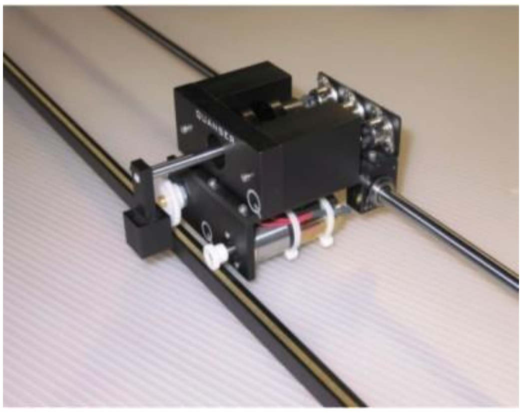
Middle East Technical University
Department of Electrical and Electronics Engineering

EE 406
Laboratory of Feedback Control Systems



Experiment #2:
Proportional-Velocity Position Control,

Preliminary Work and Laboratory Manual

Initial Version: Afşar Saranlı, Emre Tuna

Group Members:

1. Objectives

In this laboratory session, you will become familiar with the fundamentals of control system design using PID-type compensators. The challenge of the present lab is to control the *position* of your IP02 linear motion servo plant with specified control performance.

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

- How to mathematically model the IP02 servo plants from first principles in order to obtain the open-loop transfer function, in the Laplace domain.
- How to design and simulate a Proportional-Derivative (PD) position controller in the alternative Proportional-Velocity (PV) form to meet the required design specifications.
- How to tune your PV controller gains and observe their effect on the closed-loop system dynamic response through system simulation.
- How to implement your controller in real-time on actual hardware and evaluate its performance.

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 power amplifier (e.g., UPM), and your data acquisition card (e.g., Q8 or Q2), as described in References [1], [2], [3], and [4].
ii) To have successfully completed the Experiment #1. The previous laboratory should have given you some of this background. Students are therefore expected to be familiar in using QuaRC to control and monitor the plant in real-time and in designing their controller through Simulink.
iii) To be familiar with the complete wiring of your IP02 servo plant, as per dictated in Reference [1].

3. References

[1] *Quanser IP02 User Manual*.
[2] *Quanser Q2 USB Data Acquisition Device User Manual*.
[3] *Universal Power Module UPM User Manual.*
[4] *QuaRC User Manual* (type `doc quarc` in Matlab to access in the laboratory computers).
[5] *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 UPM 2405, VoltPAQ-X1
- **Data Acquisition Board:** Quanser Q8 and Q2
- **Linear Motion Servo Plant:** Quanser IP02, as shown in Figure 1 .
- **Real Time 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 will be mostly prepared for you. Your desk setup will have 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.

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 [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.

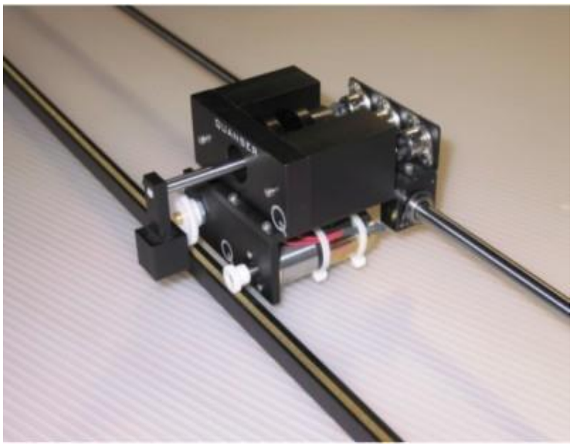


Figure 1 – The IP02 Servo Plant

5. Controller Design Specifications

In the present laboratory (i.e., during the preliminary work and lab sessions), you will design and implement a control strategy based on the Proportional-Velocity (PV) control scheme, a modified implementation of the Proportional-Derivative (PD) control approach. PD, PI and PID controllers selectively using Proportional, Derivative, and Integral terms are widely used in industrial applications for control systems. Your controller will be designed such that your controlled IP02 closed-loop system will satisfy the following time-domain performance requirements:

i) The Percent Overshoot (i.e., *PO*) should be less than 10%, i.e.:

$$PO \leq 10\ \%$$

ii) The time to first peak should be 150 ms, i.e.:

$$t_p = 0.15\ \text{s}$$

You should print out preliminary work separately from other parts.

Student Name:
Student ID:

6. Preliminary Work

6.1 Assignment #1: Open-Loop Model Block Diagram

In this part of the preliminary work, you will derive the plant transfer function of the IP02 servo plant from electrical and mechanical first principles. You will first build a model block diagram by considering each sub-component of the electro-mechanical system. Consider your EE302 training to model each component from the input to the output and gradually build the full, detailed plant block diagram. Do not simplify at this point.

Appendix A: Nomenclature gives you the names of all variables that you need to consider for this model. Note that you will need almost all the variables given inthe Nomenclature, but not all of them. A few of those parameters are not used in this experiment.

(a) Draw the schematic of the electro-mechanical system, showing all electrical and mechanical components, interconnections, and all relevant variables. (E.g., the motor circuit, gears, cart mass etc.). Provide this schematic in the box provided in Figure 3.

(b) Cleanly draw the detailed block diagram of the servo-plant in the space provided on the next page. To guide you, the skeleton of the block diagram with signal variables is given in the Figure 4.

Note:
As a reminder, your IP02 open-loop transfer function is defined by the selected plant input and plant output. As illustrated in Figure 2, the plant input is the commanded voltage to the DC motor. Since in this laboratory we want to control the cart's position, the plant output is selected as the cart linear position on the rack, as depicted in Figure 2.

