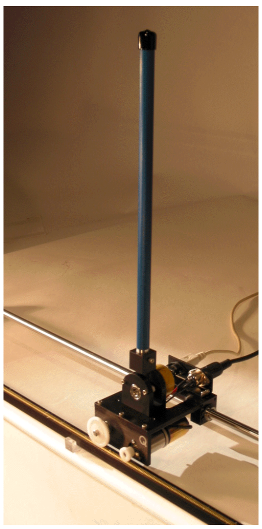
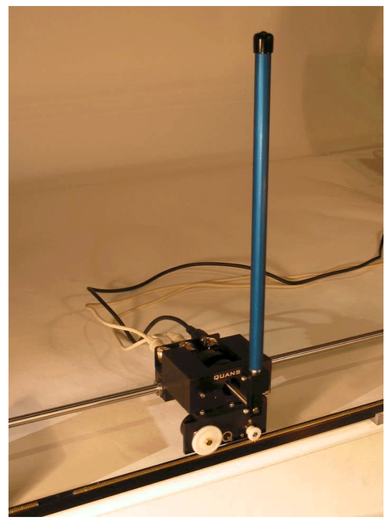
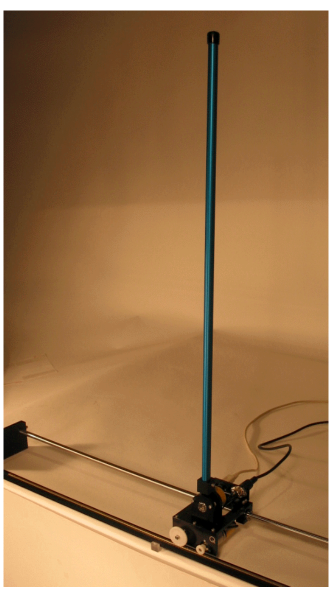
*Linear Motion Servo Plants: IP01 or IP02*

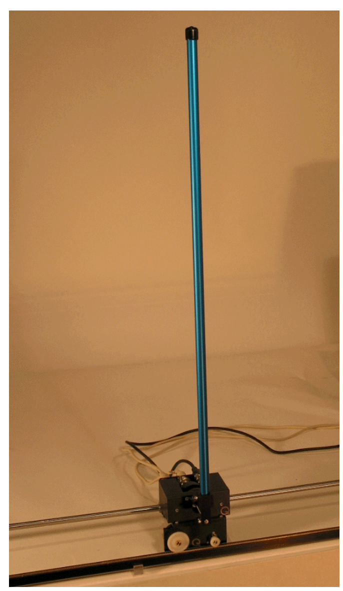
# Linear Experiment #5: LQR Control

## *Single Inverted Pendulum (SIP)*









## Student Handout

## Table of Contents

# 1. Objectives

The Single Inverted Pendulum (SIP) experiment is a "classroom example". It can be seen as balancing a broomstick on the tip of one's finger. The difference being that the broomstick is in a three-dimensional space while the pendulum is in a linear plane. In this laboratory session, the inverted pendulum is mounted on a linear cart, either the IP01 or IP02. During the course of this experiment, you will become familiar with the design and tuning principles of a Linear Quadratic Regulator (LQR). The challenge of the present laboratory is to keep a single inverted pendulum balanced and track the linear cart to a commanded position. In practice, it is interesting to point out that similar dynamics and control problem apply to rudder roll stabilization of ships.

At the end of the session, you should know the following:

- How to mathematically model the SIP mounted on the IP01 or IP02 linear servo plant, using, for example, Lagrangian mechanics or force analysis on free body diagrams.
- How to linearize the obtained non-linear equations of motion about the quiescent point of operation.
- How to obtain a state-space representation of the open-loop system.
- How to design, simulate, and tune a LQR-based state-feedback controller satisfying the closed-loop system's desired design specifications.
- How to implement your LQ Regulator in real-time and evaluate its actual performance.
- How to tune on-line and in real-time your LQR so that the actual inverted-pendulum-linear-cart system meets the controller design requirements.
- How to observe and investigate the disturbance response of the stabilized inverted-pendulum-linear-cart system, in response to a tap to the pendulum.

# 2. Prerequisites

To successfully carry out this laboratory, the prerequisites are:

i) To be familiar with your IP01 or IP02 main components (e.g. actuator, sensors), your power amplifier (e.g. UPM), and your data acquisition card (e.g. MultiQ), as described in References [1], [2], [3], and [4].

ii) To have successfully completed the pre-laboratory described in Reference [1]. Students are therefore expected to be familiar in using WinCon to control and monitor the plant in real-time and in designing their controller through Simulink.

iii) To be familiar with the complete wiring of your IP01 or IP02 servo plant, as per dictated in Reference [2] and carried out in pre-laboratory [1].

iv) To be familiar with LQRs' design theory and working principles.

# 3. References

[1] *IP01 and IP02 - Linear Experiment #0: Integration with WinCon – Student Handout*.
[2] *IP01 and IP02 User Manual.*
[3] *IP01 and IP02 - Single Inverted Pendulum User Manual.*
[4] *MultiQ User Manual*.
[5] *Universal Power Module User Manual.*
[6] *WinCon User Manual*.
[7] *IP01 and IP02 - Linear Experiment #1: PV Position Control – Student Handout.*

# 4. Experimental Setup

## 4.1. Main Components

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

- **Power Module:** Quanser UPM 1503 / 2405, or equivalent.
- **Data Acquisition Board:** Quanser MultiQ PCI / MQ3, or equivalent.
- **Linear Motion Servo Plant:** Quanser IP01, as shown in Figures 1 and 3, or IP02, as represented in Figures 2 and 4, below.
- **Single Pendulum:** Quanser 12-inch Single Pendulum, seen in Figures 1 and 2, and/or 24-inch Single Pendulum, as shown in Figures 3 and 4, below.
- **Real-Time Control Software:** The WinCon-Simulink-RTX configuration, as detailed in Reference [6], or equivalent.

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

## 4.2. Wiring

To wire up the system, please follow the default wiring procedure for your IP01 or IP02 as fully described in Reference [2]. When you are confident with your connections, you can power up the UPM.

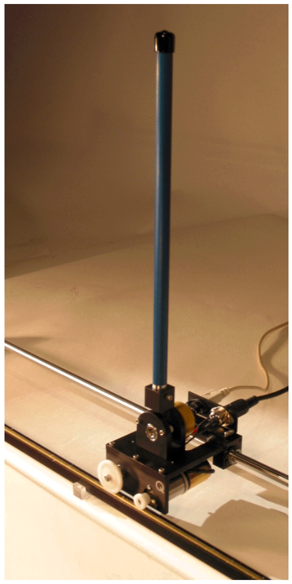


Figure 1 Medium SIP on the IP01

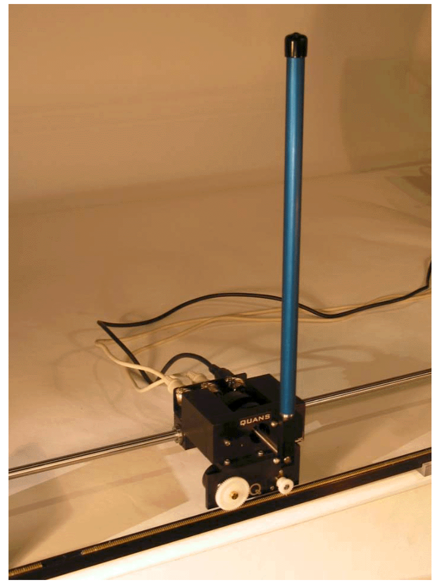


Figure 2 Medium SIP on the IP02

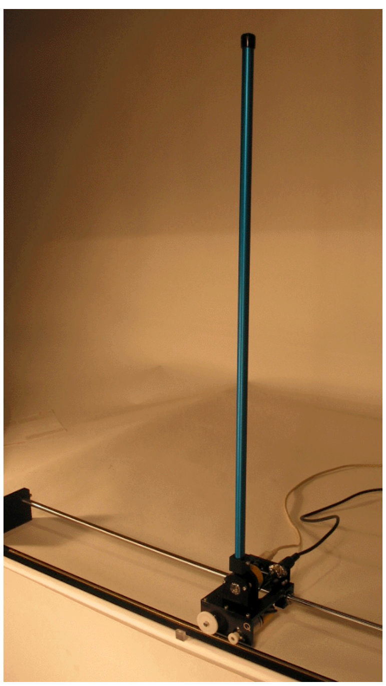


Figure 3 Long SIP on the IP01

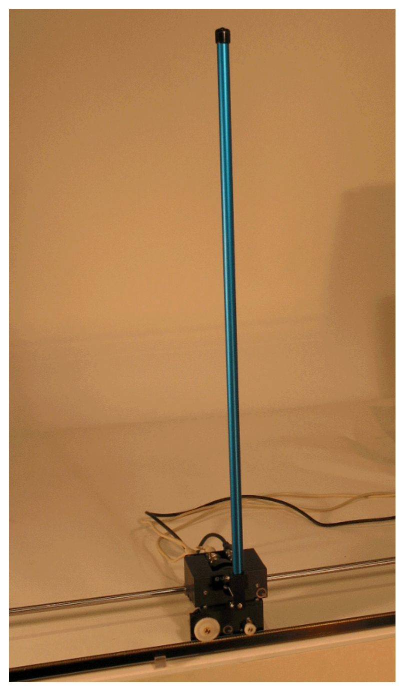


Figure 4 Long SIP on the IP02

## 5. Controller Design Specifications

In the present laboratory (i.e. the pre-lab and in-lab sessions), you will design and implement a control strategy based on the Linear Quadratic Regulator (LQR) scheme. As a primary objective, the obtained optimal feedback gain vector, *K*, should allow you to keep your single inverted pendulum balanced. At the same time, your IP01 or IP02 linear cart will be asked to track a desired (square wave) position setpoint. The corresponding control effort should also be looked at and minimized.

