LABORATORY MANUAL

ECE 360

CONTROL SYSTEMS I

This Manual was prepared by:

Dr. P. Agathoklis

Revised 2010, 2011, 2013, 2016, 2018, 2020, 2023

TABLE OF CONTENTS

A. SAFETY REGULATIONS

Safe practice in the laboratory requires an open attitude and knowledgeable awareness of potential hazards. Safety is a collective responsibility and requires full cooperation from everyone in the laboratory. This cooperation means each student and instructor must observe standard safety precautions and procedures and should:

1. Follow all safety instructions carefully.
2. Become thoroughly acquainted with the location and use of safety facilities such as fire extinguishers, first aid kits, emergency showers, eye- wash stations and exits. See marked floor plan posted near exit staircases for equipment locations.
3. Become familiar with experimental procedures and all potential hazards involved before beginning an experiment.

ELECTRICAL HAZARDS

General Safety Principles

Electrical currents of astonishingly low amperage and voltage under certain circumstances may result in fatal shock. Low-voltage dc circuits do not normally present a hazard to human life, although severe burns are possible. Voltage as low as 24 volts ac can be dangerous and present a lethal threat. The time of contact with a live circuit affects the degree of damage, especially where burns are concerned. Very small electrical shocks, even small "tingles", should serve as a warning that an electrical problem exists and that a potentially dangerous shock can occur. The equipment and circuits in use must be immediately disconnected and not re-used until the problem is discovered and corrected by the instructor and/or technologist.

1. When handling electric wires, never use them as supports and never pull on live wires.
2. Report and do not use equipment with frayed wires or cracked insulation and equipment with damaged plugs or missing ground prongs.
3. Report and do not use receptacles with loose mountings and/or weak gripping force.
4. Avoid pulling plugs out of receptacles by the cord and avoid rolling equipment over power cords.
5. Be sure line-powered equipment has 3-wire grounding cords and that you know how to use the equipment properly. Ask for help and instruction when needed.
6. Any electrical failure or evidence of undue heating of equipment should be reported immediately to the instructor and/or technologist. If you smell over-heating components or see smoke coming from any circuit or equipment, switch the power off immediately.
7. Ensure all equipment is powered-off at the end of each experiment.
8. Only qualified ECE personnel should maintain electric or electronic equipment.
9. Cardiopulmonary resuscitation (CPR) often will revive victims of high-voltage shock. Only qualified people should attempt CPR.

FIRST AID

1. For simple cuts or minor first aid use the First Aid Kits available in each room. The University Health Services may also be contacted at **8492**. All injuries, no matter how minor, should be reported to your instructor and/or technologist.
2. For medical emergencies call **911** <u>and</u> Traffic and Security at **7599**.

INDIVIDUAL RESPONSE PROCEDURES FOR FIRES

If **<u>you discover</u>** a fire, smoke or an explosion:

1. Shout for assistance.
2. Activate the nearest fire alarm.
3. If it is a small fire, attempt to put it out with available fire equipment. See the marked floor plan posted near exit staircases for equipment locations.
4. If the fire is out of control and it is too large to handle with one fire extinguisher, isolate the fire by closing the doors and windows behind you as you leave. Do **not** lock the doors.
5. Warn others and leave the building with reasonable speed using recommended exits. Assist disabled and injured persons in reaching assembly areas when conditions permit.
6. Stand by to identify yourself and provide information to fire personnel.

If a <u>**Fire Alarm sounds**</u>:

1. Secure any equipment you are using and switch the power off.
2. Close windows and doors behind you as you leave. Do **not** lock doors.
3. Leave the building with reasonable speed using recommended exit.
4. Follow instructions of your floor warden or deputy. Wardens will ensure evacuation of assigned rooms.
5. **DO NOT** use elevators for evacuation.
6. **DO NOT** re-enter the building until allowed to do so by the Fire Department.

INDIVIDUAL RESPONSE PROCEDURES FOR EARTHQUAKES

IF INDOORS:

Take action at the first indication of ground shaking.

1. Stay inside; move away from windows, shelves, heavy objects and furniture that may fall. Take cover under a table or a desk, or in a strong doorway (anticipate that doors may slam shut).
2. In halls, stairways or other areas where no cover is available, move to an interior wall. Turn away from windows, kneel alongside the wall, bend your head close to your knees, clasp your hands firmly behind your neck covering the sides of your head with your elbows.
3. Elevators must not be used. They are extremely vulnerable to damage from earthquakes. Ground shaking may cause counterweights and other components to be torn from their connections, causing extensive damage to the elevator cabs and operating mechanisms.
4. When exiting a building, move quickly through exits and away from buildings. Parapets and columns supporting roof overhangs may fall.
5. Assemble away from gas, sewer and power lines.

IF OUTDOORS:

1. Move to an open space away from buildings, trees and overhead power lines.
2. Lie down or crouch low to the ground (legs will not be steady) and constantly survey the area for additional hazards.

B. LABORATORY OPERATION GUIDELINES

During the operation of the laboratories, the following simple procedures and guidelines are essential and **must** be adhered to by all students.

- **FOOD, DRINKS and SMOKING** are **NOT** permitted in the laboratories.
- Before starting your experiment, make sure proper equipment and circuit connections are made as per instructions in the laboratory manual. Verify your set-up with your instructor.
- All damaged or missing equipment or parts should be reported as soon as possible to the technologist. A Repair Request form (available in each lab) must be completed before equipment can be serviced.
- Equipment should not be removed from the lab station. If equipment is required elsewhere, it is to be returned to the lab station once the requirement is finished.
- All electronic components, such as capacitors, resistors, transistors etc., must be returned to their respective storage trays when the experiment is finished.
- All leads and cables are to be returned to the wall racks when the lab is finished and oscilloscope probes are to remain with the oscilloscope.
- Benches and equipment set-ups are to be tidied up after each lab session. All garbage is to be placed in the garbage cans provided. No writing on equipment or benches is permitted.
- If students have any questions about the experiment, they should consult the instructor first and then the technologist.
- Abide by all safety rules and regulations of the laboratory.

ELECTRICAL AND COMPUTER ENGINEERING LABORATORIES

ACKNOWLEDGMENT FORM

NAME:

COURSE:

LAB INSTRUCTOR:

Experiments conducted in the Electrical and Computer Engineering laboratories involve the handling of electric and electronic equipment and circuits. Failure to handle this equipment properly may lead to injury or even fatal shock. For the safety of everyone, it is required that you understand and follow the appropriate laboratory procedures as outlined in the laboratory manuals and by your laboratory instructor.

Your signature below is your acknowledgment that you have read the *Safety Regulations* and the *Laboratory Operation Guidelines* and agree to abide by them.

STUDENT'S SIGNATURE

DATE

STUDENT NUMBER

Introduction to ECE 360 Labs

Automatic control systems are today an integral part of many modern manufacturing and industrial processes. They are used in a variety of processes ranging from controlling simple variables like position speed, temperature, etc. to complex guidance controls like aircraft and/or ship autopilots.

The objective of experiments 1 to 3 is the familiarization with the theory and the practical implementation of closed-loop control systems. Closed-loop control systems, like the one in Figure. 0.1, are used to control the output variable, which can be speed, position, temperature, etc., in such a way that it follows the input signal as close as possible. This is done by measuring the actual output signal and comparing it with the input signal (often called the reference signal). The difference between these two signals (called the error signal) is fed to the controller so as to reduce the error and bring the output of the system to the desired value. The design of a controller is usually done by considering linearized approximation models for the system components.

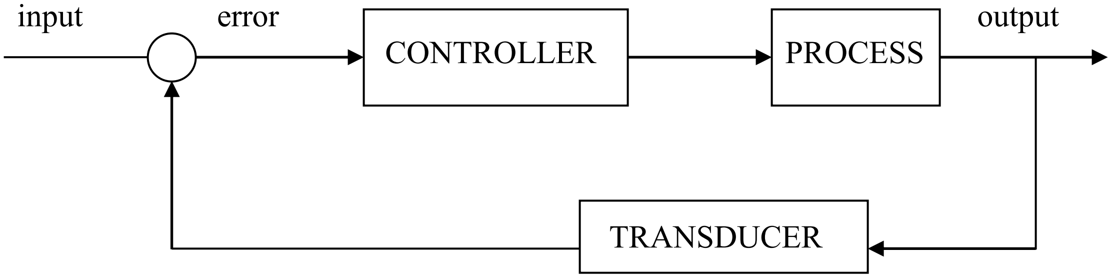


Fig. 0.1 Closed-loop Control System

The equipment to be used in experiments 1 to 3 will be a DC Motor Control Trainer (DCMCT) developed by Quanser.