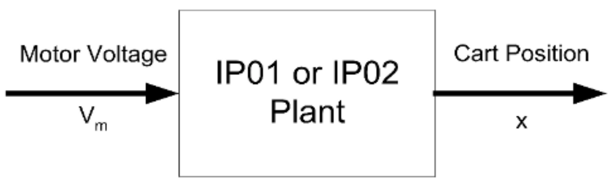


Figure 2 – The IP02 Plant Input and Output

6

Figure 3 – The IP02 Servo **Plant Schematics (Use the page in Landscape orientation)**

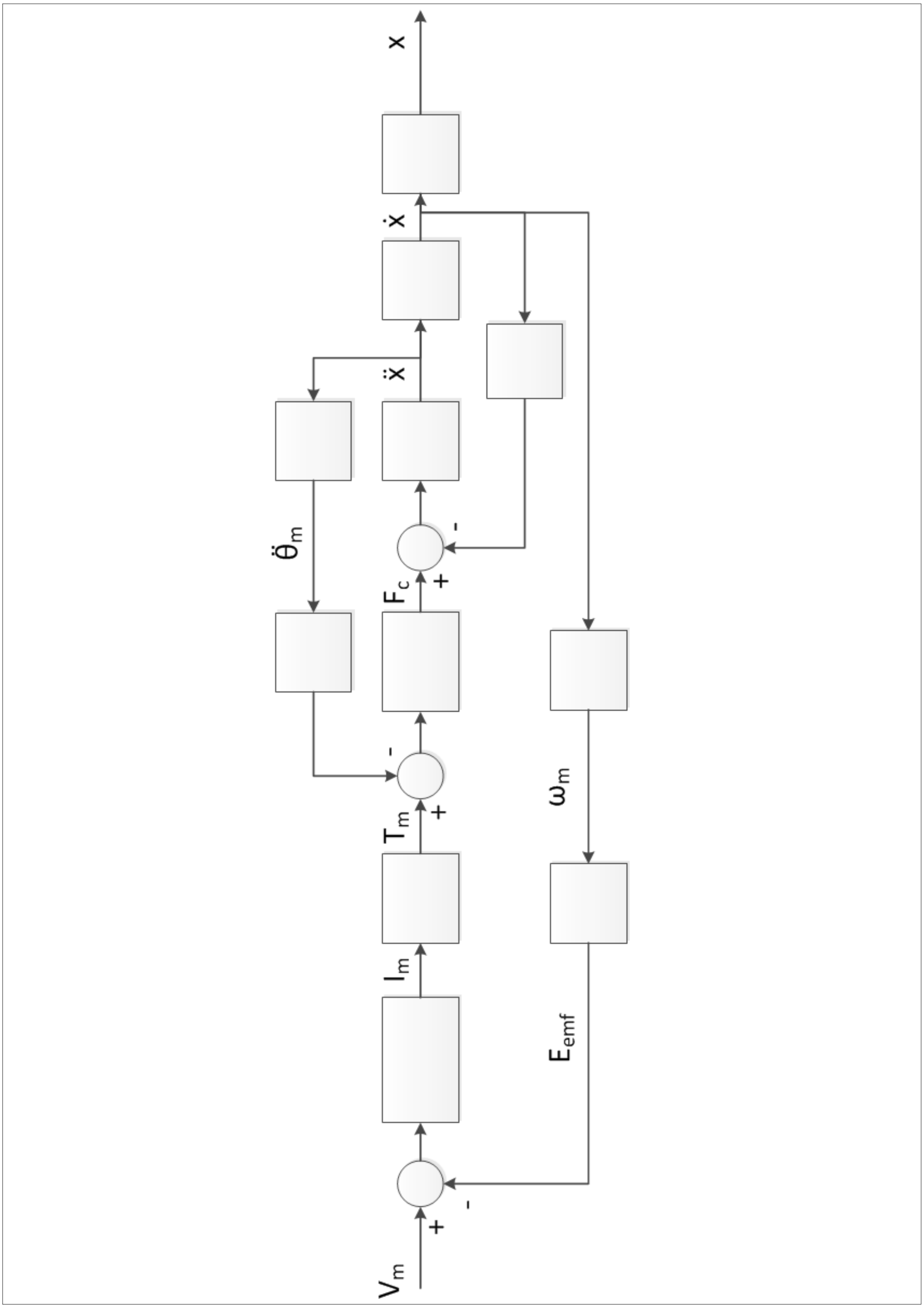


Figure 4 – The IP02 Servo Plant Detailed Block Diagram

(Use the page in Landscape orientation)

6.2. Assignment #2: Open-Loop Transfer Function

Now that you have a detailed block diagram of the open-loop plant, obtain the overall transfer function from the input to the output. This part is very important, because you will be using the same mechanical setup throughout the semester. If you understand the system well and can derive the model correctly you will use only the results of this section (your model) in the remaining experiments. Remember that the transfer function *G(s)* is given by:

$$G(s)=\frac{X(s)}{V_m(s)} \tag{1}$$

(a) Write down this overall open-loop plant transfer function in the box provided below in terms of all necessary variables defined in Appendix A. Briefly show your derivation. Do not create mid-variables that will simplify the equations, use only the given symbols. Be as clear as possible. Final transfer function should be in a nice form, i.e. (As^2+Bs+C)/(Ds^3+Es^2+Fs+G), capital letters being mathematical combination of given symbols. Also note that this is only an example, system transfer function does not necessarily have 2 zeros and 3 poles. If necessary, do your first derivations on a scrap paper and transfer only clear steps in this sheet.

