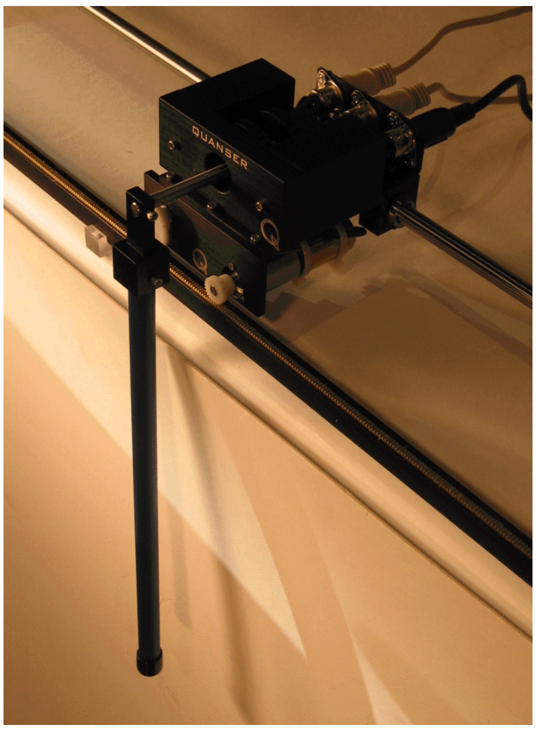
*Linear Motion Servo Plant: IP02*

# Linear Experiment #4: Pole Placement

## *Single Pendulum Gantry (SPG)*




## Student Handout

## Table of Contents

# 1. Objectives

The Single Pendulum Gantry (SPG) experiment presents a single pendulum rod which is suspended in front of an IP02 linear cart. The challenge of the present laboratory is to design a control system that makes the pendulum tip follow a commanded position on a swift and accurate manner. In other words, such a controller tracks the linear cart to a commanded position while minimizing the swing of the suspended pendulum. A few real-world applications of the gantry problem include, for example, a crane lifting and moving a heavy payload, or a pick-and-place gantry robot of an assembly line, or again the nozzle head of an inkjet printer. Therefore, the gantry response performance is assessed in terms of speed of response, minimum oscillation, and position accuracy. During the course of this experiment, you will also become familiar with the design of a full-state-feedback controller using pole placement.

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

- How to mathematically model the SPG mounted on the 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 pole-placement-based state-feedback controller satisfying the closed-loop system's desired design specifications.
- How to implement your state-feedback controller in real-time and evaluate its actual performance.
- How to use integral action to eliminate steady-state error.
- How to tune on-line and in real-time your pole locations so that the actual suspended-pendulum-linear-cart system meets the controller design requirements.
- How to observe and investigate the disturbance response of the stabilized suspended-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 IP02 main components (e.g. actuator, sensors), your data acquisition card (e.g. MultiQ), and your power amplifier (e.g. UPM), as described in References [2], [4], and [5].

ii) To be familiar with your Single Pendulum module, as described in Reference [3].

iii) 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. WinCon is fully documented in Reference [6].

iv) 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].

v) To be familiar with the pole placement design theory with regard to state-feedback controllers.

# 3. References

[1] *IP01 and IP02 – Linear Experiment #0: Integration with WinCon – Student Handout*.
[2] *IP01 and IP02 User Manual.*
[3] *IP02 – Single Pendulum Gantry (SPG) - 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 IP02, as represented in Figures 1 and 2.
- **Single Pendulum:** Quanser 12-inch Single Pendulum, seen in Figure 1, and/or 24-inch Single Pendulum, as shown in Figure 2.
- **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 IP02 as fully described in Reference [2]. When you are confident with your connections, you can power up the UPM.

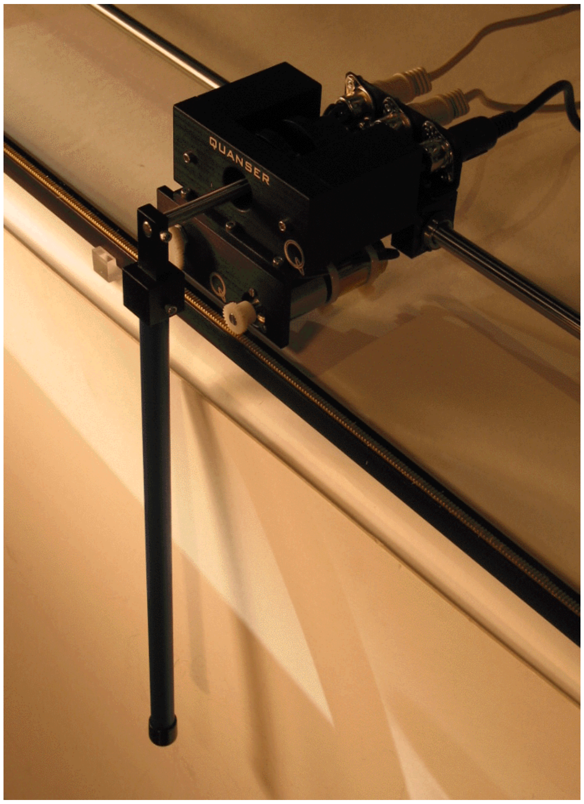


Figure 1 Medium Pendulum Suspended in Front of the IP02


Figure 2 Long Pendulum Suspended in Front of 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 full-state feedback and pole placement. As a primary objective, the obtained feedback gain vector, $K$, should allow you to minimize the swing of your single suspended pendulum. At the same time, your 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 pole placement (a.k.a. pole assignment) design theory and the corresponding implementation aspects of it. Generally speaking, the purpose of pole placement is to place the system's closed-loop eigenvalues (i.e. poles) at user-specified locations. It can be shown that for single input systems, the design that assigns all closed-loop poles to arbitrary locations, if it exists, is unique.

For our application, the suspended pendulum response performance will be assessed in terms of speed of response, minimum oscillation, and position accuracy. Consequently, the suspended-pendulum-plus-cart closed-loop system should satisfy the following design performance requirements:

1. The Percent Overshoot (*PO*) of the pendulum tip response along the x-coordinate, $x_t$, should be less than 5%, i.e.:

   $PO \leq 5\ \%$

2. The 2% settling time of the pendulum tip response along the x-coordinate, $x_t$, should be less than 2.2 seconds, i.e.:

   $t_s \leq 2.2\ [s]$

3. Zero steady-state error on the pendulum tip response, $x_t$, i.e.:

   $e_{ss} = 0$

4. The "Percent Undershoot" (i.e. *PU*) of the pendulum tip response along the x-coordinate, $x_t$, should be less than 10% of the step size, i.e.:

   $PU \leq 10\ \%$

   As it will be seen later in the in-lab session, the undershoot (i.e. an initial decrease in the response) is due to the presence of zero(s) in the Right-Half-Plane: such a system is qualified as non-minimum phase.
5. The commanded motor input voltage $V_m$ (proportional to the control effort produced) should not make the power amplifier (e.g. UPM) go into saturation.

The previous specifications are given in response to a ±30 mm square wave cart position setpoint. *PO* and *PU* are defined to limit the relative endpoint position of the gantry.