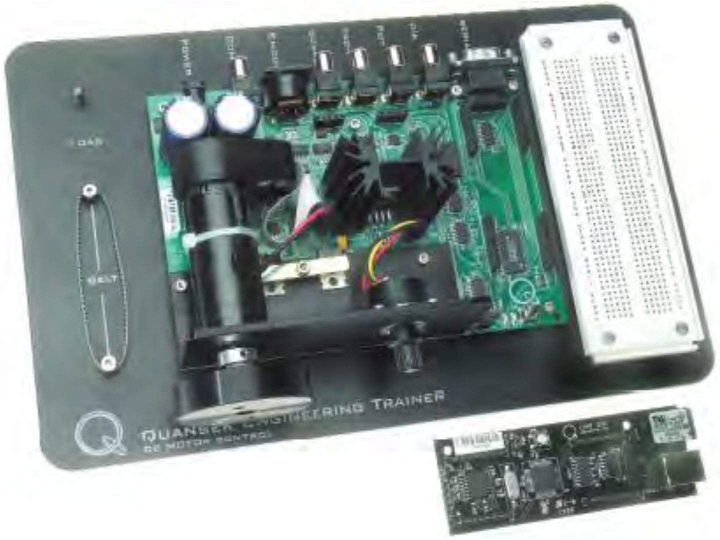


Fig. 0.2 Quanser's DC Motor Control Trainer

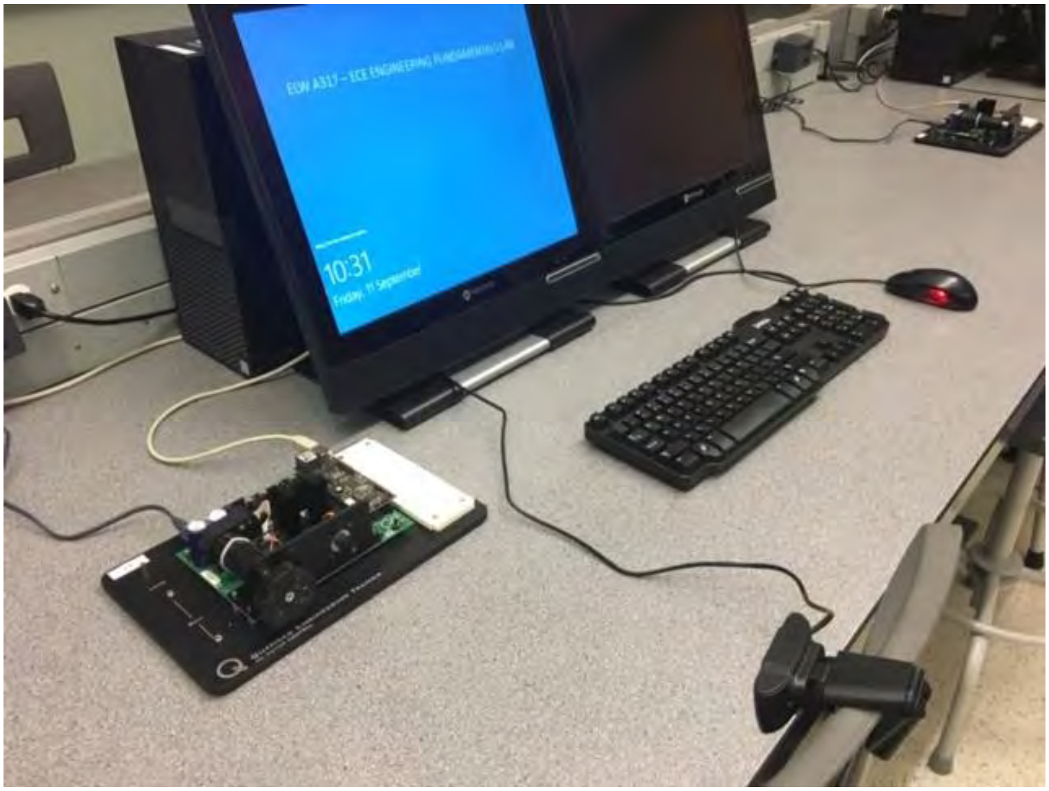
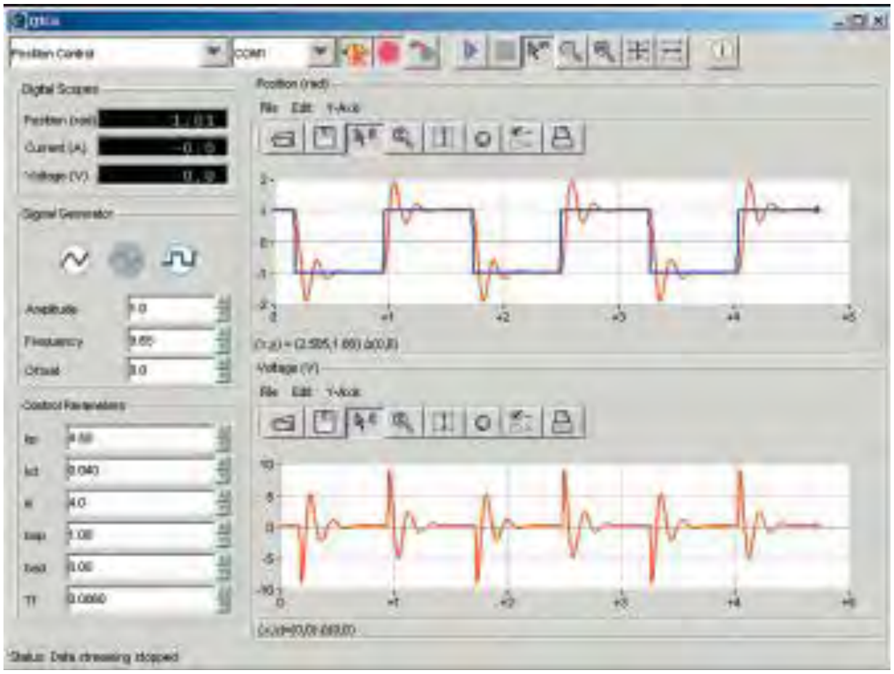


Fig. 0.3 View of a Quanser Lab station in the Control Lab

The trainer can be controlled and monitored using the software package USB QICii (USB QIC interactive interface) which runs on a PC and is connected with the trainer using a USB port. A guide for QIC can be downloaded from the ECE 360 webpage.



Position (rad) 1.01
Current (A) -0.0
Voltage (V) 0.0

Amplitude 1.0
Frequency 0.65
Offset 0.0

kp 4.50
kd 0.040
ki 4.0
bsp 1.00
bsd 0.00
Tf 0.0060

Status: Data streaming stopped

Fig. 0.3. QIC Interactive Interface Window

The experiments carried out using the DC motor trainer will deal with the following:

Experiment 1:	Modeling and Parameter Identification of a DC Motor
Experiment 2:	Speed Control Using a D.C. Motor
Experiment 3:	Position Control Using a D.C. Servo System

Experiment 4 deals with the programming of a simple robot arm and is carried out using a Lab-Volt servo robot system.

Preparation for the Laboratory Session

Each laboratory is structured in the form of **a small project consisting of a theoretical part and the implementation of the results of theory in the lab**. The theory is explained in the lab description and understanding this part is essential to be able to complete the lab.

There are also 3 youtube videos explaining the theory and discussing the Pre-Laboratory Assignments which are very helpful to understand the material. They can be found at:

https://www.youtube.com/playlist?list=PLJ-OcUCIty7ewNzOhHwqw20VoEXBHBmlz

Study the description for each experiment and **complete the Pre-Laboratory Assignments**. You will be requested to present your results at the beginning of the lab session.

IF YOUR PREPARATION IS CONSIDERED INADEQUATE, YOU WILL BE ASKED TO LEAVE AND RESCHEDULE THE EXPERIMENT IN ANOTHER LAB SESSION SUBJECT TO AVAILABILITY.

Laboratory Report

DEADLINE: A report has to be submitted for each experiment within a week after the experiment. The deadline is 6:00pm one week after the day of the lab session. If the report is handed in late, marks will be deducted.

LAB SUBMISSION: Students are required to use their Lab Section site in Brightspace to submit these reports This can be done as follows:

- Students log in their respective Lab Section in Brightspace.
- Students submit the report (as an assignment) before the deadline. Here is a video example: https://www.youtube.com/watch?v=HY0ogyh-IhQ

Laboratory Report Writing Guidelines

FORMAT: The front page of a report is shown on page 0-8 and should be used for all reports submitted.

The report should be divided in the following parts:

Summary
Introduction (Describe the objective of the lab)
Answers to the Pre-laboratory Assignments
Experimental Results (including data charts, graphs, etc)
Discussion
Conclusions

SUGESTION: Bring with you a Memory Stick to download Screendumps and Plots to use in your Report.

The following are some points you should consider while preparing your lab report:

Technical Content

Are all procedures and questions covered?
Did you discuss your results and justify your conclusions?

Presentation:

Is your write-up easy to follow, logical and complete?
Did you make an effort to write your report in precise and concise style?
Are the tables, graphs and charts clearly described and referenced (titles, units, etc.)?

Appearance

Is the general appearance of your report pleasing to the reader?
(i.e. proper spelling, grammar, printed text)

Remember:

A report is communication with other persons.
Keep in mind the reader's point of view when writing your report.
Adhere to the schedule for labs and submission of lab reports.

Acknowledgements

The help of Najith Liyanage and Lynn Palmer in the preparation of the original version of this Manual is gratefully acknowledged. The help of Paul Fedrigo, Ed Parker and Ioana Sevcenco is also acknowledged.

The use of the Quanser Manuals and the RoboCIM manual is acknowledged.

UNIVERSITY OF VICTORIA

Department of Electrical and Computer Engineering
ECE 360 – Control Systems I

Laboratory

Experiment no.: ______________________________

Title: ______________________________

Date of Experiment: ______________________________
(should be as scheduled)

Report Submitted on: ______________________________
(should be within one week from the date of the experiment)

To: ______________________________

Laboratory Group No.: ______________________________

Names: (please print)

1. ______________________________

2. ______________________________

Experiment 1: Modeling and Identification of a DC Motor

1. Objective

The objective of this laboratory project is to develop an understanding of modeling a DC motor using the Quanser's DC Motor Control Trainer (DCMCT). The model developed in this experiment will be used for control design in experiments 2 and 3.

2. Introduction

Modeling is an essential aspect of control. The key elements of modeling are:

- Get an overview of the system and its components.
- Understand the system and how it works.
- Derive a mathematical model for the system from physical laws.
- Estimating the parameters of the model using experimental tests.

An overview of the system can be obtained by pictures, schematic diagrams, and block diagrams. This gives representations of a system which emphasizes the aspects of the system that are relevant for control and suppresses many details. The block diagram gives a natural partition of the system and the mathematical descriptions of the behaviour of the subsystems representing each block can be used to obtain a complete model. In control it is often sufficient to work with linearized models where dynamics are represented by transfer functions. These transfer functions can be obtained by applying the basic physical laws that describe the subsystems or by experiments on a real system. Such modeling requires a good knowledge of the physical phenomena involved and a good sense for reasonable approximations.

Experiments on the actual physical system are a good complement to modeling. It is good practice to start the experiments by determining the static input-output characteristics of the system. For systems with several inputs as the motor in the DCMCT one can often obtain additional insight by exploiting all inputs. Models can also be obtained by exciting the system with an input and observing the system variables. The so called "bump-test" is a simple method based on a step response. Frequency response is another useful technique where a sinusoidal input is applied and the steady-state response is observed.

When a model is obtained, it is a good practice to assess the validity of the model by running simulations of the model and comparing its response with that of the actual system. Model validation is an important step that should be performed in order to give a level of confidence in the expected performance of the closed-loop system.

3. Modeling of a DC Motor

The following nomenclature, as described in Table 1.1, is used for the modeling of the DC motor:

Symbol	*Description*	*Units*
ω_m	Motor angular velocity	rad/s
u_m	Voltage from amplifier which drives the motor	V
u_e	Back-emf voltage	V
T_d	Disturbance torque externally applied to the inertial load	N m
T_m	Torque generated by motor	N m
i_m	Motor armature current	A
i_f	Motor field current	A
k_m	Motor torque constant	N.m/A
R_m	Motor armature resistance	Ω
L_m	Motor armature inductance	mH
J_m	Moment of inertia of motor rotor	kg m^2
J_l	Moment of inertia of inertial load	kg m^2
J_{eq}	Total moment of inertia of motor rotor and the load	kg m^2
K	Open-loop steady-state gain	rad/(V s)
τ	Open-loop time constant	s
M_l	Inertial load disc mass	kg
R_l	Inertial load disc radius	m
s	Laplace operator	rad/s
t	Continuous time	s

Table 1.1. DC motor system nomenclature

An essential component of many electrical position or speed control system is a DC motor with some associated power supply and amplifier stage to control the power input to the motor. A common arrangement of the DC motor is what is being called *armature connection* and is illustrated in Fig. 1.1. In armature connection, the motor speed ω_m can be controlled by the input voltage, u_m directly.

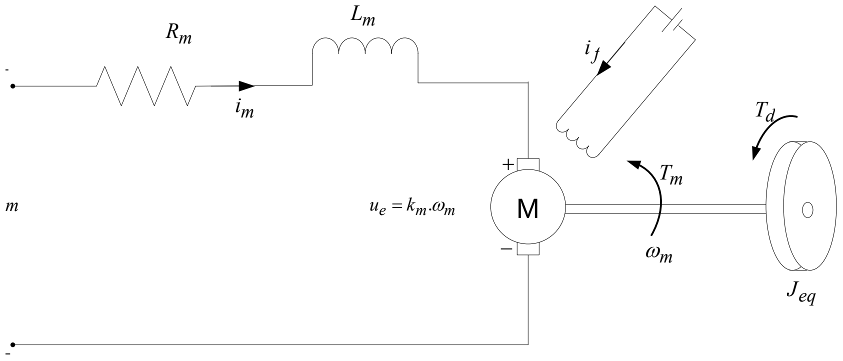


Figure 1.1. DC motor in Armature Connection

If the input voltage is increased, the speed will increase for constant load. If the motor load is increased, then the speed falls and the current increases. The increased current generates a higher torque to keep the load moving. The generated back-electro-motive-force voltage (back-emf) is proportional to the motor angular velocity:

$$u_e(t) = k_m \omega_m(t) \tag{1}$$

Note that the torque constant and the back-electro-motive-force constant is numerically equal in the International Units (SI) system and both will be denoted by *k*$_m$ here. Using the Kirchhoff's voltage law for the armature circuit of the DC motor, the following equation is obtained:

$$u_m(t) = R_m i_m(t) + L_m \left(\frac{\partial}{\partial t} i_m(t) \right) + u_e(t) \tag{2}$$

and, using eq. (1), it can expressed in the Laplace domain:

$$U_m(s) = R_m I_m(s) + sL_m I_m(s) + k_m \, \Omega_m(s) \tag{3}$$

The torque generated by the current in the armature circuit T_m is given by:

$$T_m(t) = k_m i_m(t) \tag{4}$$

Neglecting the friction in the system, the equation of motion for the DC motor's rotor (armature) is given by:

$$J_{eq} \dot{\omega}_m(t) = k_m i_m(t) + T_d(t) \tag{5}$$

where J$_{eq}$ can be obtained from:

$$J_{eq} = J_m + J_l = J_m + \frac{1}{2} M_l r_l^2 \tag{6}$$

and using the values of the system parameters given in Table A.1 in Appendix 1.1, J$_{eq}$ becomes:

$$J_{eq} = 2.21 \, x \, 10^{-5} \left[kg \; m^2 \right] \tag{7}$$

4. Pre Laboratory Questions:

4.1. Study the ***"Getting Started Guide"*** for the **QICii** interface.

You may find the short videos discussing the theory behind the preparatory questions useful:

https://www.youtube.com/watch?v=RTCznxu3W0Q&list=PLJOcUCIty7ewNzOhHwqw20VoEXBHBmlz&index=1

4.2. Static Relations

It is useful to start with a simple exploration of the system and with determining the static relations between system variables. Use the values of the system parameters given in Table A.1 in Appendix 1.1.

