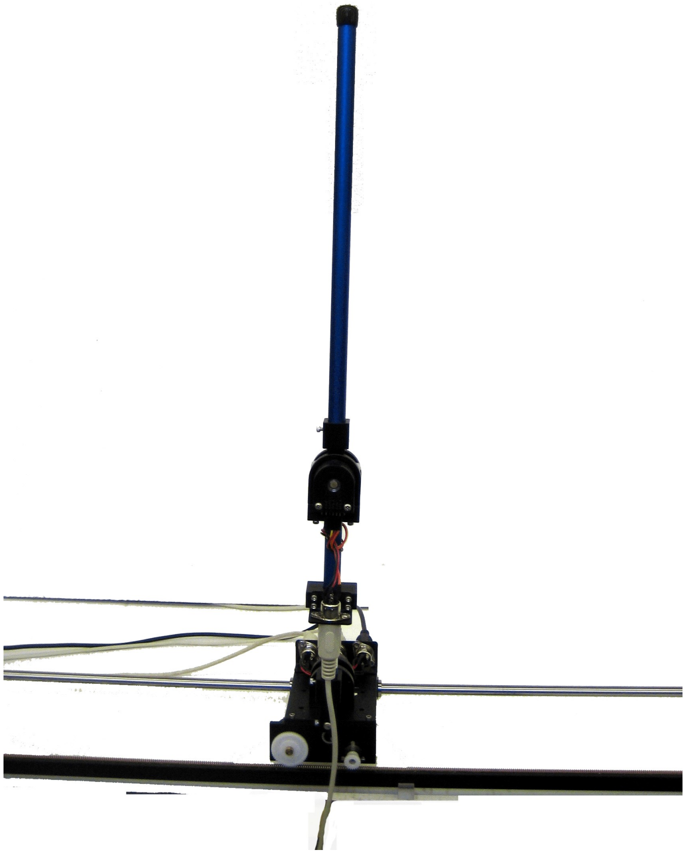
Linear Motion Servo Plant: IP01_2

Linear Experiment #15: LQR Control

Double Inverted Pendulum (DBIP)



All of Quanser's systems have an inherent open architecture design. It should be noted that the following experimental setup, accompanying files, and configuration are merely one of the many possible uses of this product.

Laboratory Manual

Table of Contents

1. Laboratory Objectives

The challenge in this laboratory is to design a state-feedback control system that keeps a double inverted pendulum (consisting of two pendula coupled vertically) balanced and tracks the IP01_2 cart to a commanded position.

2. Prerequisites

To successfully carry out this laboratory, the prerequisites are:

- To be familiar with your IP01 or IP02 main components (e.g. actuator, sensors), your data acquisition card (e.g. Q8), and your power amplifier (e.g. VoltPAQ), as described in References [1], [2], and [3], respectively.
- 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.
- To be familiar with the complete wiring and operating procedure of your IP01 or IP02 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.

File Name	***Description***
IP01_2 DBIP Equations.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.
quanser_tools.mws	Executing this worksheet generates the *quanser* repository containing the *Quanser_Tools* package. The two package files are named: *quanser.ind* and *quanser.lib*. The *Quanser_Tools* 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.

File Name	***Description***
quanser_tools.rtf	Rich Text Format presentation of the quanser_tools.mws file. It allows to view the content of the Maple worksheet without having Maple 9 installed. No modifications to the Maple procedures can be performed when in this format.
setup_lab_ip01_2_dbip.m	The main MATLAB script that calls *setup_sp_configuration.m*, and *DBIP_ABCD_eqns.m*, to set the model and configuration parameters and design the controller. **Run this file only to setup the laboratory.**
setup_ip01_2_configuration.m	Returns the model parameters *Jm*, *Kt*, *Eff_m*, *Mc*, *r_mp*, and *Beq*, the encoder resolution constant *K_EC* and *K_EP*, the amplifier current gain *Ka*, and the voltage and current saturation of the amplifier.
setup_sp_configuration.m	Returns the Single Pendulum (SP) model parameters *Mp*, *Lp*, *lp*, *Jp*, and *Bp* according to the pendulum specified.
DBIP_ABCD_eqns.m	MATLAB script file generated using the Maple worksheet *IP01_2 DBIP Equations.mws*. It sets the *A*, *B*, *C*, and *D* matrices for the state-space representation of the DBIP open-loop system used in the Simulink models *s_dbip_lqr_ip02.mdl* and *q_dbip_lqr_ip02.mdl*.
s_dbip_lqr_ip02.mdl	Simulink file that simulates the closed-loop DBIP system using its linear equations of motion model and an LQR controller.
q_dbip_lqr_ip02.mdl	Simulink file that implements the real-time state-feedback controller for the DBIP system.