# 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 Pendulum Gantry (SPG) mounted on an IP02 linear cart is represented in Figure 3. The SPG-plus-IP02 system's nomenclature is provided in Appendix A. As illustrated in Figure 3, the positive sense of rotation is defined to be counter-clockwise (CCW), when facing the linear cart. Also, the zero angle, modulus $2\pi$, (i.e. $\alpha = 0$ rad $[2\pi]$) corresponds to a suspended pendulum perfectly vertical and pointing straight down. Lastly, 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 3.

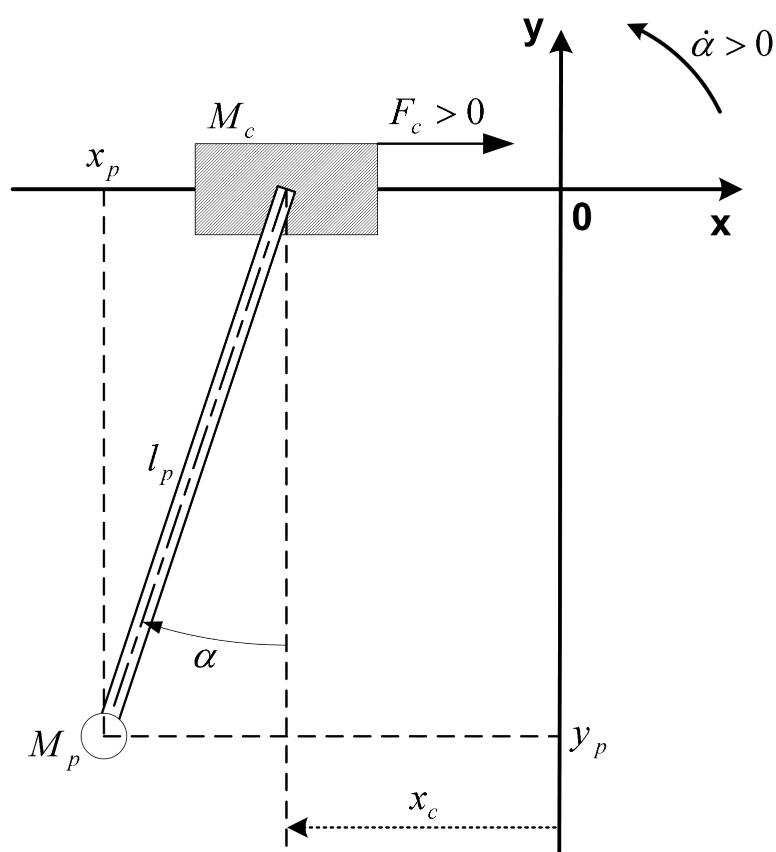



Figure 3 Schematic of the SPG Mounted in Front of the IP02 Servo Plant

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

The determination of the SPG-plus-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 3. 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 suspended 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 state-feedback controller 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 of <u>linear</u> differential equations that describes 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 suspended pendulum (a.k.a. gantry), the operating range corresponds to small departure angles, $\alpha$, from the vertical position with the pendulum pointing downwards. 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 SPG-plus-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 suspended 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 the 24-inch (i.e. "long") single pendulum mounted on the IP02 cart with the additional weight placed on top.

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? What about the damping present in the system? Do you see the need for a closed-loop controller in order to minimize the suspended pendulum swing? 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*.

5. The SPG-plus-IP02 system can be seen as a Single Input Multiple Output (SIMO) system with the two following outputs: $x_c$ and $\alpha$. However in the gantry configuration, the main variable to be controlled is $x_t$, the displacement of the pendulum tip along the x-axis. This control objective is expressed by the fact that the design specifications only bear on the response characteristics with regard to $x_t$. Therefore to evaluate the performance of our system, it will be more adequate, in the following, to consider a Single Input Single Output (SISO) system instead. As a consequence, the system's output can be set to be $x_t$, as expressed by the following relationship:
   $$Y = x_t \qquad [10]$$

   **Question:**
   Determine the state-space matrices *C* and *D* in agreement with Equation [10].
   ***Hint #1:***
   By convention of the state-space representation, the matrices *C* and *D* are defined by the following relationship:
   $$Y = C\,X + \mathrm{D}\,U \qquad [11]$$
   ***Hint #2:***
   Just like for the determination of the *A* and *B* matrices, linearization using the small angle approximation can be performed in order to obtain *C* and *D*.

## 6.3. Assignment #3: Pole Placement Design

In order to meet the design specifications previously stated, our system's closed-loop poles need to be placed judiciously. This section shows one methodology to do so. From here on, let us name the system's closed-loop poles (a.k.a. eigenvalues) as follows: $p_1$, $p_2$, $p_3$, and $p_4$.

Answer the following questions:

1. The feedback control theory states that any arbitrary set of closed-loop poles can be achieved by a constant state-feedback gain vector, *K*, if and only if the pair (*A*, *B*) is controllable. First, calculate numerically your SPG-plus-IP02 system's controllability matrix, *Co*, based on the *A* and *B* matrices evaluated in Assignment #2. Determine then if your system is controllable.
   ***Hint:***
   For a system to be controllable, its controllability matrix, *Co*, must have full rank, which is to say that its determinant must be different from zero. For our system, the controllability matrix, *Co*, can be expressed as shown below:
   $$Co = [B, A\,B, A^2\,B, A^3\,B] \qquad [12]$$

2. The pole placement method used in this laboratory consists of locating a dominating pair of complex and conjugate poles, $p_1$ and $p_2$ as illustrated in Figure 4 below, that satisfy the desired damping (i.e. *PO*) and bandwidth (i.e. $t_s$) requirements. The remaining closed-loop poles, here $p_3$ and $p_4$, are then assigned on the real axis to the left of this pair, as seen in Figure 4. The dominating pair of poles $p_1$ and $p_2$, as shown in Figure 4, can be expressed by the following equations:
   $$p_1 = -\zeta\,\omega_n + j\,\beta\,\omega_n \quad \text{and} \quad p_2 = -\zeta\,\omega_n - j\,\beta\,\omega_n \qquad [13]$$
   where β is defined as:
   $$\beta = \sqrt{1-\zeta^2} \qquad [14]$$
   since:
   $$\zeta = \cos(\phi) \quad \text{and} \quad \beta = \sin(\phi) \qquad [15]$$

   **Question:**
   Determine the locations of the dominating pair of complex and conjugate poles, $p_1$ and $p_2$, that satisfy the percent overshoot and settling time design requirements, as previously stated.