4.2.1. Assuming no disturbance T_d and zero friction, determine the motor maximum velocity ω_{max}. (*Hint:* When the velocity is constant and no external Torque is applied, then i_m is constant and $i_m = 0$ and eq. (2) can be used to obtain ω_{max} using u_{m_max} from table A.1.)

4.2.2. Determine the motor maximum current i_{m_max} and maximum generated torque T_{max}. (*Hint:* Maximum current is reached when $\omega_m = 0$ and can be obtained from eq. (2) using u_{m_max} from table A.1.)

4.2.3. During the in-laboratory session you will be experimentally estimating the motor resistance R_m. This can be done by applying constant voltages to the motor armature circuit and measuring the corresponding current while holding the motor shaft stationary. Derive an expression that will allow you to solve for R_m under these conditions.

4.2.4. During the in-laboratory session you will be experimentally estimating the motor torque constant k_m. This can be done by applying constant voltages to the motor armature circuit and measuring both the corresponding steady-state current and speed (in radians per second). Assuming the motor resistance is known, derive an expression that will allow you to solve for k_m. (*Hint:* A constant voltage causes the rotor to run at constant speed and a constant current i_m. Eq. (2) can be used to obtain k_m)

4.3. Derivation of the transfer function

4.3.1. Using eqs (1) to (5) and assuming $T_d(t) = 0$, derive the transfer function $G(s)$ between $U_m(s)$, the Laplace transform of the input voltage $u_m(t)$, and $\Omega_m(s)$, the Laplace transform of the angular velocity of the rotor $\omega_m(t)$.

4.3.2. Considering that $R_m >> L_m$, simplify the transfer function obtained in 4.3.1.

4.3.3. Express $G(s)$ obtained in 4.3.2. as a function of parameters a and b, defined such as:

$$G(s) = \frac{b}{s+a} \tag{8}$$

and evaluate a and b numerically using the system parameter values from table A.1.

4.3.4. Express $G(s)$ obtained in 4.3.2. as a function of parameters K and τ, defined in eq. (9) and evaluate their numerical values using the system parameter values from table A.1.

$$G(s) = \frac{K}{\tau s + 1} \tag{9}$$

4.3.5. Using eqs (1) to (5) and assuming $u_m(t) = 0$ and $L_m = 0$, derive the transfer function $G_d(s)$ between $T_d(s)$, the Laplace transform of the torque disturbance $T_d(t)$, and $\Omega_m(s)$, the Laplace transform of the angular velocity of the rotor $\omega_m(t)$.

4.3.6 Using $G(s)$ derived in 4.3.4. and $G_d(s)$ derived in 4.3.5. represent the system using the open-loop block diagram depicted in Fig. 1.2. Fill up the empty blocks. (*Hint: G(s)* can be expressed as $G(s) = (k_m/R_m)\ G_d(s)$).

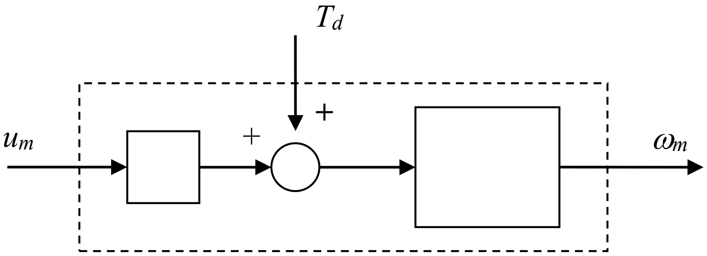


Figure 1.2. Block diagram template

4.4. Pre-Laboratory Results Summary Table

Complete Table 1.2. before you come to the in-laboratory to perform the experiments.

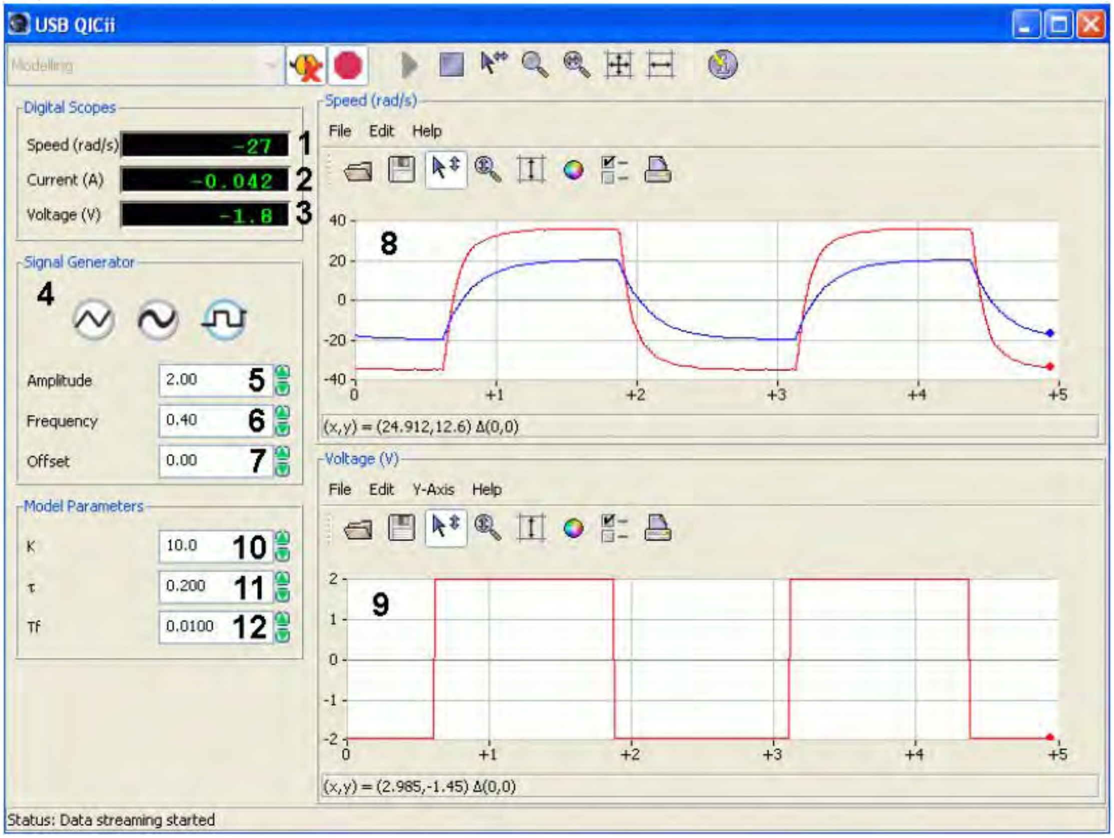
Description	*Symbol*	*Value*	*Units*
Moment of inertia of the load (see eq. (6))	J_l		kg m^2
Total moment of inertia (see eq.(7))	J_{eq}		kg m^2
Static Relations			
Motor maximum velocity	ω_{max}		rad/sec
Motor maximum current	i_{m_max}		A
Motor maximum torque	T_{max}		N m
Dynamic Models			
Open-loop model parameter *a*	a		kg m/(W s^4)
Open-loop model parameter *b*	b		rad/(V s)
Open-loop steady-state gain	K		rad/(V s)
Open-loop time constant	τ		s

Table 1.2. Modeling pre-laboratory assignment results

5. In-Laboratory Session

5.1. QICii Modeling Module

The *Modeling* module of the QICii software is the main tool for this lab. This module runs the process in open-loop using as input to the DC motor the voltage generated by the signal generator. There are two graphs that show the time histories of motor speed and motor voltage respectively.



Speed (rad/s) -27

Current (A) -0.042

Voltage (V) -1.8

Amplitude 2.00

Frequency 0.40

Offset 0.00

K 10.0

τ 0.200

Tf 0.0100

(x,y) = (24.912,12.6) Δ(0,0)

(x,y) = (2.985,-1.45) Δ(0,0)

Status: Data streaming started

Figure. 1.3 Interface of the *Modeling* module

Table 1.3 lists and describes the main elements of the *Modelling* module interface.

ID #	*Label*	*Parameter*	*Description*	*Unit*
1	Speed	ω_m	Motor Speed Numeric Display	rad/s
2	Current	i_m	Motor Armature Current Numeric Display	A
3	Voltage	u_m	Motor Input Voltage Numeric Display	V
4	Signal Generator		Type of Generator for the Angle Reference Signal: Sawtooth Wave or	

			Square Wave	
5	Amplitude		Generated Signal Amplitude Input Box	V
6	Frequency		Generated Signal Frequency Input Box	Hz
7	Offset		Generated Signal Offset Input Box	V
8	Speed	ω_m	Scope with Actual (in red) and Simulated (in blue) Angles	rad/s
9	Voltage	u_m	Scope with Applied Motor Voltage (in red)	V
10	K	K	Motor Model Steady-State Gain Input Box	rad/(V s)
11	τ	τ	Motor Model Time Constant Input Box	s
12	Tf	T_f	Time Constant of Filter for obtaining Speed from Position Measurements	s

Table 1.3 QICii *Modeling* Module Nomenclature

A simulation of the system runs in parallel with the hardware. The output of the simulation can be used for model fitting and validation. The input of the simulation is equal to the motor voltage and the output of the simulation is displayed (blue trace) in the same graph as the actual motor speed (red trace). The simulation model parameters K and τ can be adjusted from the front panel. The simulated motor speed, ω_s, is obtained from the simulated transfer function and actual motor voltage as follows:

$$\Omega_s(s) = \frac{K U_m(s)}{\tau s + 1} \qquad (10)$$

The implemented digital system in the QIC runs at 100 Hz. Thus the sampling interval is:

$$h = 0.01 \text{ [s]}.$$

The motor position is measured by an encoder generating 4096 counts per revolution. The speed (derivative of position) is obtained by filtering the position signal using the following filter:

$$\Omega_m(s) = \frac{s\,\Theta_m(s)}{T_f\,s + 1} \qquad (11)$$

where $\Theta_m(s)$ is the Laplace transform of the position. A good value for T_f is 0.01 and should be left unchanged during the experiment.

Read the Module start-up instructions in the ***"Getting Started Guide"*** **for the QICii interface.**

5.2. Static Relations

5.2.1. Initial experimental tests

The objective of this part is to get familiarized with the software interface and to make sure that a system functions properly. Please follow the steps described below:

Step 1. Set the input voltage amplitude to zero and vary the offset voltage and observe the result (do this for 3 different trials). Then, fix the offset to 5V and vary the input voltage amplitude and observe the result (do this for 3 different trials). What is the difference between varying these two parameters?

Step 2. Determine the maximum velocity and compare with calculations from 4.2.1. ***Note:*** *Do this by setting amplitude to 0V and offset to the maximum voltage to have a DC input.*

5.2.2. Estimate the motor resistance

Some of the parameters of the mathematical model of the system can be determined by measuring how the steady-state velocity and current changes with the applied voltage. To experimentally estimate the motor resistance, follow the steps described below:

Step 1 Set the generated signal amplitude to zero. If the signal offset is not zero then the motor will rotate at a constant speed due to the constant voltage applied. You can change the applied voltage by entering the desired value in the *Offset* numeric control of the *Signal Properties* box. You can also read the actual motor current from the digital display. The value is in Amperes. Fill Table 1.4. For each measurement ***hold the motor shaft stationary*** by grasping the inertial load to stall the motor.