Please refer to your in-class notes, as needed, regarding the LQR design theory and the corresponding implementation aspects of it. Generally speaking, the purpose of optimal control is to allow for best trade-off between performance and cost of control.

Tune the LQR controlling the inverted-pendulum-and-linear-cart system in order to satisfy the following design performance requirements:

1. Regulate the pendulum angle around its upright position and never exceed a ±1-degree-deflection from it, if an IP02 is used, or a ±1.5-degree-deflection, if you have an IP01, i.e.:

   $|\alpha| \leq 1.0\ [deg]$ with an IP02.

   or:

   $|\alpha| \leq 1.5\ [deg]$ with an IP01.

2. Have a maximum rise time, $t_r$, on the cart position response less than 1.5 second, i.e.:

   $t_r \leq 1.5\ [s]$

3. Minimize the control effort produced, which is proportional to the motor input voltage $V_m$. The power amplifier (e.g. UPM) should not go into saturation in any case.

The previous specifications are given in response to a ±20-to-30 mm square wave cart position setpoint. They also apply to any type of pendulum (i.e. medium or long).

As a remark, it can be seen that the previous design requirements bear on the system's two outputs: $x_c$ and α. Therefore our inverted-pendulum-linear-cart system consists of two outputs, for one input.

# 6. Pre-Lab Assignments

## 6.1. Assignment #1: Non-Linear Equations Of Motion (EOM)

### 6.1.1. System Representation and Notations

A schematic of the Single Inverted Pendulum (SIP) mounted on an IP01 or IP02 linear cart is represented in Figure 5. The SIP-plus-IP01-or-IP02 system's nomenclature is provided in Appendix A. As illustrated in Figure 5, the positive sense of rotation is defined to be counter-clockwise (CCW), when facing the linear cart. Also, the zero angle, modulus 2π, (i.e. $\alpha = 0$ rad [2π]) corresponds to the inverted pendulum perfectly vertical and pointing upward. Furthermore, the positive direction of linear displacement is to the right when facing the cart, as indicated by the global Cartesian frame of coordinates represented in Figure 5.

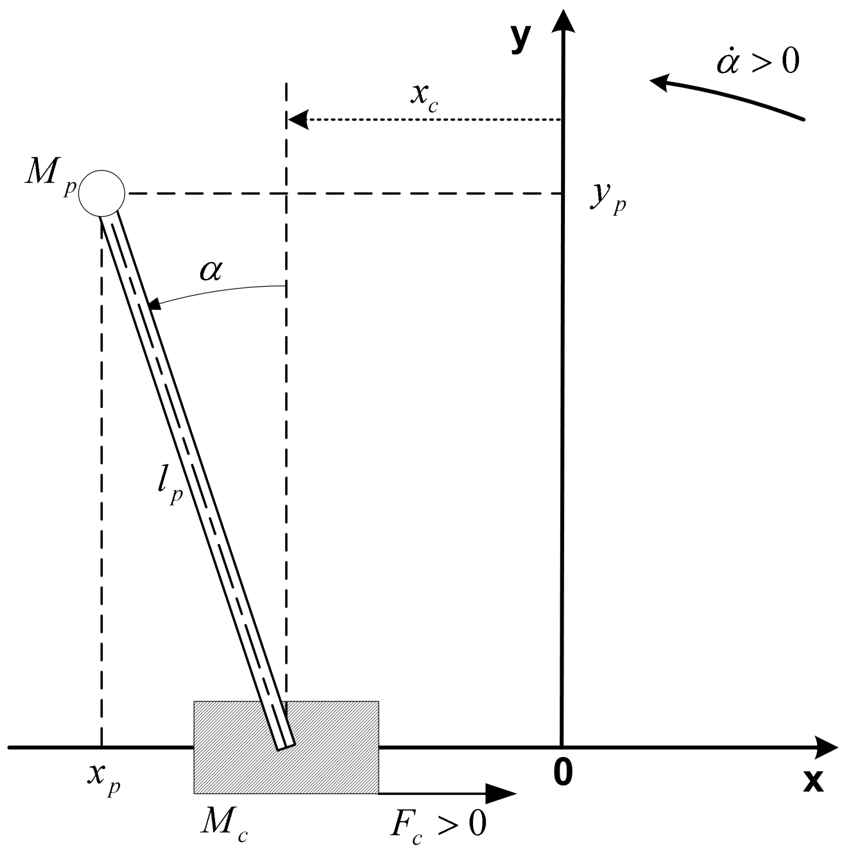



Figure 5 Schematic of the SIP mounted on an IP01 or IP02 Servo Plant

### 6.1.2. Assignment #1: Determination of the System's Equations Of Motion

The determination of the SIP-plus-IP01-or-IP02 system's equations of motion is derived in Appendix B. If Appendix B has not been supplied with this handout, derive the system's equations of motion following the system's schematic and notations previously defined and illustrated in Figure 5. Also, put the resulting EOM under the following format:

$$\frac{\partial^2}{\partial t^2} x_c = \left(\frac{\partial^2}{\partial t^2} x_c\right)(x_c, \alpha, F_c) \qquad \text{and} \qquad \frac{\partial^2}{\partial t^2} \alpha = \left(\frac{\partial^2}{\partial t^2} \alpha\right)(x_c, \alpha, F_c) \qquad [1]$$

***Hint #1:***
The mass of the single inverted pendulum is assumed concentrated at its Centre Of Gravity (COG).

***Hint #2:***
You can use the method of your choice to model the system's dynamics. However, the modelling developed in Appendix B uses the energy-based Lagrangian approach. In this case, since the system has two Degrees-Of-Freedom (DOF), there should be two Lagrangian coordinates (a.k.a. generalized coordinates). The chosen two coordinates are namely: $x_c$ and $\alpha$. Also, the input to the system is defined to be $F_c$, the linear force applied by the motorized cart.

## 6.2. Assignment #2: EOM Linearization and State-Space Representation

In order to design and implement a LQ Regulator for our system, a state-space representation of that system needs to be derived. Moreover, it is reminded that state-space matrices, by definition, represent a set linear differential equations that describe the system's dynamics. Therefore, the EOM found in Assignment #1 should be linearized around a quiescent point of operation. In the case of the inverted pendulum, the operating range corresponds to small departure angles, $\alpha$, from the upright vertical position. Answer the following questions:

1. Using the small angle approximation, linearize the two EOM found in Assignment #1.
   ***Hint:***
   For small angles $\alpha$, you can use the second-order generalized series expansions, as shown in the following:

   $$\cos(\alpha) = 1 + O(\alpha^2) \qquad \text{and} \qquad \sin(\alpha) = \alpha + O(\alpha^2) \qquad [2]$$

2. Determine from the previously obtained system's linear equations of motion, the state-space representation of our SIP-plus-IP01-or-IP02 system. That is to say, determine the state-space matrices *A* and *B* verifying the following relationship:

$$\frac{\partial}{\partial t} X = A\, X + B\, U \qquad [3]$$

where *X* is the system's state vector. In practice, *X* is often chosen to include the generalized coordinates as well as their first-order time derivatives. In our case, *X* is defined such that its transpose is as follows:

$$X^T = \left[ x_c(t),\ \alpha(t),\ \frac{d}{dt} x_c(t),\ \frac{d}{dt} \alpha(t) \right] \qquad [4]$$

Also in Equation [3], the input *U* is set in a first time to be $F_c$, the linear cart driving force. Thus we have:

$$U = F_c \qquad [5]$$

3. From the system's state-space representation previously found, evaluate the matrices *A* and *B* in case the system's input *U* is equal to the cart's DC motor voltage, as expressed below:

$$U = V_m \qquad [6]$$

To convert the previously found force equation state-space representation into voltage input, you can use the following hints:

***Hint #1:***

In order to transform the previous matrices *A* and *B*, it is reminded that the driving force, $F_c$, generated by the DC motor and acting on the cart through the motor pinion has already been determined in previous laboratories. As shown for example in Equation [B.9] of Reference [7], it can be expressed as:

$$F_c = -\frac{\eta_g K_g^2 \eta_m K_t K_m \left( \frac{d}{dt} x_c(t) \right)}{R_m r_{mp}^2} + \frac{\eta_g K_g \eta_m K_t V_m}{R_m r_{mp}} \qquad [7]$$

***Hint #2:***

The single inverted pendulum's moment of inertia about its centre of gravity is characterized by:

$$I_p = \frac{1}{12} M_p L_p^2 \qquad [8]$$

***Hint #3:***

Evaluate matrices *A* and *B* by using the model parameter values given in References [2] and [3]. Ask your laboratory instructor what system configuration is going to be set up in your in-lab session. In case no additional information is provided, assume that your system is made of a long single pendulum mounted on an IP02 cart with no additional weight.

4. Calculate the open-loop poles from the system's state-space representation, as previously evaluated in question 3. What can you infer regarding the system's dynamic behaviour? Is it stable? Do you see the need for a closed-loop controller in order to maintain the pendulum balanced in the inverted position? Explain.
   ***Hint:***
   The characteristic equation of the open-loop system can be expressed as shown below:
   $$\det(s\,I - A) = 0 \qquad [9]$$
   where *det()* is the determinant function, *s* is the Laplace operator, and *I* the identity matrix. Therefore, the system's open-loop poles can be seen as the eigenvalues of the state-space matrix *A*.

# 7. In-Lab Procedure

## 7.1. Experimental Setup

Even if you don't configure the experimental setup entirely yourself, you should be at least completely familiar with it and understand it. If in doubt, refer to References [1], [2], [3], [4], [5], and/or [6].

### 7.1.1. Check Wiring and Connections

The first task upon entering the lab is to ensure that the complete system is wired as fully described in Reference [2]. You should have become familiar with the complete wiring and connections of your IP01 or IP02 system during the preparatory session described in Reference [1]. If you are still unsure of the wiring, please ask for assistance from the Teaching Assistant assigned to the lab. When you are confident with your connections, you can power up the UPM. You are now ready to begin the lab.