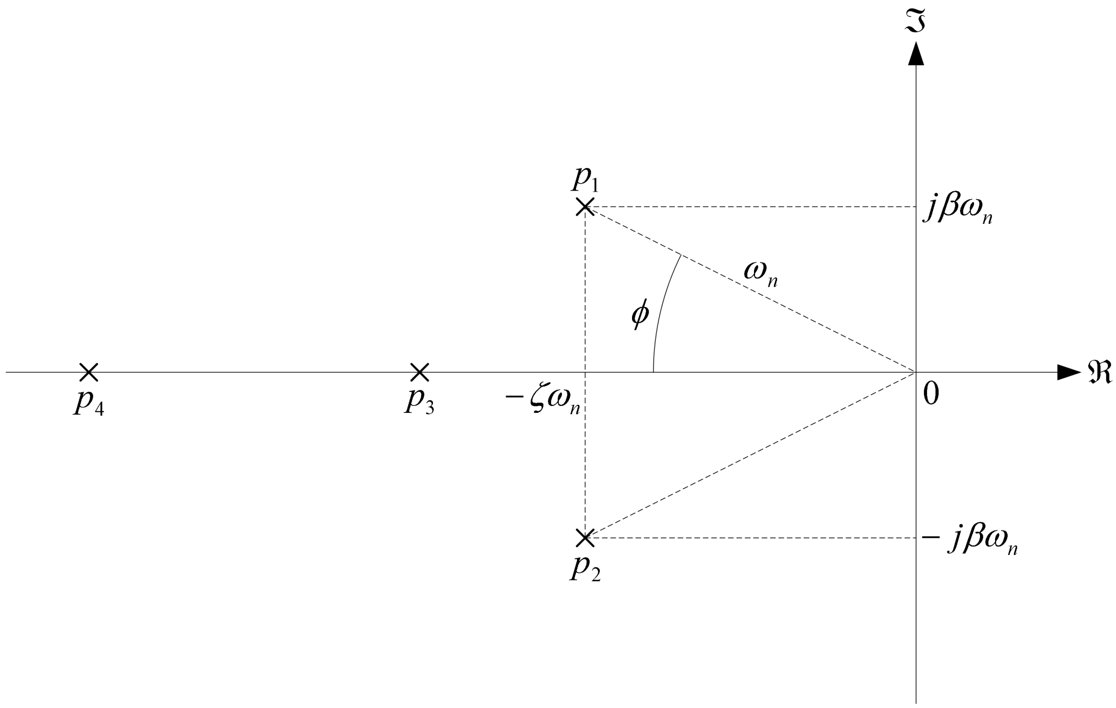



Figure 4 Closed-Loop Pole Locations in the S-Plane

The following hint formulae are provided.

***Hint formula #1:***

$$PO = 100\,e^{\left(-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \quad [16]$$

***Hint formula #2:***

$$t_s = \frac{4}{\zeta\,\omega_n} \quad [17]$$

# 7. In-Lab Procedure

## 7.1. Experimental Setup: Check Wiring and Connections

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].

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 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 laboratory.

## 7.2. Matlab Simulation of the Pole-Placement-Based Controller

### 7.2.1. Objectives

- To investigate the properties of the SPG-plus-IP02 open-loop model like, for example, its pole-zero structure.
- To set the locations of the two remaining closed-loop poles, $p_3$ and $p_4$, in order to meet the design specifications.
- To investigate the properties of the SPG-plus-IP02 closed-loop model. More specifically, the pole-zero structure and step response will be looked at.

### 7.2.2. Procedure

In this first in-lab section, you will use some of the Matlab's Control System Toolbox functions to simulate your SPG-plus-IP02 system and obtain more information and a deeper insight about your model. Follow the procedure described below:

Step1. Use Matlab to find the pole-zero locations of the SISO system previously determined in Assignment #2 – question 5. ***Hint:*** You can use Matlab's *'ss2zp'* function.

Step2. What is the open-loop pole-zero location of the SISO system in the s-plane? What are your observations? What conclusion can you draw? ***Hint:*** You can use Matlab's *'pzmap'* function.

Step3. As illustrated in Figure 4 above, assign the two remaining closed-loop poles, $p_3$ and $p_4$, to arbitrary locations to the left of the dominating pair, $p_1$ and $p_2$, as calculated in Assignment #3. $p_3$ and $p_4$ may be on the real axis since the desired damping requirement should already be achieved by the designed $p_1$ and $p_2$.

Step4. Use Matlab's *'place'* function to calculate the state-feedback gain vector, *K*, required to obtain the four closed-loop pole locations that you determined and chose.

Step5. To check your result, calculate the closed-loop poles of the obtained state-feedback system. Do the closed-loop eigenvalues match their pre-assign locations? Review your design procedure if they do not. ***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 follows: *A-B*K*.

Step6. Obtain from Matlab the closed-loop pole-zero location of the SISO system in the s-plane? What conclusion can you draw? ***Hint:*** You can use the Matlab's *'pzmap'* function.

Step7. Simulate with Matlab the closed-loop response of the SISO system to a unit step. Plot the pendulum tip response along the x-coordinate (i.e. $x_t$). Include this plot to your lab report. You should obtain a plot similar to the one shown in Figure 5, below. ***Hint:*** You can use Matlab's *'step'* function.

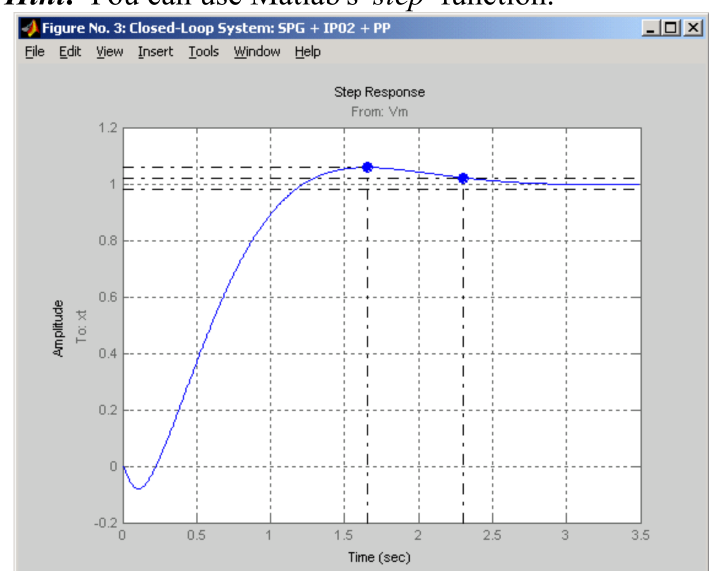



Figure 5 Simulated Step Response of the SISO Closed-Loop System: $x_t$

Step8. Does your simulated response match the design specifications with regard to *PO* and $t_s$? If it does not, re-iterate your design by modifying the location assignment for $p_3$ and $p_4$. After a few iterations, is your response closer to the design requirements? If your system still does not meet them, can you find a possible explanation for it? Moreover, how can you explain the undershoot, as illustrated in Figure 5, at the beginning of the response? ***Hint:*** To obtain the response characteristics from the Matlab plot generated by the *'step'* function, you can right-click on the figure and open from the pop-up menu the *Characteristics* selection list. You can then select the *Peak Response* and *Settling Time* submenu items.

## 7.3. Simulink Simulation and Design of the Pole-Placement-Based Controller

### 7.3.1. Objectives

- To implement in a Simulink diagram the open-loop model of the SPG-plus-IP02 system with a full-state feedback.
- To investigate, by means of the model simulation, the closed-loop performance and corresponding control effort, as a result from the chosen assignment for the poles.
- Refine/tune the chosen locations of the two remaining closed-loop poles, $p_3$ and $p_4$, meeting the design specifications as well as respecting the system's physical limitations (e.g. saturation limits).
- To infer and comprehend the basic principles involved in the pole placement design technique.

### 7.3.2. Presentation of the Simulation Diagram

In this section, the Simulink model file named *s_spg_pp.mdl* is used to assess the performance of your pole-placement-based feedback vector *K*. The model *s_spg_pp.mdl* is represented in Figure 6. It also forms, in a first time, the basis for refining/tuning your the pole locations of $p_3$ and $p_4$.