Table 1 Laboratory Design Files

4. DBIP Controller Design

4.1. DBIP System Modeling

As already mentioned in Table 1, above, the Maple worksheet named *IP01_2 DBIP Equations.mws* (or its HTML equivalent *IP01_2 DBIP Equations.html*) derives both nonlinear and linear models of the DBIP system. The free body diagram that accompanies the dynamic model derivation of the system is given in Figure 1.

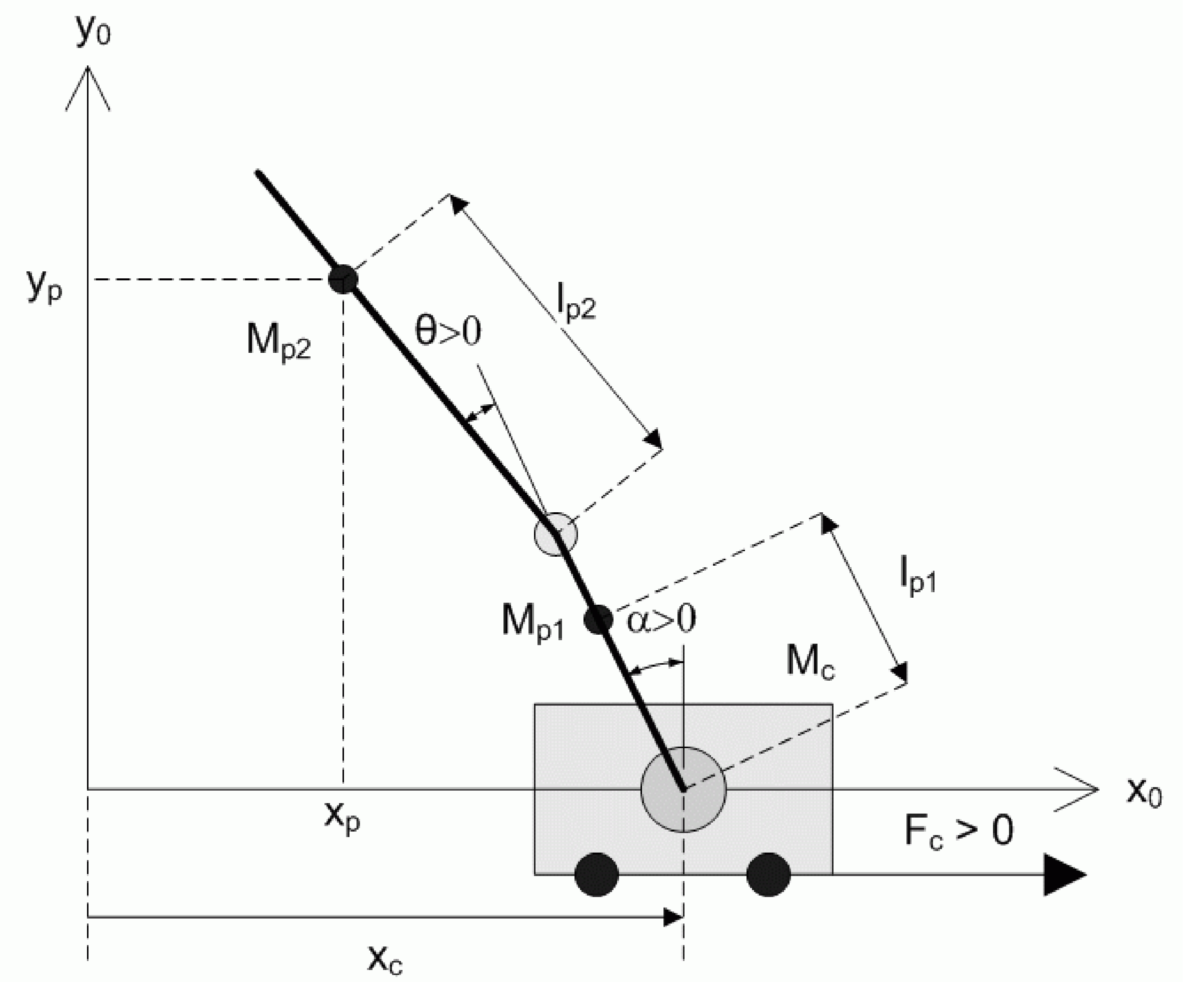


Figure 1 Free body diagram of DBIP system.