(b) Now, refer to the IP02 User Manual and use the numerical values for all the parameters for this plant to actually evaluate the open-loop plant transfer function. The viscous damping coefficient of the cart, $\mathbf{B_{eq}}$ is not given in the calculations, **take as 4.3 N.s/m**. By noting that the motor inductance is much smaller than motor internal resistance, you may from this point on, assume that $\boldsymbol{L_m=0}$. This reduces the order of the system by one and will help us with the following analysis. Write down the evaluated G(s) below. Write the final result in the **monic** form. Consider that the system does not have extra mass, i.e., mass of the cart (M_{c2}) is your only mass.

(c) Now, determine the open-loop poles, zeros, and the DC gain of the plant. Writethem down in the box provided below.

Zeros:

Poles:

DC Gain:

(d) Now repeat part b with extra mass, i.e., consider the mass of the system as (M_{c2} + M_w). You will observe the effect of extra mass in the experiment.

6.3. Assignment #3: PV Controller Design

This section deals with the design of a closed-loop controller in order to control the position of your IP02 in a quick and accurate manner (namely to satisfy the design requirements for a controlled system).

6.3.1. Standard Closed-Loop System

Figure 5, below, depicts a standard closed-loop position control system with a feed-back loop:

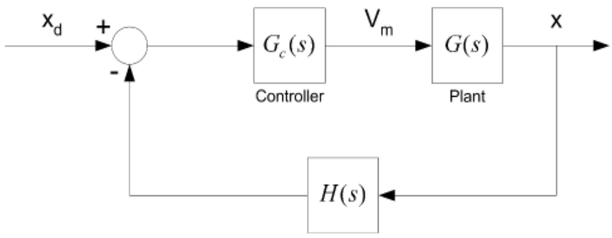


Figure 5 – Standard closed-loop position control system.

For such a closed-loop system, as represented in Figure 5, the closed-loop transfer function, *T(s)*, is given by the following well-established equation:

$$\frac{x(s)}{x_d(s)} = \frac{G_c(s)\,G(s)}{1+G_c(s)G(s)H(s)} \qquad (2)$$

For our case, we assume a unity feedback loop, therefore $H(s) = 1$. Your work for the previous question should have revealed that the plant has no zeros and two poles, i.e., a second order denominator in s. Moreover, in order to design controllers satisfying given performance requirements, Control Theory provides approximate design formulas. These formulas are based on a standard system transfer function form, namely for quadratic systems with no zeros. This standard form is given by the following equation:

$$T(s) = \frac{K_{dc} w_n^2}{s^2 + 2\zeta w_n s + w_n^2} \qquad (3)$$

where K_{dc} is the system's DC gain, w_n is the natural frequency and ζ is the damping ratio.

The characteristic equation of the closed-loop transfer function expressed in its standard form by Equation (3) is given as:

$$s^2 + 2\zeta w_n s + w_n^2 \ . \qquad (4)$$

6.3.2. Proportional-Derivative (PD) Control Scheme

In the classical sense, a Proportional-Derivative (PD) controller has the transfer function:

$$G_c(s) = K_p + K_d s \qquad (5)$$

As expressed by Equation (5), placing such a controller into the forward path would result in introducing a zero in the closed-loop transfer function. As a result of introducing this zero, the closed-loop transfer function would no longer match the standard form of Equation (3). Therefore, the design formulae derived from Equation (3) would also no longer exactly apply to the thus obtained closed-loop transfer function, and it would become more challenging to analytically design a controller that can exactly meet the user-defined time specifications.

Question: Derive error signal, E(s), for this PD Control Scheme. Rules in Assignment #2, part a, also apply here except the fact that you will be using numerical results found in part b of the same assignment as the transfer function. The only parameters are Kp, Kd and input signal Xd(s).

Question: In our case, adding an integral term (I) to the forward path does not have to be considered for achieving zero steady-state error for unit step input. Briefly explain why?

6.3.3. Proportional-Velocity (PV) Control Scheme

To work around the "undesired" zero introduced by a PD controller, this laboratory involves designing a Proportional-Velocity (PV) position controller for the IP02 servo plant. Such a controller introduces two corrective terms: one is proportional (by K_p) to the position error and the other is proportional (by K_v) to the velocity (or the derivative of the position) of the plant. Coincidentally, the characteristic equations of the PV and PD controller closed loop transfer functions are equal.

Equation (6), below, expresses the PV control law, where x_d is the reference signal (i.e., the desired position to track):

$$V_m(t) = K_p\big(x_d(t) - x(t)\big) - K_v\big(\frac{d}{dt}x(t)\big) \tag{6}$$

Figure 6, below, depicts the block diagram of the PV control scheme, as it will be implemented in this lab:

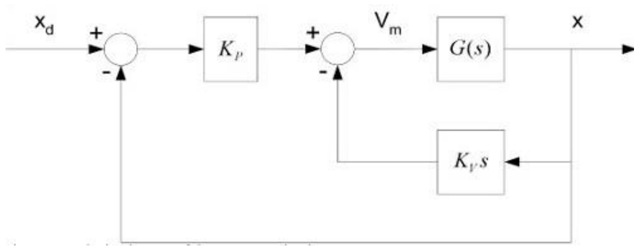


Figure 6 – Block diagram of the PV Control Scheme

In order to determine and calculate K_p and K_v, to design the controller to the given specifications, answer the following questions:

a) Perform block diagram reduction on Figure 6 to combine the PV control scheme with the plant transfer function $G(s)$ that you have already obtained. Obtain the overall closed-loop transfer function $T(s)$ of your controlled IP02 system. Write it down below. Briefly show your derivation. Make your derivation and result as clear as possible. (See the guideline in assignment #2, part a)

b) Extract from the previously obtained closed-loop transfer function the system's characteristic equation.

