Middle East Technical University
Department of Electrical and Electronics Engineering

EE 406
Laboratory of Feedback Control Systems

Experiment #4:
Crane Control using Pole Placement Design

Preliminary Work and Laboratory Manual

Initial Version: Afşar Saranlı, Emre Tuna

Group Members:

____

____

1 Objectives

The “crane system” that we will study in this experiment, also known as a Single Pendulum Gantry (SPG) consists of a rigid rod hanging from our movable IP02 cart. As the cart moves in response to the motor action, the pendulum swings in interaction with the motion of the cart. In a crane system, a skilled operator is tasked to move the load at the lower end of the crane to a target position as fast as possible and with minimum oscillations. The challenge of the present laboratory is to design a control system that would, similarly to a crane operator bring the tip of the pendulum at the commanded position in a quick and accurate manner with minimum resulting oscillations. Apart from the example of a crane moving a heavy payload, there are other real-life applications of this control system such as a pick-and-place gantry robot in an assembly line. The “crane” response performance will be characterized by the speed of the response (rise time, peak-time) as well as minimum oscillations (maximum overshoot and settling time) and position accuracy (steady state error).

At the end of this lab, you should:

- Become familiar with the nonlinear equations of motion for the swinging pendulum + IP02 cart open-loop system,
- Become familiar with how to linearize the open-loop system to obtain its linear state-space equations,
- Learn how to design, simulate, and tune a pole-placement based state-feedback controller, satisfying the closed-loop control system design specifications,
- Learn how to implement your state feedback controller in real-time and evaluate its actual performance,
- Learn how to use integral action to eliminate steady-state error,
- Learn how to tune on-line and in real-time your pole locations so that the actual closed-loop controller + swinging pendulum + IP02 cart system meets the controller design specifications,
- Observe and investigate the disturbance rejection behavior of the stabilized closed-loop controller + swinging pendulum + IP02 cart system in response to a manual tap to the pendulum.

2 Prerequisites

To successfully carry out this laboratory, the prerequisites are:

i) To be familiar with the IP02 main components (e.g., actuator, sensors), the power amplifier (e.g., UPM), and the data acquisition card (e.g., Q8 or Q2), 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 QuaRC to control and monitor the plant in real-time and in testing their controller through Simulink.
iii) To be familiar with the complete wiring of the IP02 servo plant, as per dictated in Reference [2] and carried out in the pre-laboratory for Experiment #1[1].

3 References

[1] *Experiment #1: Control Hardware and Software Setup, Signal Interfaces*

[2] *Quanser IP02 User Manual*.

[3] *Quanser Q2 USB Data Acquisition Device User Manual.*

[4] *Universal Power Module UPM User Manual.*

[5] *QuaRC User Manual* (type `doc quarc` in Matlab to access in the laboratory computers).

[6] *QuaRC Installation Manual*.

4 Experimental Setup

4.1.Main Components

This laboratory is composed of the following hardware and software components:

- **Power Module:** Quanser UPM 1503 or 2405, VoltPAQ-X1
- **Data Acquisition Board:** Quanser Q8 and Q2
- **Linear Motion Servo Plant:** Quanser IP02 cart system
- **Single Pendulum:** Quanser 12-inch and 24-inch metal pendulums, as illustrated in Figure 1
- **RealTime Control Software:** The QuaRC-Simulink configuration, as detailed in the Reference [5].
- **A Computer to Run Matlab-Simulink and the QuaRC software**

The setup for this laboratory is similar to the one you have used in Experiment #3 except for the fact that the pendulum will be mounted to the encoder shaft in front of the IP02 cart. The setup will be mostly prepared for you but the pendulum itself may not be mounted at the beginning of the lab. Please do not mount the pendulum until you are ready to test your real-time (hardware) controller. Your desk setup will have all the necessary components including all necessary software components pre-installed. You will follow the procedure of this laboratory session to setup and use the system.

When it is time to mount the pendulum, make sure you carefully fit the pendulum to the encoder shaft and reasonably tighten the set-screw using the provided key.

4.2. Wiring

To wire up the system, please follow the default wiring procedure for your IP02 as fully described in Reference [1]. When you are confident with your connections, you can power up the UPM. (If you are uncertain of any parts of the hardware, seek help from your lab assistant.)

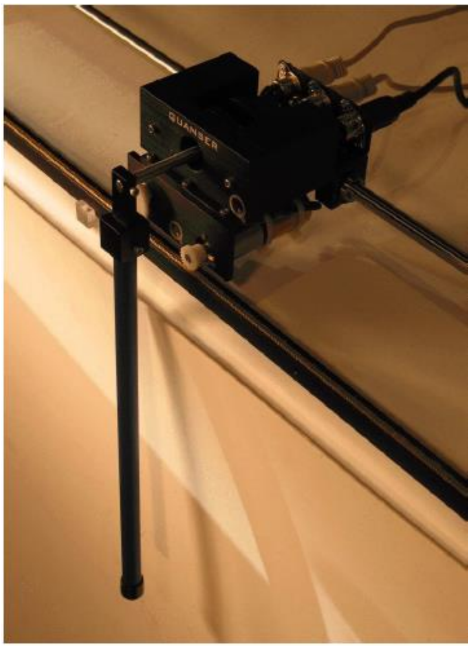


Figure 1 – The IP02 Servo Plant fitted with the 12 inch pendulum

5 Controller Design Specifications

In the preliminary work and in-lab session of the present laboratory, you will design and implement a control strategy based on full-state feedback and pole placement theory. The IP02 cart will be required to track a desired position set-point (given as a square-wave reference signal). The primary objective of our pole placement-based controller design is to find a linear state feedback gain vector K that would result for the closed-loop system to meet the design specifications on the behavior of the pendulum tip, in terms of the speed of the response (rise time, peak-time), amount of oscillations (maximum overshoot and settling time) and position accuracy (steady state error). The corresponding control effort (motor voltage) should also be checked and minimized via simulation experiments.

You should consult your EE302 Feedback Systems notes and reference texts to review pole placement-based design procedure. Generally speaking, the purpose of pole placement is to place the closed-loop eigenvalues (i.e., poles) at user specified locations. Note that for single-input controllable systems, once these locations are specified, the feedback gain vector K can be uniquely determined.

For our application, we will define the output of interest for our system to be the pendulum tip x-axis coordinate (See Figure 2) and specify the performance requirements on this output, in terms of our standard controller performance criteria, as follows:

1. The Percent Overshoot (PO) of the pendulum tip x_p along the x-axis from the commanded reference position should be

   $PO \leq 5\%$

2. The 2% band settling time for the pendulum tip x_p along the x-axis should be

   $t_s \leq 2.2$ secs.

3. Zero steady-state position error on the pendulum tip response x_p:

   $e_{ss} = 0.$

4. The "percent-undershoot" (PU) of the pendulum tip response x_p along the x-axis should be less than 10% of the step size. I.e.,

   $PU \leq 10\%$

   As it will be later seen in the lab session, the undershoot is an initial decrease in the response to a step input and is due to the presence of zero(s) in the right-half plane. Such a system is known as a non-minimum phase system.

5. The control effort produced (commanded motor input voltage Vm) should not drive the motor power amplifier (UPM) into saturation.

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

You should print out preliminary work separately from other parts.