### 7.1.2. IP01 or IP02 Configuration

In case you use the IP02 for this laboratory, this experiment is designed for an IP02 cart without the extra weight on it.
However, once a working controller has been tested, the additional mass can be mounted on top the cart in order to see its effect on the response of the system. As an extension to the present lab, the designed controller could be modified in order to account for the additional weight.

### 7.1.3. Single Inverted Pendulum Configuration

This laboratory uses the long single pendulum as a default configuration. However, the same pre-lab and in-lab sessions could be applied, as long as they are consistent, to a different configuration using, for example, the medium single pendulum.
Likewise, once a working controller has been developed and tested for one pendulum, another type of pendulum can be mounted on top the cart in order to observe the resulting system response with regard to, for example, the controller robustness to modelling errors. As an extension to the present lab, the previously designed controller could then be modified in order to account for the new pendulum, of different characteristics.

## 7.2. Assessment of the State-Space Model Operating Range

It is obvious that linearized models, such as the inverted pendulum system state-space model, are only approximate models. Therefore, they should be treated as such and used with appropriate caution, that is to say within the valid operating range and/or conditions. However, linearized models generally give valuable insights on the system's dynamics and usually allow for the development of fairly straightforward control strategies.

This section attempts to assess the operating range of the system's state-space representation derived in the pre-lab session. It does so by simulating and comparing the inverted pendulum response to an initial non-zero angle (i.e. free fall and free swing) obtained from both state-space model and original nonlinear equations of motion.

### 7.2.1. Simulation Diagram

Opening the file ***s_sip_freefall_ss_vs_eom.mdl*** should display a diagram similar to the one shown in Figure 6, below. It implements, in parallel, the two inverted pendulum models previously discussed. Using this diagram allows to compare the two inverted pendulum responses, to a small non-zero initial angle $\alpha_0$, obtained from the two models.

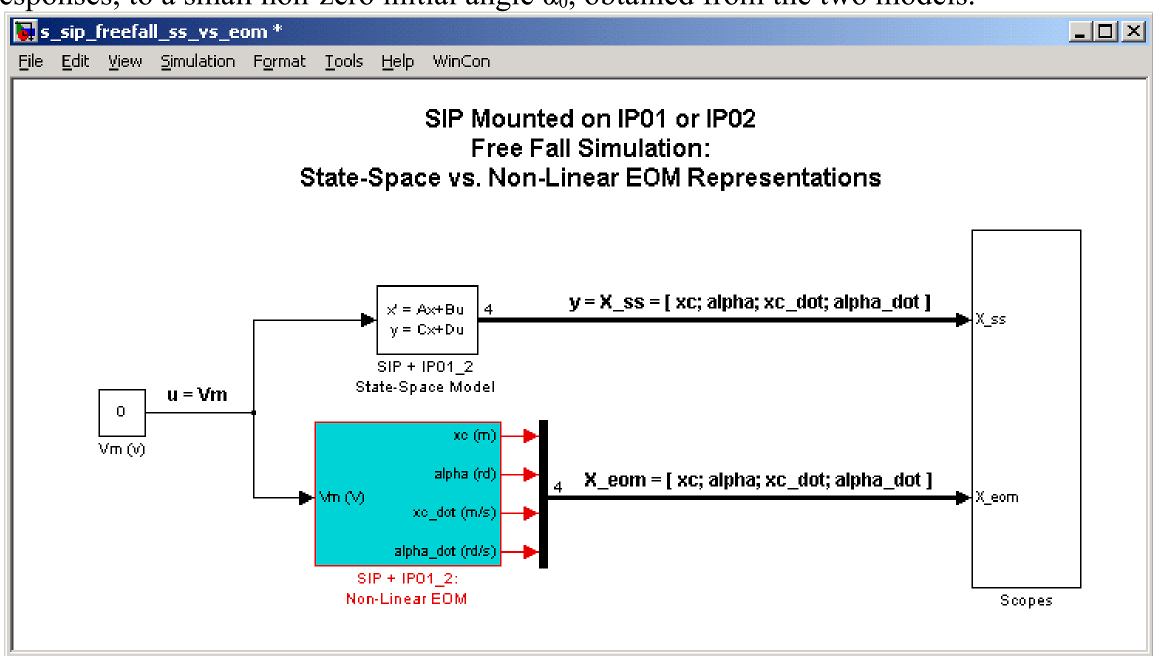


Figure 6 SIP Free Fall Simulation Diagram

### 7.2.2. State-Space Model Implementation

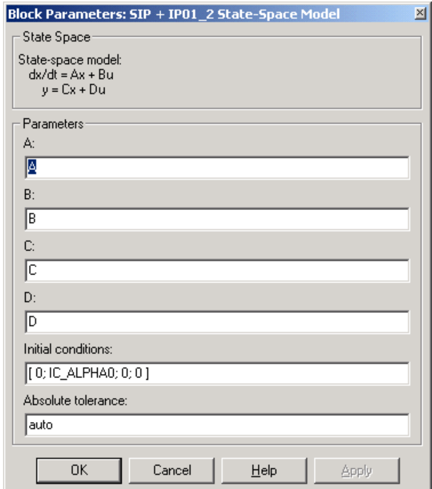


Figure 7 State-Space Block Parameters

Opening the state-space block named *SIP + IP01_2 State-Space Model* in the *s_sip_freefall_ss_vs_eom.mdl* file should show a dialog box similar to that displayed in Figure 7. The model state-space matrices *A* and *B* should correspond to the matrices evaluated in the pre-lab Assignment #2, question 3. It should also be noted that the initial condition on α, $\alpha_0$, is setup inside the *Initial conditions:* input field. That initial angle is initialized in the Matlab workspace as the variable *IC_ALPHA0*.

### 7.2.3. Simulink Representation of the Two EOM

After obtaining of the system's two EOM in Assignment #1, they can be represented by a series of block diagrams, as illustrated in Figures 8, 9, and 10, belowbelow. Opening the subsystem block named *SIP + IP01_2: Non-Linear EOM* in the file *s_sip_freefall_ss_vs_eom.mdl* should show something similar to Figure 8. This mostly corresponds to the familiar IP01 or IP02 open-loop transfer function representation, as previously discussed in, for example, Reference [7].