c) Fit the obtained characteristic equation to the standard form given in Equation (4), by identifying the parameters ω_n and ζ. Thus, you should obtain 2 equations expressing ω_n and ζ as functions of K_p and K_v as these are the only 2 variables (i.e. controller parameters) in your system.

d) Using your newly obtained formulae and referring to your EE302 notes, what changes to your IP02 response would you expect to see by varying the values of K_p and K_v? Keep your answers simple (i.e., will ω_n and ζ increase or decrease?). How would this translate in terms of changes in t_p and Percent Overshoot (*PO*)? Also relate these changes to the physical behavior of your closed-loop system. *Hint:* You can use the equations given in the next subsection. Specifically, answer the following:

   i. Assuming K_v constant, what happens to ω_n and ζ when you increase/decrease K_p?
   ii. Assuming K_p constant, what happens to ω_n and ζ when you increase/decrease K_v?

   Clearly show your work and give the equations to support your answers.

e) Using the formulae previously obtained, determine the analytical expressions and numerical values for K_p and K_v in order to meet the previously specified time requirements. The following are provided as a reminder of your EE302 material.

   i. Percent Overshoot:

$$PO = 100 e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{7}$$

   ii. Peak time:

$$t_p = \frac{\pi}{w_n\sqrt{(1-\zeta^2)}} \tag{8}$$

Clearly show your derivations.

Kp:

Kv:

f) Briefly repeat the PV controller design for the IP02 system with extra mass. This time, you don't need to show every derivation in detail. Just emphasize what changes when the system transfer function change, show related calculation steps referring to your detailed design in the previous steps and numerically calculate Kp & Kv for the new heavier system.

Kp for the heavy cart:

Kv for the heavy cart:

This is the end of your preliminary work for the laboratory. Examine the following material to get an insight about what you will perform during the lab experiment.

7. In-Lab Experimental Procedure

In the beginning of the lab session, have your Teaching Assistant check your results and in particular, your controller parameters K_p and K_v. If there is serious disagreement with the expected values, your Assistant will guide you how to proceed.

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.

Teaching Assistant Review Notes: (Transfer Function and Design check)
Assistant Name: Signature:

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.

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. As an extension to the lab, the first PV controller design could be modified in order to account for the additional weight. For the time being, please carefully remove the extra weight if it is mounted on top of the cart.

7.2. Closed-Loop System Actual Requirements

As already stated in the preliminary work, this lab requires you to design a Proportional-Velocity (PV) controller to control the position of your IP02 cart with the following performance specifications:

i. The Percent Overshoot should be equal to 10 %: *PO* = 10 %, i.e., $\zeta = 0.59$.
ii. The peak time should be 150 ms: $t_p = 0.15$ s

These specifications are the same as the ones you previously used in the preliminary work to calculate the corresponding PV controller gains K_p and K_v.

7.3. Simulation of the Servo Plant with PV Controller

7.3.1. Objectives

- To simulate with a Simulink diagram your IP02 model and to close the servo loop by implementing a Proportional-Velocity (PV) position controller.
- To change, during the simulation, the two gains, K_p and K_v, of the PV controller and observe the effect on the position response.

7.3.2. Experimental Procedure

If you have not done so yet, you can start-up Matlab and then Simulink now and follow the steps described below:

Step 1. In Simulink, open a model called *s_position_pv_ip01_2.mdl*. This diagram should be similar to the one shown in Figure 7. It includes a subsystem containing your IP02 modeled plant, as well as the PV controller two feedback loops. In order to be conveniently changed on-the-fly, the two controller gains K_p and K_v are both set by slider gains. Check that the signal generator block properties are properly set to output a square wave signal, of amplitude 1 and of frequency 2/3 Hz.