Student Name: Özgür Gülsuna

Student ID: 2307668

6 Preliminary Work

6.1 Review the Non-Linear Equations of Motion

A schematic of our “crane” model, namely the single pendulum gantry (SPG) composed of a pendulum rod mounted on the IP02 is illustrated in Figure 2. The set of system parameters and variables relating to this system is given in Appendix A. As given in the figure, axis conventions are such that the positive sign of pendulum rotation is counter-clockwise when facing the cart, the cart and pendulum tip positions as well as the input force are defined as positive towards the right. Finally, α=0 is defined such that the pendulum is perfectly perpendicular to the ground (along the direction of gravity).

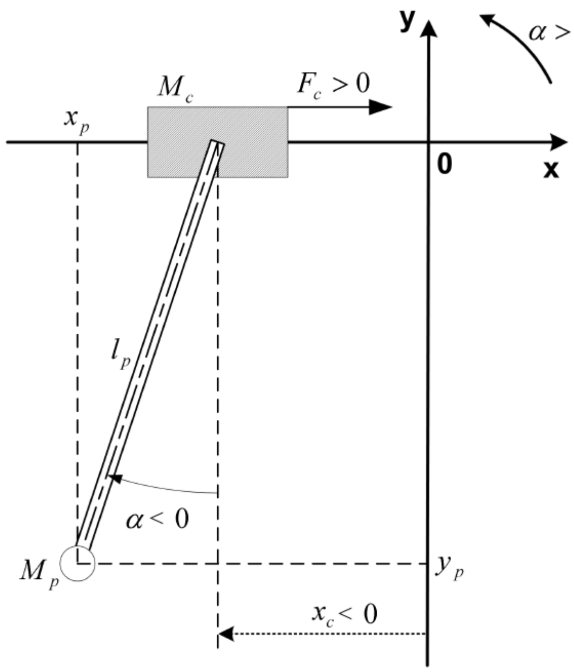


Figure 2 – Schematic of the SPG mounted in front of the IP02 servo plant

This system has two *degrees-of-freedom* (DOF) indicating the minimum number of variables to describe the system configuration. These variables are known as *generalized coordinates*. The system state vector is composed of these variables and their derivatives and completely captures the dynamics of the system. It is given as $\mathbf{z} = [z_1\ z_2\ z_3\ z_4]^T = [x_c\ \alpha\ \dot{x}_c\ \dot{\alpha}\ ]^T$ and the task of deriving the system equations of motion amounts to determining the expressions for the two variables $\ddot{x}_c$ and $\ddot{\alpha}$ in the following form

$$\ddot{x}_c = g_3(\mathbf{z}, V_m) \tag{1a}$$

$$\ddot{\alpha} = g_4(\mathbf{z}, V_m) \tag{1b}$$

Where g_2 and g_4 are two nonlinear functions of the state and the input. Here, the mechanical system input is the force F_c to the IP02 cart, but this in turn is a function of the overall plant input, i.e., the motor voltage V_m. The relation between F_c and V_m is established through the DC motor model. This means that our overall nonlinear plant has the DC motor voltage V_m as its input, i.e. u. Remember also that the plant output on the other hand is the pendulum tip x-axis position x_p, as discussed in the controller performance specifications.

The nonlinear state-space representation of the plant is given by

$$\dot{z}_1 = g_1(\mathbf{z}, V_m) = z_3 \tag{2a}$$

$$\dot{z}_2 = g_2(\mathbf{z}, V_m) = z_4 \tag{2c}$$

$$\dot{z}_3 = g_3(\mathbf{z}, V_m) \tag{2b}$$

$$\dot{z}_4 = g_4(\mathbf{z}, V_m) \tag{2d}$$

while the plant output is given by the nonlinear relation

$$y = x_p(t) = z_1 + L_p \sin(z_2)\,. \tag{3}$$

The derivation of the system equations of motion corresponding to the mechanical part can be based either on free-body diagrams combined with Newton's Second Law or an energy based approach using the Lagrangian formulation. A derivation using the latter approach is given in Appendix B. The relation between F_c and V_m is given by the armature controlled DC motor model. This is also summarized in Appendix B.

The resulting system equations of motion are given by

$$\frac{d^2}{dt^2}x_c(t) = \frac{\left(\begin{array}{l} -\left(I_p + M_p l_p^2\right)B_{eq}\left(\frac{d}{dt}x_c(t)\right) + \left(M_p^2 l_p^3 + I_p M_p l_p\right)\sin(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)^2 \ldots \\ + \left(I_p + M_p l_p^2\right)\left(-\frac{\eta_g K_g^2 \eta_m K_t K_m \left(\frac{d}{dt}x_c(t)\right)}{R_m r_{mp}^2} + \frac{\eta_g K_g \eta_m K_t V_m(t)}{R_m r_{mp}}\right)\ldots \\ + M_p l_p \cos(\alpha(t)) B_p \left(\frac{d}{dt}\alpha(t)\right) + M_p^2 l_p^2 g \cos(\alpha(t))\sin(\alpha(t)) \end{array}\right)}{\left(\left(M_c + M_p\right)I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin(\alpha(t))^2\right)} \tag{4}$$

$$\frac{d^2}{dt^2}\alpha(t) = \frac{\begin{pmatrix} -(M_c + M_p)M_p\, g\, l_p \sin(\alpha(t)) - (M_c + M_p)B_p\left(\frac{d}{dt}\alpha(t)\right)\dots \\ -\left(-\frac{\eta_g K_g^2 \eta_m K_t K_m \left(\frac{d}{dt}x_c(t)\right)}{R_m r_{mp}^2} + \frac{\eta_g K_g \eta_m K_t V_m(t)}{R_m r_{mp}}\right) M_p\, l_p \cos(\alpha(t))\dots \\ -M_p^2 l_p^2 \sin(\alpha(t))\cos(\alpha(t))\left(\frac{d}{dt}\alpha(t)\right)^2 + M_p\, l_p \cos(\alpha(t)) B_{eq}\left(\frac{d}{dt}x_c(t)\right) \end{pmatrix}}{\left((M_c + M_p)I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin(\alpha(t))^2\right)} \quad (5)$$

In this formulation, the mass of the pendulum was assumed to be concentrated at the center-of-mass of the pendulum rod.

6.2 Assignment #1: Linearization and Linear State-Space Approximation

The pole-placement based design requires a linear state-space representation for the system given by the standard form

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}u \quad (6a)$$
$$y = \mathbf{C}\mathbf{z} \quad (6b)$$

Our pendulum system is non-linear in nature and a linear approximation needs to be found in order to apply our pole-placement design procedure.