The model will not be summarized since the state-space matrices are too large to be printed in this document. It may be useful to view the model summary given in Reference [5] for the single-inverted pendulum system.

4.2. State-Feedback Controller Design

Like the single inverted pendulum device, the double-inverted pendulum is an unstable, non-minimum phase system that is controllable through feedback. The cart is to follow a reference position while balancing the double-link pendulum. That is, the cart position, x_c, tracks a desired setpoint position, $x_{c,d}$, and both upright pendulum angles, α and θ, are stabillized about 0. Assuming no actuator saturation, i.e. $I_m = u$, consider the state-feedback law

$$u = -K\zeta \qquad [9]$$

where

$$\zeta^T = \left[x_c - x_{c,d}, \alpha, v_x, v_\alpha, \int x_c - x_{c,d}\, dt \right] \qquad [10]$$

is the augmented error state. The cart velocity, v_x, is found using the high-pass filter

$$V_x(s) = \mathrm{D}_1(s)\, X_c(s), \qquad [11]$$

the angular rate of the pendulum 1, v_α, is calculated using

$$V_\alpha(s) = \mathrm{D}_2(s)\, \alpha(s) \qquad [12]$$

and the angular rate of the pendulum 2, v_α, is calculated using

$$V_\theta(s) = \mathrm{D}_2(s)\, \theta(s) \qquad [13]$$

where the high-gain observers $D_1(s)$ and $D_2(s)$ are defined in Reference [1]. The closed-loop system implemented is depicted in Figure 3.

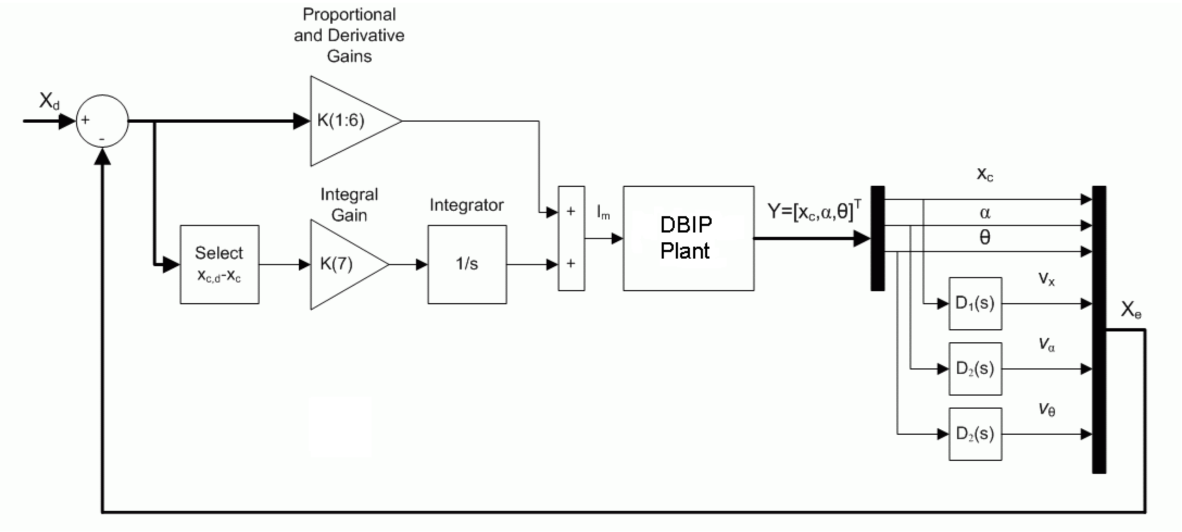


Figure 2 Closed-loop DBIP system.

Given the derived state-space matrices and using the Matlab *lqr* command with the weighting matrices

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \quad [14]$$

and

$$R = 0.1 \quad [15]$$

With the controller defined in [9], the ζ state converges towards zero. Thus the cart position approaches its setpoint, $x_c(t) \to x_{c,d}(t)$, while both pendulum angles are stabilized about the origin, i.e. $\alpha(t) \to 0$ and $\theta(t) \to 0$. The integration term improves the steady-state error of the cart position.