Sample #	**u_m [V]**	**Offset in measured current i_{bias} [A]**		
0	0			
Sample #	**u_m [V]**	**Measured current i_{m_meas} [A]**	**Corrected for bias i_m [A]**	**Resistance R_m [Ω]**
1	-5			
2	-2			
3	1			
4	2			
5	5			
Average resistance R_{avg} [Ω]				

Table 1.4. Motor resistance; experimental results

Note that for zero Volts you will measure a current, i_{bias}, that is possibly non-zero. This is an offset in the measurement which you need to subtract from subsequent measurements in order to obtain the right current. Note also that the current value shown in the digital display is filtered and you must wait for the value to settle.

Step 2 The system parameters are given in Table A.1 in Appendix 1.1. Compare the estimated value for R_m (i.e. R_{avg}) with the specified value and discuss your results.

5.2.3. Estimate the motor torque constant

To experimentally estimate the motor back-emf constant (k_m) follow the following steps:

Step 1 ***With the motor free to spin*** apply the same procedure as the one used to measure motor resistance and fill Table 1.5. You can read the value for the motor angular velocity from the digital display. Wait a few seconds after you enter a new voltage value as the displayed velocity values are low-pas filtered. Calculate the motor back-emf constant k_m for each measurement and then calculate the average. Note that i_m.is small and can be neglected.

Sample #	**u_m [V]**	**Measured speed ω_m [rad/s]**	**k_m [V.s/rad]**
1	5		
2	2		
3	1		
4	-2		
5	-5		
Average back-emf constant $k_{m\ avg}$ [V.s/rad]			

Table 1.5. Back-emf constant; experimental results

Step 2 The system parameters are given in Table A.1. Compare the estimated value for k_m, k_{m_avg}, with the specified value and discuss your results

5.2.4. Obtain the motor transfer function

From the above estimates, obtain a numerical expression for the motor open-loop transfer function $G(s)$. What are the estimated open-loop steady-state gain K and time constant τ? How do they compare with the open-loop transfer function you obtained in the pre-laboratory questions.

$$G(s) = \frac{K}{\tau s + 1} \tag{12}$$

5.3. Dynamic model: Experimental Determination of System Dynamics

A linear model of a system can also be determined purely experimentally. The idea is simply to observe how much a system reacts to different inputs and to change structure and parameters of the model until a reasonable fit is obtained. The inputs can be chosen in many different ways and there is a large variety of methods.

5.3.1. The bump-test

The bump-test is a simple test based on a step response of a stable system. It is carried out in the following way. A constant input is applied. A stable system will then reach equilibrium. The input is then changed rapidly to a new level and the output is recorded. A simple model of the form:

$$G(s) = \frac{K}{\tau s + 1} \tag{13}$$

can be easily fitted to the output data. A bump-test is illustrated in Fig. 1.4.

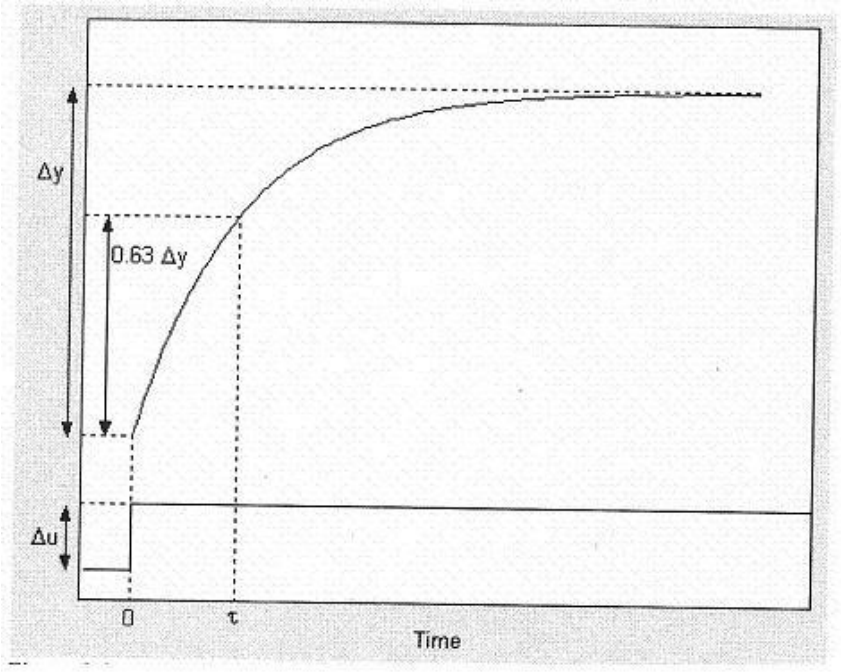


Figure 1.4. Bump-test input and output

Assuming that the input change is *Δu,* the output of the system given by eq. (13) will be given by

$$y(t) = \Delta u K(1 - e^{\frac{-t}{\tau}}) \tag{14}$$

With *Δy*, the change in the steady-state output, an estimate of the steady-state gain is given by

$$K = \frac{\Delta y}{\Delta u} \tag{15}$$

Using eqs (14) and (15) follows that

$$y(\tau) = \Delta y(1 - e^{-1}) = \Delta y \; 0.63 \tag{16}$$

i.e, the time constant τ can be is approximately given by the time necessary for the output to reach 63% of its steady-state level.

Experimental procedure:

Step 1 Set the values of the simulated system *K* to 10 and τ to zero. Apply a square wave of amplitude 2 V, offset 3 V and frequency of 0.4 Hz.

Step 2 Determine the parameters *K* and τ of the model defined in (13) (see fig. 1.4) and compare them with the model obtained by in the pre-laboratory questions.

5.4. Model validation

A simple form of model validation can be done because the module *Modeling* contains a first-order simulation, whose model expressed in (10) is driven by the actual open-loop motor voltage. This model is running in parallel with the motor which can be used for model fitting. The simulation parameters *K* and τ can be adjusted from the GUI. The output of the model is displayed together with the actual motor speed. You can explore this in the following procedure.

Experimental procedure:

Step 1. Set the *K* and τ values that were determined in the pre-lab exercise into the model parameter. Compare the model with the actual motor response, and take screen shot of the of the model and motor response plot.

Step 2. Repeat Step 1 with the model parameters that you have obtained from section 5.2.4.

Step 3. Set *K* and τ to the values you previously estimated from the bump-test in 5.3.1. Alter the simulation parameters *K* and τ □□and observe the effect.

Step 4. Try to improve the simulated response match by adjusting the simulation parameters K and τ on-the-fly until you have a good fit. This procedure is called model fitting. Record the values that give you the best fit. Compare this value with the values that you have obtained from the other methods. Comment on the differences.

6. Concluding Remarks

An important question to be asked is how accurately is the linear model developed in this experiment representing the behaviour of the real system. To answer this question it is useful to consider the unmodeled dynamics of the system. The system has some obvious nonlinearities:

- Saturation of the motor amplifier at 15 V.
- Coulomb friction in the motor. The motor starts turning when the input voltage is above about 0.4 V required to counteract friction.
- Quantization error due to the finite resolution of the encoder.

Further, L_m has been assumed to be zero, the output position signal is sampled and the system has no sensor for velocity. The velocity is obtained by filtering the position measurements. This leads to a delay of 1 to two sampling points and possible additional noise due to filtering.

Among all the above, saturation, friction and the computing of the speed from position measurements are the unmodeled dynamics which have the most effect. It can be said that:

the first order linear model, eq. (9), developed in this experiment gives an acceptable approximation of the system behaviour in most cases, provided that the system is operated above the voltage where friction is dominant and below the voltage where saturation occurs.

6. References

[1] K. Ogata, Modern Control Engineering, 5th Edition, Prentice Hall, Englewood Cliffs, N.J., 2010.

[2] K. J. Astrom, J. Apkarian, H. Lacheray, USB QICii Laboratory Workbook, Quanser Engineering Trainer Series.

7. Appendix 1.1

Table A.1 - System Parameters

	Value	**Units**	**Symbol**
Motor			
Torque constant	0.0502	Nm/Amp	k_m
Terminal resistance	10.6	Ω	R_m
Terminal Inductance	0.82	mHenry	L_m
Rotor Inertia	11.6	g cm^2	J_m
Max Torque	0.07	Nm	T_{max}
Mechanical Time Constant	0.005	s	t_m
Inertial load disc mass	0.068	kg	M_l
Inertial load disc radius	0.0248	m	r_l
Linear Amplifier			
Gain	3.0	V / V	G_a
Max output voltage	15	V	u_{m_max}
Max current	1.5	Ampere	i_{m_max}
Max output power	22	Watt	P_{max}
Max dissipated power (with heat sink) R_{load} = 4 Ohm	8	Watt	P_{dis}
Current sense			
Current sensitivity (+/- 10%)	0.556	Amp/Volt	Gcurr
Encoder			
Lines per revolution	1024	Lines	
Resolution-Quadrature	0.0879	Deg / count	G_{Enc}
Type		TTL	
Signals		A, B, Index	
Tachometer (Analog output - digitally derived from Encoder)			
Sensitivity	1.5	V / 1000 RPM	G_{tach}

Experiment 2: Speed Control using a DC Motor

1. Objective

The objective of this laboratory is to develop an understanding of Proportional and Integral (PI) control as applied to a speed control application. In particular you will explore:

- Qualitative properties of proportional and integral action.
- Design of PI controllers for given specifications.
- Response of a PI controlled system to load disturbances.

2. Introduction

The following nomenclature, as described in Table 2.1, is used for the system modeling and control design.

Symbol	*Description*	*Units*
ω_m	Motor angular velocity	rad/s
u_m	Voltage from amplifier which drives the motor	V
u_e	Back-emf voltage	V
T_m	Torque generated by motor	N m
T_d	Disturbance torque externally applied to the inertial load	N m
V_d	Disturbance voltage corresponding to T_d	V
V_{sd}	Simulated disturbance voltage	V
i_m	Motor armature current	A
k_m	Motor torque constant	N.m/A
R_m	Motor armature resistance	Ω
J_{eq}	Total moment of inertia of motor rotor and the load	kg m^2
K	Open-loop steady-state gain	rad/(V s)
τ	Open-loop time constant	s
ω_n	Undamped Natural Frequency	rad
ζ	Damping Ratio	
T_s	2% Settling Time	*s*
h	Sampling interval	s
k_p	Proportional gain	V s/rad
k_i	Integral gain	V/rad
u	Control signal	V
r	Reference signal	rad/s
y	Measured process output	rad/s

Table 2.1. System speed control nomenclature

Consider a DC motor whose angular velocity *$\omega_m(t)$* is supposed to follow a certain reference signal *$r(t)$*. Assuming that the speed of the motor is measured, the difference between the measured speed *$\omega_m(t)$* and the desired speed *$r(t)$* can be used to control the system. The structure of such a feedback control system is given in Fig. 2.1. The block labeled "Motor" represents a DC motor such as the one studied in experiment 1 and has voltage u_m and disturbance torque T_d as inputs and motor speed ω_m as the output. The controller is usually designed so that the output of the closed-loop system satisfies certain specifications with respect to transient and steady-state performance. Another effect of the controller is to reduce the influence of disturbances. A torque on the motor axis, such as T_d, is a typical example of a load disturbance for a speed control system. In this experiment, the disturbance torque is typically a torque that you apply manually to the inertial load. The effects of such a disturbance will also be considered in this experiment.

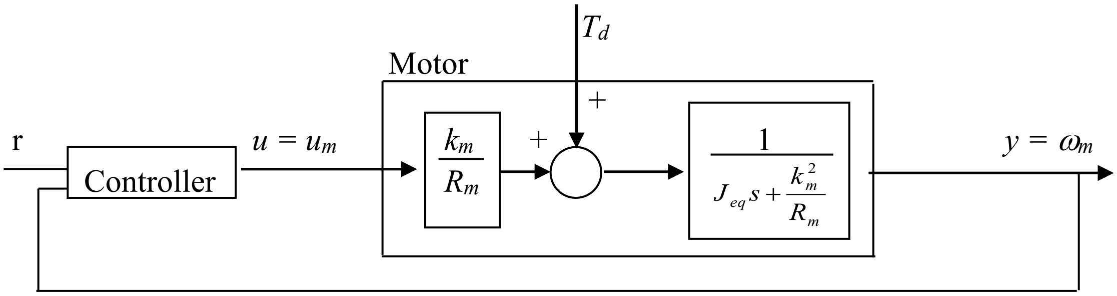


Figure 2.1. Block diagram of a speed control system

Some of the most commonly used control algorithms are Proportional (P), Integral (I) and Proportional-Integral (PI) control. The PI controller is used for a variety of purposes and it often works very well for systems with simple dynamics with respect to transient and steady-state response. For systems with complicated dynamics, it can often give good performance provided that specifications are not too demanding. Better performance can be obtained by using more complicated controllers (PID, led-lag compensators etc.).