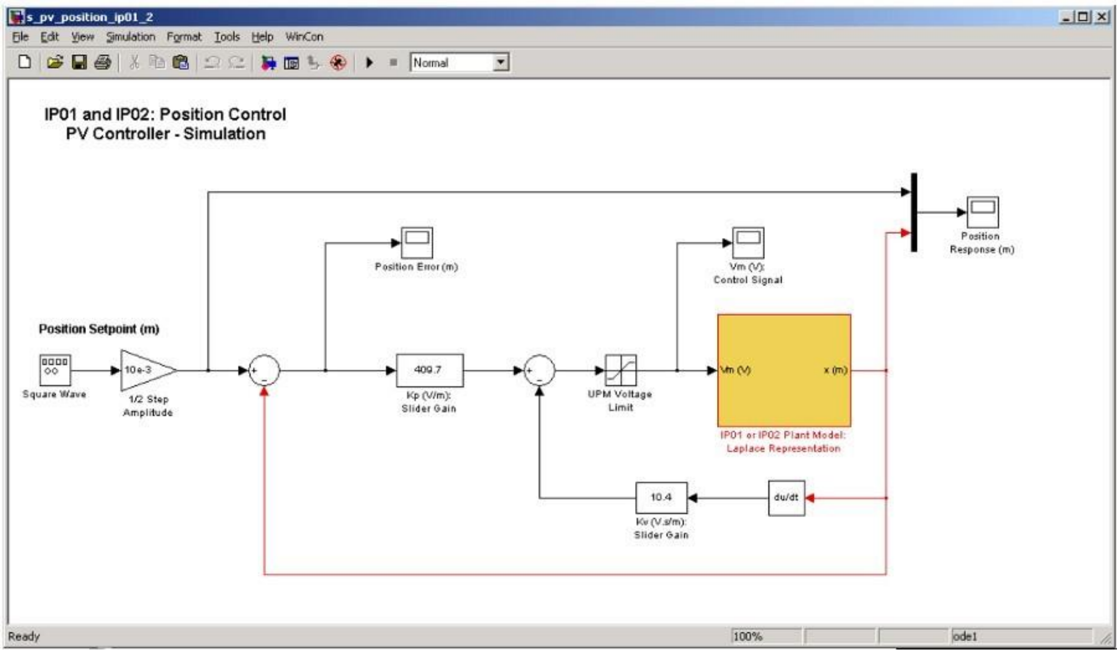


Figure 7 – Simulink Diagram used for the Simulation of the PV Control System

Step 2. Before you begin, you must run the Matlab script called *setup_lab_ip01_2_position_pv.m*. This file initializes all the IP02 system parameters and configuration variables used by the Simulink diagrams. Note that the controller gains that are loaded by the model are arbitrary.

Step 3. Modify the Simulink model as in Figure 8. The given model very close to the real system. As a second step you will implement a new subsystem (labeled as IP02 in Figure 8) as you did in the first experiment. But this time, you will make a sub-block for simulation, not for real implementation. Inside ofyour IP02 sub-block will be similar to the Figure 4 of this experiment.

You can use variables generated on Matlab Workspace, so that you would not need to write long variables and you can avoid typos. Then for the third step you will use "Open Loop Transfer Function" of IP02 as you calculated in assignment #2,part b of the preliminary work. Finally, you will directly use "Close Loop Trans-fer Function" you calculated for PV control scheme. If you did all the calculations correctly all 4 simulated systems should give the same output, provided that Kp and Kv are set to the same value.

REMARK: Go inside of the given IP02 sub-block and try to understand the blocks in it.

TIPS & TRICKS: After this experiment, you should be able to easily create the IP02 subblock at home. And you can test the controllers that you will design as preliminary works throughout the semester before coming to the laboratory session. Testing correctness of your controller design will gain you quite a lot of precious time that you can spend on the real inverted pendulum setup.