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

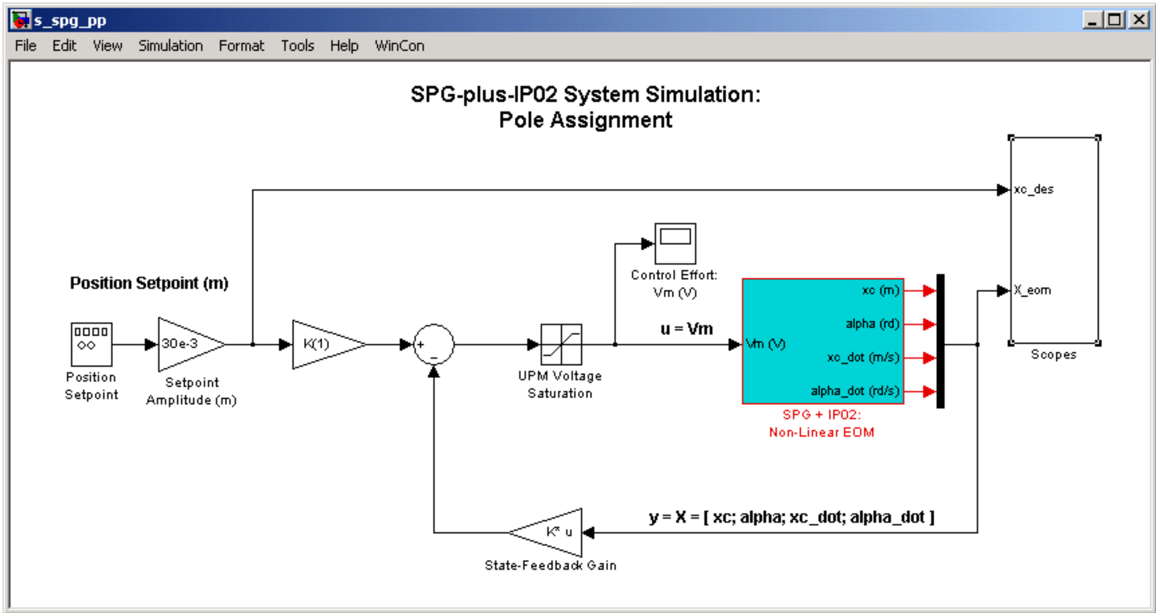



Figure 6 Simulation Diagram of the SPG-plus-IP02 System with State-Feedback Controller

