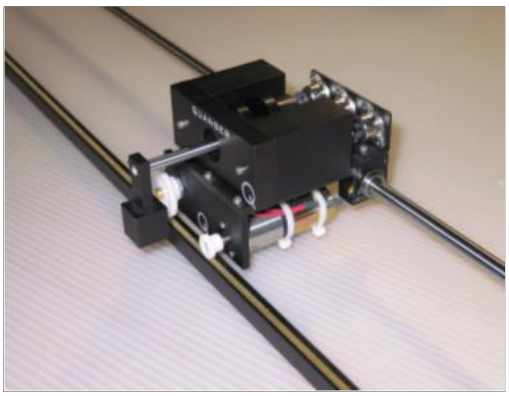
Middle East Technical University
Department of Electrical and Electronics Engineering

EE 406
Laboratory of Feedback Control Systems



Experiment #3:
Phase-Lead Compensator Based Speed 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 phase-lead types of compensators. The challenge of the present lab is to control the velocity, i.e., speed, of your IP02 linear motion servo plant. In practice, the phase-lead types of compensators are important because they are very common for analog controllers since they can be realized using an RC network with an amplifier.

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

- How to mathematically model the IP02 servo plant from first principles in order to obtain the open-loop transfer function, in the Laplace domain. (The experiment includes this step for those who have been unable to come up with the correct TF in Experiment #2. Those who already have the correct TF can use their results from the previous experiment and skip this modeling step).
- How to design and simulate a Phase-Lead-based speed controller to meet the required design specifications.
- How to use Bode plots and the phase margin method to design your speed controller.
- How to implement your controller in real-time and evaluate its actual performance.

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, as shown in *Fig. 1* below.
- **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 identical to the one you have used in Experiment #2. The setup 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 the IP02 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.)

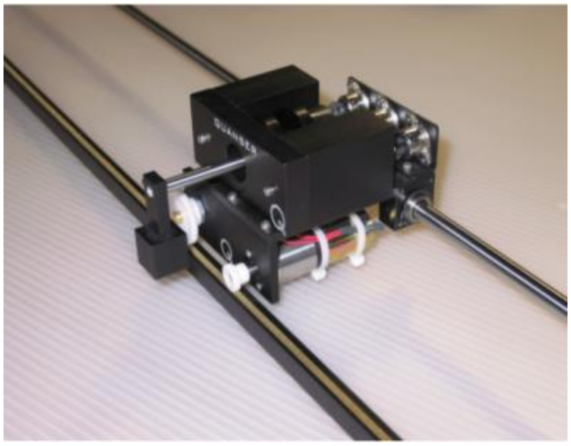


Fig. 1 – The IP02 Servo Plant

5 Controller Design Specifications

In the present laboratory (i.e., the preliminary work and in-lab sessions), you will design and implement a control strategy based on the phase-lead compensation method, in order for your compensated IP02 closed-loop system to satisfy three pre-defined performance requirements. These are given below. The purpose of these performance specifications is first to reduce, or eliminate, any steady-state error present in your system. Second and last, to improve the transient response of the system by means of increasing the bandwidth (in this laboratory, we consider the Gain crossover frequency w_c as the bandwidth of the system) and phase margin Φ_m of the system.

The phase-lead-based closed-loop control system including an IP02 plant should meet the following performance requirements:

i) The steady-state error to a unit step input should be zero.

$$|e_{ss}| = 0 \text{ m/s}$$

ii) The bandwidth of the compensated open-loop system should be:

$$\omega_c = 80 \text{ rad/s}$$

iii) The phase margin of the compensated open-loop system should be:

$$\Phi_m = 85^o$$

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. Note that the plant model is almost the same with the model in Experiment #2, except for the fact that **the plant output is now the velocity of the cart instead of the cart position**. Therefore, the preliminary work of the previous experiment may be useful here.

You will first build a model block diagram by considering each sub-component of the electro-mechanical system. Consider your EE302 course notes 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.

(a) This step is a refresher of Experiment #2: 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. This step applies to all students and is a refresher if you already have the correct TF in a previous experiment.

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 velocity, the plant output is now selected as the cart velocity on the rack, as depicted in Figure 2.

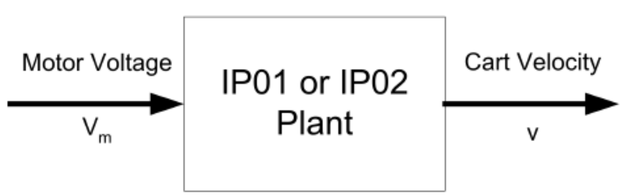


Fig. 2 – The IP02 Plant Input and Output for Velocity control

Fig. 3 – The IP02 Servo Plant Schematics (Use the page in Landscape orientation)