Finding a linear approximation to the system equations (i.e., *linearization*) is covered in your EE302 lectures. The **A** matrix is the Jacobian matrix **J** (essentially composed of the derivatives of the non-linear state functions $g_i(.)$ w.r.t all state variables) evaluated at the equilibrium point **z**=0. Since we have a single input, **B** is a vector also composed of the derivatives of the same functions with respect to this input, evaluated again at **z**=0.
Substituting the numerical values of all system parameters and evaluating the derivatives at the equilibrium point of the system, state equations of the linear approximation of our system is given by:

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1.5216 & -11.6513 & 0.0049 \\ 0 & -26.1093 & 26.8458 & -0.0841 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 1.5304 \\ -3.5261 \end{bmatrix}u \quad (7)$$

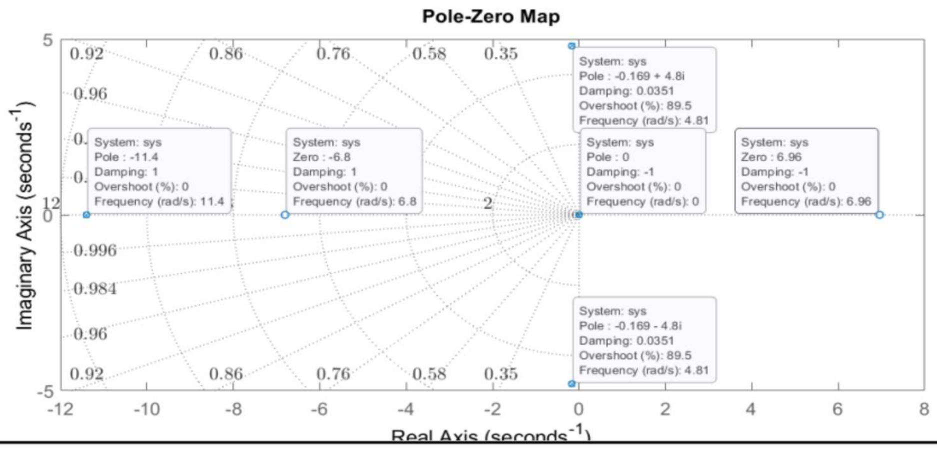
Derive the matrix **C**

$$\text{Jacobian of the output equation} \Rightarrow H = \nabla h = \begin{bmatrix} 1 & L_p cos(\alpha) & 0 & 0 \end{bmatrix}\Big|_{\alpha=0}$$

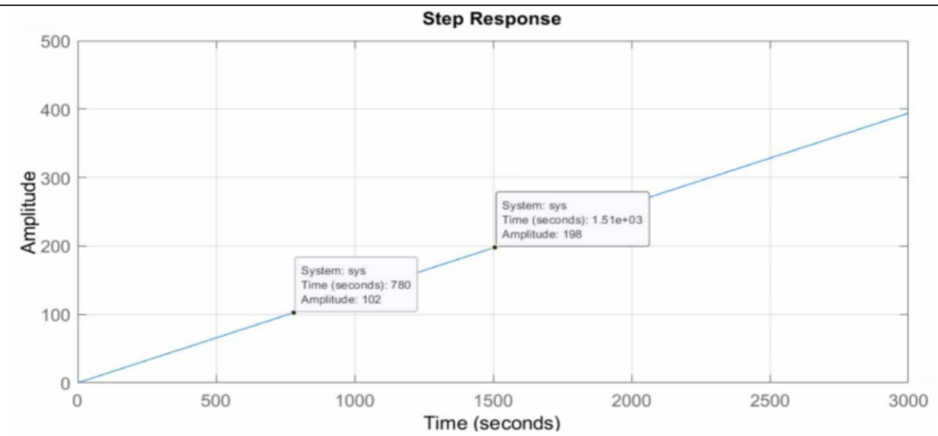
$$C = \begin{bmatrix} 1 & 0.6413 & 0 & 0 \end{bmatrix}$$

6.3 *Assignment #2: Analysis of Open-Loop System*

As a first step, we would like to analyze the behavior of the open-loop system using its linear approximation. The first step is to determine the pole-zero configuration of the



Now in Matlab, define the state-space open-loop system using the `ss(.)` command. Obtain the unit step response of the open-loop system, using the `step(.)` command. Sketch the observed response in the space provided below, indicating all necessary details.



Comment on the expected and observed behavior of the open-loop system in response to a unit step-input. Relate you discussion with the pole and zero locations.

Observing the pole-zero map of the system reveals a zero in the origin, this also means that an integrator is built in the system. Unit step response will ramp up to infinity.

6.4 *Assignment #3: Pole Placement Design*

In order to meet the design specifications previously defined, our system's closed-loop poles need to be placed with methodically. In this experiment, we will explore one such methodology for design of a controller. Let us assume that the closed-loop poles (eigenvalues) of our controlled system are named as p_1, p_2, p_3 and p_4.

6.4.1 Checking Controllability

From EE302, we know that one can arbitrarily place the closed-loop poles of a system, using a method known as pole-placement by state-feedback. However, this is only possible if the system is controllable (as determined by the pair (**A**, **B**) of the linear approximation given above).

Check whether the SPG-plus-IP02 system is controllable. Explain how C_0 is calculated. (Hint: Compute the controllability matrix and verify that it is full rank)

Controllability Matrix **Co:**

$$1000 \cdot \begin{bmatrix} 0 & 0.0015 & -0.0178 & 0.2028 \\ 0 & -0.0035 & 0.0414 & -0.3906 \\ 0.0015 & -0.0178 & 0.2028 & -2.38018 \\ -0.0035 & 0.0414 & -0.3906 & 4.3966 \end{bmatrix} = \begin{bmatrix} 0 & 1.5 & -17.8 & 202.8 \\ 0 & -3.5 & 41.4 & -390.6 \\ 1.5 & -17.8 & 202.8 & 2380.18 \\ -3.5 & 41.4 & -390.6 & 4396.6 \end{bmatrix}$$

Rank {**Co**} = 4

6.4.2 Placing the "Dominant Poles" of the System

We know from EE302 that a system's closed loop behavior can be approximated by a second order system if two of the poles "dominate" the remaining poles of the system. This is the case if the dominant poles are much closer to the *jw* axis as compared with the remaining poles of the system. In this step, we will specify the locations of the dominant pair of complex conjugate poles based on our design specifications while keeping the other pair of poles sufficiently far from these two. This is illustrated in Figure 3, where p_1 and p_2 are selected as the dominant pair.

Hint: From EE302, we know that for a second order system of the standard from we have the Percent Overshoot and Settling Time given by the expressions

$$PO(\%) = 100e^{\left(-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right)}; \; t_s = \frac{4}{\zeta w_n}.$$

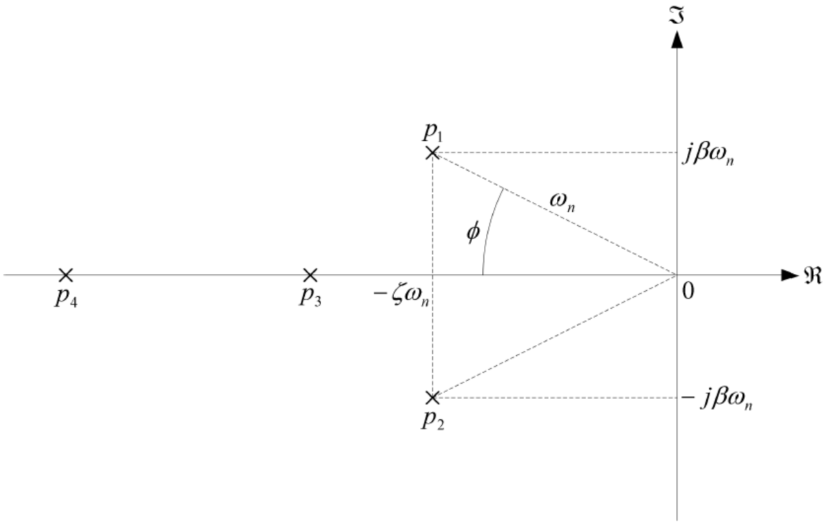


Figure 3 – Closed-Loop Pole positions in the s-plane (p_1 and p_2 are the dominant complex conjugate pair)