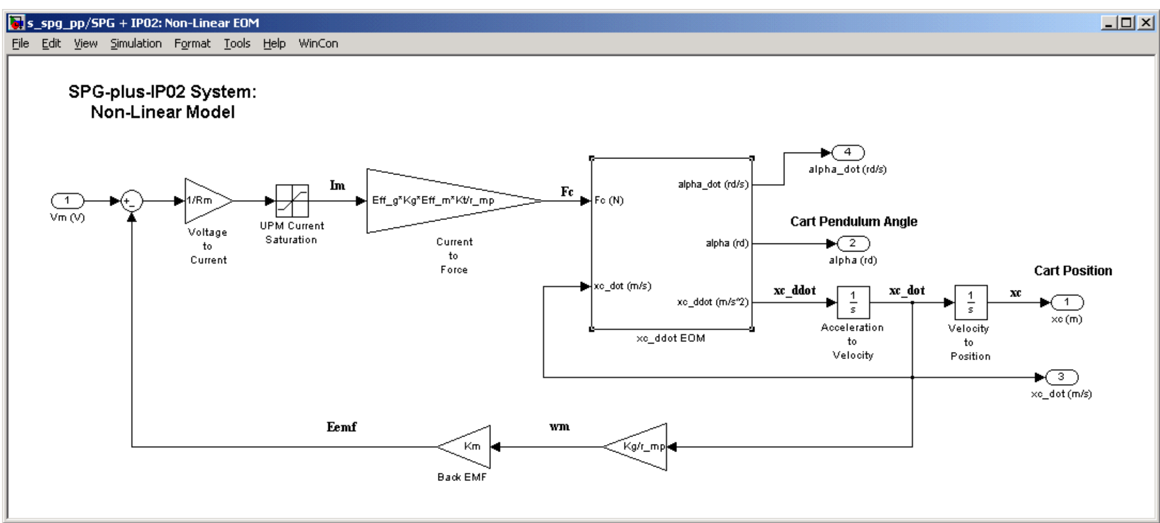



Figure 7 Simulink Subsystem Representing the IP02 Model

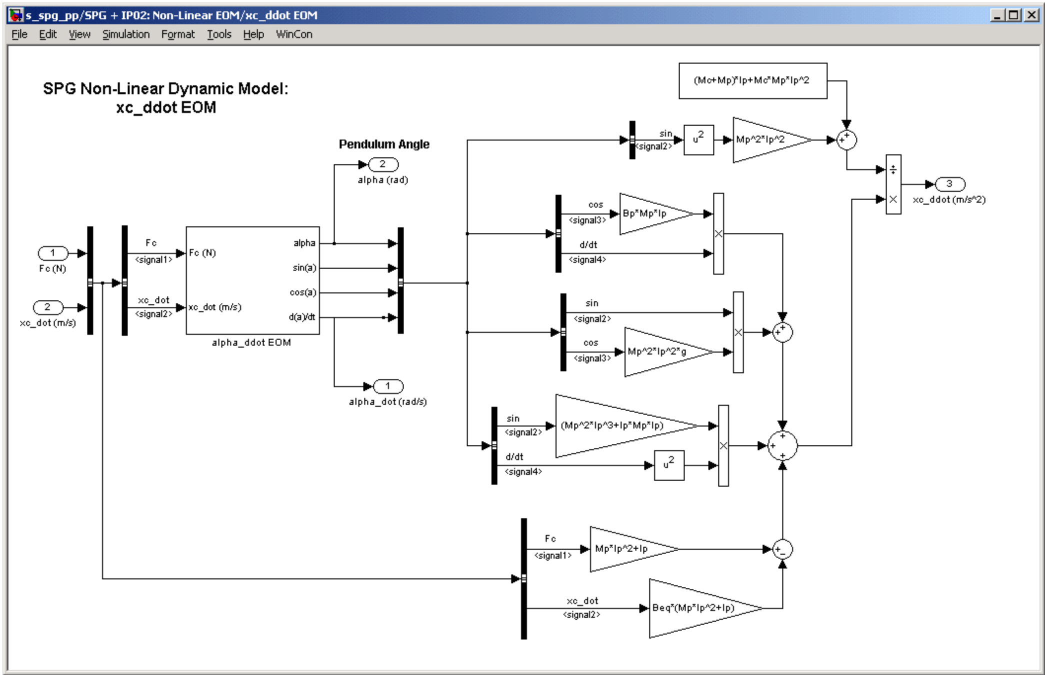



Figure 8 Simulink Subsystem Representing the First EOM

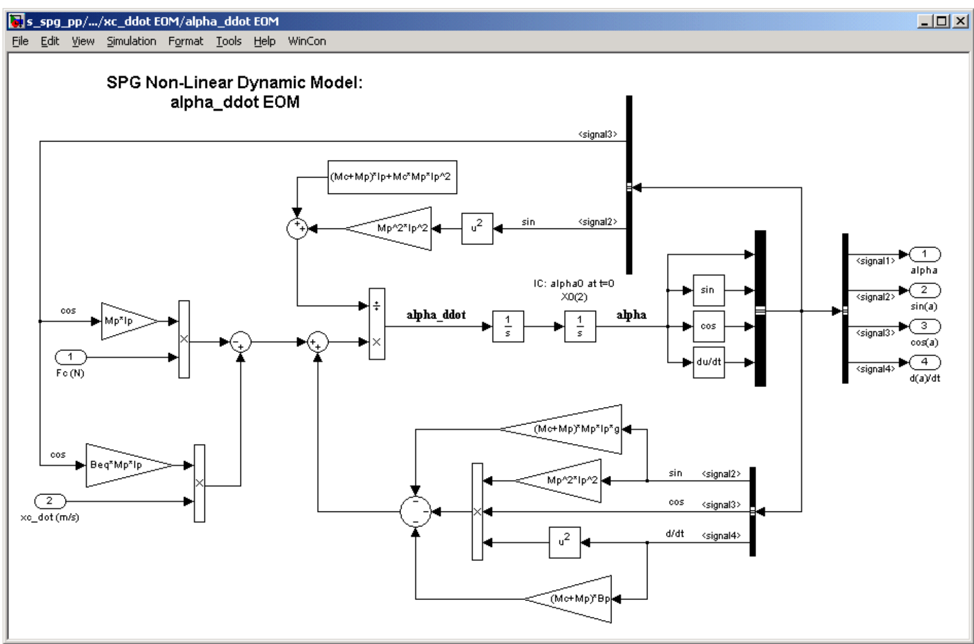



Figure 9 Simulink Subsystem Representing the Second EOM

Figure 8, above, represents the first EOM, as expressed in Equation [B.17] and of the following form:

$$\frac{\partial^2}{\partial t^2} x_c = \left( \frac{\partial^2}{\partial t^2} x_c \right) (x_c, \alpha, F_c) \quad [18]$$

Likewise, Figure 9, above, represents the second EOM, as expressed in Equation [B.18] and of the following form:

$$\frac{\partial^2}{\partial t^2} \alpha = \left( \frac{\partial^2}{\partial t^2} \alpha \right) (x_c, \alpha, F_c) \quad [19]$$

Of interest in Figure 9, it should be noted that the initial condition on α, $\alpha_0$, is contained inside the Simulink integration block located between the time derivative of α and α. That initial angle is referred in the Matlab workspace as the variable *X0(2)*.

Optionally, 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 8 and 9.

As a remark, an alternative and replacement to Figures 8 and 9, above, could have been to build two Matlab S-functions corresponding to the system's EOM, as expressed in Equation [B.17] and [B.18].

### 7.3.3. Experimental Procedure

Please follow the steps described below:

Step1. Before beginning the simulation, you must run the Matlab script called *setup_lab_ip02_spg.m*. The *setup_lab_ip02_spg.m* file initializes all the SPG-plus-IP02 model parameters and user-defined configuration variables needed and used by the Simulink diagrams. Lastly, it also calculates the state-space matrices, *A*, *B*, *C*, and *D*, corresponding to the SPG-plus-IP02 system configuration that you defined. Check that the *A* and *B* matrices set in the Matlab workspace correspond to the ones that you evaluated in Assignment #2, question 3.

Step2. If you have not done so yet, open the Simulink diagram titled *s_spg_pp.mdl*. You should obtain a diagram similar to the one shown in Figure 6. Take some time to familiarize yourself with this model: it will be used for the our pole-placement tuning simulation. For a plant simulation valid over the full angular range, the SPG-plus-IP02 model has been implemented without linear approximation through its two equations of motion [B.17] and [B.18], as previously discussed. This model is represented in *s_spg_pp.mdl* by the subsystem block titled *SPG + IP02: 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 0.1 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 (since we are using a linear controller). It can also be noticed in *s_spg_pp.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 feedback vector, named *K*, has four elements, corresponding to the four system's states defined in Equation [4].

Step3. In the Matlab workspace, set the vector *K* to the values determined in Step 4 of the previous Matlab simulation section. Alternatively, you can use the following Matlab command line to set it up:

*>>K = place( A, B, [ p1 p2 p3 p4 ] )*

Step4. Set the *Stop Time:* in the Simulink *Simulation parameters...* (Ctrl+E) to 10 seconds. You can now *Start* (Ctrl+T) the simulation of your diagram.

Step5. After the simulation run, open the four Scopes titled Scopes/*Tip Horizontal Pos (mm)*, Scopes/*xc (mm)*, Scopes/*alpha (deg)*, and especially *Control Effort: Vm (V)*. What do you observe? Does your system response meet all the design requirements? Acceptable responses are shown in Figure 10 and Figure 11, below. ***Hint:*** In order to satisfy the design requirement on the control effort produced (i.e. specification 5), the commanded motor input voltage, $V_m$, should always be between ±10V. In such a case, the actual system (e.g. power amplifier) will not go into saturation.

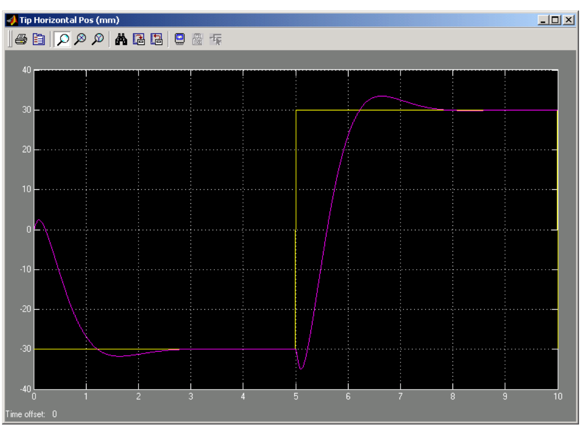



Figure 10 *Tip Horizontal Pos (mm)* Simulink Scope

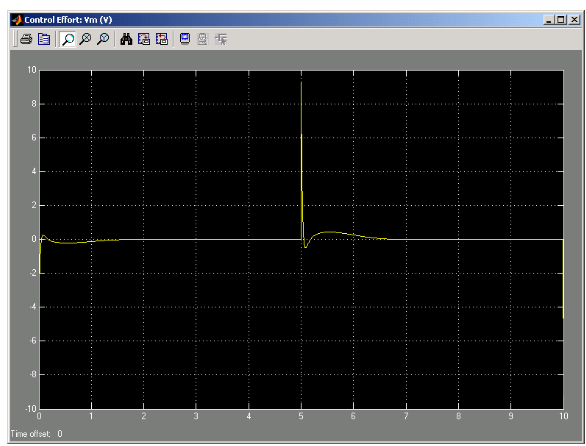



Figure 11 *Control Effort: Vm (V)* Simulink Scope

Step6. Infer the relationship between the locations of $p_3$ and $p_4$, and the resulting gain vector K, the achieved system performance, and the resulting control effort supplied by the input, i.e. the power spent.

Step7. If your responses do not meet all the desired design specifications of Section Controller Design Specifications on page 4, you should re-iterate your location assignment of $p_3$ and $p_4$, re-calculate the corresponding $K$ and re-run the simulation until the achieved performance and cost of control are satisfactory. A trade-off is probably to be found between the response performance of $x_t$ and the cost of control.