5. In-Lab Procedure

5.1. Experimental Setup Components

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.
- **Linear Motion Servo Plant:** Quanser IP01_02
- **Double Pendulum:** Quanser Double Inverted Pendulum,
- **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. System Wiring and Hardware Setup

5.2.1. System Wiring and Setup

Refer to Reference [1] for the setup information required to carry out this control laboratory. Reference [1] provides the specifications and a description of the main components composing the IP01-02 system. It also fully describes the wiring conventions and the default wiring procedure for the IP01_02 plant. More importantly, Reference [1] gives some safety operating guidelines and the start-up procedure to properly wire the system.

5.2.2. Double Pendulum Wiring and Setup

Follow the steps described below:

1. Mount the double-pendulum to the cart front pendulum shaft. Tighten the set-screw enough such that the pendulum T-Fitting is securely fastened to the metal shaft.

2. **Connect cable #7** from the double-pendulum encoder connector to the "Encoder Input #2" connector on the Terminal Board. This connection is done using the 5-pin-stereo-DIN to 5-pin-stereo-DIN encoder cable described in Reference [1]. It carries the signals required to measure the angle of the longer, bottom link of the double-pendulum assembly and its associated variable is θ. This connection is summarized in Table 2.

Cable #	*Cable Type*	*From*	*To*	*Function*
7	Encoder Cable: 5-pin-stereo-DIN to 5-pin-stereo-DIN	Encoder Connector on Double-Pendulum	Encoder #2 on the Terminal Board	Measures angle of longest link, θ.

Table 2 Wiring Summary

5.3. Controller Simulation

The Simulink model entitled *s_dbip_lqr_ip02* is shown in Figure 3 and is used to simulate the DBIP system. The cart setpoint position can be changed in the Setpoint Amplitude (m) block.

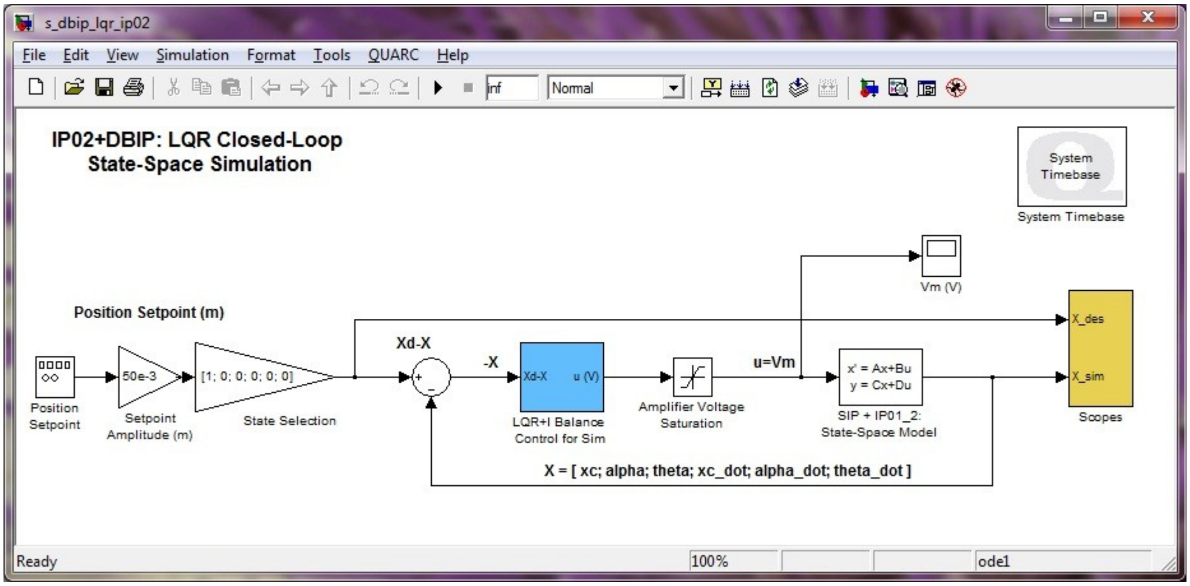


Figure 3: Simulink model used to simulate the IP02 + DBIP system.