Write down in the space provided below, the complex conjugate pair of dominant poles.

$$5 \geqslant PO(\%) = 100e^{\left(-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \implies \zeta \geqslant 0.6901$$

$$\text{pick}: \; \zeta = 0.7071 = \frac{\sqrt{2}}{2}$$

$$t_s \leqslant 2.2\,s = \frac{4}{0.7071 \cdot \omega_n} \implies \omega_n \geqslant 2.5713$$
$$\text{pick}: \omega_n = 2.6$$

$$\omega_n^2 = \zeta^2\omega_n^2 + \beta^2\omega_n^2 \text{ then } \beta = \sqrt{1-\zeta^2}$$
$$for\ \zeta = 0.7071,\ \beta = 0.7071$$

$$p_1 = -1.8385 + j1.8385$$
$$p_2 = -1.8385 - j1.8385$$

Desired Dominant Poles:

$p_1 = -1.8385 + j1.8385$ $p_2 = -1.8385 - j1.8385$

When the non-dominant poles are added to the system, the overall closed-loop behavior will deviate to some small extend from that of the second order standard form. We will now select a possible pair of non-dominant poles, and check whether our design specifications are violated for the resulting state-feedback based closed-loop 4th order system. In order to do that, we will (a) use Matlab's `place()` function to design a state-feedback gain vector K, (b) determine the closed-loop system matrix $\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{BK}$ and (c) examine the resulting unit step response to check for whether the two design specifications are satisfied. (You can read $PO(\%)$ and t_s values from the Matlab step response plot). Note that the closed loop system parameters are $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$ with $\tilde{\mathbf{A}}$ given above and $\tilde{\mathbf{B}} = \mathbf{B}$, $\tilde{\mathbf{C}} = \mathbf{C}$.

Tabulate your trials below, until you find a suitable p_3 and p_4 .

p_3	p_4	$K = [\,k_1 \;\; k_2 \;\; k_3 \;\; k_4\,]$	$PO(\%)$	t_s
-2	-3	[1.17,-0.87,-6.00,-1.74]	0	2.7240
-3	-4	[2.35,-4.11,-4.98,-1.86]	1.6976	2.0316
-4	-5	[3.91,-7.75,-3.74,-1.89]	2.8060	2.6703
-2.5	-8	[3.91,-9.31,-3.45,-2.19]	1.3799	1.9801
-2.22	-20	[8.68,-26.23,1.42,-3.40]	0.8037	2.0169
-3+i	-3-i	[1.95,-2.69,-5.38,-1.75]	1.7768	2.0577

Discuss the feasibility of the K values you have obtained.

For more distant excess poles, overshoot increases. Generally larger overshoot means faster system, but when we observe the settling time, for higher overshoot settling time also increase since it also waits for oscillations to be damped.

7 In-Lab Experimental Procedure

In the beginning of the lab session, have your Teaching Assistant check your results and in particular, your closed loop pole locations.

Teaching Assistant Review Notes:	
Assistant Name:	Signature:

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 in order not to damage any of the components. If in doubt, refer to References [1], [2], [3], [4], and/or [5] or ask for help from your lab Teaching Assistant .

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 [1]. You should have become familiar with the complete wiring and connections of your IP02 system during the previous experiment. If you are still unsure of the wiring, please ask for assistance from the Teaching Assistant. When you are confident with your connections, you can power up the UPM. You are now ready to begin the lab. The pendulum that we will use in the lab should NOT initially be mounted on the setup. To ensure there is no damage to the pendulum and/or the encoder connection, it should only be mounted when the actual hardware experiment steps are conducted.

7.1.2 IP02 Configuration

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. For the time being, please carefully remove the extra weight if it is mounted on top of the cart.

7.2 *Simulink Simulation of the SPG+IP02 Plant and Pole-Placement Controller*

7.2.1 Objectives

- To implement, in a Simulink diagram, the open-loop model of the SPG-plus-IP02 system with full-state feedback,
- To investigate, by means of the simulation of the model, the performance of the compensated closed-loop system and the corresponding control effort as a result of the particular choices for the non-dominant pole locations p_3 and p_4.
- Refine/tune the locations of these two non-dominant poles, using the Simulink model, so as to meet the design specifications while respecting the system's physical limitations (for our case, the saturation limits for the motor voltage).

7.2.2 Simulink Model of the State-Feedback Controlled Closed-Loop System

In this part, the Simulink model named `s_spg_pp.mdl` is used to analyze and assess the performance of your pole-placement-based state-feedback controller design (the gain vector *K*). Please open, in your experiment directory, this Simulink model file, which should look like Figure 4. Spend some time to familiarize yourself with the model.

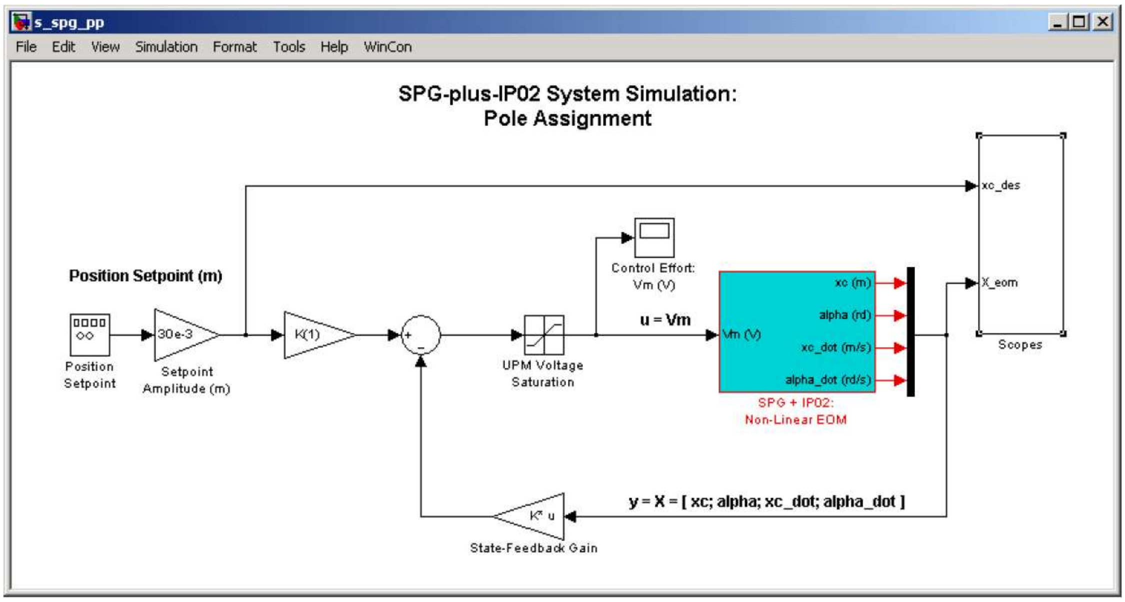


Figure 4 – Simulink diagram of the SPG+IP02 system with state-feedback controller.