Step8. Once you found acceptable values for $p_3$ and $p_4$ satisfying the design requirements, save them for the following of this in-lab session as well as the obtained value of the feedback gain vector $K$. Have your T.A. check your values and simulation plots. Include in your lab report your final $p_3$, $p_4$, and $K$, as well as the resulting response plots of $x_t$, $x_c$, $\alpha$, and $V_m$.

Step9. Once you feel comfortable regarding the design principles involved in the pole placement technique and you found acceptable values for $p_3$ and $p_4$ satisfying the design requirements, you can proceed to the next section. That section deals with the implementation in real-time of your pole-placement-obtained state-feedback controller on your actual SPG-plus-IP02 system.

## 7.4. Real-Time Implementation of the State-Feedback Controller

### 7.4.1. Objectives

- To implement with WinCon a real-time state-feedback controller for your actual SPG-plus-IP02 plant.
- To refine the chosen placements of closed-loop poles so that the actual system meets the desired design specifications.
- To run the state-feedback closed-loop system simulation in parallel and simultaneously, at every sampling period, in order to compare the actual and simulated responses.
- To eliminate any steady-state error present in the actual responses by introducing an integral control action.
- To tune on-the-fly the integral gain $K_i$.
- To investigate the effect of partial state-feedback on the closed-loop responses.

### 7.4.2. Experimental Procedure

After having gained insights, through the previous closed-loop simulation, on the placement procedure of closed-loop poles for your SPG-plus-IP02 plant, and checked the type of responses obtained from the system's main output, $x_t$ (i.e. the pendulum tip horizontal position), you are now ready to implement your pole-placement-designed controller in real-time and observe its effect on your actual linear-cart-suspended-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_spg_pp_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 12. The model has 2 parallel and independent control loops: one runs a pure simulation of the state-feedback-controller-plus-SPG-plus-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 suspended pendulum mounted in front of your 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. You should check that the position setpoint generated for the cart and pendulum tip x-coordinate to follow is a square wave of amplitude 30 mm and frequency 0.1 Hz. Lastly, your model sampling time should be set to 1 ms, i.e. $T_s = 10^{-3}$ s.

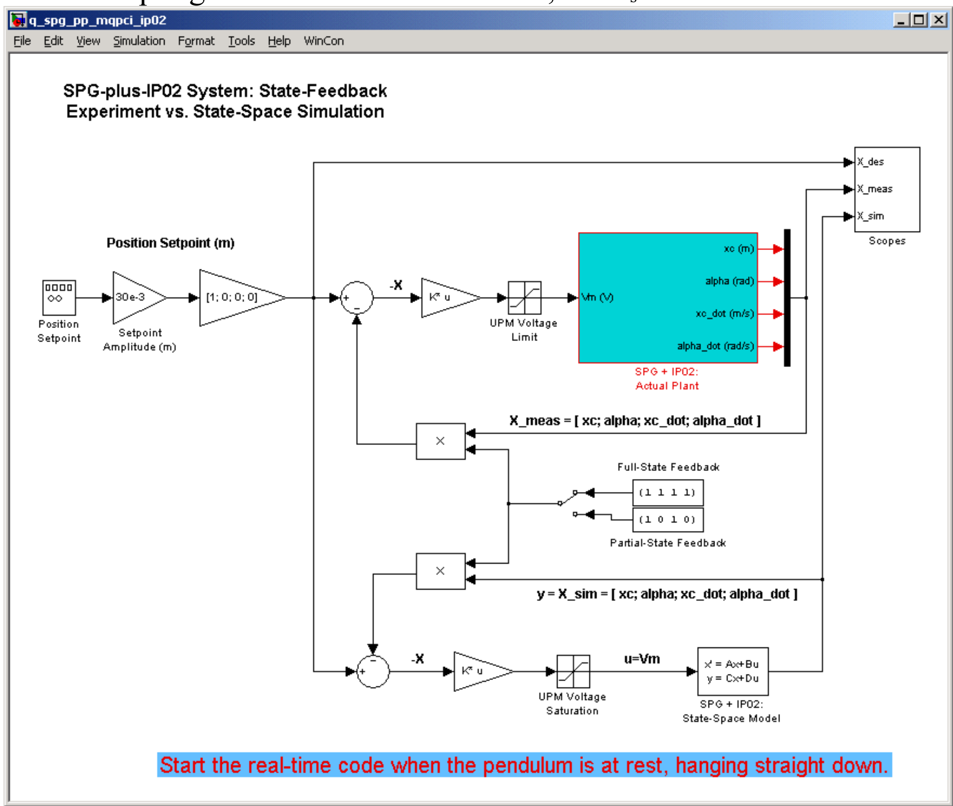


Figure 12 Diagram used for the Real-Time Implementation of the State-Feedback Closed-Loop

Step 2. Ensure that your 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 = place( A, B, [ p1 p2 p3 p4 ] )*

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 starting the real-time code, follow the SPG starting procedure described in the following Step.

Step 4. **Single Pendulum Gantry starting procedure:** the real-time code should only be started when the suspended pendulum is hanging at rest in its equilibrium position, and pointing straight down. The pendulum starting procedure is important in order to properly initialize the encoder counts to zero (this is automatically done at real-time code start-up) only when the perfectly vertical position with pendulum pointing downwards is reached.

Step 5. Position the IP02 cart around the mid-track position and wait for the suspended pendulum to come to complete rest. Ensure that the system is free to move over its workspace. You can now start your real-time controller by clicking on the START/STOP button of the WinCon Server window. Your IP02 cart position should now be tracking the desired square wave setpoint while minimizing the swing of the suspended pendulum.

Step 6. In a WinCon Scope, open the *Scopes/Pend Tip Pos (mm)* sink. For more insight on your actual system's behaviour, also open the two sinks named *Scopes/xc (mm)* and *Scopes/Pend Angle (deg)* in two other separate WinCon Scopes. Finally, you should also check the system's control effort with regard to saturation, as mentioned in the design specifications. Do so by opening the *V Command (V)* scope located, for example, in the following subsystem path: *SPG + IP02: Actual Plant/IP02 - MQPCI Plant/*. On the *Pend Tip Pos (mm)* scope, you should now be able to monitor on-line, as the cart and pendulum move, the actual pendulum end position as it tracks your pre-defined reference input, and compare it to the simulation result produced by the SPG-plus-IP02 state-space model. Such a response is shown in Figure 13, below.
***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. *Pend Tip Pos (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.