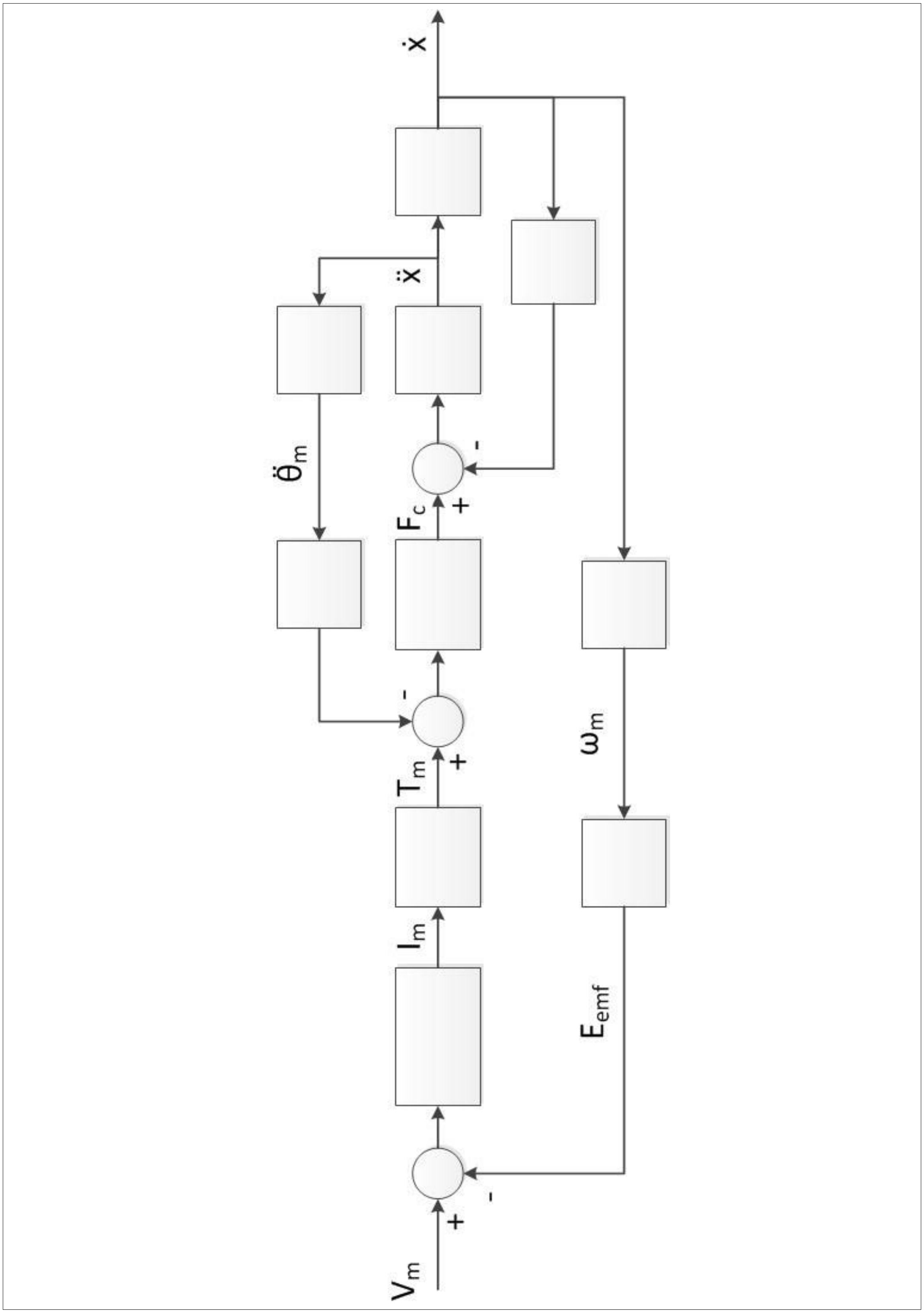


Fig. 4 – The IP02 Servo Plant Detailed Block Diagram

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. Remember that the transfer function *G(s)* is given by

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

and is now defined for a velocity output.

(a) Derive and write down this overall open-loop plant transfer function in the box provided below in terms of all the variables defined in Appendix A.

(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. By noting that the motor inductance is much smaller than motor internal resistance, you may from this point on, assume that 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.

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

Zeros:

Poles:

DC Gain:

(d) What is the Type of the system? Assuming uncompensated unity-feedback structure (no controller in the forward path) can the steady-state error for this system for a unit-step input be zero? Derive E(s) and find $|e_{ss}|$ for unit step and unit ramp inputs.

Type of the system:

$|e_{ss}|$ for unit step: $|e_{ss}|$ for unit ramp:

6.3 Assignment #3: Phase-Lead Compensator Design

This section deals with the design of a phase-lead based closed-loop controller in order to control the velocity of your IP02 cart so as to satisfy the design requirements. We will use the following approach:

i. First, we will try eliminating the system's steady-state error $|e_{ss}|$.
ii. Second, we will find the gain K such that the open-loop bandwidth requirement is met.
iii. Lastly, we will find the resulting phase margin and introduce the phase-lead compensator in order to obtain the desired closed-loop relative stability (desired open-loop phase margin) for the system.

6.3.1 Eliminating Steady-State Error

Suppose we want to control the plant using the unity-feedback structure given in *Fig. 5*. Find the simplest possible $H(s)$ using such that we satisfy the steady-state error design specification. Hint: Use $E(s)$ and $|e_{ss}|$.

$H(s) =$

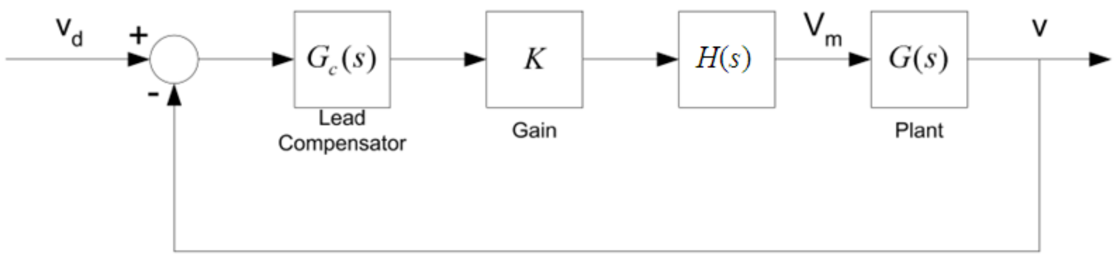


Fig. 5 – Lead-Compensator based Speed Control structure

6.3.2 Satisfying Desired Bandwidth

We have found the necessary $H(s)$ in *Fig. 5* such that we have zero steady-state error to a unit step input. One design criterion related with the system transient response is given as the open-loop bandwidth of the system. We would like to first find what the present bandwidth (Gain crossover frequency) is, followed by determining the required gain K to set the bandwidth to the desired value.