5.3.1. Objectives

- Investigate the closed-loop performance of the controller using the nonlinear model of the DBIP system.
- Ensure the controller does not saturate the actuator before implementing it on the actual system.

5.3.2. Procedure

Follow these steps to simulate the controller:

1. Open Simulink model *s_dbip_lqr_ip02.mdl* shown in Figure 3 above.
2. Run the MATLAB script *setup_lab_ip01_2_dbip.m* to set the model parameters and the control gain *K*.
3. Click on the black arrow located in the toolbar (or click on Simulation-Start in the menu) to simulate the closed-loop system. As shown in Figure 4, the purple trace in

the xc (mm) scope is the simulated cart position and the yellow trace is the desired position. The position of the cart should track the commanded square position signal, $x_c(t) \rightarrow x_{c,d}(t)$, while balancing the pendulum. Thus the bottom link, shown in Figure 5, and the top link, shown in Figure 6, should both be stabilized about zero, $\alpha(t) \rightarrow 0$ and $\theta(t) \rightarrow 0$.

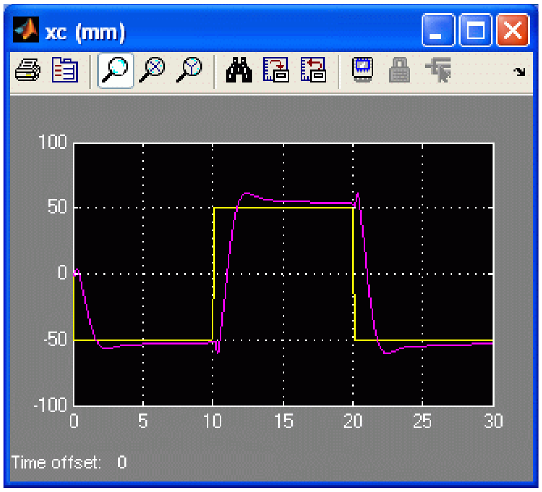


Figure 4 Simulated cart response.

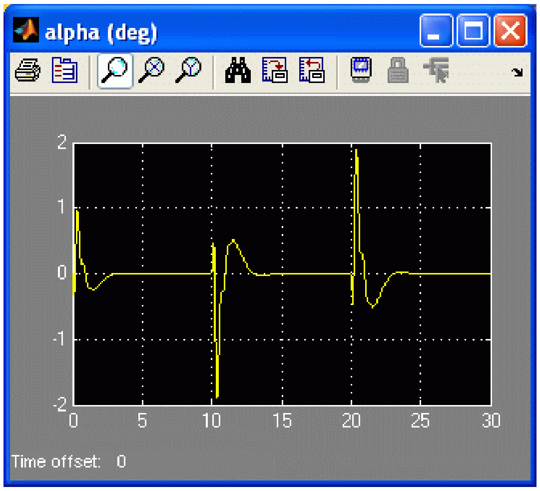


Figure 5 Simulated response link 1.

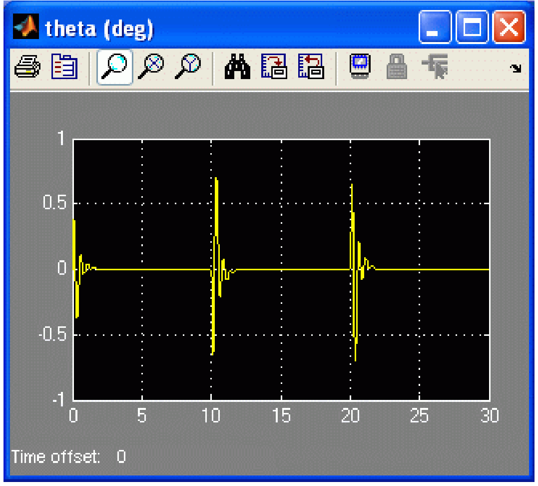


Figure 6 Simulated response of link 2.

4. Verify that the motor input current shown in the *Im (A)* scope is relatively smooth and not saturated, as depicted in Figure 7.