3. PI Control Law

The controller function in Fig. 2.1 to be used in this lab can be described using the following equation:

$$u_m(t) = k_p(b_{sp}r(t) - \omega_m(t)) + k_i\int_0^t (r(\tau) - \omega_m(\tau))d\tau \qquad (1)$$

where $u_m(t)$ is the control signal, $r(t)$ the reference signal (also called the set point), $\omega_m(t)$ the measured output and

k_p = the proportional gain
k_i = the integral gain
b_{sp}= reference signal weight

This is a linear Proportional and Integral (PI) controller and its response is governed by the above three parameters. The Proportional (P) controller is the special case of (1) with $k_i = 0$ and $b_{sp} = 1$ while Integral (I) controller is the case with $k_p = 0$.

The error signal $e(t)$ is an important signal when evaluating the performance of the system. It is given by

$$e(t) = r(t) - \omega_m(t) \tag{2}$$

An important property of the controller with integral action is that always leads to the correct steady-state value, provided that the closed-loop system is stable and therefore there exists an equilibrium. With this assumption, at steady-state we have

$$r(t\,) = r_{ss} \qquad u(t) = u_{ss} \qquad \text{and} \qquad \omega_m(t) = \omega_{mss} \tag{3}$$

Eq. (1) then implies that

$$u_{ss} = k_p(r_{ss} - \omega_{mss}) + k_i(r_{ss} - \omega_{mss})t \tag{4}$$

Since the left hand side is a constant, the right hand side must also be a constant which implies that

$$\omega_{mss} = r_{ss} \quad and \quad e_{ss} = 0 \tag{5}$$

where e_{ss} is the steady-state error.

3.1. Manual Tuning of PI Controller: Ziegler-Nichols

Manual tuning procedures are generally used when no mathematical model of the system is available to perform control system design. The Ziegler-Nichols method is a classical tuning rule. Typically in manual control tuning, we first set integral gain to zero and increase proportional gain until the system reaches the stability boundary. At this point a stable output oscillation is achieved. The critical gain k_{pc}, where this occurs and the frequency of the oscillation T_{pc} are determined. A similar test with pure integral control gives k_{ic} and T_{ic}. The values k_{pc} and k_{ic} give the ranges for the gains. Suitable values can then be determined empirically or by traditional tuning rules.

The Ziegler-Nichols closed-loop method recommends the following PI controller gain tuning:

$$k_p = 0.4\, k_{pc} \tag{6}$$

$$T_i = 0.8\, T_{pc} \tag{7}$$

$$k_i = k_p \,/\, T_i \qquad \text{or} \qquad k_i = 0.5 * k_{pc} \,/\, T_{pc} \tag{8}$$

J.G. Ziegler and N.B. Nichols experimentally developed in the early forties the above tuning rules based on closed-loop tests. However the Ziegler-Nichols method suffers from one major drawback: the physical system has to tolerate to be brought into a critically stable state without catastrophic consequences. For example, sustained oscillation is generally out of the question for many industrial processes limiting the use of the Ziegler Nichols tuning rules.

4. Pre-Laboratory Assignments

You may find the short videos discussing the theory behind the preparatory questions useful:
https://www.youtube.com/watch?v=aY3V3Cl5yAk&list=PLJ-OcUCIty7ewNzOhHwqw20VoEXBHBmlz&index=2

4.1. Proportional Control

4.1.1. Using the open-loop transfer function of the DC motor with the values of *K* and *τ* as obtained in experiment 1

$$G(s) = \frac{\Omega_m(s)}{U_m(s)} = \frac{K}{\tau s + 1} \tag{9}$$

and proportional control,

$$u_m(t) = k_p(r(t) - \omega_m(t)) \tag{10}$$

obtain the closed-loop transfer function $G_P(s)$ between the reference signal *r(t)* as input and the motor speed ω_m as output (Assume disturbance torque $T_d = 0$).

4.1.2. Determine the location of the poles of the closed-loop system when the proportional gain k_p is changed. That is, derive the poles of the closed loop system as a function of k_p. How does the unit step response of the system change when k_p is changed?

4.1.3. Consider a step with amplitude r_0 for *r(t)* and use the Final Value Theorem to find the value of the output signal ω_m of the closed-loop system at steady-state. How does it compare to the steady-state value of the input signal *r(t)*?

4.2. Integral Control

4.2.1. Using the open-loop transfer function of the DC motor obtained in experiment 1, eq. (9) and integral control

$$u_m(t) = k_i \int_0^t (r(\tau) - \omega_m(\tau))d\tau \tag{11}$$

obtain the closed-loop transfer function *$G_I(s)$* between the reference signal *$r(t)$* as input and the motor speed ω_m as output (Assume disturbance torque $T_d = 0$).

4.2.2. Determine the location of the poles of the closed-loop system when the integral gain k_i is changed. That is, derive the poles of the closed loop system as a function of k_i. How does the unit step response of the system change when k_i is changed?

4.2.3. Consider a step with amplitude r_0 for $r(t)$ and use the Final Value Theorem to find the value of the output signal ω_m of the closed-loop system at steady-state. How does it compare to the steady-state value of the input signal $r(t)$?

4.3. Proportional and Integral Control

4.3.1. Using the open-loop transfer function of the DC motor obtained in experiment 1, eq. (9) and PI control

$$u_m(t) = k_p(b_{sp} r(t) - \omega_m(t)) + k_i \int_0^t (r(\tau) - \omega_m(\tau))d\tau \tag{12}$$

obtain the closed-loop transfer function *$G_{PI}(s)$* between the reference signal *$r(t)$* as input and the motor speed ω_m as output (Assume disturbance torque $T_d = 0$).

4.3.2. One possible way to design a controller is to choose controller gains which give a specified denominator polynomial for the closed-loop system (i.e characteristic polynomial). For a second order system, the controller gains can be chosen in a way resulting in the following transfer function:

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n\, s + \omega_n^2} \tag{13}$$

Determine the PI controller parameters k_p, b_{sp} and k_i so that the closed-loop system, i.e *$G_{PI}(s)$,* satisfies the specified transfer function (13). That is, derive k_p, b_{sp} and k_i as functions of ω_n, ζ, K, and τ.

4.3.3. Determine k_p, b_{sp} k_i, that give ω_n = 16 [rad/sec] and $\zeta=1$. Determine also the corresponding poles of the closed-loop system and the 2% settling time T$_s$

$$T_s = \frac{4}{\varsigma\ \omega_n} \tag{14}$$

4.3.4. Using the Final Value Theorem, compute ω_{ss_PI}, the steady-state value of the output signal for a unit step in the input. Roughly plot the step response of the closed-loop system.

4.4. Closed-loop System's response to disturbances

4.4.1. In Fig. 2.1. consider the case where r*(t)* = 0 and a PI controller is being used. Determine the closed-loop system transfer function $G_D(s)$

$$G_D(s) = \frac{\Omega(s)}{T_d(s)} \tag{15}$$

where a disturbance torque T_d is applied on the inertial load as input and the motor speed ω_m is the output. Express $G_D(s)$ as a function of the following system parameters: k_p, k_i, K, τ and J_{eq} .

4.4.2. When a proportional controller is used ($k_p \neq 0$ and $k_i = 0$), apply the Final Value Theorem to calculate the steady-state velocity, ω_{ss_P}, in response to a step input disturbance torque of amplitude T_{d0}. Comment.

4.4.3. When a integral controller is used ($k_p = 0$ and $k_i \neq 0$), apply the Final Value Theorem to calculate the steady-state velocity, ω_{ss_I}, in response to a step input disturbance torque of amplitude T_{d0}. Comment.

4.5. Pre-Laboratory Results Summary Table

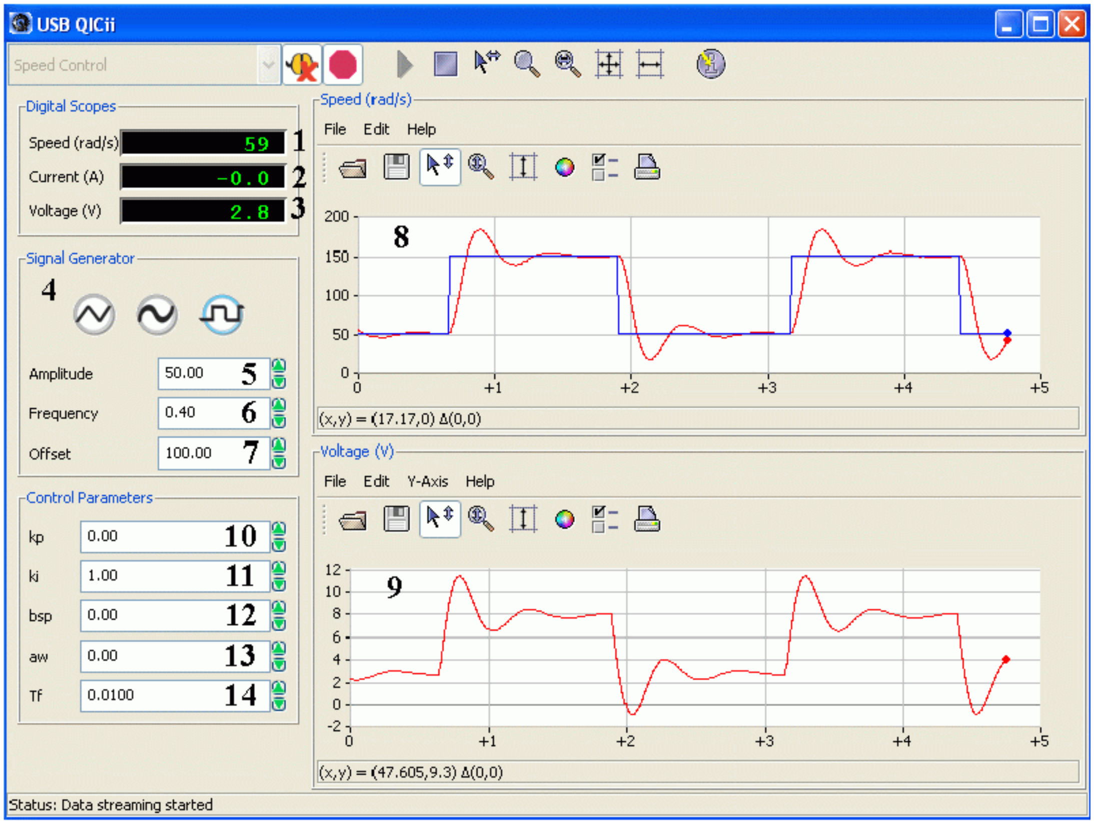
<u>Complete Table 2.2. before you come to the in-laboratory to perform the experiments.</u>

Description	*Symbol*	*Value*	*Units*
Open-loop steady-state gain t	K		rad/(V s)
Open-loop time constant	τ		s
PI Controller Design			
Given Damping Ratio	ζ		
Given Undamped Natural Frequency	ω_n		
Proportional Gain	k_p		(V s)/rad
Integral Gain	k_i		V/rad
Closed-Loop Poles			
2% Settling Time	T_s		s
Output at Steady-State using PI control	ω_{ss_PI}		rad
Response To Load Disturbances			
Steady-State Velocity, P control	ω_{ss_P}		rad
Steady-State Velocity, I control	ω_{ss_I}		rad

Table 2.2. Speed Control pre-laboratory assignment results

5. In-Laboratory Session

The *Speed Control* module of the QICii software package is the main tool for this laboratory.



Speed (rad/s) 59

Current (A) -0.0

Voltage (V) 2.8

Amplitude 50.00

Frequency 0.40

Offset 100.00

kp 0.00

ki 1.00

bsp 0.00

aw 0.00

Tf 0.0100

(x,y) = (17.17,0) Δ(0,0)

(x,y) = (47.605,9.3) Δ(0,0)

Status: Data streaming started

Figure 2.2. Interface of the *Speed Control* module

Table 2.3 lists and describes the main elements of the *Speed Control* module interface.