(a) Sketch the magnitude part of the bode plot of the cascade system $H(s)G(s)$ in the space provided below, indicating all critical points. Determine the gain crossover frequency w_c, hence the present bandwidth of this intermediate system. You can always use MATLAB bode plots for checking your results. But, in this step analytical calculations are expected for bode plots. Especially, calculate gain cross-over frequency and phase at gain crossover frequency.

w_c =

Phase @ gain cross-over frequency =

(b) Calculate the value of the multiplier Gain, K, to bring w_c of $KH(s)G(s)$ to the specified value.

$K =$

(e) Using your knowledge from EE302 and assuming that the gain you have found is the combined gain of the open-loop system $KH(s)G(s)$, discuss the effect of changing K (increase/decrease) on the transient behavior of the closed-loop system (rise time t_r and maximum overshoot M_p%). (Note: For this step, ignore the compensator block, i.e., assume $G_c(s)=1$).

K	t_r	M_p%

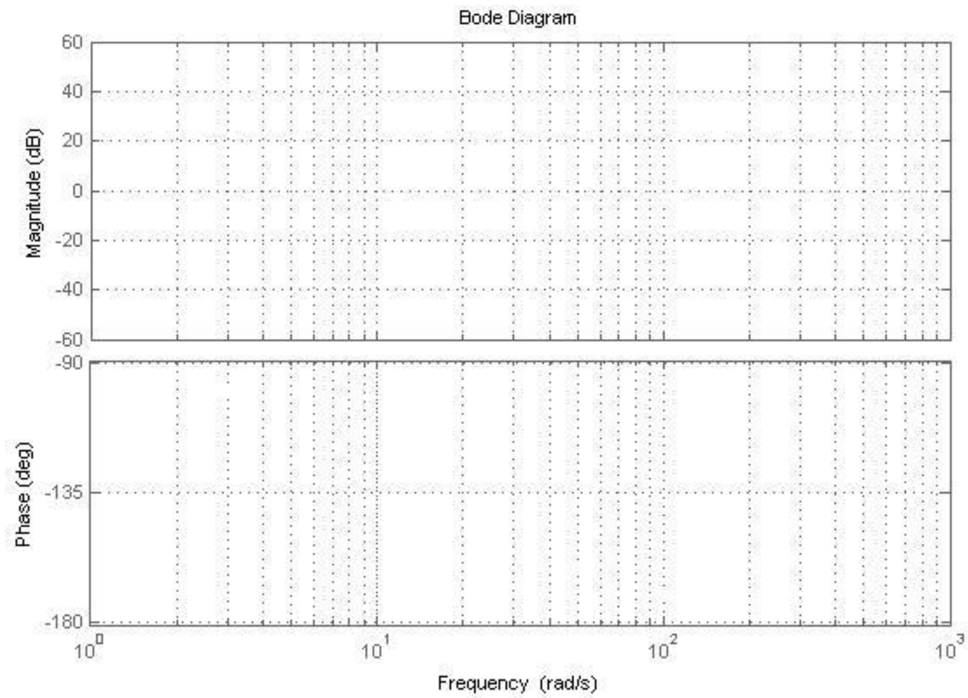
(f) Considering the relation between the gain, K, and the bandwidth, w_c , state the effect of bandwidth on the transient behavior of the closed-loop system.

w_c	t_r	M_p%

6.3.3 Satisfying Desired Phase Margin

From EE302, the purpose of the *Lead Compensator* block in *Fig. 5* is to increase relative stability of the system, by increasing the phase margin of the closed-loop system.

(a) We would like to know what the present phase margin of the system is without the help from the $G_c(s)$ block in *Fig. 5*. In the space provided below, sketch magnitude and phase of the bode plot of the uncompensated open-loop system $KH(s)G(s)$. In this step you can use MATLAB to plot the desired diagrams. Do not forget to mark all critical points on your plots.



(b) On the plot, indicate the gain crossover frequency w_c and determine the phase margin of the system.

$\Phi_m =$

(c) Given the design specification on phase margin, determine the extra phase, Φ_c, required to meet the compensated phase margin of the system. In EE302 you were also adding 5° error margin while calculating required compensator phase. In this step do not add any error margin.

$\Phi_c =$

The phase-lead compensator that we will design has a transfer function $G_c(s)$ of the form

$$G_c(s) = \frac{\alpha\left(s + \dfrac{w_c^{lead}}{\alpha}\right)}{s + w_c^{lead}\alpha} \tag{2}$$

Note that this compensator structure is slightly different from the form we are familiar from our EE302. The main difference of the present form is that it exposes the frequency at which the gain is 1.

(d) Show that $|G_c(jw)| = 1$ at $w = w_c^{lead}$.

(e) Show that the maximum phase $\angle G_c(jw)$ is achieved when $w = w_c^{lead}$.

(f) Determine this maximum phase as a function of the compensator parameters.

(g) By equating the maximum phase of the compensator to the extra phase required for the design, find the suitable parameter values of the compensator.

$w_c^{lead} =$ $\alpha =$ $G_c(s) =$

(h) You have found values of all blocks in Figure 5. Now, calculate open-loop transfer function of the system shown in Figure 5.

(i) Derive E(s) for the final compensated system. Calculate $|e_{ss}|$ for unit step and unit ramp inputs.

(j) Comment on the effect of the compensator on the steady-state error performance of the closed-loop system (Hint: recall the type of the open-loop system before compensation)

(k) Comment again on the effect of the compensator on the steady-state error of the closed-loop system if the integrator block was not considered in our design.

(l) Calculate the closed-loop transfer function of the whole compensated system.

(m) Why didn't we add the error margin of 5° to the required compensator phase as we did in EE302? Explain.

7 In-Lab Experimental Procedure

In the beginning of the lab session, have your Teaching Assistant check your results and in particular, your open-loop Plant transfer function and compensator parameters w_c^{lead} and α. If there is serious disagreement with the expected values, your Assistant will guide you how to proceed.

