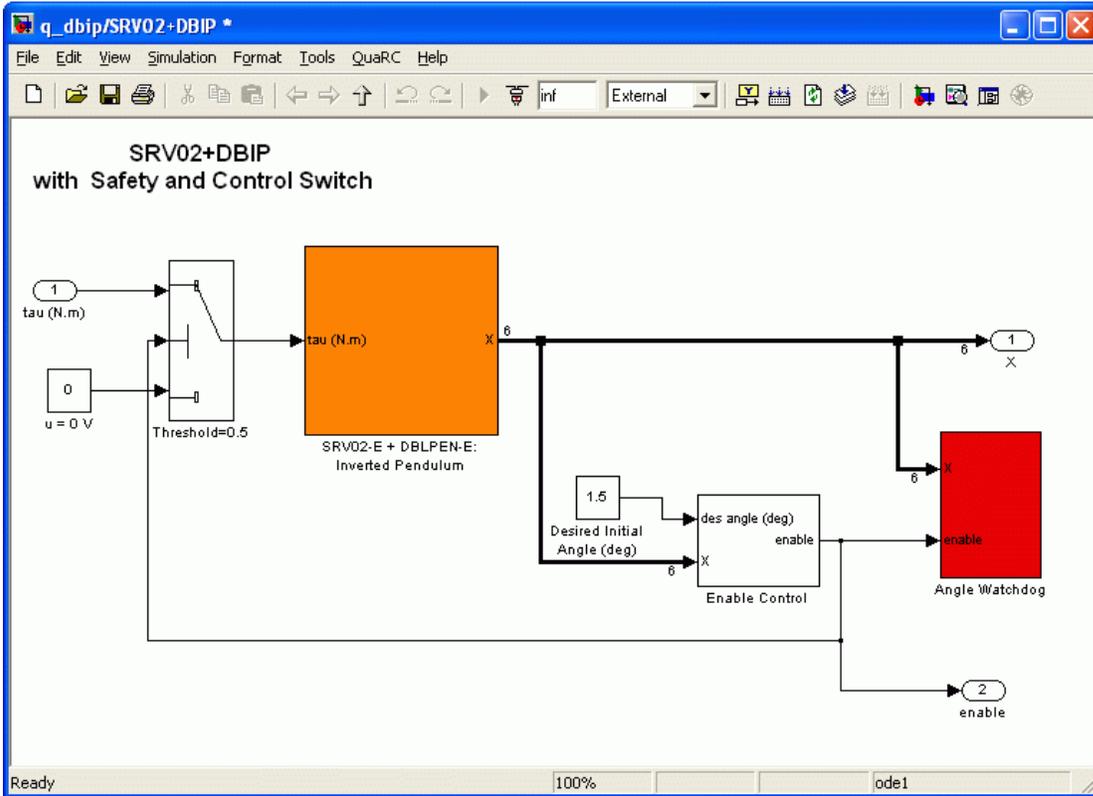


Figure 14 SRV02+DBIP subsystem.

Step 3. The green SRV02-E + DBLPEN-E block in Figure 14 interfaces with the actual hardware and its content is shown in Figure 15. The V_{MAX_AMP} , K_AMP , and V_{MAX_DAC} are all set in the `setup_srv02_exp09_dbip.m` script depending on the type of amplifier and data-acquisition board chosen. The amplifier gain does not have to be included in the model since the control signal is divided by the gain of the amplifier in order to have the same reference and applied voltage. The *Direction Convention* gain block makes a positive control signal result in the rotary arm turning counter-clockwise as specified in the model conventions. The V_m (V) scope displays the reference voltage applied to the DC motor by the amplifier. The gain K_ENC converts counts to radians. Recall that the control design is based on a model that defines $\alpha = 0$ when the bottom pendulum is in the upright position and $\gamma = 0$ when the top pendulum is in the upright position relative to the bottom pendulum. Therefore given that the bottom pendulum begins in the suspended position an offset of $-\pi$ rad (-180°) is applied to angle α . Lastly, the velocities of the angles are calculated by taking their derivative and filtering the result, as explained in Section 4.3, using the transfer function shown in [15]. This is implemented using the Simulink transfer function blocks. The low-pass filter cutoff frequency for the arm angle, wcf_1 , and the cutoff frequency for both pendulula, wcf_2 , are specified in the `setup_srv02_exp09_dbip.m` script.