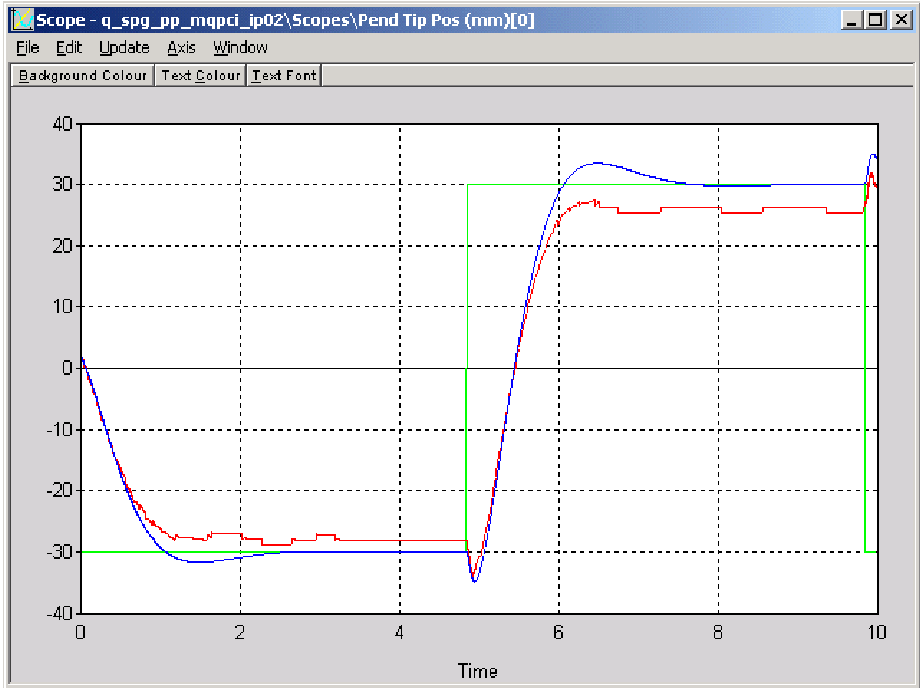



Figure 13 Actual and Simulated Pendulum End Effector (i.e. $x_t$) Position Response

Step 7. Analyze your system response at this point, as shown on your *Pend Tip Pos (mm)* scope, in terms of *PO*, $t_s$, and steady-state error. You can support your theory by also observing the *Scopes/xc (mm)* and *Scopes/Pend Angle (deg)* scopes, showing. Do not forget to mention the corresponding control effort spent, by monitoring the *V Command (V)* scope. Does your system meet all the design specifications? Include the values of your system's performances in your lab report, as well as a plot of your *Scopes/Pend Tip Pos (mm)* scope.

Step 8. Try to achieve the design specifications as closely as possible by refining your two real poles' positions, $p_3$ and $p_4$. Make sure to re-calculate (using the Matlab *'place'* function) your gain vector *K* and to apply it to your real-time code.

Step 9. Do you notice a steady-state error on your actual cart position response? Does that surprise you? 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. As you might have already figured out, to meet the zero steady-state error requirement in our crane, we could introduce integral action on the linear cart's position state, $x_c$. Our goal is to eliminate the steady-state error seen in Figure 13, above. Such an integrator on $x_c$ has already been implemented for you in a new controller diagram. Depending on your system configuration, open the Simulink model file of name type *q_spg_pp_I_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 14, below. Apart from the added integrator loop, this model should be the same in every aspect, including the I/O connections, as the one you previously used.

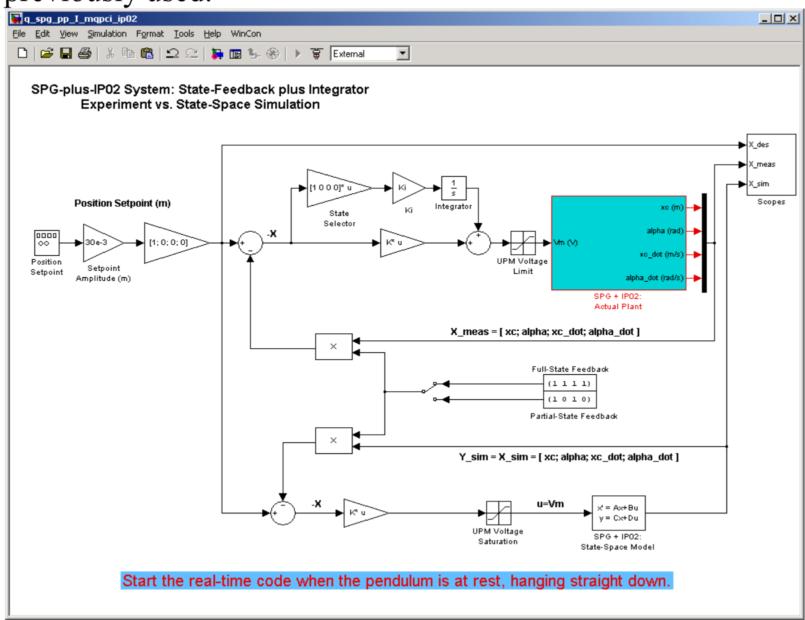



Figure 14 Actual State-Feedback Closed-Loop with Integral Action

Step 11. The integral gain is named $K_i$. First set $K_i$ to zero in your Matlab workspace. Then, compile and run your state-feedback controller with integral action on $x_c$. Re-open the previous Scopes of interest. Finally by monitoring your system actual response plotted in the *Scopes/Pend Tip Pos (mm)* Scope, tune $K_i$ to eliminate the steady-state error.

***Hint:***

You can use the *'wc_update'* WinCon script to update your real-time code with the new $K_i$.

Step 12. Did you achieve zero-steady-state error? Include your value of $K_i$ in your lab report. Your pendulum end position response should look similar to the one displayed in Figure 15.

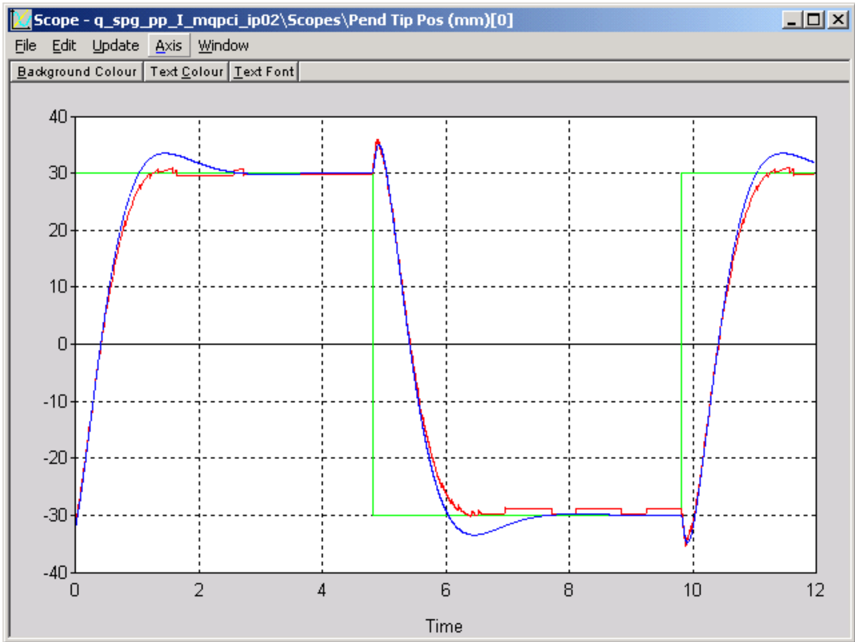



Figure 15 Actual and Simulated $x_t$ Response: State-Feedback with Integrator

Step 13. Also ensure that the actual commanded motor input voltage $V_m$ (which is proportional to the actual control effort produced) does not go into saturation. As an example, an acceptable command voltage $V_m$ is illustrated in Figure 16, below. ***Hint:*** No sign of saturation should be seen on the *V Command (V)* scope.