Teaching Assistant Review Notes: (Transfer Function, and Design check, K, $H(s)$, $G_c(s)$)   Assistant Name:

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.

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 Simulation of the Servo Plant with a Phase-Lead Controller

7.2.1 Objectives

a) To simulate with Matlab the different time and frequency responses involved in the design of a lead-based speed controller for the IP02 servo plant.
b) To simulate with a Simulink diagram your IP02 model and to close the velocity loop by implementing a phase-lead speed controller.

7.2.2 Experimental Procedure

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

Step 1. Before you begin, you must run the Matlab script called *setup_lab_ip01_2_speed.m*. This file initializes all the IP02 system parameters and configuration variables used by the Simulink diagrams.

Step 2. In order to check the phase-lead-controller design that you carried out in the pre-lab Assignment #3, start by verifying the Bode plots for your open-loop IP02 servo plant using your derived transfer function expression and Matlab. (Hint: You can use the Matlab function: 'bode'.) You can then move on by generating the Bode plots for the Plant + Integrator system. Determine how much gain K must be introduced to achieve the desired system's crossover frequency. You can also iteratively change K to achieve the desired w_c. Does the value match what you had calculated in Assignment #3? If so, you can proceed to the next step. If not, the value you calculate here may be verified with your Lab Assistant and used for the subsequent steps.

Help on MATLAB bode: g=tf([num coefficients],[den coefficients]);

bode(g,{0.01,100});

Step 3. In order to determine the phase-lead needed to be added by the compensator to meet the phase margin requirement, generate the Bode plots for your open-loop IP02 servo Plant + Integrator + Gain K. (Hint #1: Your new open-loop transfer function of your system is: $KG(s)/s$. Hint #2: To plot Bode plots and obtain at the same time a reading of the phase margin of your system, you can use the Matlab function: 'margin'.) What is the compensator's phase lead, Φ_c, required to achieve the phase margin design specification? Sketch your observed bode plot (magnitude and phase) in the space provided, indicating critical values. Does the value match what you had calculated in Assignment #3? If so, you can proceed to the next step. If not, the value you calculate here may be verified with your Lab Assistant and used for the subsequent steps.

Step 4. In order to verify your lead compensator design, generate its Bode plots. If the obtained phase lead is consistent with your design goals, then also generate the Bode plots of the final, compensated system, that is to say: the IP02 servo plant plus Integrator plus Gain *K* plus Lead Compensator. **Hint: To create Bode plots and at the same time obtain a reading of the phase margin of your system, you can use the Matlab function: 'margin'*. Sketch your observed bode plot of final compensated system (magnitude and phase) in the space provided, indicating critical values.

Does your resulting compensated system meet the required performance specifications, in terms of crossover frequency and phase margin? If so, you can proceed to the next step. If you have not yet met the design specifications, you should go back and redesign your speed controller.

Step 5. Finally, generate with Matlab the time response of the compensated system to a unit velocity step. (Hint: You can use the Matlab function: 'step') Sketch the time response in the space provided. Indicate all critical time-response parameters.

Step 6. In Simulink, open the model called *s_speed_lead_ip01_2.mdl*. This diagram should be similar to the one shown in *Fig. 6*. It includes a subsystem containing your IP01 or IP02 modeled plant, as well as the lead-based controller, i.e., the integrator, the gain, and the phase-lead compensator. Check that the signal generator block properties are properly set to output a square wave signal, of amplitude 1 and of frequency 1 Hz.

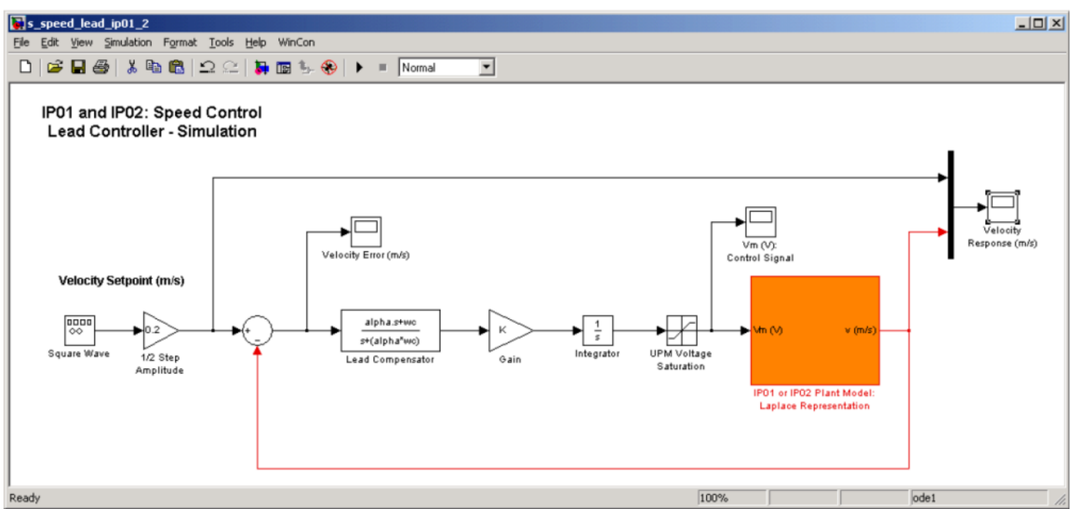


Fig. 6 – Simulink diagram used for the simulation of the phase-lead based speed control system

Step 7. Enter at the Matlab prompt, the designed controller parameters *K* and *alpha*, as previously calculated and/or determined. You can use LeadCompParams.m file and fill the variables that you calculated in the preliminary work.

Step 8. Ensure that the Simulink simulation mode is set to *Normal*. Click on *Simulation | Start* from the Simulink menu bar and bring up the *Velocity Response (m/s)* scope. Observe the system's velocity response. You can also bring up the *Velocity Error (m/s)* as well as the *Vm (V): Control Signal* scopes. Does the system response react as how you had simulated in Step 5?