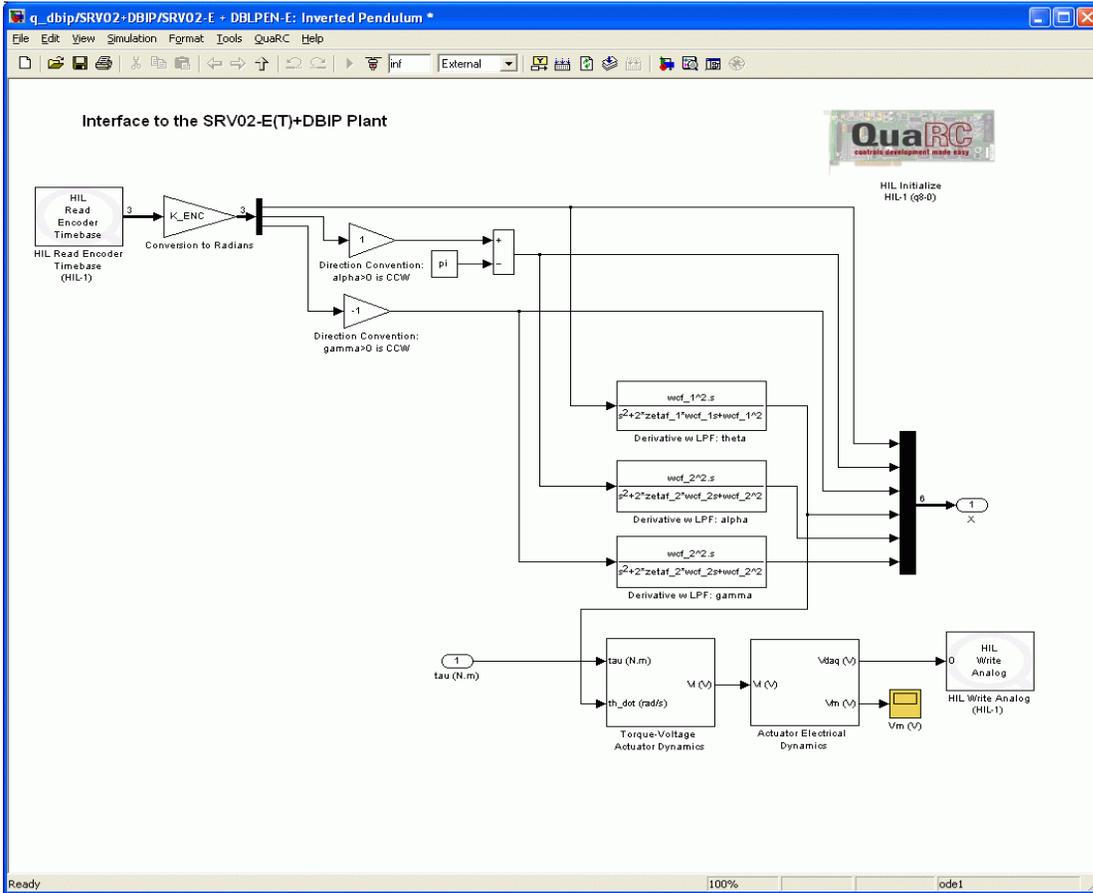


Figure 15 Interface to SRV02-E + DBPEN-E system.

Step 4. **Configure DAQ:** Ensure the HIL Initialize block in the *SRV02-E+DBLPEN-E* subsystem shown in Figure 15 is configured for the DAQ device that is installed in your system. By default, the block is setup for the Quanser Q8 hardware-in-the-loop board. See Reference [4] for more information on configuring the HIL Initialize block.

Step 5. **Setup script:** Open the design file *setup_srv02_lab09_dbip.m* and ensure the settings are properly set.



CAUTION: Ensure the AMP_TYPE is set to the correct amplifier you are using. **If using the VoltPAQ, make sure the gain is set to 1 and the K_AMP =1 in setup_srv02_lab09_dbip.m.**

NOTE: Make sure the ETS_OPTION is set to YES or NO depending on whether or not your SRV02 has a slip-ring (i.e SRV02-ETS option).

- Step 6. Execute the design file *setup_srv02_lab09_dbip.m* to setup the workspace before compiling the diagram and running it in real-time with QUARC. As already mentioned, this file sets the feedback gain vector k in the Matlab workspace as well as various other parameters such as the filter cutoff frequencies, safety limits, and the maximum output voltage of the amplifier.
- Step 7. The controller is activated when both pendula are rotated until their angles are within the tolerance specified in the box labeled *Desired Initial Angle (deg)* shown in Figure 14. By default the tolerance is set to 1.5 degrees.
- Step 8. The real-time code corresponding to the diagram can now be built by selecting the *QUARC | Build* item from the Simulink menu bar. After successful compilation and download you should be able to run your actual system in real-time.
- Step 9. From the Scopes subsystem, double-click on the *theta (deg)*, *alpha (deg)* and *gamma (deg)* sinks that display the measured arm, bottom pendulum, and upper pendulum angles, respectively. Also open the V_m (V) scopes to view the voltage being fed to the servo motor.
- Step 10. Make sure the SRV02 base is firmly clamped down onto a rigid table, as shown in Figure 16 below.
- Step 11. Ensure the double pendulum is in the suspended position as depicted in Figure 16 below. When both pendulums are motionless, start the real-time controller by clicking on the *QUARC | Start* (or the play button in the tool bar).
- Step 12. As illustrated in Figure 16, slowly rotate the entire double pendulum assembly **counter-clockwise** until it is vertical. When both the bottom and the top pendulum are within 1.5 degrees of their vertical position (pointing upwards), the motion control loop will automatically start and balance the double pendulum.