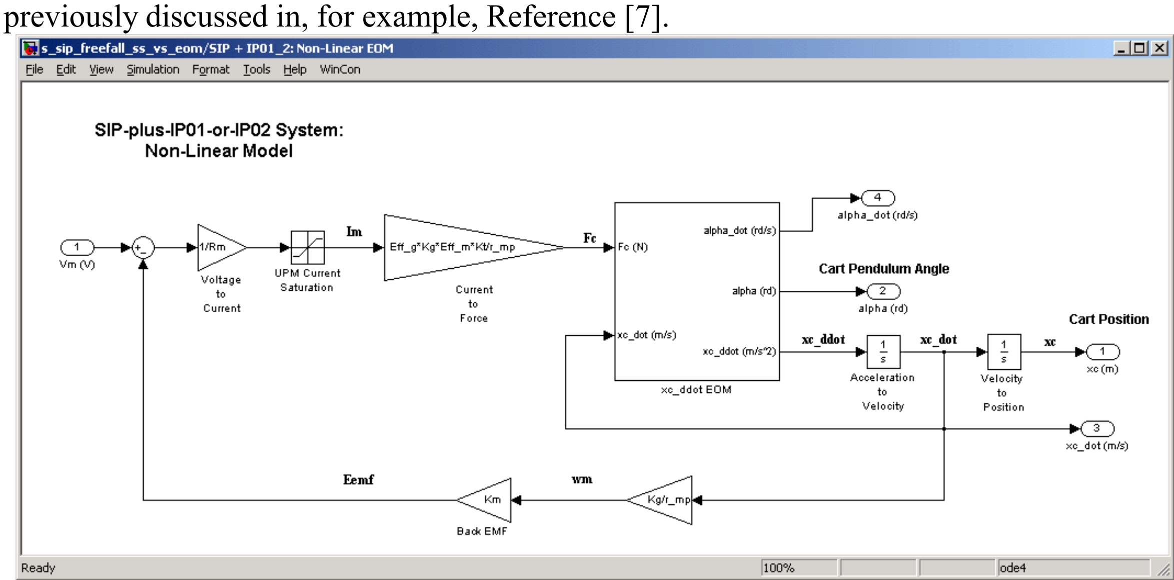


Figure 8 Simulink Subsystem Representing the IP01 or IP02 Model

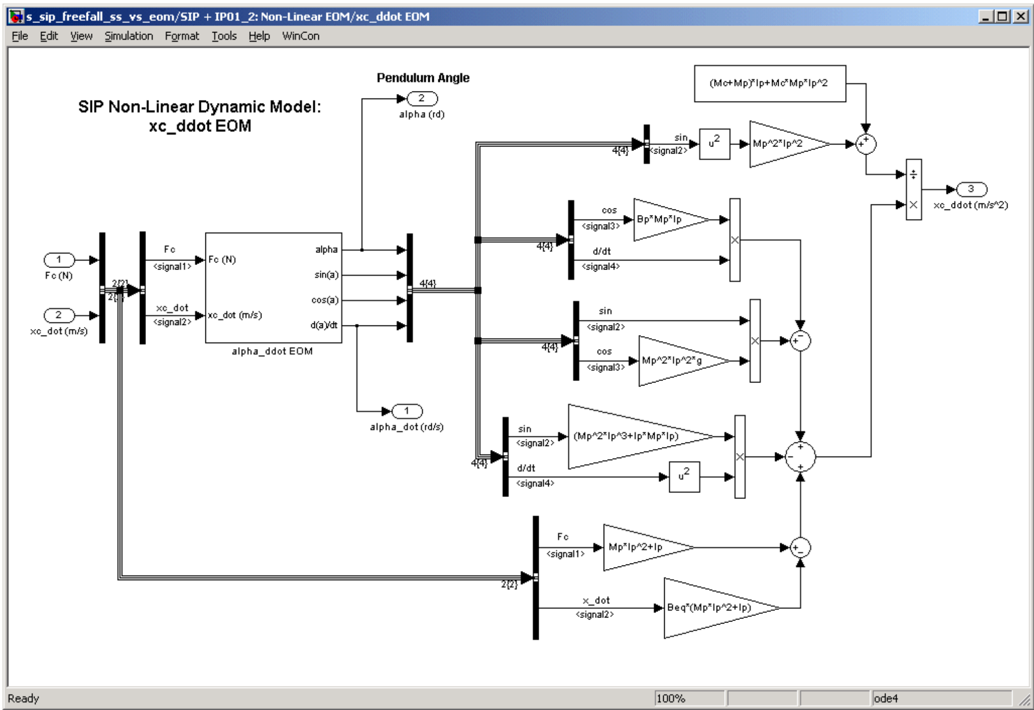



Figure 9 Simulink Subsystem Representing the First EOM

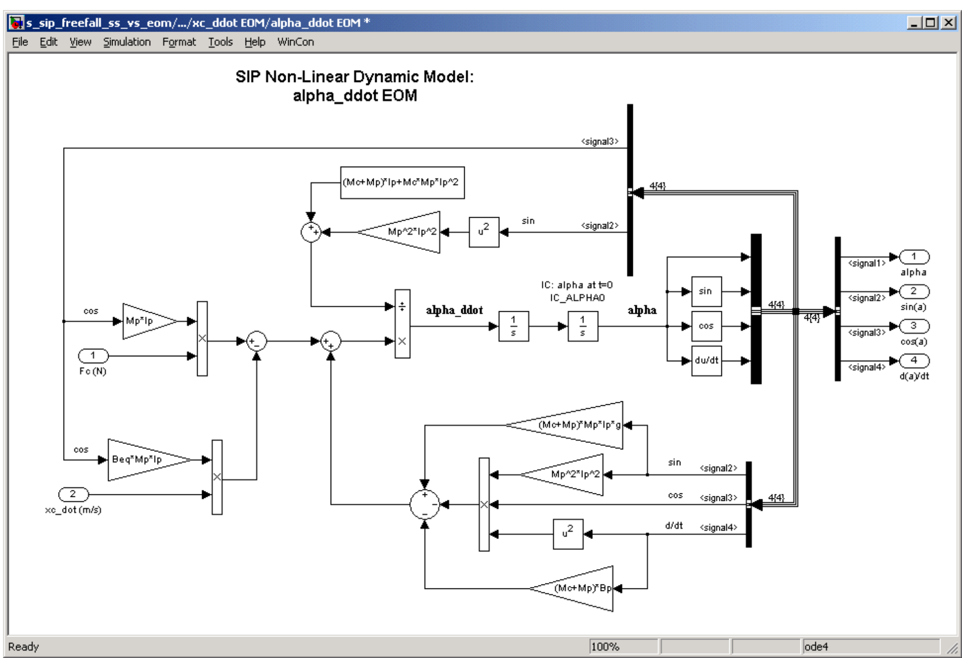



Figure 10 Simulink Subsystem Representing the Second EOM

Figure 9, above, represents the first EOM, as expressed in Equation [B.17] and of the following form:

$$\frac{\partial^2}{\partial t^2} x_c = \left(\frac{\partial^2}{\partial t^2} x_c\right)(x_c, \alpha, F_c) \qquad [10]$$

Likewise, Figure 10, above, represents the second EOM, as expressed in Equation [B.18] and of the following form:

$$\frac{\partial^2}{\partial t^2} \alpha = \left(\frac{\partial^2}{\partial t^2} \alpha\right)(x_c, \alpha, F_c) \qquad [11]$$

Of interest in Figure 10, it should be noted that the initial condition on $\alpha$, $\alpha_0$, is contained inside the Simulink integration block located between the time derivative of $\alpha$ and $\alpha$. That initial angle is referred in the Matlab workspace as the variable *IC_ALPHA0*.

By carrying out block diagram reduction, you can check that the two EOM that you determined in Assignment #1 correspond to the Simulink representations shown in Figures 9 and 10.

As a remark, an alternative and replacement to Figures 9 and 10, above, could have been to build two Matlab S-functions corresponding to the system's EOM, as expressed in Equations [B.17] and [B.18].

## 7.2.4. Simulation of the Inverted Pendulum Free Fall

### 7.2.4.1. Objectives

- To simulate with Simulink the two inverted pendulum open-loop models previously discussed, namely the state-space linearized model and the two original EOM as expressed in Equations [B.17] and [B.18].
- To assess the operating range of the linearized, i.e. state-space, model. Therefore, assert the validity conditions of the linear approximation.

### 7.2.4.2. Experimental Procedure

Please follow the steps described below:

Step1. If you have not done so yet, you can start-up Matlab now and open the Simulink diagram titled *s_sip_freefall_ss_vs_eom.mdl*.

Step2. Before beginning the simulation, you must run the Matlab script called *setup_lab_ip01_2_sip.m*. This file initializes all the SIP and IP01 or IP02 model parameters and user-defined configuration variables needed and used by the Simulink diagram.

Step3. Initialize at the Matlab prompt *IC_ALPHA0* to be equal in radians to +0.1 degree. Set the *Stop Time:* in the Simulink *Simulation parameters...* (Ctrl+E) to 3 seconds. You can now *Start* (Ctrl+T) the simulation of your diagram.

Step4. After the simulation run, open the two Scopes titled *Cart Position (mm)* and *Pend Angle (deg)*. What do you observe? Is there a good match between the state-space and the EOM responses? Over the first second of the simulation? Over the 3-second simulation range?
***Hint:*** In reality, the pendulum actual behaviour would first be like a free fall about its pivot then followed by a free swing around the "hanging down" equilibrium position.

Step5. Draw your conclusions and try to infer the state-space model valid operating range.

## 7.3. Simulation and Design of a Linear Quadratic Regulator (LQR)

### 7.3.1. Objectives

- To use the obtained SIP-plus-IP01-or-IP02 state-space representation to design the following kind of state-feedback controller: Linear Quadratic Regulator (LQR).
- To tune on-the-fly and iteratively the LQ Regulator by simulating the closed-loop system in real-time with WinCon.
- To infer and comprehend the basic principles at work during LQR tuning.

***Note:***
Please refer to your in-class notes regarding the LQR design theory and associated working principles.

### 7.3.2. Experimental Procedure

Please follow the steps described below:

Step 1. Before you begin, you must run the Matlab script called *setup_lab_ip01_2_sip.m*. However, ensure beforehand that the *CONTROLLER_TYPE* flag is set to *'MANUAL'*. This mode initializes, before starting on-line the tuning procedure, the optimal gain vector *K* to zero, i.e. *[0,0,0,0]*. The *setup_lab_ip01_2_sip.m* file (re-)initializes all the SIP and IP01 or IP02 model parameters and user-defined configuration variables needed and used by the Simulink diagrams. Lastly, it also calculates the state-space matrices, *A* and *B*, corresponding to the SIP-plus-IP01-or-IP02 system configuration that you defined. Check that the *A* and *B* matrices thus set in the Matlab workspace correspond to the ones that you evaluated in Assignment #2, question 3. It is reminded that the LQR design is based of the plant's state-space (i.e. linear) model. Finally, you should also check that the variable *IC_ALPHA0* has been set back to zero in the Matlab workspace (from its value during the previous simulation).

Step 2. Open the Simulink diagram titled *s_sip_lqr.mdl*. You should obtain a diagram similar to the one shown in Figure 11, below. Take some time to familiarize yourself with this model: it will be used for the LQR tuning simulation. For a plant simulation valid over the full angular range, the SIP-plus-IP01-or-IP02 model is implemented without linear approximation through its equations of motion [B.17] and [B.18], as previously discussed. This model is represented in *s_sip_lqr.mdl* by the subsystem block titled *SIP + IP01_2: Non-Linear EOM*. You should also check that the signal generator block properties are properly set to output a square wave signal, of amplitude 1 and of frequency 0.2 Hz. As a remark, it should be noted that the reference input (a.k.a. setpoint) should be small enough so that our system remains in the region where our linearization is valid. It can also be noticed in *s_sip_lqr.mdl* that the setpoint needs to be scaled in order to accommodate for the feedback vector, located in the feedback loop. By definition, it is reminded that the LQR feedback vector, named *K*, has four elements, corresponding to the four system's states defined in Equation [4].

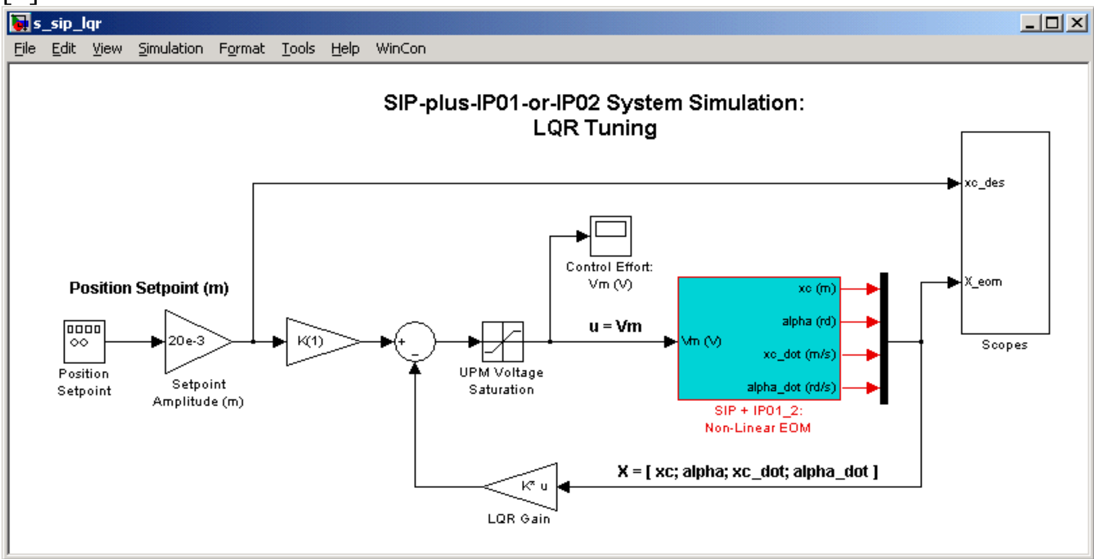



Figure 11 LQR Tuning Simulation Diagram

Step 3. You are now ready to build the real-time code corresponding to your diagram, by using the *WinCon | Build* option from the Simulink menu bar. After successful compilation and download to the WinCon Client, you should see the green START button available on the WinCon Server window. You can now start your LQR closed-loop simulation in real-time by clicking on the START/STOP button of the WinCon Server window.