Step 9. Now, modify the simulation model shown in Figure 6 to observe different approaches for modeling a system. Final model should look like Figure 7. There should be 3 new control paths: i. Same loop with plant transfer function, ii. Open-loop transfer function with unity feedback, iii. Closed-loop transfer function. Run the simulation and comment on results. Comment on advantages and disadvantages of different modeling approaches.

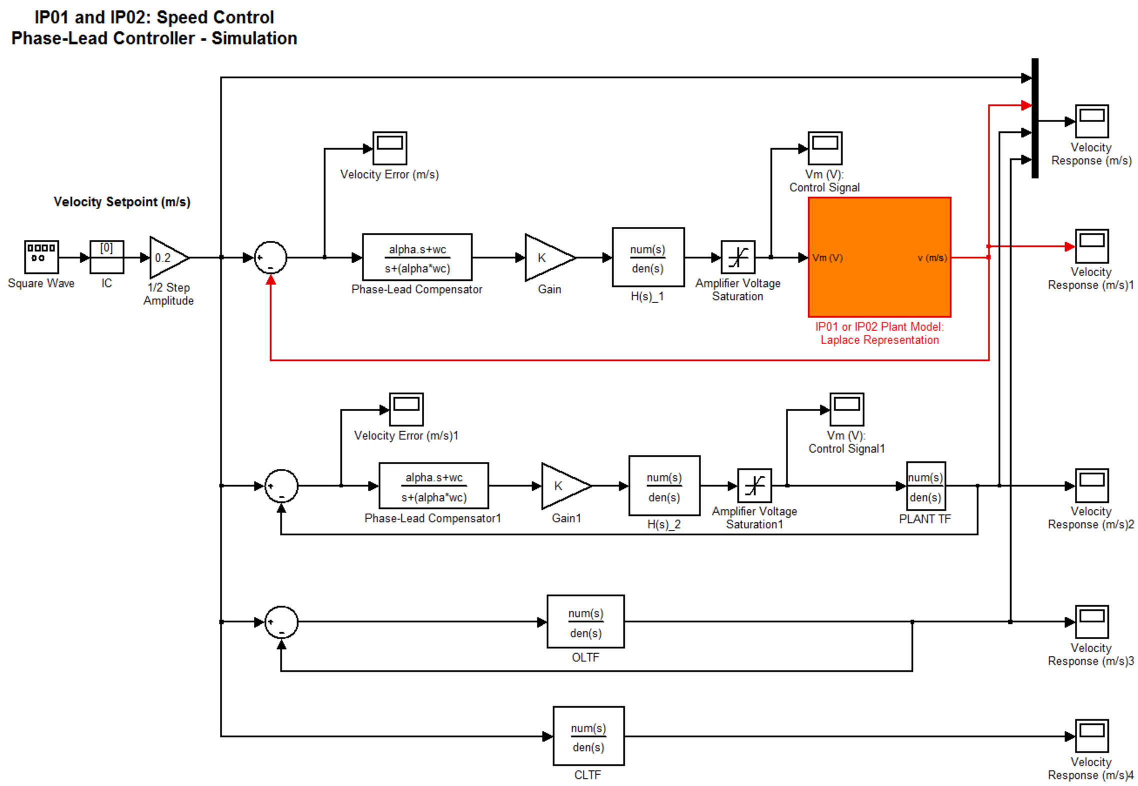


Fig. 7: Modeling the same system with different approaches.

Step 10. Before proceeding, make sure that laboratory assistant checked your simulation results.

Step 11. Now, start a new model. As a first control loop, you can copy and paste the model that is already given to you as illustrated in Figure 6. In this step, you will observe the time response effects of each phase-lead compensator design steps. Modify your model as seen in Figure 8. Sketch the simulation result for each control loop. Please don't forget to add appropriate titles, axis values and labels. Comment on results you observe. Note that you already observed their bode plots.