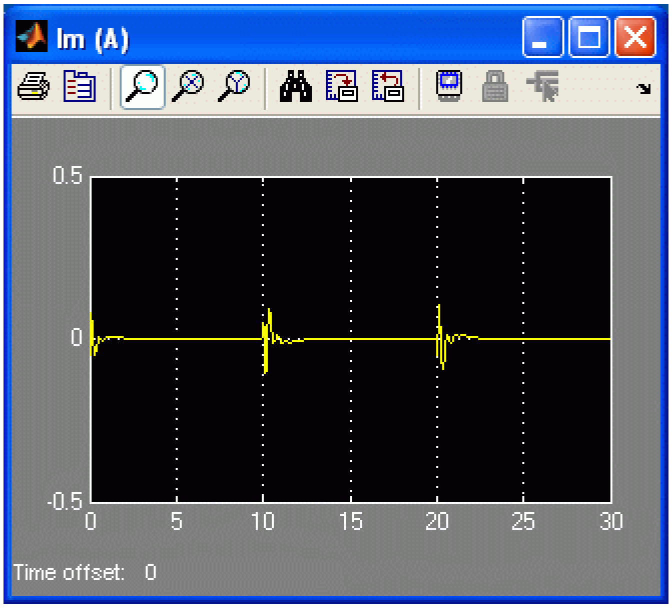
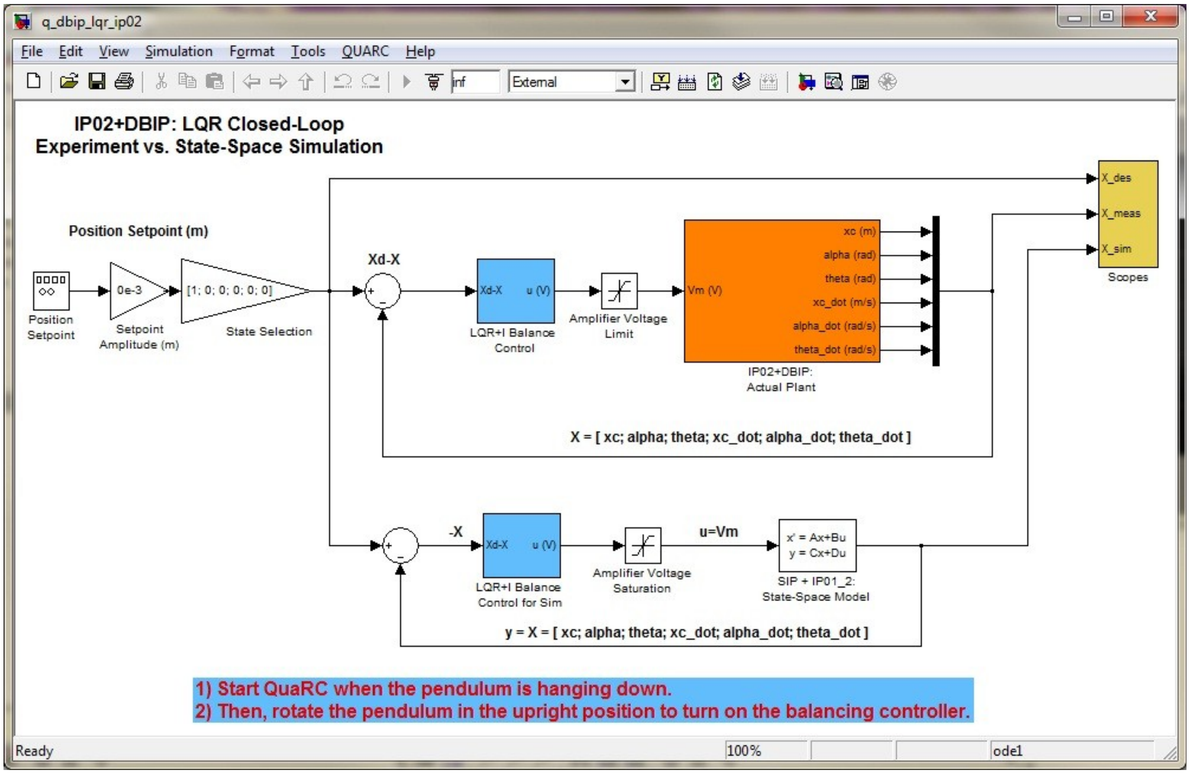


Figure 7 Simulated control input.

Alternatively, the user may also set the setpoint to zero and view the response of the system when subject to a set of initial conditions.

5.4. Controller Real-Time Implementation

The *q_dbip_lqr_ip02.mdl* Simulink model shown in Figure 8 implements the previously designed balance controller.