ID #	***Label***	***Parameter***	***Description***	***Unit***
1	Speed	ω_m	Motor Speed Numeric Display	rad/s
2	Current	i_m	Motor Armature Current Numeric Display	A
3	Voltage	u_m	Motor Input Voltage Numeric Display	V
4	Signal Generator		Type of Generator for the Angle Reference Signal: Sawtooth Wave or Square Wave	
5	Amplitude		Generated Signal Amplitude Input Box	rad/s
6	Frequency		Generated Signal Frequency Input Box	Hz

7	Offset		Generated Signal Offset Input Box	rad/s
8	Speed	ω_m	Scope with Actual (in red) and Reference (in blue) Angles	rad/s
9	Voltage	u_m	Scope with Applied Motor Voltage (in red)	V
10	kp	k_p	Control Proportional Gain Input Box	(V s)/rad
11	ki	k_i	Control Integral Gain Input Box	V/rad
12	bsp	b_{sp}	Controller Set-Point Weight Input Box	
13	aw	a_w	Controller Windup Protection Parameter Input Box	
14	$\mathrm{T_f}$	T_f	Time Constant of Filter for obtaining Speed from Position Measurements	s

Table 2.3 QICii *Speed Control* Module Nomenclature

The *Speed Control* module runs the DC motor in closed-loop with the motor reference speed generated by the signal generator. The motor speed is obtained by filtering the position signal using:

$$\Omega_m(s) = \frac{s\,\Theta_m(s)}{T_f\,s+1} \tag{16}$$

T_f should be set at 0.01 and all three should be left unchanged during the experiment. Further, $\mathrm{a_w}$ is a parameter used to protect the integrator output from saturation and should be set to zero during this experiments.

5.1. Properties of Proportional and Integral Control

The goal of the following procedures is to develop an intuitive feel for the properties of proportional and integral control and compare the experimental results with the analysis in the pre-laboratory assignments.

5.1.1. Proportional Control

Start by exploring the properties of pure proportional control. Please follow the steps below:

Step 1 Set the reference signal to a square wave. Reasonable amplitude would be 50 rad/s. When you change the reference signal level ensure that the control signal does not saturate. It may be useful to adjust the *Offset* of the signal generator so that the signal is not clipped. Set $b_{sp} = 1$ and the proportional gain to 0.04 V.s/rad to start with. Ensure that the following parameters in the GUI, as displayed in Table 2.4, are set properly.

Signal Type	***Amplitude* [rad/s]**	***Frequency* [Hz]**	***Offset* [rad/s]**	**k_i [V/rad]**	**k_p [V.s/rad]**
Square Wave	50	0.4	50	0	0.04

Table 2.4. Default parameters for the speed control module

Step 2 Change the proportional gain k_p from 0.1 V s/rad to 0.4 V s/rad by incremental steps of 0.1 V s/rad to investigate the closed-loop system for proportional controllers (k_i = 0) with different gains. What are your observations?

Step 3 Observe and describe the steady-state error to a step input, as k_p is increased.

Step 4 Summarize your observations in your report and include some representative results, screen captures, and plots. Discuss how your plots compare with the analysis in 4.1.

5.1.2. Integral Control

Step 1 Set the proportional gain to zero. Set the integral gain to 0.4 V/rad to start with. Ensure that the parameters are set as listed in Table 2.5.

Signal Type	***Amplitude* [rad/s]**	***Frequency* [Hz]**	***Offset* [rad/s]**	**k_p [V.s/rad]**	**k_i [V/rad]**
Square Wave	50	0.4	100	0	0.4

Table 2.5. Module parameters for the pure integral control test

Step 2 Change the integral gain by by increasing or decreasing it with steps of about 0.5 V/rad to investigate the closed-loop system for integral controllers (k_p = 0) with different gains. What are your observations?

Step 3 Determine a value of integral gain which gives the quickest response without overshooting. Determine the settling time for this closed loop system.

Step 4 Summarize your observations in your report. Select some representative results, screen captures, and plots. Discuss how your plots compare with the analysis in 4.2.

5.1.3. Proportional and Integral Control

The combination of proportional and integral control will now be explored. Please follow the steps below.

Step 1. Set the parameters as listed in Table 2.6.

Signal Type	*Amplitude* **[rad/s]**	*Frequency* **[Hz]**	*Offset* **[rad/s*]***
Square Wave	50	0.4	100

Table 2.6. Module parameters for the proportional and integral control test

Step 2. Set b_{sp} = 1, proportional gain to k_p = 0.1 V s/rad, and change integral gain k_i in the range of 0.5 to 5 V/rad. Observe the tracking error (difference between input and output signals) and the control signal.

Step 3. Set integral gain to k_i = 0.5 V/rad, b_{sp} = 1 and change proportional gain k_p in the range of 0.05 to 0.3 V s/rad. Observe the tracking error and the control signal.

Step 4. Set b_{sp}, the proportional and integral gains to the values obtained in section 4.3.3. Observe the tracking error and the control signal.

Step 5. Summarize your observations in your report. Select some representative results, screen captures, and plots. Discuss how your plots compare with the analysis in 4.3.

5.2. Close-loop System's Response to Disturbances

This session carries out an experimental investigation of the system response to load disturbances. A load disturbance can be introduced by manually applying a torque to the inertial load (Use your finger to slow down the wheel without bringing it to a complete stop).

Please follow the steps below

Step 1. The response to disturbance T_d at a constant reference speed of 150 rad/s, is investigated. Set the signal generator module parameters to 0 [rad/s] Amplitude and 150 [rad/s] Offset.

Step 2. Choose a pure proportional controller (k_i = 0, b_{sp} = 1) with gain k_p = 0.20 V·s/rad. Apply a torque manually by gently touching the inertial load with your finger. Observe what happens when you change the gain of the controller.

Step 3. Choose a controller with pure integral action ($k_p = 0$), such that $k_i = 1.0$ V/rad. Apply a disturbance torque manually and observe what happens.

Step 4. Observe the response of the system output ω_m to the external disturbance when using proportional control and when using integral control. Summarize your observations and your calculations in your report. Select some representative results, screen captures, and plots.

5.3. Manual Tuning of PI Controller: Ziegler-Nichols

This part of the experiment should illustrate the performance of the closed-loop system with a manually tuned PI controller and compare its performance with the previous controllers.

Please follow the steps below.

Step 1. Determine the critical gain, k_{pc}, ($k_i = 0$, $b_{sp} = 1$) where the system becomes critically stable and a stable oscillation is achieved. Also determine the critical period T_{pc} of the corresponding oscillations (Refer to section 3.1 for procedures). Using these values determine the Ziegler-Nichols controller gains using the equations in 3.1.

Description	*Symbol*	*In-Lab Result*	*Units*
Properties of PI Control			
Critical proportional gain	k_{pc}		V·s/rad
Critical period for k_{pc}	T_{pc}		s
Ziegler-Nichols design			
Proportional gain	k_p		V·s/rad
Integral gain	k_i		V/rad

Step 2. Set the parameters of the signal generator module window as listed in Table 2.7.

Signal Type	***Amplitude* [rad/s]**	***Frequency* [Hz]**	***Offset* [rad/s]**
Square Wave	50	0.5	150

Table 2.7. Module parameters for the Ziegler-Nichols-tuned PI controller

Set $b_{sp} = 1$, both proportional and integral gains to their Ziegler-Nichols values as calculated. above. What are your observations?

Step 3. Adjust proportional and integral gain manually to give a very slightly under-damped response with no saturation of the control signal (the system is saturated when changing the control signal does not have any effect on the output signal). Comment on the new gain values.

Step 4. How does the response of the gain values of the Ziegler-Nichols compare with the previous controllers in 5.1.1, 5.1.2. and 5.1.3.? Explain your observations.

Step 5. Summarize your observations and your calculations in your report. Select some representative results, screen captures, and plots.

6. References

[1] K. Ogata, Modern Control Engineering, 5th Edition, Prentice Hall, Englewood Cliffs, N.J., 2010.

[2] K. J. Astrom, J. Apkarian, H. Lacheray, USB QICii Laboratory Workbook, Quanser Engineering Trainer Series.

Experiment 3: Position Control Using a DC Motor

1. Objective

The objective of this laboratory project is to develop an understanding of Proportional and Derivative (PD) Control as applied to a position control application. In particular you will explore:

- Qualitative properties of proportional and derivative action.
- Design of controllers for specifications on the set-point response.
- Tracking of triangular signals.

2. Introduction

The following nomenclature, as described in Table 3.1, is used

Symbol	*Description*	*Units*
θ_m	Motor angle	rad
ω_m	Motor angular velocity	rad/s
u_m	Voltage from amplifier which drives the motor	V
u_e	Back-emf voltage	V
T_m	Torque generated by motor	N m
T_d	Disturbance torque externally applied to the inertial load	N m
V_d	Disturbance voltage corresponding to T_d	V
V_{sd}	Simulated disturbance voltage	V
i_m	Motor armature current	A
k_m	Motor torque constant	N.m/A
R_m	Motor armature resistance	Ω
J_{eq}	Total moment of inertia of motor rotor and the load	kg m^2
K	Open-loop steady-state gain	rad/(V s)
τ	Open-loop time constant	s
ω_n	Undamped Natural Frequency	rad
ζ	Damping Ratio	
k_p	Proportional gain	V s/rad
k_d	Derivative gain	V/rad
b_{sp}	Set-Point Weight on proportional Control	
b_{sd}	Set-Point weight on derivative Control	
u	Control signal	V
r	Reference signal	rad/s
y	Measured process output	rad/s

Table 3.1. Nomenclature used for position control

Consider a DC motor whose angular position *$\theta_m(t)$* is supposed to follow a reference signal *$r(t)$*. Such a system can be be represented by the block diagram in Figure 3.1. This block diagram illustrates the parts of the system that are relevant for position control:

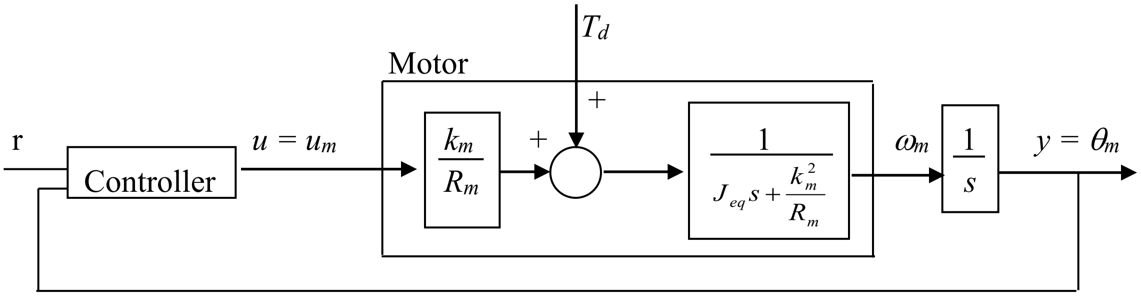


Figure 3.1. Block diagram of a position control system

The process is represented by a block which has input voltage u_m and torque T_d as inputs and motor angle θ_m as the output. The torque is typically a disturbance torque similar to the disturbance torque used in experiment 2. The controller to be used in this experiment will be proportional and derivative (PD) control.

2.1. PD Control Law

The controller function in Fig. 3.1 can be described using the following equation:

$$u_m(t) = k_p(b_{sp}r(t) - \theta_m(t)) + k_d[b_{sd}\{\frac{\partial}{\partial t}r(t)\} - \{\frac{\partial}{\partial t}\theta_m(t)\}] \qquad (1)$$

where

k_p	Proportional Gain
k_d	Derivative Gain
b_{sp}	Reference Signal Weight for Proportional Part
b_{sd}	Reference Signal Weight for Derivative Part

The derivative term can be seen as a predictor of future measurements and improves the possibility of introducing damping. One possible disadvantage of derivative control is that it may be noise sensitive due to differentiation.