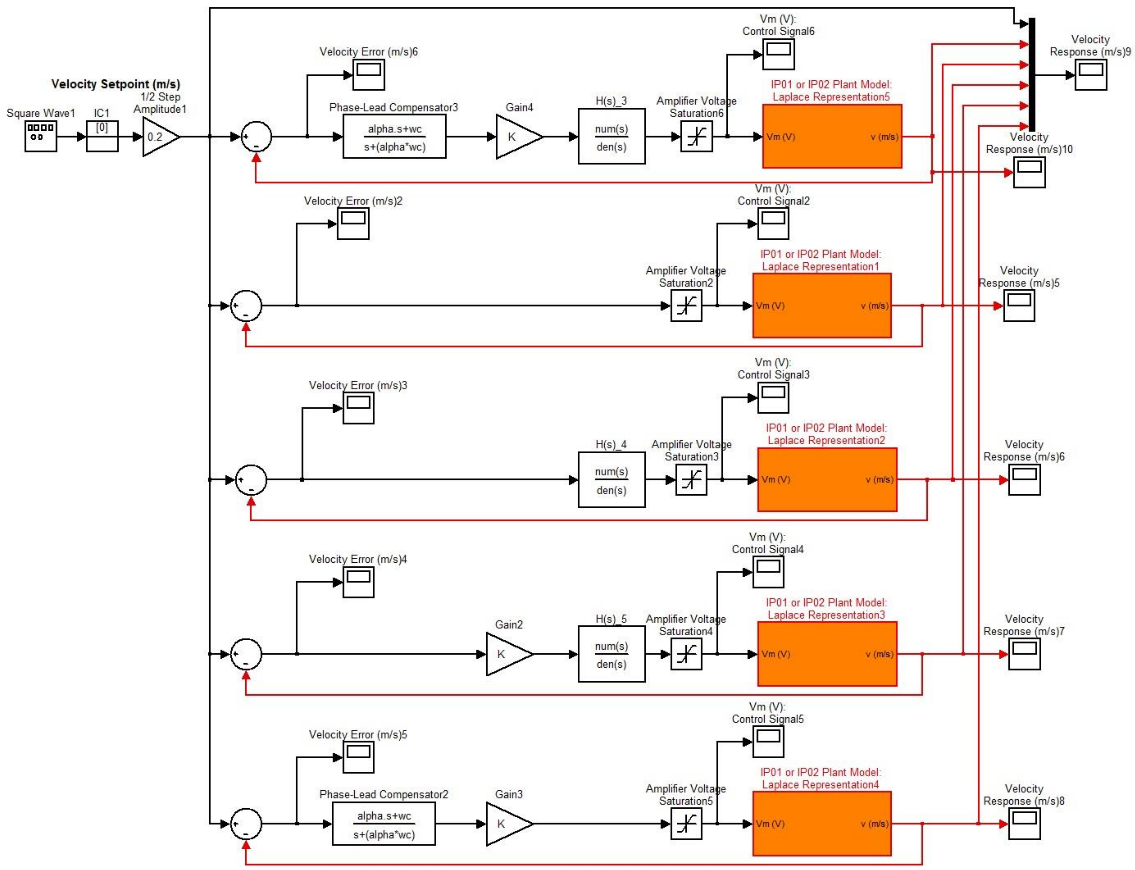


Fig. 8: A Simulink model for observing phase-lead compensator design steps and effects of added blocks.

7.3 *Real-Time Implementation of the Phase-Lead Controller*

7.3.1 Objectives

i) To implement with QuaRC the previously designed phase-lead speed controller in order to command IP02 hardware servo plant.
ii) To run the simulation simultaneously with the electro-mechanical hardware, at every sampling period, in order to compare the actual hardware and simulated responses.

7.3.2 Experimental Procedure

After having designed your phase-lead controller, calculated its parameters and gain satisfying the desired performance requirements, and checked the speed 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 electro-mechanical hardware plant. To achieve this, please follow the steps described below:

Step 1. Open the Simulink model file *q_speed_lead_ip02.mdl*. Ask your Lab Assistant if in doubt. You should obtain a diagram similar to the one shown in Figure 9. The model has 2 parallel and independent control loops: one runs a pure simulation of the lead-based controller connected to the same plant model as the one developed in Assignment #2 of the preliminary work section. 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.

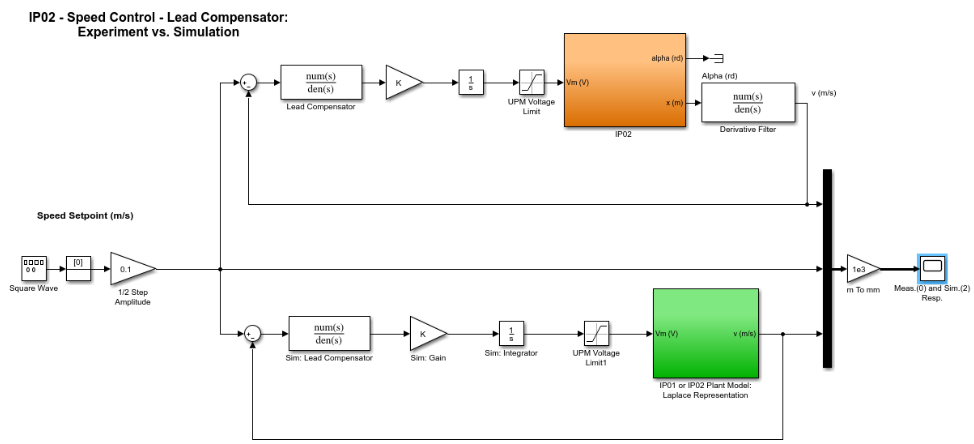


Fig. 9 – Simulink diagram used for the real-time implementation of the phase-lead based speed control system

Step 2. Check that the speed setpoint 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 0.5 Hz. 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 loop of the actual IP02 plant is obtained by first differentiating the position signal (e.g., encoder counts or potentiometer voltage), 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 70 Hz for the IP02 Plant.

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. See Reference [6] 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 *alpha* and *K* in the Matlab workspace.

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, you should be able 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 your real-time controller. To do this, click on the QuaRC | Start from the Simulink model menu bar. Your cart speed should now be tracking the desired setpoint values (e.g., square wave of ±0.1 m/s).

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 speed as it tracks your pre-defined reference input and compare it to the simulation result produced by the IP02 model. Sketch both observed plots in the space provided by using separate line styles.

Step 8. Specifically discuss in your lab report the following points:

- How does your IP02 cart actual speed compare to the simulated response?
- Is there a discrepancy in the results? If so, find some of the possible reasons.

Step 9. Once your results are in agreement with the one displayed in Figure 10, you can conclude that the design requirements are met. Remember that there is no such thing as a perfect model, and that your calculated controller gains were based on a theoretical and ideal plant model.