Step 14. Refine your two real poles' positions, $p_3$ and $p_4$, as necessary so that your actual suspended-pendulum-linear-cart closed-loop system implementation meets the set design specifications as closely as possible. Iterate your manual tuning as many times as necessary so that your actual system's performances meet the desired design requirements. If you are still unable to achieve the required performance level, ask your T.A. for advice.

Step 15. Does your final actual closed-loop implementation meet the desired design specifications? What values do you obtain for *PO*, $t_s$, $e_{ss}$, $V_m$ (minimum and maximum), and *PU*? Include the corresponding plots in your lab report to support your observations. Does your system perform better or worse than expected? Ensure to properly document all your results and observations before moving on to the next section.

Step 16. Outline the most prominent differences between the theoretical (i.e. simulated) and actual response. Remember that there is no such thing as a perfect model.

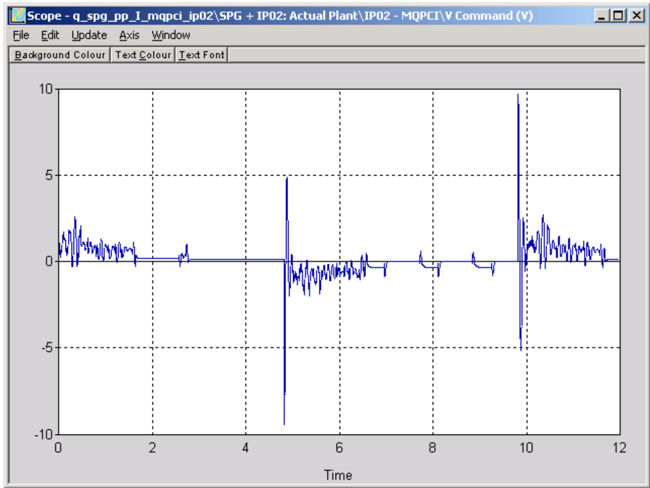


Figure 16 Actual Command Voltage: State-Feedback with Integrator

Step 17. On a side note, double-click on the manual switch located around the centre of your diagram. This should move the switch from the "up" to the "down" position. In the down position, some of the fed-back state vector elements (i.e. states) are multiplied by zero, therefore canceling their feedback: your closed-loop becomes then with partial-state feedback. Start the real-time controller again and observe the effect of canceling the two states (position and velocity) characterizing the pendulum angle, α. Your pendulum end position response should look similar to the one displayed in Figure 17, below. Comment.

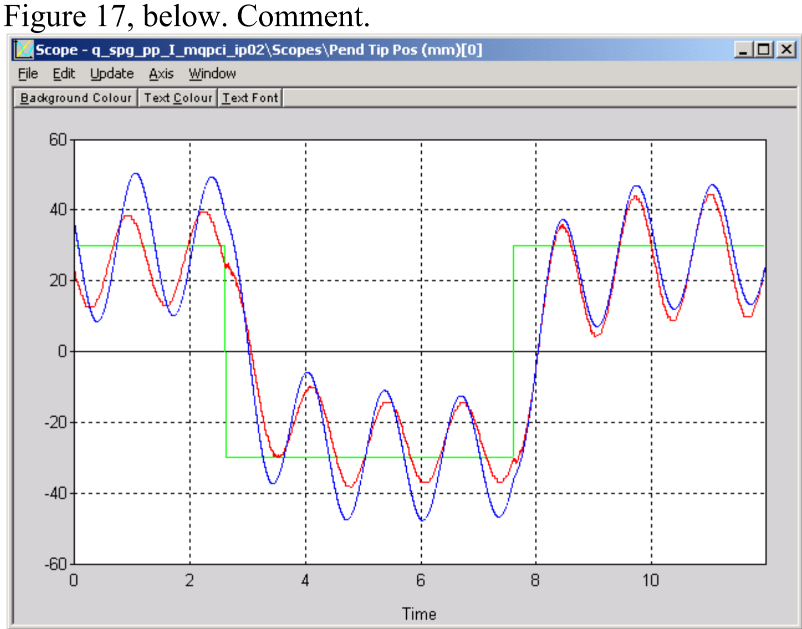


Figure 17 Actual and Simulated $x_t$ Responses: Partial-State Feedback

## 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 suspended-pendulum-linear-cart system in response to a tap to the pendulum.
- To study the full-state feedback effectiveness in stabilizing the swing of the suspended pendulum.

### 7.5.2. Experimental Procedure

Follow the experimental procedure described below:

Step 1.Start and run your suspended-pendulum-linear-cart system around the mid-stroke position (make sure to let the pendulum come to rest before starting the real-time code). Use the same full-state feedback with integral loop controller as the one previously developed. However this time, set the cart position setpoint amplitude to zero, so that the controller now 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 suspended pendulum one the side towards one end of the track. Do not apply a tap of more than (plus or minus) 20 degrees from its equilibrium (i.e. vertical) position. Visually observe the response of the linear cart and its effect on the pendulum angle. Plot these two outputs in two WinCon Scopes. Additionally, also open a WinCon Scope to plot the resulting pendulum end position and another one for 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 *Pend Tip Pos (mm)* Scope, or *1]* for the *Pend Angle (deg)* Scope).

Step 3.How do the four responses behave, as a result to the tap, in the dampening of the suspended pendulum's swing? 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 $x_t$, $x_c$, $\alpha$, 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 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 |
| $M_{c2}$ | IP02 Cart Mass (Cart Alone) | Mc2 |
| $M_w$ | IP02 Cart Weight Mass | Mw |
| M | 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 | x |
| $\frac{\partial}{\partial t} x_c$ | Cart Linear Velocity | x_dot |

Table A.1 IP02 Model Nomenclature

Table A.2, below, provides a complete listing of the symbols and notations used in the mathematical modelling of the single suspended 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 Hanging Down 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 | |
| $x_t$ | Absolute x-coordinate of the Pendulum Tip | |

Table A.2 Single Suspended Pendulum Model Nomenclature

Table A.3, below, provides a complete listing of the symbols and notations used in the pole-placement-plus-integrator design, as presented in this laboratory.

| ***Symbol*** | ***Description*** | ***Matlab / Simulink Notation*** |
|---|---|---|
| A, B, C, D | State-Space Matrices of the SPG-plus-IP02 System | A, B, C, D |
| X | State Vector | X |
| $X_0$ | Initial State Vector | X0 |
| Y | System Output Vector | |
| *PO* | Percent Overshoot | PO |
| $t_s$ | 2% Settling Time | ts |
| $\zeta$ | Damping Ratio | zeta |
| $\omega_n$ | Undamped Natural Frequency | wn |

| *Symbol* | *Description* | *Matlab / Simulink Notation* |
|---|---|---|
| K | State-Feedback Gain Vector | K |
| $K_i$ | Integral Gain | Ki |
| U | Control Signal (a.k.a. System Input) | |
| t | Continuous Time | |

Table A.3 State-Feedback Nomenclature

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

This Appendix derives the general dynamic equations of the Single Pendulum Gantry (SPG) module mounted on the 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 3, 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 gantry pendulum (since the SPG'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:

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

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:

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

*Hint #1* says that the mass of the single 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:

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

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

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

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

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

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

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

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:

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

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:

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

and:

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

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

$$L = T_T - V_T \quad \text{[B.13]}$$

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:

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

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.