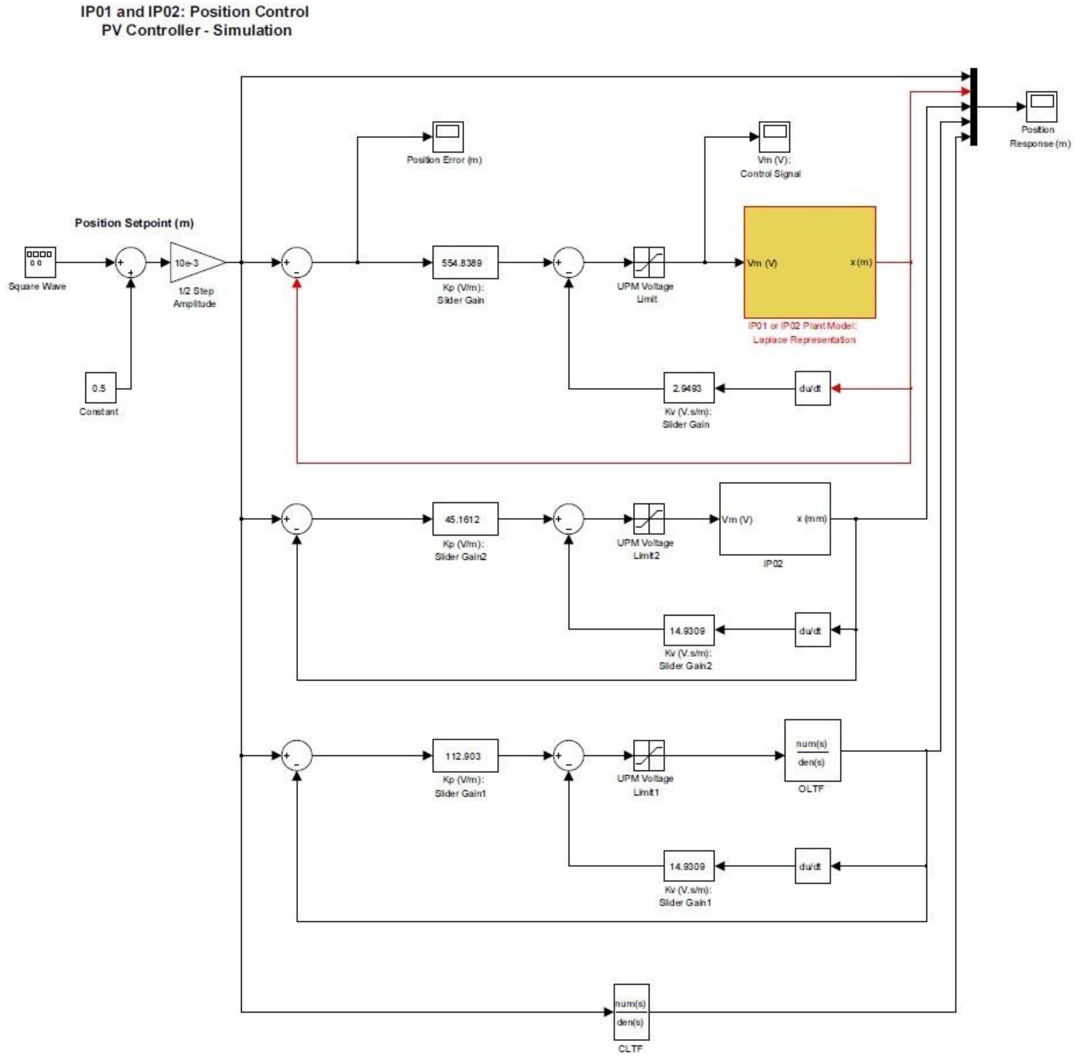


Figure 8: Modified model which includes open and close loop transfer functions

Step 4. Ensure that the Simulink simulation mode is set to *Normal*. Click on *Simulation* | *Start* from the Simulink menu bar and bring up the *Position Response (m)* scope. As you monitor the position response, adjust K_p and K_v using the slider gains, as depicted in Figure 9. Try a variety of combinations and note the effects of varying each gain (one at a time) on the system response. **Doing itonly for the given system will be enough**, don't do it for the parts you added instep 3.

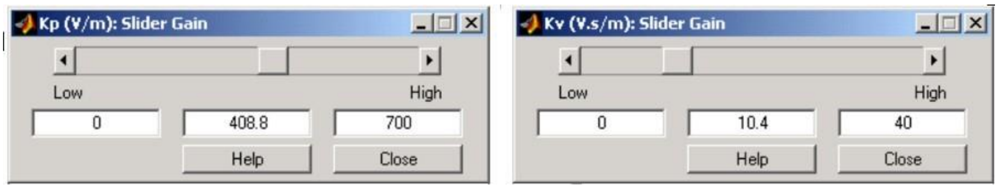


Figure 9 – Slider gains for K_p and K_v parameters

Step 5. Bring up the *Position Error (m)* as well as the *Vm (V): Control Signal* scopes. Discuss the effect of varying K_p and K_v (one at a time) on the resulting position error and the commanded voltage applied to your IP02 DC motor. Please refer to transient response properties in your discussion.

Step 6. To specifically include in your lab report:

i) Make a short table to describe the changes in the system response characteristics t_p and *PO* with respect to changes in K_p and K_v. *Note: Hold one gain constant while changing the other within the preset range.*

ii) Does the system response follow the behavior you had predicted in Assignment #3-part d (6.3.3) If not, briefly explain what you observe.

iii) Repeat part i for a few extreme cases of Kp and Kv of your own choice. Try to use different configurations to observe different damping characteristics. Comment on results.

Step 7. Now that you are familiar with the effects of each one of the two controller gains, enter in the designed K_p and K_v that you have calculated in Assignment #3 to meet the system requirements. *Note: the values should fall within theslider limits.*

Step 8. After running the simulation with the gains set to their calculated values, specify in space provided, the following:

i) Does the system response look like what you had expected? If not, what are the differences?

ii) What is its Percent Overshoot, *PO*? Measure its rise time, t_p. *Hint: To get a better resolution when measuring t_p, decrease the time range under the parameters option of the scope.*

iii) Do they match the design requirements?

Step 9. If the simulated response is as expected, you can move on to the next step. If your response is close to meeting the set requirements, try fine-tuning the controller gains to achieve the desired response. If the system response is far from the specifications, then you have to re-iterate your design process and re-calculate your controller gains K_p and K_v, as requested in Assignment #3.

Step 10. Now increase the cart mass on MATLAB workspace. Run the simulation and observe the effects of heavier cart. Observing only the output of system in Figure 7 will be enough. Then, update Kp and Kv values according to your calculations in the very last part of preliminary work. Run the simulation once more. Write down your observations in this step. First, comment on the performance of controller when cart mass is increased but control parameters are not updated, then, also consider effects of Kp & Kv modification.

Step 11. Now, you can move on to the next section in order to implement a real-time controller.

7.4. Real-Time Implementation of the PV Controller

7.4.1. Objectives

- To implement with QuaRC real-time environment, the previously designed PV position controller in order to command your IP02 servo plant.
- To run the simulation simultaneously, at every sampling period, in order to compare the actual and simulated responses.
- To change on-the-fly the two controller gains, K_p and K_v, and observe the effect on the actual position response of your physical IP02 system.

7.4.2. Experimental Procedure

After having designed your PV controller, calculated its two gains satisfying the desired time requirements, and checked the position response of the obtained closed-loop system through simulation, you are now ready to implement your designed controller in real-time and observe its effect on your actual IP02 plant. To achieve this, please follow the steps described below:

Step 1. Open the Simulink model file *q_position_pv_ip02.mdl*. You should obtain a diagram similar to the one shown in Figure 10. The model has 2 parallel and independent control loops: one runs a pure simulation of the PV controller connected to the same plant model as the one developed in Assignment #2 of the preliminary work. The other loop directly interfaces with your hardware and runs your actual IP02 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 taking note of the I/O connections.

Step 2. Check if the position set point generation signal is coming from the signal generator block called "Square Wave". Also check that the signal generator block properties are properly set to output a square wave signal, of amplitude 1 and of frequency 2/3Hz. Moreover, your model sampling time should be set to 1 ms, i.e., $T_s = 10^{-3}$ s.