This model will be used for running simulations in order to tune our state feedback-controller, in particular, the locations of the two non-dominant poles p_3 and p_4.

At this point, you may also wish to open the Simulink block corresponding to the non-linear model of the SPG+IP02 system and investigate its contents. Note that although we used the linear approximation of the system to design our controller, the simulation of the system uses the original non-linear model of the plant. This ensures that the simulation remains accurate for the full angular range of the pendulum rather than small angles only.

7.2.3 Experimental Procedure

Step 1. Before you begin, you must run the Matlab script called `setup_lab_ip02_spg.m`. This file initializes all the SPG+IP02 system parameters and user-defined configuration variables needed and used by the Simulink diagrams. The file also defines the **A**, **B**, **C** and **D** matrices that correspond to the linear approximation of the SPG+IP02 plant that you have studied in the preliminary work. After executing the m-file, check that these matrices, now defined in the Matlab workspace, correspond to the ones you have investigated in your preliminary.

Step 2. Bring the `s_spg_pp.mdl` model to the front. Check that the signal generator block properties are properly set to output a square wave signal of amplitude 1 and frequency 0.1 Hz. As a remark, the input indicated in Figure 4 (position setpoint) should be small enough in magnitude such that the “small angle” assumption that we have used to find the linear approximation to the system remains valid.

Take some time to familiarize yourself with this model. This simulation model will be used when we will fine-tune our controller. Note that although we have used the linear approximation to the plant for designing our pole-placement based controller, in the simulation, the full non-linear model of the plant (given in Equations (4) and (5)) is used instead. This model is represented in the Simulink model file `s_spg_pp.mdl` as the block `SPG+IP02:Nonlinear EOM`. This guarantees that the simulation remains valid for the entire range of pendulum angles. (This does not mean the controller will perform to the design specifications when pendulum angle violates the small-angle assumption). 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 states defined in Section 6.1.

Step 3. The setup function `setup_lab_ip02_spg.m` determines a proper set of gain values for this experiment. However, you will perform this simulation experiment with your own set of gain values determined in Section 6.4.2 of the preliminary work. Remember that given a set of poles, you can use the Matlab `place` function to determine the state-feedback gain vector *K* as in

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

Step 4. Set the *Stop Time* in the Simulink Simulation Parameters to 10 secs. You can now start the simulation of your model.

Step 5. After the Simulation completes, open the four scopes titled:

Scopes/Tip Horizontal Pos (mm),

Scopes/xc (mm),

Scopes/alpha (degree),

Control Effort Vm(V).

Does your system meet all the design requirements? Acceptable responses are shown below in Figure 5. Note that, the control effort is the voltage applied to the motor is bounded by the physical capabilities of the power amplifier. The Simulink block includes a saturation block to impose this requirement. You can check the block properties to see the saturation value. Your design (mostly determined by your p_3 and p_4 selection) should not saturate the power amplifier.

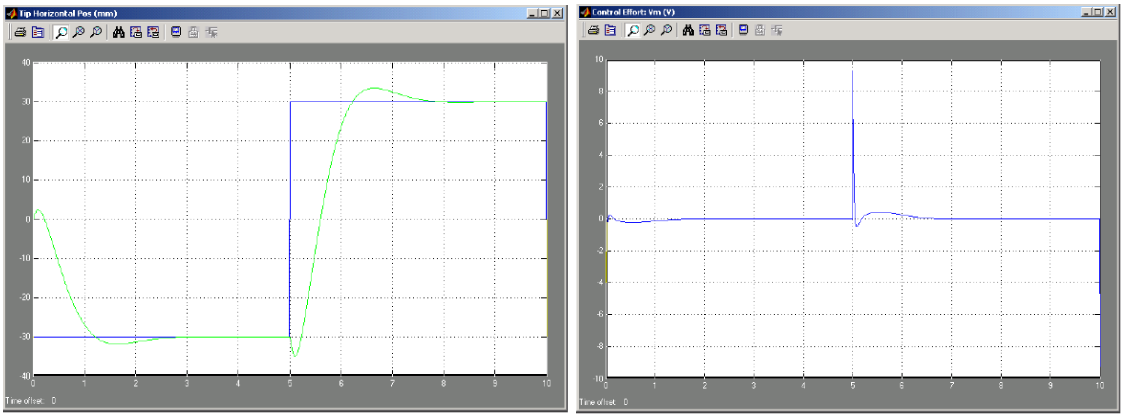


Figure 5 – Expected Tip Horizontal Pos (mm) and Control Effort Vm(V) plots.

Please comment on the possible negative effects of a saturating control effort on the performance of the overall system.

Step 6. Experiment with your p_3 and p_4 values using the full non-linear simulation model while observing the resulting response plots. Comment in detail about the effect of the choice of these non-dominant poles on the performance of the overall system (in relation to the design specifications).

Step 7. If your responses do not meet all the design specifications, you should re-iterate your location assignments of p_3 and p_4 and re-calculate the corresponding K as well as re-running the simulation. You will do this until the results are satisfactory. A trade-off is probably required between the response performance of x_t and the control effort required.

Note that even if you had obtained these pole values by similar trials at the end of Section 6.4.2, the presence of the full non-linear model may lead to slightly different results here.

Step 8. At this point, log your determined p_3 and p_4 pole locations as well as the feedback gain vector K below and have your lab assistant check them. You will use these values for the actual hardware experiments in the following sections. Real-Time Implementation of the State-Feedback Controller

7.3 *Objectives*

i) To implement with QuaRC areal-time state-feedback controller for your actual SPG+IP02 Servo Plant.
ii) To refine the chosen locations of the closed-loop poles so that the actual system meets the desired design specifications.
iii) To run the state-feedback closed-loop system simulation in parallel and simultaneously with the hardware setup, at every sampling period, in order to compare the actual and simulated responses.
iv) To eliminate any steady-state error present in the actual responses by introducing an integral control action.
v) To tune, on-the-fly the integral gain K_i.
vi) To investigate the effect of partial state feedback on the closed-loop responses.

7.3.1 Experimental Procedure

Through the previous closed-loop simulation on the placement procedure of the closed-loop poles for your SPG+IP02 Servo Plant, you have gained insight into the operation of the system 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 effects on your actual linear-cart-suspended-pendulum (SPG+IP02) system. To achieve this, please follow the experimental steps below:

Step 1. Open the Simulink model file `q_spg_pp_ip02.mdl`. You should obtain a view similar to the one shown in Figure 6. The model has two parallel and independent control loops: One runs a pure simulation of the state-feedback controller+SPG+IP02 system, using the plant's state-space representation. Since full-state-feedback is used, ensure that the output matrix **C** for the SPG+IP02 state-space model is a 4x4 identity matrix. (If you wish, you may use `C=eye(4)` to generate such a matrix.) The other loop directly interfaces with your hardware and runs you 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 sub-systems and investigate their contents. While doing this, also take note of the Input-Output (IO) connections of the corresponding blocks.

Step 2. Check that the position reference waveform generated for the cart and pendulum tip x-coordinate is a square-wave of amplitude 30mm and frequency 0.1 Hz. Lastly, your model sampling time should be set to 1 ms, that is T_s=10^{-3}.