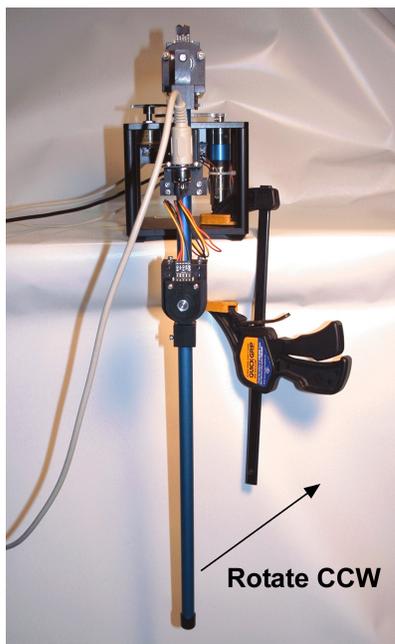


Figure 16 Starting Position of the SRV02+DBPEN system.

Step 13. Release the pendulum as soon as the controller is activated. Any obstructions are not compensated for and may lead to instability.

Step 14. A sample response is shown in

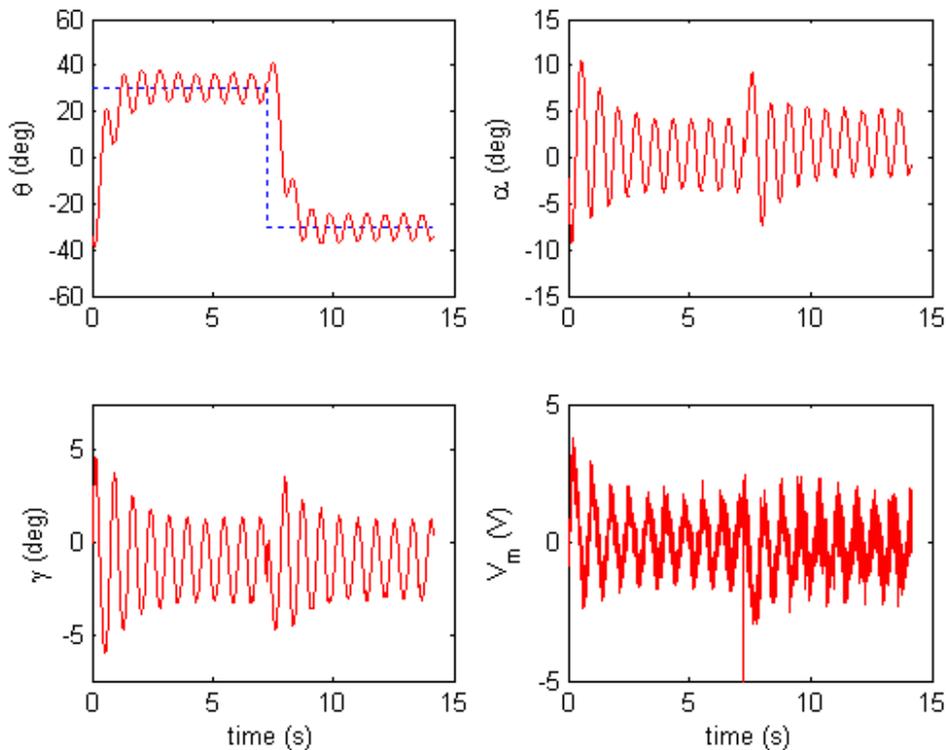


Figure 17: Typical response of DBIP when servo is tracking a +/- 30 degree step.

Step 15. Click on the *Stop* button located in the *q_dbip* Simulink diagram menu bar to stop running the controller.



CAUTION: Make sure you catch the double-pendulum when stopping the controller to avoid having the device strike any obstructions that may cause it damage.

Step 16. Shut off the amplifier if the laboratory session is over.

6. References

- [1] SRV02 User Manual.
- [2] DAQ User Manual.
- [3] Amplifier User Manual
- [4] QUARC User Manual (type `doc quarc` in Matlab to access).
- [5] SRV02 QUARC Integration