Step 4. In order to observe the system's responses from the real-time simulation, open the three following WinCon Scopes: *xc (mm)*, *alpha (deg)*, and *Control Effort: Vm (V)*. You should now be able to monitor on-the-fly the simulated cart position as it tracks your pre-defined reference input, the simulated pendulum angle, as well as the corresponding control effort. However at this stage, the responses obtained should all be zero since the feedback gain has been initialized to zero in Step 1.
***Hint:*** To open a WinCon Scope, click on the Scope button of the WinCon Server window and choose the display that you want to open (e.g. *xc (mm)*) from the selection list.

Step 5. You are now ready to start the on-line tuning of your LQR. First, the Matlab script *setup_lab_ip01_2_sip.m* should be edited in order to set the *CONTROLLER_TYPE* flag to *'LQR_GUI_TUNING'*. Press F5 to execute the modified file. In this mode, the *setup_lab_ip01_2_sip.m* script calls the custom function *d_gui_lqr_tuning.m* that creates a Matlab input dialog box similar to the one shown in Figure 12, below.

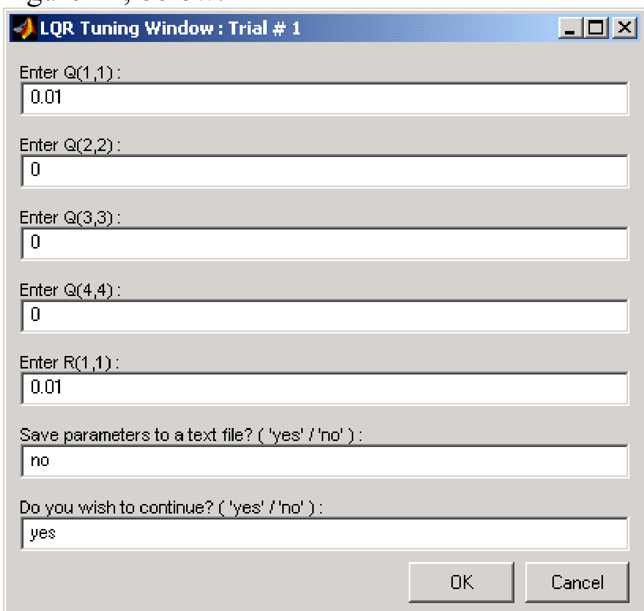



Figure 12 On-Line LQR Tuning Dialog Box

The dialog box pictured in Figure 12 allows the user to enter the diagonal elements of the LQR weighting matrices *Q* and *R*. In this laboratory, *Q* is defined to be a pure diagonal matrix of 4-by-4 size (since the system has four states). Also, because we are studying a single input system, *R* is a scalar (a.k.a. 1-by-1 matrix), of strictly positive value. It is reminded that, by design, the LQR attempts to return the system to a zero state vector by minimizing the cost function defined by *Q* and *R*. At this stage, the user, i.e. you, is entered into an iterative trial and error simulation loop. Once the tuning values of *Q* and *R* from the dialog box have been validated by clicking on the OK button, the tuning procedure then automatically calculates the corresponding LQR feedback vector *K*, by calling the function *'lqr'*, from the Matlab's Control System Toolbox. The effect of the newly calculated feedback vector can be seen right away in the real-time simulation run by WinCon. At his point, the tuning dialog box of Figure 12 re-appears on the screen for a new trial, if desired/needed. The user can click on the Cancel button (or write *'no'* in the last text input field) at any time to stop the tuning iterations. Also, if the user is satisfied by the simulated response of a particular *Q* and *R* tuning, those tuning values can be saved to a text file, called *lqr_tuning_logfile.txt*, for future reference. To do so, the user just needs to write *'yes'* in the *Save parameters to a text file ( 'yes' / 'no' ):* input field.

Step 6. Keep the intial, untuned, values for *Q* and *R* and click OK. This will compute the corresponding LQR feedback vector *K*. Observe the effect of the newly-determined *K* on the system simulated responses displayed in the three WinCon Scopes previously opened.

Step 7. In brief, the LQR tuning principle is as follows: by modifying the elements in *Q* and *R*, one can change the relative gain/weight of each state error, and the amount of control effort supplied by the input, i.e. the power spent.

The objective of this simulation is not really to make you find the perfect LQR tuning for the SIP-plus-IP01-or-IP02 system. This will be done during the actual implementation part of this laboratory. It is more to make you infer, feel, and comprehend the basic principles at work during LQR tuning.

It is also reminded the LQR design objective are enunciated in Section Controller Design Specifications, on page 4. The primal goal is, of course, to keep the pendulum balanced in the inverted position.

Now, you should decrease and/or increase the values of *Q(1,1)*, *Q(2,2)*, and *R(1,1)*, individually, or in conjunction, and observe the resulting effect on the three system responses simulated in real-time and being plotted on the Scopes by WinCon. Leave *Q(3,3)* and *Q(4,4)* equal to zero. Experiment with values included, for example, between 0.01 and 10 for *Q(1,1)* and *Q(2,2)* and between 0.0001 and 1 for *R(1,1)*. To keep a linear behaviour, ensure that the motor voltage $V_m$ never goes into saturation. Also, the pendulum angle should stay within the validity range of the linearized state-space model, as determined in the inverted pendulum free fall simulation.

Try to figure out trends on how our three tuning parameters *Q(1,1)*, *Q(2,2)*, and *R(1,1)*, affect the simulated cart position, pendulum angle, and control effort spent. Provide WinCon plots as needed to support your conjunctures.

***Hint #1:*** *Q(1,1)* can be seen as a position error gain, *Q(2,2)* as an angle error gain, and *R(1,1)* as a control effort factor.

***Hint #2:*** You can also refer to the definition of the LQR cost function to be minimized.

Step 8. Now, finalize the tuning procedure in order to meet the desired closed-loop specifications as stated in Section Controller Design Specifications. For the purpose of this simulation, limit to $V_m$ ±3-to-4 V. What is your final feedback gain vector, *K*, satisfying the design requirements? What are the corresponding weighting matrices *Q* and *R*? Calculate the location of the corresponding closed-loop poles. Compare them to the location of the open-loop poles found in Assignment #2, question 3.

***Hint #1*:** Use the Matlab function *'eig'* to determine the eigenvalues of the closed-loop state-space matrix.

***Hint #2*:** The closed-loop state-space matrix can be expressed as: *A-B*K*.

Step 9. As an example, your simulated responses could look like those plotted in Figures 13 and 14, below. If your responses do not meet the desired design specifications of Section Controller Design Specifications on page 4, you should continue your iterative tuning as described in Steps 7 and 8, above. A trade-off is probably to be found between the response performance of $x_c$ of that of $\alpha$. Once you found acceptable values for *Q* and *R* satisfying the design requirements, save them for the following of this in-lab session as well as the corresponding value of the feedback gain vector *K*. Have your T.A. check your values and simulation plots. Include in your lab report your final *Q*, *R*, and *K*, as well as the resulting response plots of $x_c$, $\alpha$, and $V_m$.

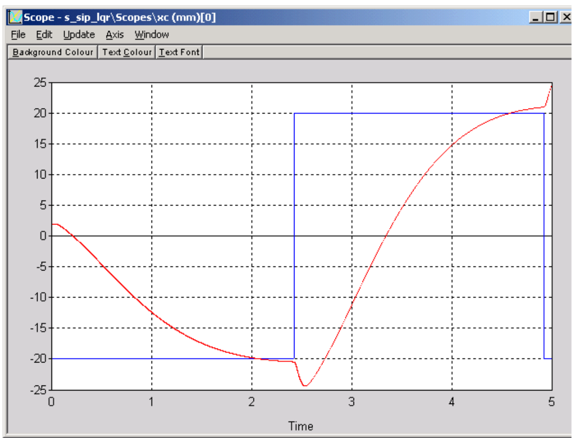



Figure 13 *xc (mm)* WinCon Scope

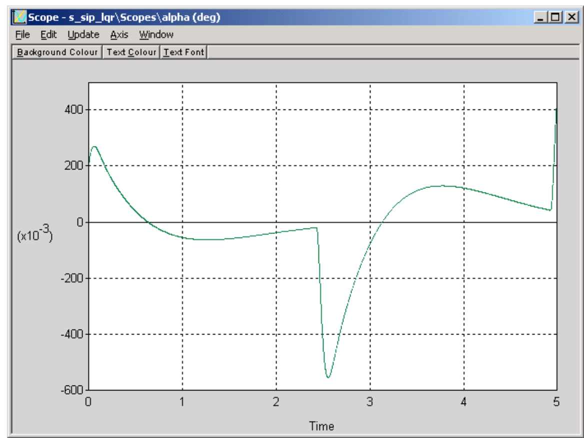



Figure 14 *alpha (deg)* WinCon Scope

Step 10. Once you feel comfortable regarding the working principles of LQR tuning and you found acceptable values for *Q* and *R* satisfying the design requirements, you can proceed to the next section. The following section deals with the implementation in real-time of your LQR closed-loop on an actual SIP-plus-IP01-or-IP02 system.

## 7.4. Real-Time Implementation of the LQ Regulator

### 7.4.1. Objectives

- To implement with WinCon a real-time LQR for the actual SIP-plus-IP01-or-IP02 plant.
- To tune on-the-fly and iteratively the LQ Regulator from the actual system response.
- To run the LQR closed-loop system simulation in parallel and simultaneously, at every sampling period, in order to compare the actual and simulated responses.