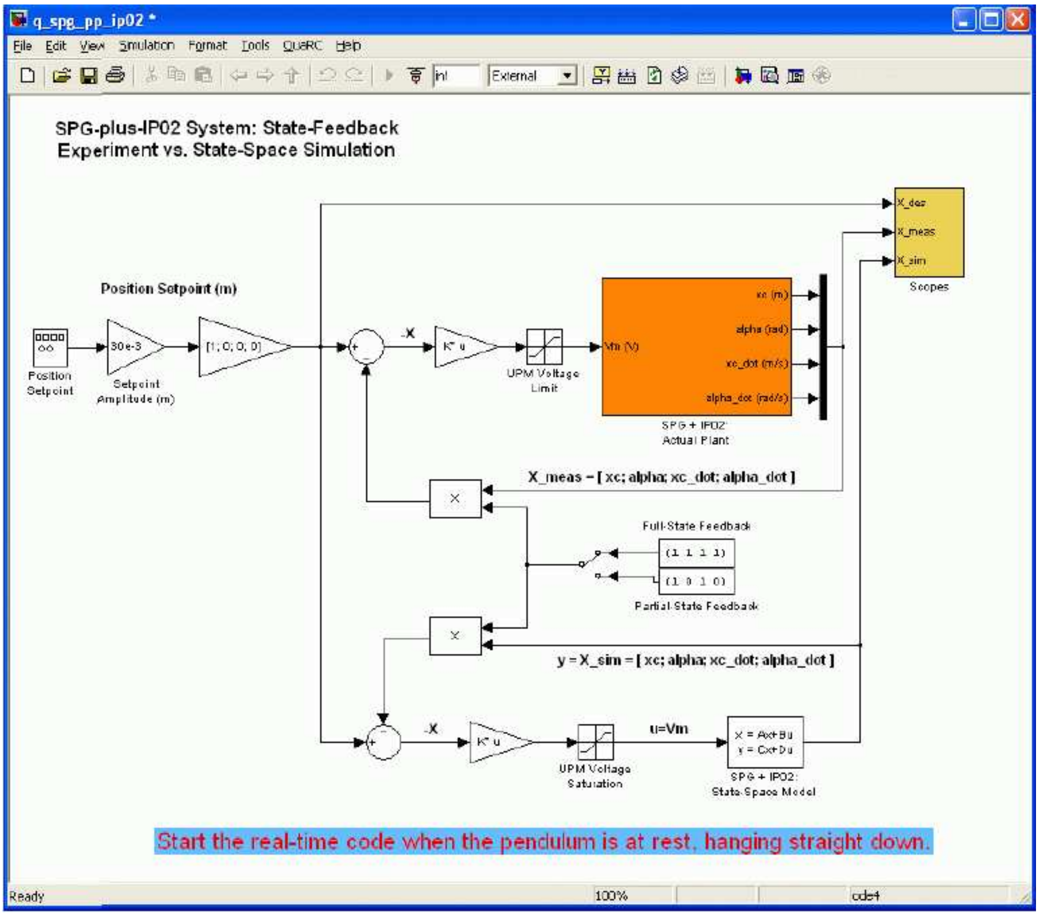


Figure 6 – Illustration of the Simulink model used to compare simulation-based and actual hardware implementation of the state-feedback controller.

Step 3. **Configure DAQ**: Double-click on the `HIL-Initialize` block inside the SPG+IP02 Actual Plant\IP02 sub-system and ensure it is configured for the DAQ device that is installed in your system (Most likely "Q4" in our case). See Reference [6] for more information.

Step 4. Ensure that your feedback gain vector K satisfying the system performance specifications as determined in the previous section's simulations is still set in the Matlab workspace (you may use `whos` command or just type `K` to the workspace). If not, re-initialize K to correspond to the vector that you have found in your work.

Step 5. You are now ready to build the real-time code corresponding to your diagram, by using the `QuaRC | Build` option from the Simulink menu-bar. After successful compilation and download (to the execution directory), you should be able 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 6. **Single Pendulum Gantry (SPG) 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) at this "home" configuration.

Note: Incremental sensors such as encoder counters always need an initial position where their counters are reset. They also may accumulate errors over time and may also need occasional repetition of this initialization procedure.

Step 7. Position the IP02 cart around the mid-point of the track and wait for the suspended pendulum to come to complete rest. Ensure that the system is free to move within its sideways motion range without anything, including connecting cables limiting its motion.

Step 8. Start the real-time controller by clicking on `QuaRC | Start` from the Simulink Model menu-bar. Your IP02 cart position should now be tracking the desired square-wave reference signal while minimizing the swing of the suspended pendulum. (Note: Whenever you are ready to log your observation in the subsequent steps, do not forget to stop the controller by clicking the Stop button in the Simulink model toolbar (or by selecting `QuaRC | Stop` menu item). This will prevent any accidental damage to the hardware)

Step 9. Open the `Scopes/Pend_Tip_Pos_(mm)` sink. For more insight on your actual system's behavior, also open the two sinks named `Scopes/xc_(mm)` and `Scopes/Pend_Angle_(deg)`. 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 in the sub-system `SPG+IP02: Actual Plant\IP02`. On the `Pend_Tip_Pos_(mm)` scope, you should now be able to monitor online, as the cart and pendulum move, the actual pendulum tip position as it tracks your pre-defined reference values defined by the square-wave reference signal. Compare the actual plant system output with the response generated by the `SPG+IP02 State-Space Model` simulation model. Such a response is shown in Figure 7.

Step 10. 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 e_{ss}. You can verify your design by also observing the `Scopes/xc_(mm)` and `Scopes/Pend_Angle_(deg)` scopes. Do not forget to also review the corresponding control effort, by monitoring the `V_Command_(V)` scope. Does your system meet all the design specifications? In the space provided, specify the observed values of *PO*(%), *PU*(%), t_s, and steady-state error e_{ss}. Also sketch a plot of the content of your `Scopes/Pend_Tip_Pos_(mm)` scope, indicating these criticial values.

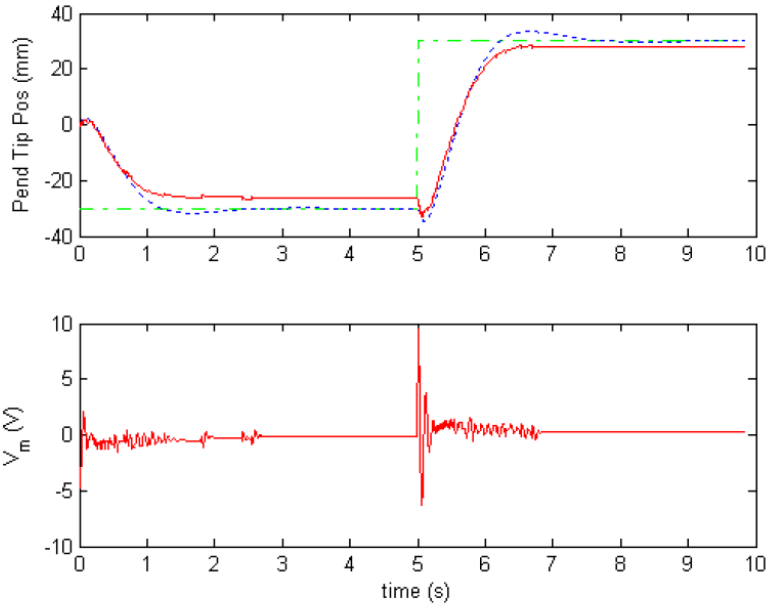


Figure 7 – Actual and simulated response of the state-feedback controlled system. In the top plot, the green dash-dot line is the reference waveform, the blue dot line is the simulation, and the solid-red line is the measured pendulum position. The bottom plot is the motor input voltage (i.e., the control effort).