3. Pre-Laboratory Assignments

You may find the short videos discussing the theory behind the preparatory questions useful: https://www.youtube.com/watch?v=QX4b-9BcIY4&list=PLJ-OcUCIty7ewNzOhHwqw20VoEXBHBmlz&index=3

3.1 Proportional Control

3.1.1. The open-loop transfer function between the control input $u_m(t)$ and the position signal $\theta_m(t)$ of the DC motor in Fig. 3.1 can be given by

$$G(s) = \frac{\Theta_m(s)}{U_m(s)} = \frac{K}{\tau s + 1} \times \frac{1}{s} \tag{2}$$

Using the values of K and τ obtained in experiment 1 and proportional control,

$$u_m(t) = k_p(r(t) - \theta_m(t)) \tag{3}$$

(i.e. $k_d = 0$, $b_{s\,p}=1$ in eq. (1)) obtain the closed-loop transfer function $G_P(s)$ between the reference signal $r(t)$ as input and the motor position $\theta_m(t)$ as output (Assume disturbance torque $T_d = 0$).

3.1.2. Determine the location of the poles of the closed-loop system when the proportional gain k_p is changed. That is, derive the poles of the closed loop system as a function of k_p. How does the unit step response of the system change when k_p is changed?

3.1.3. Consider a step of amplitude r_0 for $r(t)$ and use the Final Value Theorem to find θ_{SS_P} the value of the output signal of the closed-loop system at steady state. How does it compare to the input signal $r(t)$?

3.2 Design of Proportional and Derivative Control Parameters

3.2.1. Using the open-loop transfer function, eq. (2) and PD control, eq. (1) with $b_{sp}=1$, obtain the closed-loop transfer function $G_{PD}(s)$ between the reference signal $r(t)$ as input and the motor position $\theta_m(t)$ as output (Assume disturbance torque $T_d = 0$).

3.2.2. One possible way to design a controller is to choose controller parameters that give a specified transfer function. The controller parameters can be determined by using the mathematical model of the process and applying pole placement design. Determine the PD controller parameters k_p, k_d, and b_{sd} so that the closed-loop system, i.e. $G_{PD}(s)$, becomes the following quadratic transfer function:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4}$$

In other words, derive k_p, k_d, and b_{sd} as functions of ω_n, ζ, K, and τ.

3.2.3. For a second order system, eq. (4), with two complex conjugate poles (underdamped), the maximum Percentage Overshoot, *PO*, over the steady-state response is given by

$$PO = 100\, e^{-\left(\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)} \tag{5}$$

the time to first peak t_p is given by

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{6}$$

and the settling time is given by:

$$T_s = \frac{4}{\varsigma\,\omega_n} \tag{7}$$

Using the following values as design specifications:

$$PO \le 18\ \% \qquad \text{and} \qquad T_s = 0.43\ s \tag{8}$$

choose a damping ratio, ζ and a natural frequency ω_n, that satisfies the above requirements. Is the value $\zeta = 0.5$ acceptable? Explain.

3.2.4. Using the numerical results from 3.2.3. and the expressions derived for k_p, k_d, and b_{sd} in 3.2.2 compute k_p, k_d, and b_{sd} .

3.2.5. Consider a step with amplitude r_0 for $r(t)$ and use the Final Value Theorem to find θ_{ss_PD} the value of the output signal θ_m of the closed-loop system at steady state. How does it compare to the input signal $r(t)$?

3.3. Tracking Triangular Signals

3.3.1. Consider the closed-loop system with PD control, i.e. $G_{PD}(s)$, and assume that you have as input a ramp signal given by

$$r(t) = r_0\, t \quad for \quad t \ge 0 \tag{9}$$

Apply the final value theorem to calculate the steady-state error, e_{ss_PD}.

3.3.2. Compute e_{ss_PD} using $r_0 = 32$ [rad/s] and the values of k_p, k_d, and b_{sd} obtained in 3.2.4.

3.4. Pre-Laboratory Results Summary Table

<u>Complete Table 3.2. before you come to the in-laboratory to perform the experiments.</u>

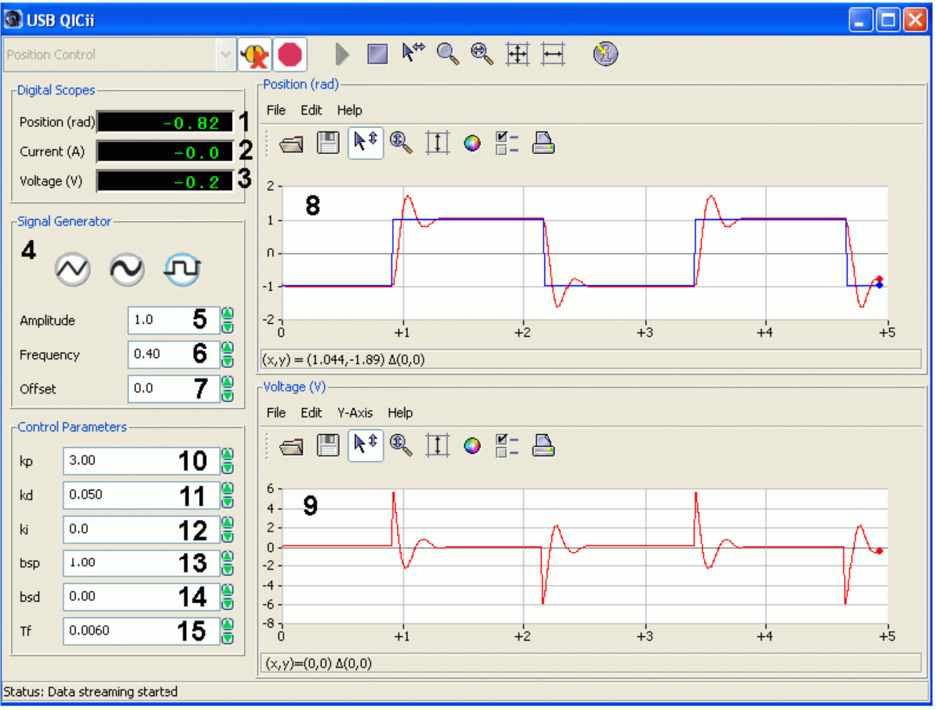
Description	*Symbol*	*Value*	*Units*
Open-loop steady-state gain t	K		rad/(V s)
Open-loop time constant	τ		s
PD Controller Design			
Proportional Gain	k_p		V/rad
Derivative Gain	k_d		V.s/rad
Set-Point Weight On Derivative Part	b_{sd}		
Desired Damping Ratio	ζ		
Desired Undamped Natural Frequency	ω_n		rad/s
Given Maximum Percentage Overshoot	PO		%
Given 2% Settling Time	T_s		s
Desired Peak Time	t_p		s
Steady-Stave Value of Position using PD	θ_{ss_PD}		rad
Tracking Triangular Signals			
Steady-State Error using PD control	e_{ss_PD}		rad

Table 3.2. Position Control pre-laboratory assignment results

4. In-Laboratory Session

4.1 Position Control Module

The *Position Control* module of the QICii software package is the main tool for this laboratory



Position (rad) -0.82
Current (A) -0.0
Voltage (V) -0.2

Amplitude 1.0
Frequency 0.40
Offset 0.0

kp 3.00
kd 0.050
ki 0.0
bsp 1.00
bsd 0.00
Tf 0.0060

(x,y) = (1.044,-1.89) Δ(0,0)

(x,y)=(0,0) Δ(0,0)

Status: Data streaming started

Figure 3.2. Interface of the *Position Control* module.

Table 3.3. lists and describes the main elements of the *Position Control* module interface.

ID #	***Label***	***Parameter***	***Description***	***Unit***
1	Position	θ_m	Motor Output position Numeric Display	rad
2	Current	i_m	Motor Armature Current Numeric Display	A
3	Voltage	u_m	Motor Input Voltage Numeric Display	V
4	Signal Generator		Type of Generator for the Angle Reference Signal: Sawtooth Wave or Square Wave	
5	Amplitude		Generated Signal Amplitude Input Box	rad
6	Frequency		Generated Signal Frequency Input Box	Hz
7	Offset		Generated Signal Offset Input Box	rad
8	Speed	ω_m	Scope with Actual (in red) and Reference (in blue) Angles	rad
9	Voltage	u_m	Scope with Applied Motor Voltage (in red)	V
10	kp	k_p	Controller Proportional Gain Input Box	V/rad
11	kd	k_d	Controller Derivative Gain Input Box	V.s/rad
12	ki	k_i	Controller Integral Gain Input Box	V/(rad.s)
13	bsp	b_{sp}	Controller Proportional Reference Signal Weight Input Box	
14	bsd	b_{sd}	Controller Derivative Reference Signal Weight Input Box	
15	Tf	T_f	Time Constant of Filter for obtaining the Derivative of the Error Signal	s

Table 3.3. QICii *Position Control* Module Nomenclature

The *Position Control* module program runs the process in closed-loop with the motor reference position angle given by the signal generator. There are two windows that show the time histories of motor position (control output) and motor voltage (control input).

To compute the control signal, eq. (1), it is required that the term:

$$e_d(t) = \frac{\partial}{\partial t}(b_{sd} r(t) - \theta_m(t)) \tag{10}$$

is computed. The signal *$e_d(t)$* is obtained by filtering the error signal using the following filter:

$$E_d(s) = \frac{s}{T_f\ s+1}[b_{sp} R(s) - \Theta_m(s)] \tag{11}$$

where *$E_d(s)$, $R(s)$* and *$\Theta_m(s)$* are the Laplace transforms of the signals *$e_d(t)$, $r(t)$* and the position *$\theta_m(t)$*. A good value for *T_f* is 0.006 and should be left unchanged during this experiment.

4.2. Quantitative Properties of Proportional and Derivative Control

The goal of the following procedures is to develop an intuitive feel for the properties of proportional and derivative control actions.

4.2.1 Proportional Control

Step 1. Set reference signal to a square wave. A reasonable amplitude is 3 rad. When you change the reference signal level ensure that the control signal does not saturate. Set both integral and derivative gains to zero (*k_i = k_d = 0*). Set the proportional gain to 0.2 V/rad to start with. Ensure that the following parameters of the Position Control window are set properly.

Signal Type	*Amplitude [rad]*	*Frequency [Hz]*	*Offset [rad]*	*k_p [V/rad]*	*b_{sp}*
Square Wave	3	0.4	0	0.2	1

Table 3.4. Module Parameters for the Proportional Control Test.

Step 2. To investigate the closed-loop system for proportional controllers with different gains change the proportional gain to the following values: *k_p = 1, 2,* and *4* V/rad. What are your observations?

Step 3. Describe the steady-state error to a step input.

Step 4. Repeat the previous observations. Change the Amplitude of the reference signal and observe under what conditions the control signal saturates

4.3 Proportional and Derivative (PD) Control

The combination of proportional and derivative control will now be explored. Follow the steps below:

Step 1. Fix the proportional gain to 2.0 V/rad and set the derivative gain to 0.0 V.s/rad to start with ($b_{sp} = b_{sd} = 1$). Set the parameters of the QICii module window as listed in Table 3.5. Set the integral gain to zero ($k_i = 0$).

Signal Type	*Amplitude [rad]*	*Frequency [Hz]*	*Offset [rad]*	k_p *[V/rad]*	k_d *[V.s/rad]*	b_{sp}	b_{sd}
Square Wave	2	0.4	0	2.0	0	1	1

Table 3.5. Module Parameters for the Proportional and Derivative Control Test

Step 2 Change the derivative gain by incremental steps of 0.05 V.s/rad to investigate the closed-loop system for PD controllers with different derivative gains. Try the following gains: $k_d = 0, 0.05, 0.1,$ and 0.15 V.s/rad. What are your observations?

Step 3 Determine the lowest value of derivative gain k_d which gives a step response without overshoot (k_p still 2 V/rad). Determine the settling time for the closed loop system.

4.4 PD Controller Design to Given Specifications

This section provides the experimental verification of the PD controller design to given specifications, as carried out in the pre-lab assignment in Section 3.2. The performance of the closed-loop system with the control parameters obtained using the basic pole assignment method in Section 3.2. will be evaluated and compared to the performance specifications given in eq. (8). Please follow the steps below:

Step 1 Set the parameters of the QICii module window as described in Table 3.6.