### 7.4.2. Experimental Procedure

After having gained insights, through the previous closed-loop simulation, on the LQR tuning procedure for your SIP-plus-IP01-or-IP02 plant, and checked the type of responses obtained from the system's two outputs (i.e. cart position and pendulum angle), you are now ready to implement your designed controller in real-time and observe its effect on your actual inverted pendulum system. To achieve this, please follow the steps described below:

Step 1. Depending on your system configuration, open the Simulink model file of name type *q_sip_lqr_ZZ_ip01* or *q_sip_lqr_ZZ_ip02*, where *ZZ* stands for either for *'mq3'*, *'mqpci'*, *'q8'*, or *'nie'*. Ask the TA assigned to this lab if you are unsure which Simulink model is to be used in the lab. You should obtain a diagram similar to the one shown in Figure 15. The model has 2 parallel and independent control loops: one runs a pure simulation of the LQR-plus-SIP-plus-IP01-or-IP02 system, using the plant's state-space representation. Since full-state feedback is used, ensure that the *C* state-space matrix is a 4-by-4 identity matrix; enter *'C=eye(4)'* at the Matlab prompt if necessary. The other loop directly interfaces with your hardware and runs your actual inverted pendulum mounted on your IP01 or IP02 linear servo plant. To familiarize yourself with the diagram, it is suggested that you open both subsystems to get a better idea of their composing blocks as well as take note of the I/O connections. Of interest, it should be noticed that the subsystem interfacing to the IP01 cart implements a *Bias Removal* block in order to set both initial potentiometer voltages to zero upon starting the real-time controller. Also, check that the position setpoint generated for the cart position to follow is a square wave of amplitude 20 mm and frequency 0.2 Hz. Lastly, your model sampling time should be set to 1 ms, i.e. $T_s = 10^{-3}$ s.

Step 2. Ensure that your LQR feedback gain vector *K* satisfying the system specifications, as determined in the previous section's simulations, is still set in the Matlab workspace. Otherwise, re-initialize it to the vector you found.
***Hint:***
*K* can be re-calculated in the Matlab workspace using the following command line:

```
>>K = lqr( A, B, diag( [ Q(1,1), Q(2,2), Q(3,3), Q(4,4) ] ), R(1,1) )
```

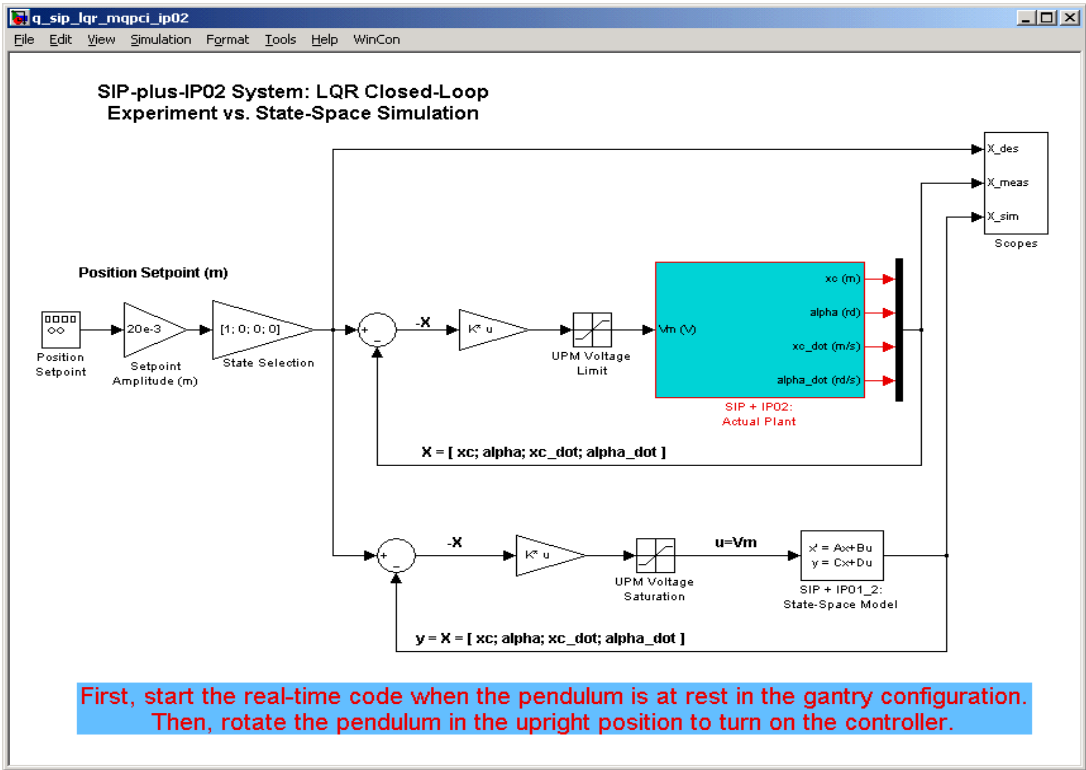



Figure 15 Diagram used for the Real-Time Implementation of the LQR Closed-Loop (IP02 with MultiQ-PCI)

Step 3. You are now ready to build the real-time code corresponding to your diagram, by using the *WinCon | Build* option from the Simulink menu bar. After successful compilation and download to the WinCon Client, you should be able to use WinCon Server to run in real-time your actual system. However, before doing so, **manually move your IP01 or IP02 cart to the middle of the track (i.e. Around the mid-stroke position) and make sure that it is free to move on both sides. Additionally, ensure that the pendulum is free to rotate over its full rotational range (±360° if configured on an IP02) anywhere within the cart's full linear range of motion.** Before starting the real-time controller, also follow the starting procedure for the inverted pendulum, as described in the following Step.

Step 4. **Inverted Pendulum Starting Procedure.** There are two different starting procedures for the inverted pendulum, depending on your type of linear cart, and its configuration. One important consideration is that encoders, and also potentiometers if a bias removal block is used, take their initial position as zero when the real-time code is started. In our linearized model, it is reminded that the zero pendulum angle corresponds to a perfectly upright position.

Therefore, the **first starting procedure is to manually hold the pendulum upright** in its equilibrium position in order to zero the initial angle measurement. Start the controller once the vertical equilibrium point is reached, and let the pendulum go when the real-time controller kicks in, without trying to help it any further. This takes the present pendulum angle measurement as zero. This is always the procedure to follow when an **IP01** cart is used. This can also be used for the IP02. Although simple, the significant inconvenient of this method is that one would have to ensure, upon starting the controller, that the pendulum is exactly vertical, pointing upright. On top of being difficult to accurately achieve, the results' repeatability between subsequent runs is highly uncertain, due to potentially different starting conditions.

That is why a **second starting procedure** has been designed. This starting procedure finds the exact vertical position from the pendulum hanging straight down, at rest, in front of the linear cart. Therefore, this method can only be practiced with an **IP02** cart, where the pendulum can be in the gantry configuration. The starting procedure **consists first of letting the pendulum come to perfect rest in the gantry configuration**, so that it is hanging straight down. Then, the real-time code can be started so that the exact ±π-radian angle is precisely known. To do this, click on the START/STOP button of the WinCon Server window. Finally, manually rotate the pendulum to its upright position. The LQR, initially turned off, automatically becomes enabled and in effect once the pendulum angle reaches zero. This starting procedure has been implemented in the *q_sip_lqr_ZZ_ip02.mdl* files, where *ZZ* stands for either for *'mq3'*, *'mqpci'*, *'q8'*, or *'nie'*.

Your cart position should now be tracking the desired setpoint while maintaining the inverted pendulum balanced.

Step 5. From the *Scopes* subsystem block, open the two sinks *Cart Position (mm)* and *Pend Angle (deg)* in two separate WinCon Scopes. You should also check the system's control effort and saturation, as mentioned in the specifications, by opening the *V Command (V)* scope located, for example, in the following subsystem path: *SIP + IP02: Actual Plant/IP02 - MQPCI/*. On the *Cart Position (mm)* scope, you should now be able to monitor on-line, as the cart moves, the actual cart position as it tracks your pre-defined reference input, and compare it to the simulation result produced by the SIP-plus-IP01-or-IP02 state-space model. On the *Pend Angle (deg)* scope, you should be able to monitor on-line, as the cart moves, the actual inverted pendulum angle and its fluctuation about the vertical axis at any given time. You can also compare it, on the same scope, to the simulated angle resulting from the state-space model in closed-loop.

***Hint #1:*** To open a WinCon Scope, click on the Scope button of the WinCon Server window and choose the display that you want to open (e.g. *Cart Position (mm)*) from the selection list.

***Hint #2*:** For a better signal visualization, you can set the WinCon scope buffer to 10 seconds. To do so, use the *Update | Buffer...* menu item from the desired WinCon scope.

Step 6. What are your observations at this point? Does your actual LQR closed-loop implementation meet the desired design specifications? If it does not, then you should finely tune the LQR weighting matrices, *Q* and *R*, in order for the actual inverted-pendulum-linear-cart system to meet the design requirements. You can do so on-the-fly and in real-time by means of the previously used on-line LQR tuning GUI. First, ensure that the Matlab script *setup_lab_ip01_2_sip.m* still has the *CONTROLLER_TYPE* flag set to *'LQR_GUI_TUNING'*. Press F5 to execute the Matlab script. In this mode, *setup_lab_ip01_2_sip.m* calls the function *d_gui_lqr_tuning.m*, which creates a Matlab input dialog box similar to the one shown in Figure 12.

Step 7. Iterate your manual LQR tuning as many times as necessary so that your actual system's performances meet the desired design specifications. If you are still unable to achieve the required performance level, ask your T.A. for advice.

Step 8. Once your results are in agreement with the closed-loop requirements, they should look similar to those displayed in Figures 16, 18, and 17, below.