Step 11. Try to achieve the design specifications as closely as possible by refining the location of the two non-dominant real poles, p_3 and p_4. Make sure to re-calculate (using the Matlab 'place' function) your gain vector K and to re-apply it to your real-time code. Comment on your performance in attaining the design specifications.

Step 12. Did you observe, in the actual system output response, a non-zero steady-state error? How about the response of the simulated system? Report your observations.

Comment on possible reasons for your observations. Can you think of any improvement on the closed-loop scheme in order to reduce, or eliminate, that steady-state error for the actual hardware system output?

Step 13. 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 that you have most likely observed in your output response for the actual system. Such an integrator on x_c has already been implemented for you in a new Simulink model. Open the Simulink model file q_spg_pp_I_ip02.mdl. You should obtain a model view that is similar to the one shown in Figure 8. Apart from the added integrator loop, this model should exactly be the same as your previous model, including the IO connections.

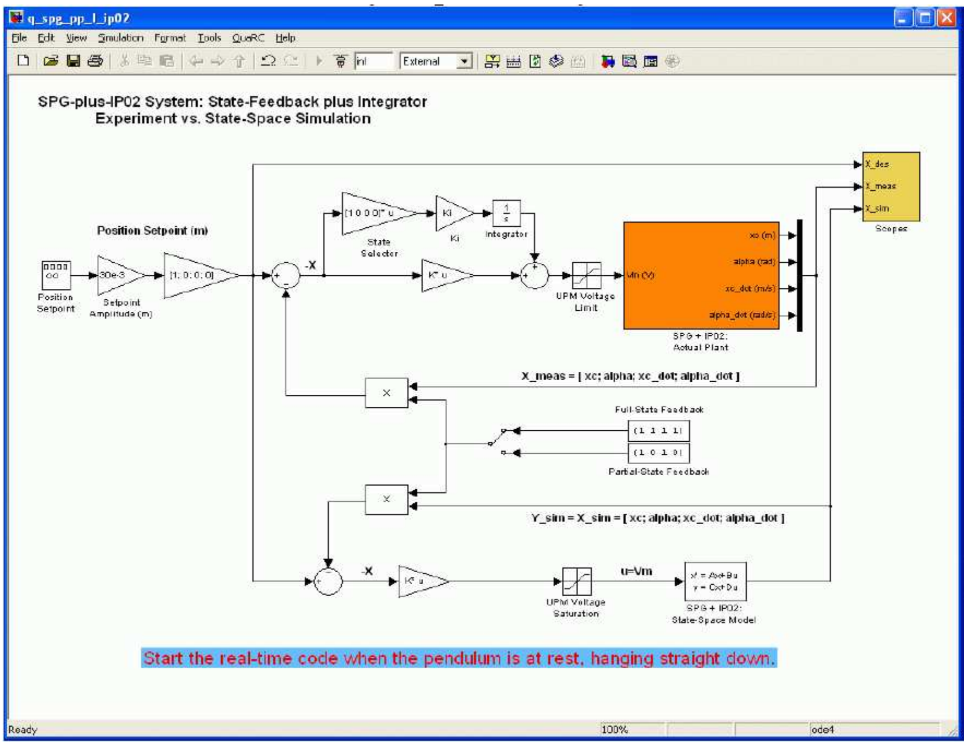


Figure 8 – Illustration of the Simulink model of the state feedback controller with added integral action for the actual hardware side.

Step 14. The integral gain is named `Ki`. First set `Ki` 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 Ki (by slowly increasing it to a positive value) to eliminate the steady-state error. (Hint: If you update the Ki value from the Matlab workspace, you need to select `Edit | Update Diagram` menu option in Simulink model window in order fr the change to take effect. Alternatively, you may change the value directly inside the model component properties for the gain block.)

Warning: You have to be careful about increasing `Ki` as it also adversely affects the stability of the system. An unstable hardware system may damage one or more of the system components.

Step 15. Did you achieve zero-steady-state error? Include your value of `Ki` in your lab report.

Step 16. Also ensure that the actual commanded motor input voltage Vm (which is proportional to the actual control effort produced) does not go into saturation. Ideally, no sign of saturation should be seen on the V Command (V) scope if the locations of the non-dominant poles are properly chosen. Comment on what you observe in your case.

Step 17. Now that you have an integrator in your system, its transfer function is modified, and system performance may have changed in return. Refine your two real poles' positions, p3 and p4, as necessary so that your system again meets the original performance specifications as closely as possible. Iterate your manual tuning as many times as necessary. If you are still unable to achieve the required performance level, ask your T.A. for advice.

Step 18. After your tuning efforts, record the final performance parameters of the actual controlled hardware system. Could you achieve desired design specifications?

Step 19. Also comment on the most notable differences between the simulated system and actual system responses as observed in all the different plots (Pendulum tip position, pendulum angle, control effort etc.)

Step 20. As a final experiment in this section, you will experiment with losing some of the feedback components from the state vector (resulting in a partial-state feedback control). For this you will use the manual switch located at around the center of your Simulink model diagram. Move the switch from the "Up" to the "Down" position, which multiplies the two pendulum states (*angular position* α and *angular rate* $\dot{\alpha}$) by zero, effectively canceling their effect in the feedback loop. This may correspond for example to an unexpected sensor failure in the system. It is quite important to gain an insight about the resulting system behavior.

Step 21. Start the real-time controller, observe the effect of canceling the state components corresponding to the pendulum state on the output performance of the controller. In the space provided, sketch the output (pendulum tip position) of the simulation and actual system as seen on the response plots on your screen. On a separate figure, also sketch the control effort applied to the plant.

7.4 Assessment of the System's Disturbance Rejection

This part of the experiment will give you some basic insights on the disturbance rejection behavior of the designed state-feedback based controller.

7.4.1 Objectives

i. To observe and investigate the disturbance response of the stabilized suspended-pendulum-linear-cart system, in response to a gentle tap to the pendulum.

ii. To study the full-state-feedback effectiveness in stabilizing the swing of the suspended pendulum.

7.4.2 Experimental Procedure

Step 1. Start and run your suspended-pendulum-linear-cart system around the mid-track position, again making sure that the pendulum is at complete rest before execution of the controller. Use the same full-state feedback controller with integral action which you have developed in the previous section. However, this time, set the cart position reference signal to zero amplitude so that the controller now regulates both cart position and pendulum angle around the zero state. This is the regulation configuration (i.e., there is no tracking of a reference signal).

Step 2. Gently tap the suspended pendulum (in the direction parallel to the track). Do not apply a tap of more than about 20˚ from its equilibrium (i.e., the vertical position). Observe the response of the linear cart (x_c) and its effect on the pendulum angle (α) in the corresponding scopes. In addition, open the scopes and observe the pendulum tip position (x_t) and the corresponding control effort (V_m). Sketch these plots in the space provided.