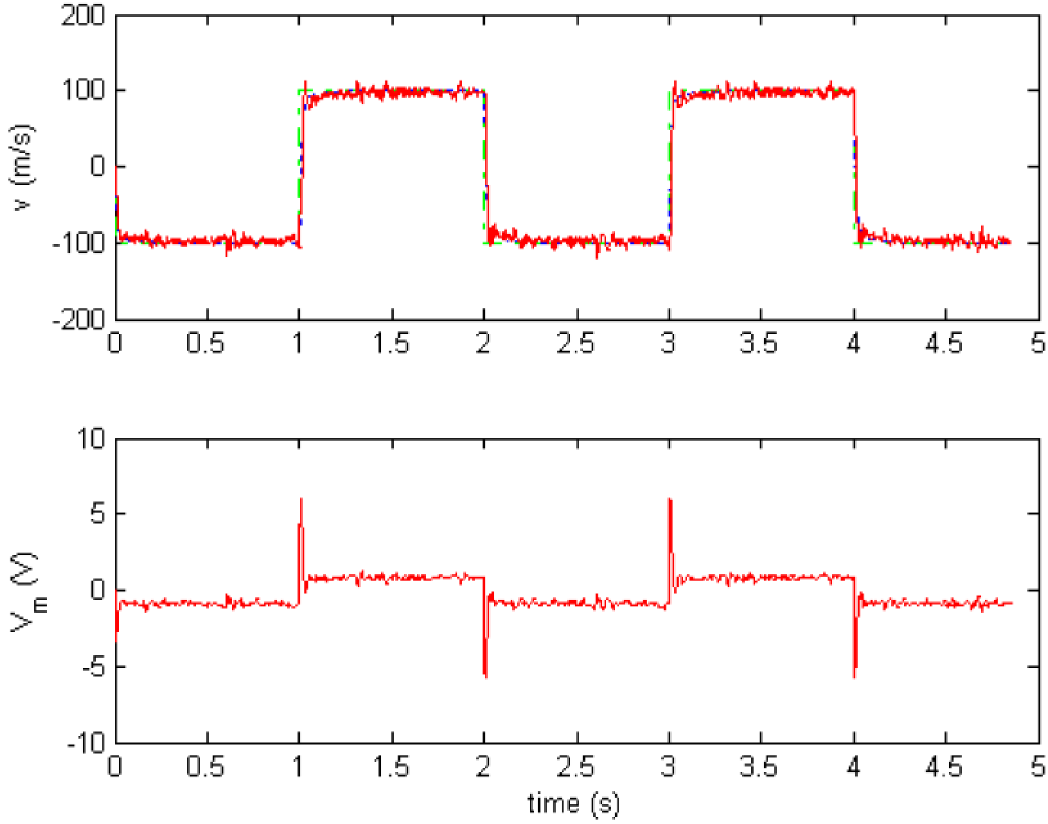


Fig. 10 – Actual and Simulated Speed Responses Obtained from the Lead Controller. In the top plot, the green dash-dot line is the setpoint, the blue dot trace is the cart simulation, and the solid red trace is the measured cart speed. The bottom plot is the motor input voltage.

Appendix A. Nomenclature

Table A.1, below, provides a complete listing of the symbols and notations used in the IP01 and IP02 mathematical modelling 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
ç_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
ç_g	Planetary Gearbox Efficiency	Eff_g
M_{c1}	IP01 Cart Mass (Cart Alone)	Mc1
M_{c2}	IP02 Cart Mass (Cart Alone)	Mc2
M_w	IP02 Cart Weight Mass	Mw
M	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
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	
ω_m	Motor Shaft Angular Velocity	

Symbol	*Description*	*Matlab / Simulink Notation*
v	Cart Linear Velocity	v
t	Continuous Time	
s	Laplace Operator	
e_{ss}	Steady-State Error	ess
ω_c	Desired Bandwidth of the Compensated System	wc
Φ_m	Phase Margin	phi_m
K	Gain	K
Φ_c	Compensator's Phase Lead	phi_c
M_c	Compensator's Amplification	Mc
α	Lead Compensator Parameter	alpha

Table A.1 IP01 and IP02 Model Nomenclature

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.

8 Report

Names, Student IDs, and Signatures of Group Students:

1) ____

____

2) ____

____

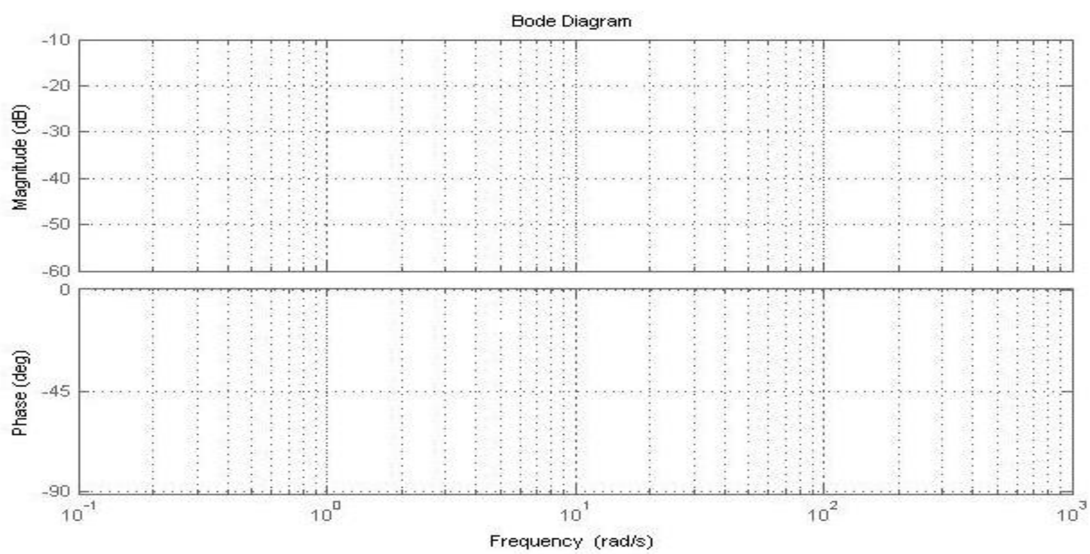
Date of Experiment:

Name of Lab Assistant:

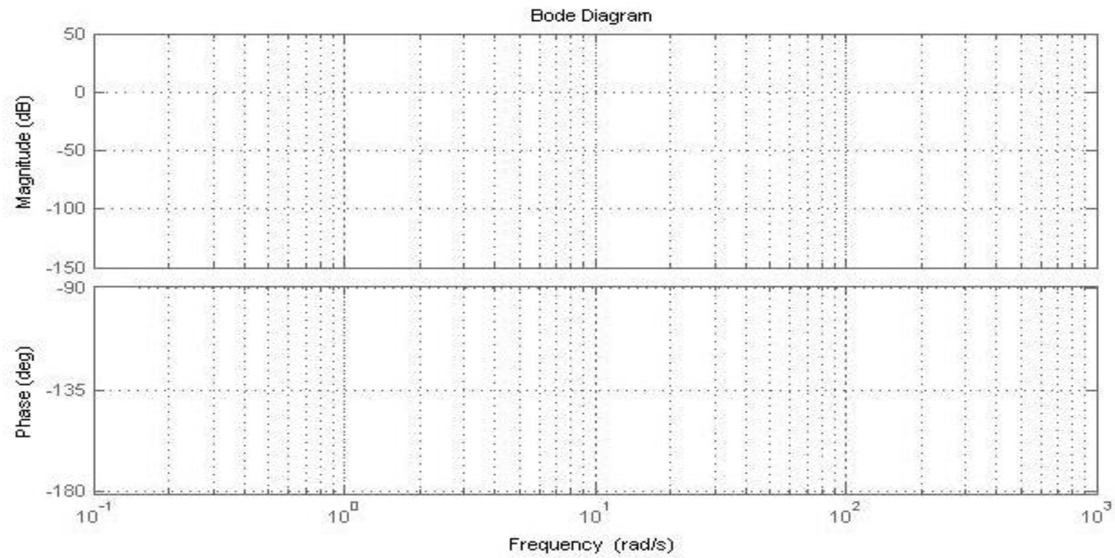
7.2. Simulation of the servo Plant with a Phase-Lead Controller

Step 2:

Plant only bode plots:



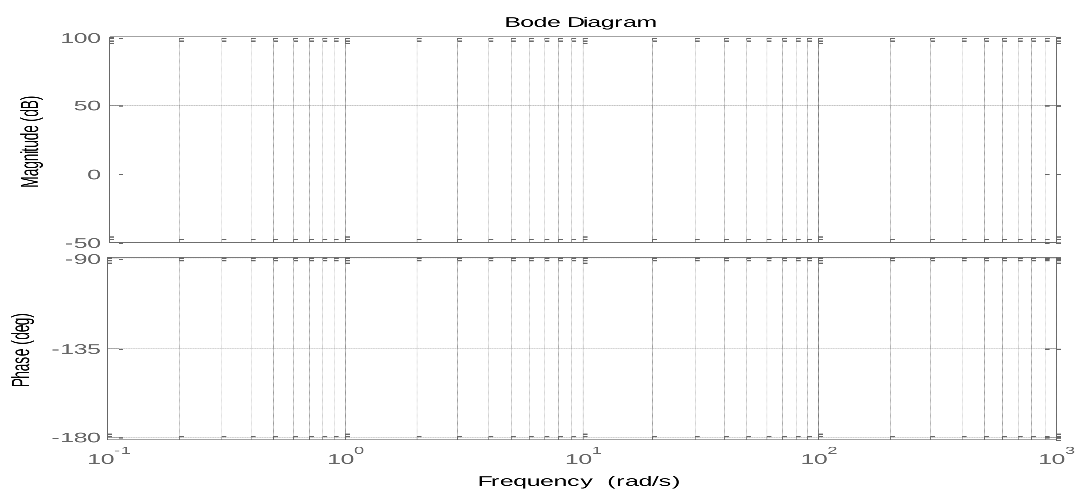
Plant with H(s) bode plots:



Gain required on dB = K =

Step 3:

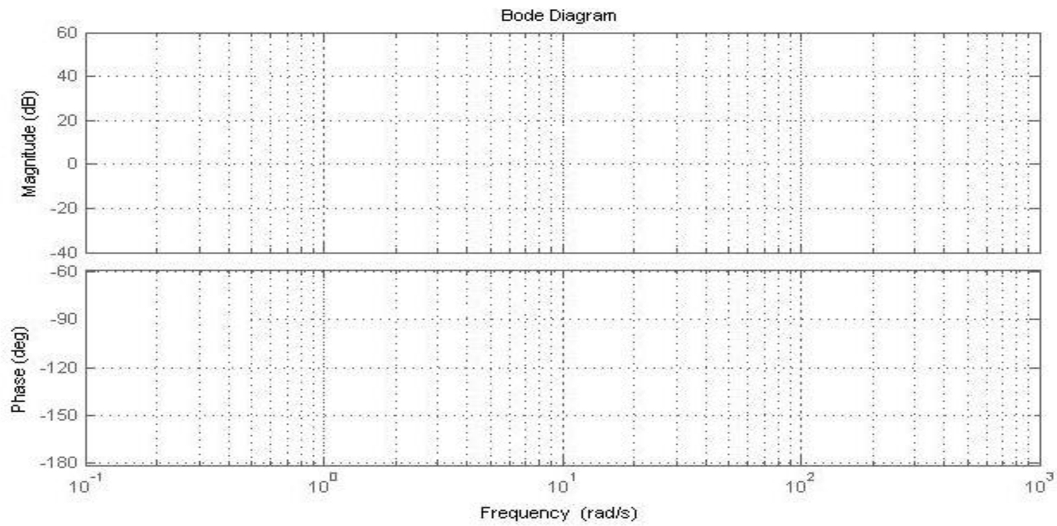
Plant with H(s) and K bode plots:



Φ_c =

Step 4:

Bode plots for compensated full system:



Step 5:

Step 8:

Step 9:

Step 11:

7.3 Real-Time Implementation of the Phase-Lead Controller

Step 7:

Step 8:

9 Knowledge Test

1. What is the gain cross-over frequency?

2. What is the phase margin? How did you decide on the value of phase added by the phase-lead compensator?

3. Why did we use H(s) block in this experiment?

4. What is the difference of using a PID type controller and phase-lead compensator?

5. Explain the phase-lead compensator design procedure.