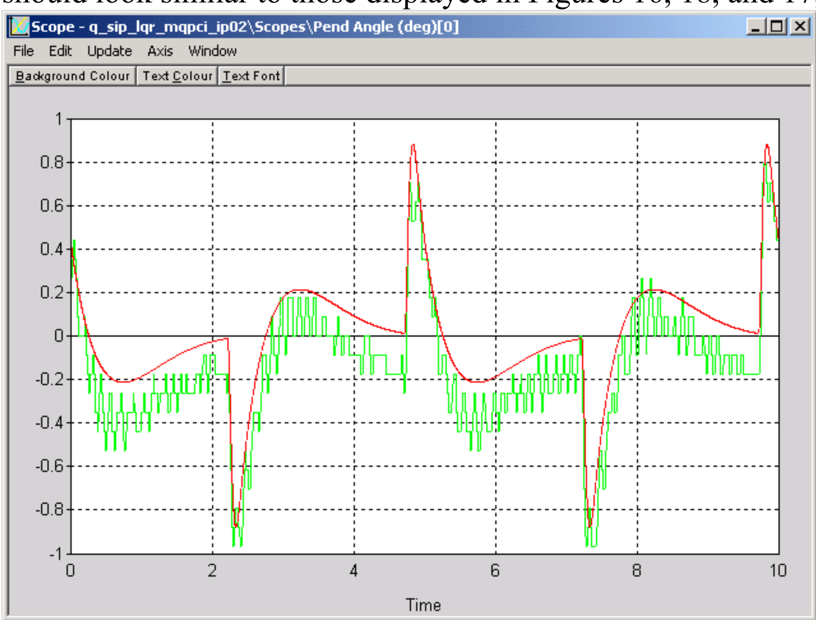


Figure 16 Actual and Simulated Pendulum Angle Responses

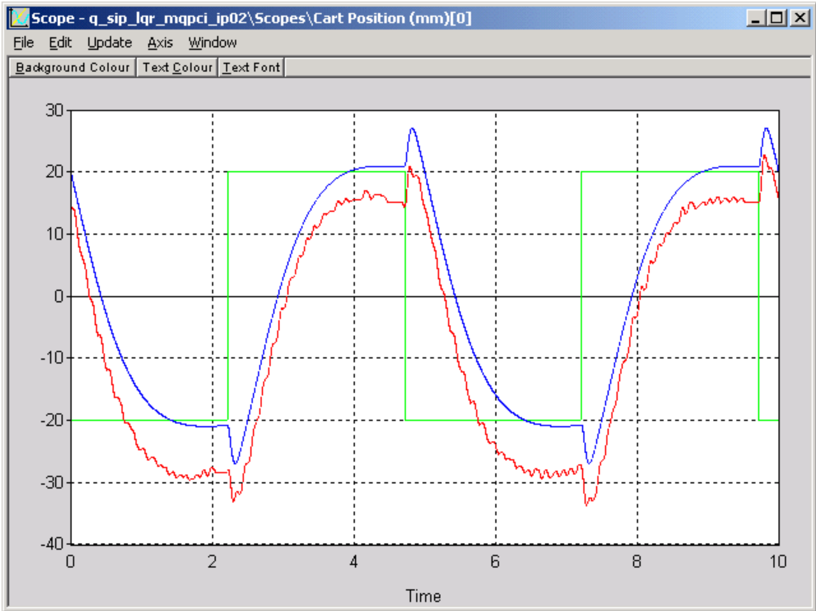


Figure 17 Actual and Simulated Cart Position Responses

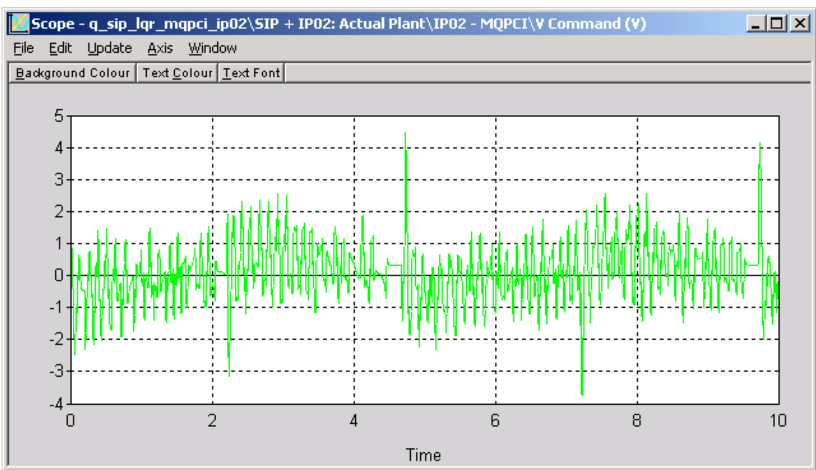


Figure 18 Actual Command Voltage

Step 9. Do you notice a steady-state error on your actual cart position response? If so, find some of the possible reasons. Can you think of any improvement on the closed-loop scheme in order to reduce, or eliminate, that steady-state error?

Step 10. Include in your lab report the WinCon plots that you obtained, as equivalent to Figures 16, 18, and 17, above. Ensure to properly document all your results and observations before moving on the the next section.

Step 11. Remember that there is no such thing as a perfect model. Specifically discuss in your lab report the following points:

i) First and foremost, how does your actual inverted pendulum angle compare to the simulated one?
ii) How does your actual IP01 or IP02 cart position compare to the simulated one?
iii) Is there any discrepancy in the results? If so, find some of the possible reasons.
iv) How much different was that the actual LQR feedback gain vector compare to the one you determined through simulation (which was based on a theoretical and ideal plant model)?

## 7.5. Assessment of the System's Disturbance Rejection

This part of the experiment is provided to give you some basic insights on the regulation problem through a few disturbance rejection considerations.

### 7.5.1. Objectives

- To observe and investigate the disturbance response of the stabilized inverted-pendulum-linear-cart system in response to a tap to the pendulum.
- To study the LQR effectiveness in maintaining the inverted pendulum in a recoverable state, that is to say in a configuration where the linear cart can rescue the pendulum from falling.

### 7.5.2. Experimental Procedure

Follow the experimental procedure described below:

Step 1. Start and run your inverted-pendulum-linear-cart system around the mid-stroke position, i.e. near the centre of the track. Use the same LQR closed-loop as the one previously developed. However this time, set the cart position setpoint amplitude to zero, so that the LQR regulates both cart position and pendulum angle around zero. This is the regulation configuration (i.e. there is no tracking).

Step 2. Once the system has stabilized, gently tap the inverted pendulum, but not more than (plus or minus) 4 degrees from its upright equilibrium position. Visually observe the response of the linear cart and its effect on the pendulum angle. Additionally, plot these two outputs in two WinCon Scopes. Also, open a WinCon Scope to plot the corresponding motor input voltage $V_m$.

***Hint #1:***

You can use the WinCon Scope's *Update | Freeze All Plots* menu item to capture, simultaneously in all the scopes, the response sweep resulting from the tap disturbance.

***Hint #2:***

Not to plot the simulated data, open up the selection list from the Scope's *File |*

*Variables...* menu item and uncheck the data coming from the simulation closed-loop (e.g. *0]* and *2]* for the *Cart Position (mm)* Scope, and *1]* for the *Pend Angle (deg)* Scope).

Step 3. How do the three responses behave, as a result to the tap, in the recovery of the inverted pendulum? How does the cart catch the pendulum? Describe the system's response from both your visual observations and the obtained response plots. Include in your lab report your plots of $x_c$, $\alpha$, and $V_m$ to support your answers.

Step 4. You can now move on to writing your lab report. Ensure to properly document all your results and observations before leaving the laboratory session.

# Appendix A. Nomenclature

Table A.1, below, provides a complete listing of the symbols and notations used in the IP01 and IP02 mathematical modelling, as presented in this laboratory. The numerical values of the system parameters can be found in Reference [2].

| ***Symbol*** | ***Description*** | ***Matlab / Simulink Notation*** |
|---|---|---|
| $V_m$ | Motor Armature Voltage | Vm |
| $I_m$ | Motor Armature Current | Im |
| $R_m$ | Motor Armature Resistance | Rm |
| $K_t$ | Motor Torque Constant | Kt |
| $\eta_m$ | Motor Efficiency | Eff_m |
| $K_m$ | Back-ElectroMotive-Force (EMF) Constant | Km |
| $E_{emf}$ | Back-EMF Voltage | Eemf |
| $J_m$ | Rotor Moment of Inertia | Jm |
| $K_g$ | Planetary Gearbox Gear Ratio | Kg |
| $\eta_g$ | Planetary Gearbox Efficiency | Eff_g |
| $M_{c1}$ | IP01 Cart Mass (Cart Alone) | Mc1 |
| $M_{c2}$ | IP02 Cart Mass (Cart Alone) | Mc2 |
| $M_w$ | IP02 Cart Weight Mass | Mw |
| $M$ | IP01 or IP02 Cart Mass, including the Possible Extra Weight | |
| $M_c$ | Lumped Mass of the Cart System, including the Rotor Inertia | Mc |
| $r_{mp}$ | Motor Pinion Radius | r_mp |
| $B_{eq}$ | Equivalent Viscous Damping Coefficient as seen at the Motor Pinion | Beq |
| $F_c$ | Cart Driving Force Produced by the Motor | |
| $x_c$ | Cart Linear Position | xc |
| $\frac{\partial}{\partial t} x_c$ | Cart Linear Velocity | xc_dot |

Table A.1 IP01 and IP02 Model Nomenclature

Table A.2, below, provides a complete listing of the symbols and notations used in the mathematical modelling of the single inverted pendulum, as presented in this laboratory. The numerical values of the pendulum system parameters can be found in Reference [3].

| *Symbol* | *Description* | *Matlab / Simulink Notation* |
|---|---|---|
| $\alpha$ | Pendulum Angle From the Upright Position | alpha |
| $\frac{\partial}{\partial t}\alpha$ | Pendulum Angular Velocity | alpha_dot |
| $\alpha_0$ | Initial Pendulum Angle (at t=0) | IC_ALPHA0 |
| $M_p$ | Pendulum Mass (with T-fitting) | Mp |
| $L_p$ | Pendulum Full Length (from Pivot to Tip) | Lp |
| $l_p$ | Pendulum Length from Pivot to Center Of Gravity | lp |
| $I_p$ | Pendulum Moment of Inertia | Ip |
| $x_p$ | Absolute x-coordinate of the Pendulum Centre Of Gravity | |
| $y_p$ | Absolute y-coordinate of the Pendulum Centre Of Gravity | |

Table A.2 Single Inverted Pendulum Model Nomenclature

Table A.3, below, provides a complete listing of the symbols and notations used in the LQR controller design, as presented in this laboratory.

| *Symbol* | *Description* | *Matlab / Simulink Notation* |
|---|---|---|
| A, B, C, D | State-Space Matrices of the SIP-plus-IP01-or-IP02 System | A, B, C, D |
| X | State Vector | X |
| K | Optimal Feedback Gain Vector | K |
| U | Control Signal (a.k.a. System Input) | |
| Q | Non-Negative Definite Hermitian Matrix | Q |
| R | Positive-Definite Hermitian Matrix | R |
| t | Continuous Time | |

Table A.3 LQR Nomenclature

# Appendix B. Non-Linear Equations Of Motion (EOM)

This Appendix derives the general dynamic equations of the Single Inverted Pendulum (SIP) module mounted on the IP01 or IP02 linear cart. The Lagrange's method is used to obtain the dynamic model of the system. In this approach, the single input to the system is considered to be $F_c$.