CAUTION:

The velocity signal used in the control inner-loop of the actual IP02 plant is obtained by first differentiating the position signal (e.g., encoder counts), and then by low-pass filtering the obtained signal in order to eliminate its high frequency content. As a matter of fact, high frequency noise, which is moreover amplified during differentiation, causes long-term damage to the motor. To protect your DC motor, the recommended cut-off frequency is 50 Hz.

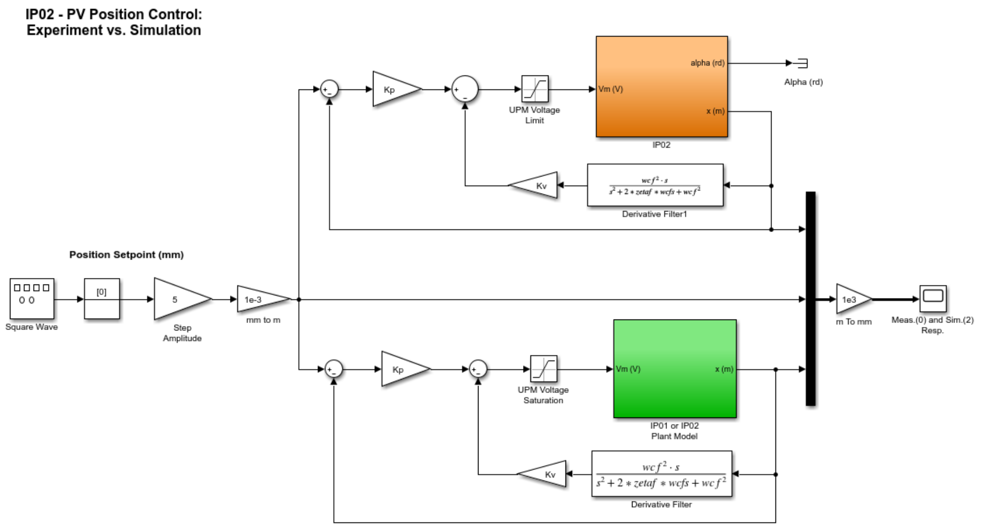


Figure 10 – Diagram used for the Real-Time Implementation of the PV Controller

Step 3. **Configure DAQ**: Double-click on the HIL Initialize block inside the *IP02* subsystem and ensure it is configured for the DAQ device that is installed in your system (The q2_usb or q8_usb board in our case). See Reference [5] for more information on configuring the HIL Initialize block.

Step 4. Before compiling the diagram and running it in real-time with QuaRC, you must enter your previously designed values of K_p and K_v in the Matlab workspace. To assign K_p and K_v, type their value in the Matlab command window.

Step 5. 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 CPU, you are ready to run your actual system in real-time. **Before doing so, manually move your IP02 cart to the middle of the track (i.e., mid-stroke position) and make sure that it is free to move on both sides.**

Step 6. Start the real-time controller by going to *QuaRC | Start* from the Simulink model menu bar. Your cart position should now be tracking the desired set point (e.g., square wave of ±5mm).

Step 7. Open the sink *Meas. (0) and Sim. (2) Resp.* by double-clicking on it. You should now be able to monitor on-line, as the cart moves, the actual cart position as it tracks your pre-defined reference input and compare it to the simulation result produced by the IP02 model.

Step 8. Now, you should try to discuss the following points:

i) How does your IP02 cart actual position compare to the simulated response?

ii) Is there a discrepancy in the results? If so, find some of the possible reasons.

iii) From the plot of the actual cart position, measure your system t_p and *PO* and record in the space provided below. Are the values in agreement with the design specifications? **Hint: You can accurately measure these parameters by saving the position traces and making the necessary calculations through Matlab.*

Step 9. Once your results are in agreement with the desired design requirements and your response looks similar to the one displayed in Figure 11, below, it means you have completed the design task and verified it on a physical system.