1) Start QuaRC when the pendulum is hanging down.
2) Then, rotate the pendulum in the upright position to turn on the balancing controller.

Figure 8: Simulink model that controls the DBIP system.

Similarly to the cart safety limits found in the IP02 + DBIP subsystem a watchdog is placed in this subsystem that stops the real-time controller when α goes beyond the ALPHA_MIN and ALPHA_MAX limits, given that ALPHA_LIM_ENABLE = 1, and when θ goes exceeds the THETA_MIN and THETA_MAX limits, given that THETA_LIM_ENABLE=1. The limits are both specified in the *setup_lab_ip01_2_dbip.m* file.

The pendulum begins in the gantry position. The control design is based on a model that defines $\alpha = 0$ when the bottom pendulum is in the upright position and $\theta = 0$ when the top pendulum is in the upright position relative to the bottom pendulum. Therefore given that the bottom pendulum begins in the gantry position an offset of $-\pi$ rad (-180°) is applied to angle α. The controller is activated when both pendula are rotated until their angles are within the tolerance specified in the box labelled "Desired Initial Angle (deg)". By default the tolerance is set to 0.5°.

5.4.1. Objectives

- Implement the previously designed controller with QUARC to balance the Double Inverted Pendulum (DBIP) system on the linear track.

5.4.2. Experimental Procedure

Follow the steps described below to implement the designed controller in real-time and observe its effect on the actual IP02 + DBIP plant:

1. Open Simulink model *q_dbip_lqr_ip02.mdl* shown in Figure 6 above.
2. Execute the design file *setup_lab_ip01_2_dbip.m* to setup the workspace before compiling the diagram and running it in real-time with QUARC. This file sets the state-space model of the IP02+DBIP system. It also calculates the feedback gain vector *K* and enters it in the MATLAB workspace. Further, various parameters used in the IP02+DBIP subsystem such as the filter cutoff frequencies, cart position safety limits and amplifier gain must also be loaded prior to compilation.
3. 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 controller on the actual system in real-time.
4. If not already opened, double-click on the sinks *xc (mm), alpha (deg)*, and *theta (deg)* found in the Scopes folder. The *xc (mm)* scope displays the cart setpoint, the measured cart position, and the simulated cart position generated by the state-space model. The *alpha (deg)* scope displays the measured and simulated angle of the bottom pendulum and the *theta (deg)* scope displays the measured and simulated angle of the top pendulum.
5. **Ensure the real-time code is ran safely by manually moving the IP02 cart to the middle of the track (i.e. home position) so it is free to move on both sides.**
6. Verify that the suspended pendulum is not moving before starting the controller.
7. Start your real-time controller by clicking on QUARC | Start from the Simulink model menu bar.
8. Slowly rotate the entire double pendulum assembly **counter-clockwise** until it is vertical. When both the bottom and the top pendulum are within 0.5 degrees of their vertical position (pointing upwards), the motion control loop will automatically start and balance the double pendulum.
9. Release the pendulum as soon as the controller is activated. Any obstructions are not compensated for and may lead to instability.
10. Figure 9 depicts the measured and simulated responses of the cart position, the

bottom pendulum angle ≃, and the top pendulum angle ┐ . Since the cart setpoint is zero, the simulation traces all remain at zero. Figure 10 displays the corresponding cart motor current used to balance the double-pendulum.