To carry out the Lagrange's approach, the Lagrangian of the system needs to be determined. This is done through the calculation of the system's total potential and kinetic energies.

According to the reference frame definition, illustrated in Figure 5, on page 5, the absolute Cartesian coordinates of the pendulum's centre of gravity are characterized by:

$$x_p(t) = x_c(t) - l_p \sin(\alpha(t)) \quad \text{and} \quad y_p(t) = l_p \cos(\alpha(t)) \qquad \text{[B.1]}$$

Let us first calculate the system's total potential energy $V_T$. The potential energy in a system is the amount of energy that that system, or system element, has due to some kind of work being, or having been, done to it. It is usually caused by its vertical displacement from normality (gravitational potential energy) or by a spring-related sort of displacement (elastic potential energy).
Here, there is no elastic potential energy in the system. The system's potential energy is only due to gravity. The cart linear motion is horizontal, and as such, never has vertical displacement. Therefore, the total potential energy is fully expressed by the pendulum's gravitational potential energy, as characterized below:

$$V_T = M_p\, g\, l_p \cos(\alpha(t)) \qquad \text{[B.2]}$$

It can be seen from Equation [B.2] that the total potential energy can be expressed in terms of the generalized coordinate(s) alone.

Let us now determine the system's total kinetic energy $T_T$. The kinetic energy measures the amount of energy in a system due to its motion. Here, the total kinetic energy is the sum of the translational and rotational kinetic energies arising from both the cart (since the cart's direction of translation is orthogonal to that of the rotor's rotation) and its mounted inverted pendulum (since the SIP's translation is orthogonal to its rotation).
First, the translational kinetic energy of the motorized cart, $T_{ct}$, is expressed as follows:

$$T_{ct} = \frac{1}{2} M \left( \frac{d}{dt} x_c(t) \right)^2 \qquad \text{[B.3]}$$

Second, the rotational kinetic energy due to the cart's DC motor, $T_{cr}$, can be characterized by:

$$T_{cr} = \frac{1}{2}\frac{J_m K_g^2 \left(\frac{d}{dt}x_c(t)\right)^2}{r_{mp}^2} \qquad [B.4]$$

Therefore, as a result of Equations [B.3] and [B.4], $T_c$, the cart's total kinetic energy, can be written as shown below:

$$T_c = \frac{1}{2}M_c\left(\frac{d}{dt}x_c(t)\right)^2 \qquad \text{where} \qquad M_c = M + \frac{J_m K_g^2}{r_{mp}^2} \qquad [B.5]$$

*Hint #1* says that the mass of the single inverted pendulum is assumed concentrated at its Centre Of Gravity (COG). Therefore, the pendulum's translational kinetic energy, $T_{pt}$, can be expressed as a function of its centre of gravity's linear velocity, as shown by the following equation:

$$T_{pt} = \frac{1}{2}M_p\sqrt{\left(\frac{d}{dt}x_p(t)\right)^2 + \left(\frac{d}{dt}y_p(t)\right)^2} \qquad [B.6]$$

where, the linear velocity's x-coordinate of the pendulum's centre of gravity is determined by:

$$\frac{d}{dt}x_p(t) = \left(\frac{d}{dt}x_c(t)\right) - l_p\cos(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right) \qquad [B.7]$$

and the linear velocity's y-coordinate of the pendulum's centre of gravity is expressed by:

$$\frac{d}{dt}y_p(t) = -l_p\sin(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right) \qquad [B.8]$$

In addition, the pendulum's rotational kinetic energy, $T_{pr}$, can be characterized by:

$$T_{pr} := \frac{1}{2}I_p\left(\frac{d}{dt}\alpha(t)\right)^2 \qquad [B.9]$$

Thus, the total kinetic energy of the system is the sum of the four individual kinetic energies, as previously characterized in Equations [B.5], [B.6], [B.7], [B.8], and [B.9]. By expanding, collecting terms, and rearranging, the system's total kinetic energy, $T_T$, results to be such as:

$$T_T = \frac{1}{2}(M_c + M_p)\left(\frac{d}{dt}x_c(t)\right)^2 - M_p\,l_p\cos(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)\left(\frac{d}{dt}x_c(t)\right) + \frac{1}{2}(I_p + M_p\,l_p^2)\left(\frac{d}{dt}\alpha(t)\right)^2 \qquad [B.10]$$

It can be seen from Equation [B.10] that the total kinetic energy can be expressed in terms of both the generalized coordinates and of their first-time derivatives.

Let us now consider the Lagrange's equations for our system. By definition, the two Lagrange's equations, resulting from the previously-defined two generalized coordinates, $x_c$ and α, have the following formal formulations:

$$\left(\frac{\partial}{\partial t\,\partial\frac{d}{dt}x_c(t)}L\right) - \left(\frac{\partial}{\partial x_c}L\right) = Q_{x_c} \qquad [B.11]$$

and:

$$\left(\frac{\partial}{\partial t\,\partial\frac{d}{dt}\alpha(t)}L\right) - \left(\frac{\partial}{\partial\alpha}L\right) = Q_{\alpha} \qquad [B.12]$$

In Equations [B.11] and [B.12], above, $L$ is called the Lagrangian and is defined to be equal to:

$$L = T_T - V_T \qquad [B.13]$$

In Equation [B.11], $Q_{xc}$ is the generalized force applied on the generalized coordinate $x_c$. Likewise in Equation [B.12], $Q_\alpha$ is the generalized force applied on the generalized coordinate α. Our system's generalized forces can be defined as follows:

$$Q_{x_c}(t) = F_c(t) - B_{eq}\left(\frac{d}{dt}x_c(t)\right) \quad \text{and} \quad Q_\alpha(t) = -B_p\left(\frac{d}{dt}\alpha(t)\right) \qquad [B.14]$$

It should be noted that the (nonlinear) Coulomb friction applied to the linear cart has been neglected. Moreover, the force on the linear cart due to the pendulum's action has also been neglected in the presently developed model.

Calculating Equation [B.11] results in a more explicit expression for the first Lagrange's equation, such that:

$$(M_c+M_p)\left(\frac{d^2}{dt^2}x_c(t)\right)-M_p\,l_p\cos(\alpha(t))\left(\frac{d^2}{dt^2}\alpha(t)\right)+M_p\,l_p\sin(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)^2= F_c-B_{eq}\left(\frac{d}{dt}x_c(t)\right) \qquad [B.15]$$

Likewise, calculating Equation [B.12] also results in a more explicit form for the second Lagrange's equation, as shown below:

$$-M_p\,l_p\cos(\alpha(t))\left(\frac{d^2}{dt^2}x_c(t)\right)+(I_p+M_p\,l_p^{\;2})\left(\frac{d^2}{dt^2}\alpha(t)\right)-M_p\,g\,l_p\sin(\alpha(t))= -B_p\left(\frac{d}{dt}\alpha(t)\right) \qquad [B.16]$$

Finally, solving the set of the two Lagrange's equations, as previously expressed in Equations [B.15] and [B.16], for the second-order time derivative of the two Lagrangian coordinates results in the following two non-linear equations:

$$\frac{d^2}{dt^2}x_c(t)=\left(-(I_p+M_p\,l_p^{\;2})\,B_{eq}\left(\frac{d}{dt}x_c(t)\right)-(M_p^{\;2}\,l_p^{\;3}+I_p\,M_p\,l_p)\sin(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)^2 - M_p\,l_p\cos(\alpha(t))\,B_p\left(\frac{d}{dt}\alpha(t)\right)+(I_p+M_p\,l_p^{\;2})\,F_c+M_p^{\;2}\,l_p^{\;2}\,g\cos(\alpha(t))\sin(\alpha(t))\right) \Big/ \left((M_c+M_p)\,I_p+M_c\,M_p\,l_p^{\;2}+M_p^{\;2}\,l_p^{\;2}\sin(\alpha(t))^2\right) \qquad [B.17]$$

and:

$$\frac{d^2}{dt^2}\alpha(t)=\left((M_c+M_p)\,M_p\,g\,l_p\sin(\alpha(t))-(M_c+M_p)\,B_p\left(\frac{d}{dt}\alpha(t)\right) - M_p^{\;2}\,l_p^{\;2}\sin(\alpha(t))\cos(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)^2-M_p\,l_p\cos(\alpha(t))\,B_{eq}\left(\frac{d}{dt}x_c(t)\right) + F_c\,M_p\,l_p\cos(\alpha(t))\right) \Big/ \left((M_c+M_p)\,I_p+M_c\,M_p\,l_p^{\;2}+M_p^{\;2}\,l_p^{\;2}\sin(\alpha(t))^2\right) \qquad [B.18]$$

Equations [B.17] and [B.18] represent the Equations Of Motion (EOM) of the system.