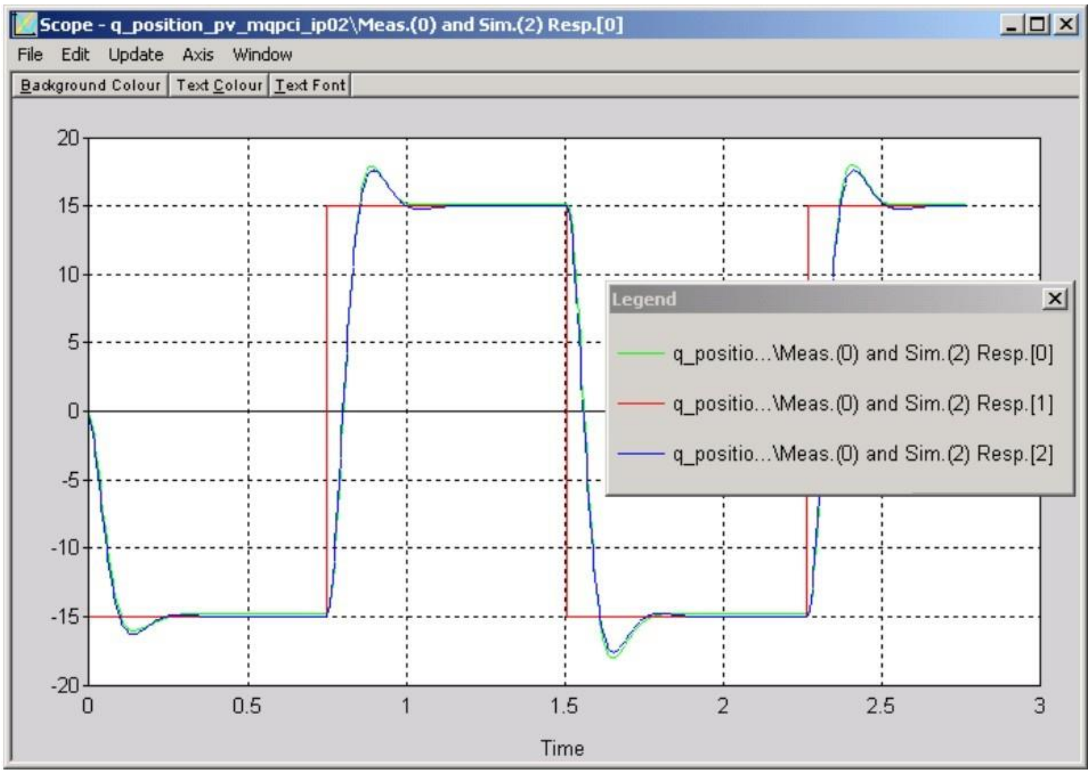


Figure 11 – Actual and Simulated Position Responses to a Square Wave Set Point. In the top plot, the green dash-dot line is the set point, the blue dot trace is the cart simulation, and the solid red trace is the measured cart position. The bottom plot is the motor input voltage.

Step 10. However, in order to perfectly meet the chosen design requirements (i.e., t_p and *PO*) of the closed-loop system, any controller design will usually involve some form of fine-tuning, which will often be an iterative process. At this point, you may want to fine tune your controller gains manually to determine whether you can improve on the designed controller. In order to do so, change constant "Gain" blocks of the real IP02 system with "Slider Gain" blocks. **Limit the maximum value of Kp to 550 and maximum value of Kv to 25 to avoid any damage to the system.** Then save and rebuild the model. Finally run your model. You should be able to change slider gains and observe the effects simultaneously. If gain blocks were not changed, for all new Kp and Kv values model would be required to rebuild and run. What are the values of fine-tuned Kp & Kv parameters?

Step 11. Now insert the extra mass onto the cart. First, control the heavy system with fine-tuned Kp and Kv for the light system. Then, use the parameters that you calculated in the very end of preliminary work. Finally, perform a fine-tuning step to obtain the best performance you can achieve. Write down your observations, comments, and values of fine-tuned Kp & Kv.

This is the end of the lab session.

8. Appendix A: Nomenclature

Table A.1, below, provides a complete listing of the symbols and notations used in the IP02 mathematical modeling and controller design 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
L_m	Motor Armature Inductance	Lm
K_t	Motor Torque Constant	Kt
η_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
η_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
P_r *	Rack Pitch	Pr
r_{mp}	Motor Pinion Radius	r_mp
N_{mp} *	Motor Pinion Number of Teeth	N_mp
r_{pp} *	Position Pinion Radius	r_pp

Symbol	***Description***	***Matlab / Simulink Notation***
N_{pp} *	Position Pinion Number of Teeth	N_pp
B_{eq}	Equivalent Viscous Damping Coefficient as seen at the Motor Pinion	Beq
T_m	Torque Generated by the Motor	
T_{mp}	Torque Applied by the Motor on the Motor Pinion	
F_c	Cart Driving Force Produced by the Motor	
F_{ai} *	Armature Rotational Inertial Force, acting on the Cart	
T_{ai} *	Armature Inertial Torque, as seen at the Motor Shaft	
ç_m *	Motor Shaft Rotation Angle	
ω_m	Motor Shaft Angular Velocity	
X	Cart Linear Position	x
PO	Percent Overshoot	PO
t_p	Peak Time	tp
T	Continuous Time	
S	Laplace Operator	
ω_n	Undamped Natural Frequency	wn
Z	Damping Ratio	zeta
K_p	Proportional Gain	Kp
K_v	Velocity Gain	Kv

Table A.1. The list of all variables used for the modeling of the IP02 servo plant and the PV controller.

Notes on variables:

Efficiency variables are multiplier factors that indicate how much energy is lost in a conversion process. For example, the Motor Torque Constant indicates how current is converted into Torque in a rotary DC motor while the Motor Efficiency variable is a multiplier smaller than one to indicate the loss in the current-to-torque conversion.

You will not need all the given parameters in this experiment. The ones that won’t be used are marked with ‘*****’ character.

9. Report

7.3. Simulation of the Servo Plant with PV Controller

Step 5:

Step 6:

i.

K_p	K_v	t_p	PO
150	4.5		
150	5.0		
250	5.0		
250	5.5		
350	5.5		
350	6.0		
275	5.0		
275	5.5		
275	6.0		
275	6.5		

ii.

iii.

K_p	K_v	Damping observation

Step 8:

i.

ii.

iii.

Step 9:

Did you need to fine-tune your controller?

What were the final Kp & Kv values? **Kp:** **Kv:**

Step 10:

7.4. Real-Time Implementation of the PV Controller

Step 8:

i.

ii.

iii.

Step 10:

Fine-Tuned Kp:

Fine-Tuned Kv:

Step 11:

Fine-Tuned Kp (with extra mass):

Fine-Tuned Kv (with extra mass):

10. Knowledge Test

1. What is the effect of increasing mass?

2. What is the reason for non-zero steady state error for step response in PV control? Remember your discussion in preliminary work. Is there any discrepancy?

3. When the system is ordered to go from x=-10cm to x=20cm (PO=20%), what will be the position of chart at the maximum overshoot time?