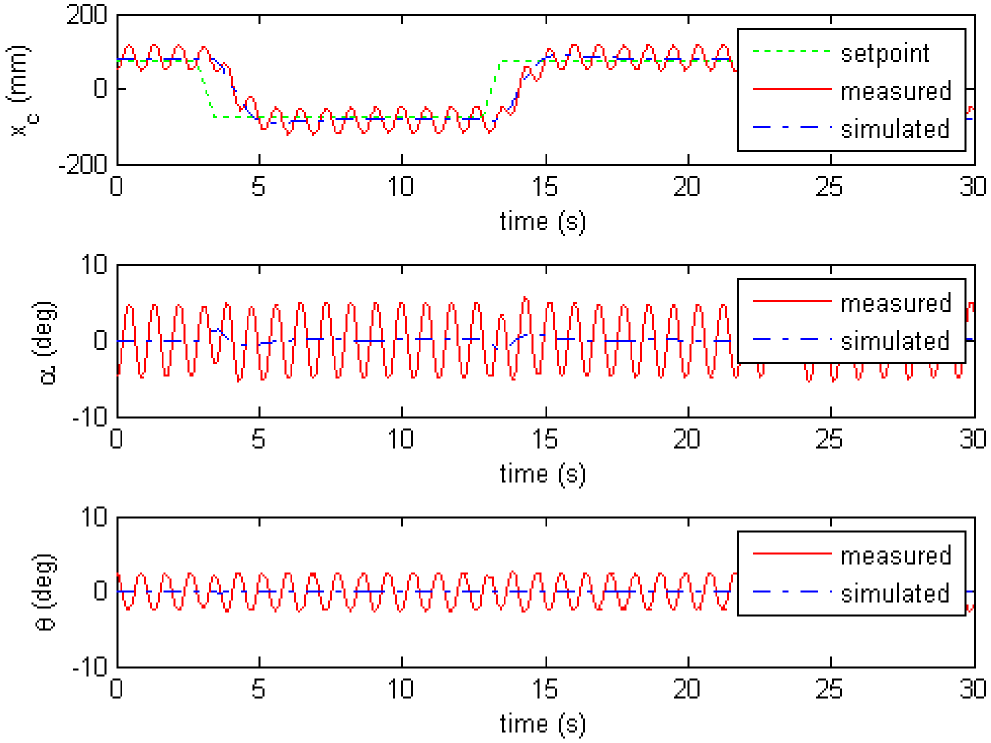


Figure 9 Measured and simulated DBIP closed-loop response.

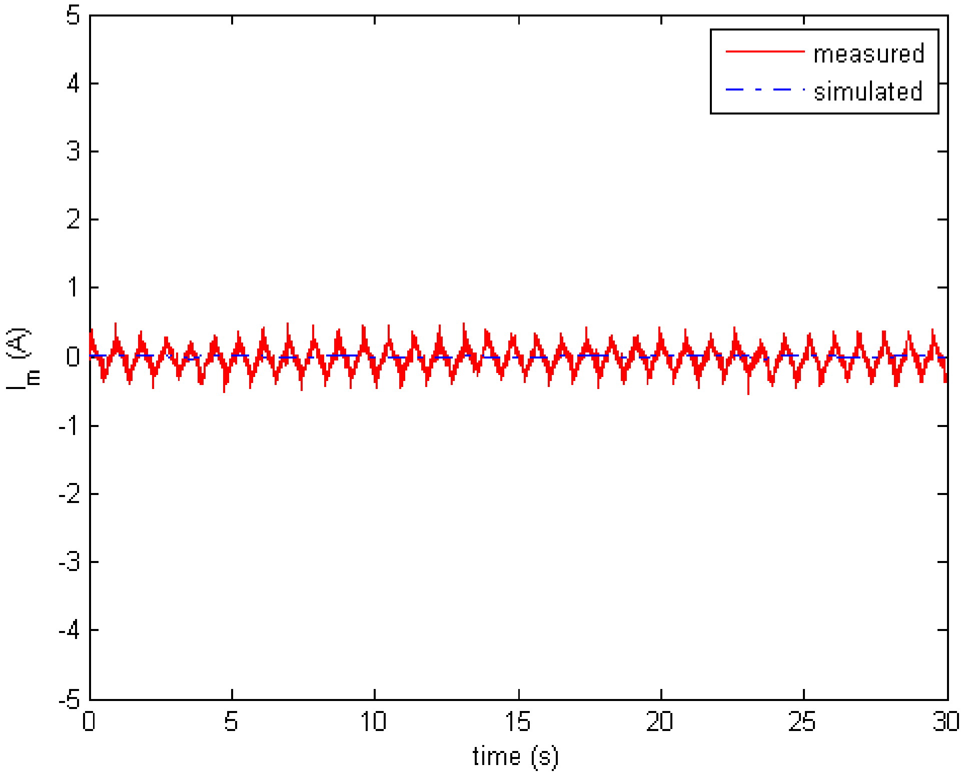


Figure 10 Measured and simulated control input.

11. Click on the *Stop* button in the Simulink model tool bar to stop running the real-time controller. Turn off the power amplifier.

6. References

[1] IP01_2 User Manual.
[2] DAQ User Manual.
[3] Power amplifier User Manual
[4] QUARC User Manual (type `doc QUARC` in the MATLAB Command Prompt).
[5] HFLC Single Inverted Pendulum Laboratory Manual.