Appendix A. Nomenclature

Table A.1, below, provides a complete listing of the symbols and notations used in the IP02mathematical modeling, as presented in this laboratory. The numerical values of the system parameters can be found in the IP02 User Manual.

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
$ç_m$	Motor Efficiency	Eff_m
K_m	Back-Electro Motive-Force (EMF) Constant	Km
E_{emf}	Back-EMF Voltage	Eemf
J_m	Rotor Moment of Inertia	Jm
K_g	Planetary Gearbox Gear Ratio	Kg
$ç_g$	Planetary Gearbox Efficiency	Eff_g
M_{c2}	IP02 Cart Mass (Cart Alone)	Mc2
M_w	IP02 Cart Weight Mass	Mw
M	Total Mass of the Cart System (i.e. moving parts)	M
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 modeling of the single suspended pendulum. The numerical values of the pendulum system parameters can be found in the SPG User Manual.

Symbol	*Description*	*Matlab / Simulink Notation*
α	Pendulum Angle From the Hanging Down Position	alpha
$\frac{\partial}{\partial t}\alpha$	Pendulum Angular Velocity	alpha_dot
α_0	Motor Armature Resistance	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	I_p
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
ζ	Damping Ratio	zeta
w_n	Undamped Natural Frequency	wn
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. Derivation of 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 frame definition, illustrated in Figure 2, the absolute Cartesian coordinates of the pendulum's center 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 is a system is the amount of 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 is only due to gravity. The cart linear motion is horizontal, and as such, never has vertical displacement. Therefore, the total potential energy if 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:

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

Mass of the single pendulum is assumed to be concentrated at its Center Of Gravity (COG). Therefore, the pendulum's translational kinetic energy, T_{pt}, can be expressed as a function of its center of gravity's linear velocity, as shown by the following equation:

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

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

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

and the linear velocity's x-coordinate of the pendulum's center of gravity is expressed by:

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

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

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

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:

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

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 equation for our system. By definition, the two Lagrange's equations, resulting from the previously-defined two generalized coordinated, x_c and α, have the following formal formulations:

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

and:

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

The first term of Equation [B.11] means taking derivative of L with respect to $\left(\frac{d}{dt} x_c(t) \right)$ and then taking derivative of the result with respect to time (t).

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

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

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

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

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:

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

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

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

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:

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

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

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

Equations [B.17] and [B.18] use force applied on cart, F_c. However, IP02 block in the laboratory takes motor voltage, V_m, as the input. But one can easily find the transform between F_c and V_m using the block diagram of IP02 setup. It can be expressed as:

$$F_c(t) = -\frac{\eta_g\, K_g^2\, \eta_m\, K_t\, K_m \left(\frac{d}{dt} x_c(t)\right)}{R_m\, r_{mp}^2} + \frac{\eta_g\, K_g\, \eta_m\, K_t\, V_m(t)}{R_m\, r_{mp}} \qquad \text{[B.19]}$$

Equations [B.17] and [B.18] should be reconfigured by integrating Equation [B.19]. So the final form of the EOMs can be written as:

$$\frac{d^2}{dt^2} x_c(t) = \frac{\left(\begin{array}{l} -\left(I_p + M_p\, l_p^2\right) B_{eq} \left(\frac{d}{dt} x_c(t)\right) + \left(M_p^2\, l_p^3 + I_p\, M_p\, l_p\right) \sin(\alpha(t)) \left(\frac{d}{dt} \alpha(t)\right)^2 \ldots \\ +\left(I_p + M_p\, l_p^2\right) \left(-\frac{\eta_g\, K_g^2\, \eta_m\, K_t\, K_m \left(\frac{d}{dt} x_c(t)\right)}{R_m\, r_{mp}^2} + \frac{\eta_g\, K_g\, \eta_m\, K_t\, V_m(t)}{R_m\, r_{mp}} \right) \ldots \\ +M_p\, l_p \cos(\alpha(t)) B_p \left(\frac{d}{dt} \alpha(t)\right) + M_p^2\, l_p^2\, g \cos(\alpha(t)) \sin(\alpha(t)) \end{array} \right)}{\left(\left(M_c + M_p\right) I_p + M_c\, M_p\, l_p^2 + M_p^2\, l_p^2 \sin(\alpha(t))^2\right)}$$

[B.20]

$$\frac{d^2}{dt^2} \alpha(t) = \frac{\left(\begin{array}{l} -\left(M_c + M_p\right) M_p\, g\, l_p \sin(\alpha(t)) - \left(M_c + M_p\right) B_p \left(\frac{d}{dt} \alpha(t)\right) \ldots \\ -\left(-\frac{\eta_g\, K_g^2\, \eta_m\, K_t\, K_m \left(\frac{d}{dt} x_c(t)\right)}{R_m\, r_{mp}^2} + \frac{\eta_g\, K_g\, \eta_m\, K_t\, V_m(t)}{R_m\, r_{mp}} \right) M_p\, l_p \cos(\alpha(t)) \ldots \\ -M_p^2\, l_p^2 \sin(\alpha(t)) \cos(\alpha(t)) \left(\frac{d}{dt} \alpha(t)\right)^2 + M_p\, l_p \cos(\alpha(t)) B_{eq} \left(\frac{d}{dt} x_c(t)\right) \end{array} \right)}{\left(\left(M_c + M_p\right) I_p + M_c\, M_p\, l_p^2 + M_p^2\, l_p^2 \sin(\alpha(t))^2\right)}$$

[B.21]

As a final remark on derivations, the single suspended pendulum's moment of inertia about its center of gravity is characterized by:

$$I_p = \frac{1}{12} M_p\, L_p^2 \qquad \text{[B.22]}$$

8 Report

Names, Student IDs and Signatures of Group Students:

1) ____

____

2) ____

____

Date of Experiment:

Name of Lab Assistant:

7.2 Simulink Simulation of the SPG+IP02 Plant and Pole-Placement Controller
Step 5:

Step 6:

Step 8:

Non-Dominant Pole Locations p_3 = p_4 =

Feedback Gain Vector K = [ ]

Measured Values of:

PO(%) =

t_s =

7.3 Real-Time Implementation of the State-Feedback Controller

Step 10:

Actual (Hardware) Performance
PO(%) = *PU(%)* = t_s = e_{ss} =
Sketch of the actual system response

Step 11:

Step 12:

Step 15:

Ki =

Step 16:

Step 18:

$PO(\%) =$

$PU(\%) =$

$t_s \quad =$

$e_{ss} \quad =$

Comments:

Step 19:

Step 21:

Output sketch:

Control effort sketch:

7.4 Assessment of the System's Disturbance Rejection

Step 2:

x_c
α
x_t
V_m

9 Knowledge Test

1. Explain how linearized A, B and C matrices of the nonlinear system of this laboratory (SPG) are derived. Also give the generic formulation.

2. What does controllability mean? Explain the calculation of controllability matrix (C_0). What is the criterion on C_0 for a controllable system?

3. Explain how the poles of a system can be extracted from state space representation parameters.

4. Explain the aim of the "Pole Placement Problem". Draw the diagram (control loop) illustrating this problem.

5. If you did not have Matlab's `place()` function, how could you find the K matrix of the "Pole Placement Problem". Explain the procedure.