Signal Type	*Frequency [Hz]*	*Amplitude [rad]*	*Offset [rad]*	b_{sp}	T_f *[s]*
Square Wave	0.4	4.5	0	1	0.006

Table 3.6. Module Parameters for PD Controller Design to Given Specifications

Ensure that the integral gain is zero ($k_i = 0$) and set b_{sd} and both proportional and derivative gains, k_p and k_d, to the values you calculated in Section 3.2, Question 3.2.4.

Step 2 Make sure that the motor input voltage is below its saturation limit. If not, adjust the square wave reference signal Amplitude. Measure the resulting Percent Overshoot (PO), settling time T_s and peak time t_p. Does the system's actual response meet the desired requirements? How close are the measurements to the values you calculated in Question 3.2.3?

Step 3 Summarize your observations and your calculations in your report. Select some representative results, screen-captures, and plots.

4.5. Tracking Triangular Signals

Step 1 Select a triangular reference signal and set the parameters of the QICii module window as listed in table 3.7. Set the controller gains to the PD controller parameters (i.e k_p, k_d, b_{sd}) to the values calculated in Question 3.2.4.

Signal Type	*Amplitude [rad]*	*Frequency [Hz]*	*Offset [rad]*	b_{sp}	*Tf [s]*
Triangular Wave	20	0.4	0	1	0.006

Table 3.7. QICii Module Parameters for Triangular Wave Tracking Test

Step 2 From the triangular wave specifications given in Table 3.7, calculate the slope r_0 of the ramp signal (see eq. (9)).

Step 3 Observe how well the output signal tracks the triangular signal (in particular in terms of the tracking error). Measure the actual asymptotic tracking error and compare it with the analytic estimate obtained in Question 3.3.2.

Step 4 Change the controller proportional gain kp by steps of ± 0.5 V/rad and explore its effect on the tracking error. Select some representative results and plots and include in your report.

5. References

[1] K. Ogata, Modern Control Engineering, 5th Edition, Prentice Hall, Englewood Cliffs, N.J., 2010.

[2] K. J. Astrom, J. Apkarian, H. Lacheray, USB QICii Laboratory Workbook, Quanser Engineering Trainer Series.

Experiment 4: Introduction to the Programming of a Robot Arm

1. OBJECTIVE

To become familiar with Labvolt 5250 robotic system and its four different programming environments.

2. THE STRUCTURE OF THE LABVOLT ROBOTIC SYSTEM

The Labvolt 5250 robotic system used in this laboratory is composed of the following parts:

- **Servo Robot**

The servo manipulator is the main part in the Labvolt 5250 robotic system. This manipulator, as shown in Figure 4.1, is a five degrees of freedom robotic manipulator. All five joints are driven by servo motors which are positioned in the robot base. Optical encoders (using LED) provide position feedback to the robot controller so that movement can be performed with high accuracy and repeatability.

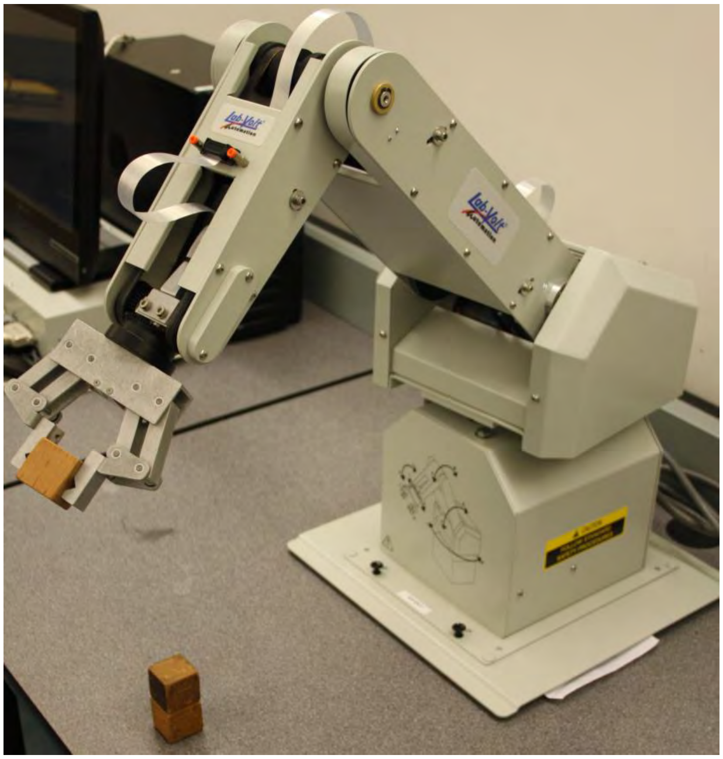


Figure 4.1 - Servo Robot

- **Robot Controller**

The controller is a specialized computer, designed to process the input commands and robot's encoders and provide the servo's with proper command and move the manipulator to the desired position. It also has several inputs and outputs to connect to external equipment such as sensors and motors. The servo robot, external PC, emergency stop button and the hand-held terminal are all connected to the robot controller. Figure 4.2 and Figure 4.3 show the robot controller.

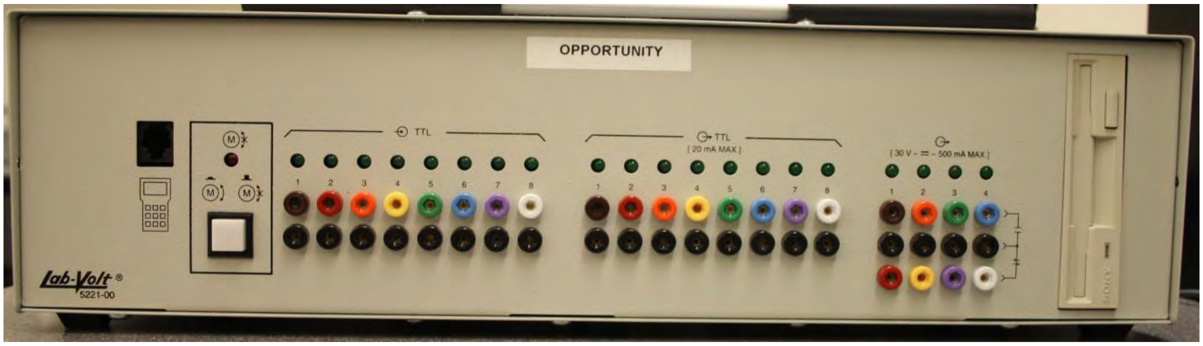


Figure 4.2 - Controller front view

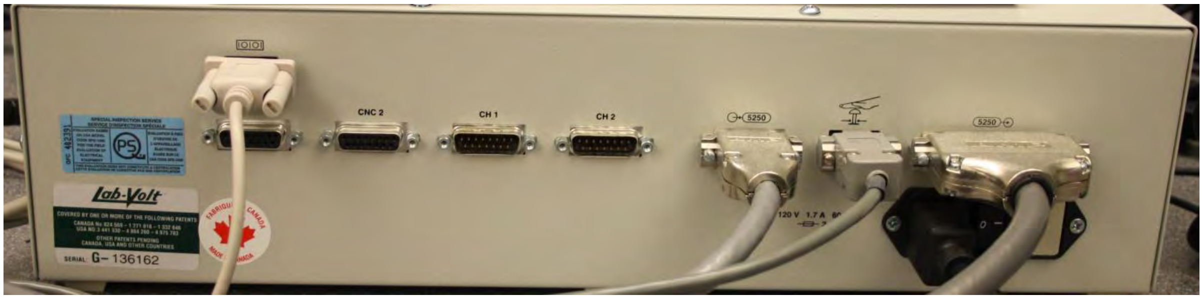


Figure 4.3 - Controller back view

- **Hand-Held Terminal**

The hand-held terminal (or the teach pendant) shown in Figure 4.4 can be used to control the manipulator without using the computer. It is used to enter commands to move the robot on a joint by joint basis (Angular movement) or move it in the Cartesian space (Linear movement). It can *"teach"* the manipulator to go through the workspace and follow the saved points to go through a saved trajectory.

- **Emergency Stop Button**

The emergency stop button, shown in Figure 4.5, contains a push button and a LED indicator. When the push button is pressed, all operations are halted and the LED indicator will turn on. The manipulator won't move unless the button is released (using the key attached to the emergency stop box) and the software release command is issued (which will be discussed later).

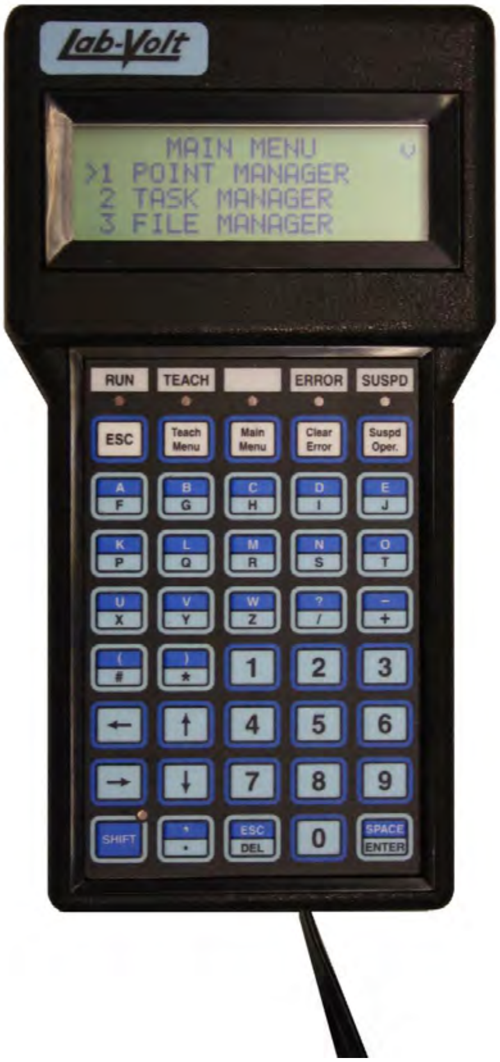


Figure 4.4 - Hand-held terminal

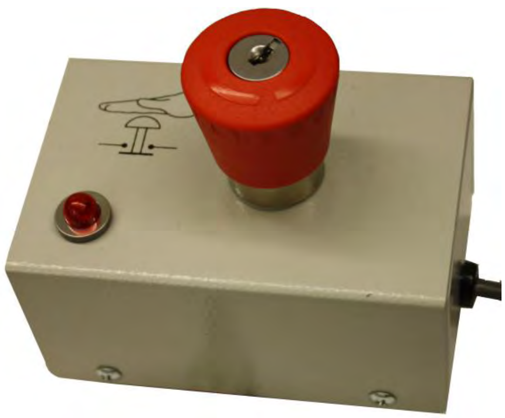


Figure 4.5 - Emergency stop button

Please note that **Whenever the robot is in operation, one of the students in the group should be ready at all times to press this button to prevent damage to the robot or its environment.** If the stop button is pressed, the lab TA should be notified before releasing the button and the release should be supervised by the lab TA. Once the switch is deactivated, on the hand-held device, hit "ESC" then "CLEAR ERROR" and then "ENTER" to resume the operation.

- **External PC**

The external computer provided with the robot can be used to control the robot movement through the "Robotics" program or using Matlab code written by the users. In such cases the robotic system is in *"PC mode"* (as opposed to the *"Manual mode"*, when the hand-held terminal is in use). The computer is connected to the robot controller via a serial RS-232 interface.

3 METHODS OF USING THE ROBOTIC SYSTEM

3.1 Hand-Held terminal

In this method, the manipulator is controlled using the hand-held device (or the teach pendant) shown in Figure 4.4. The device should be activated using the following instructions and then the position data can be stored in the device during the *"Teach"* mode.

Procedure to activate the Hand-Held Terminal:

- Make sure that the controller is switched on and the emergency stop button is not pressed (The LED should be off)
- After the controller is switched on, observe the screen on the hand-held terminal displaying the *Main* menu. (Note that there are lots of options available, but you will see four options on the screen because of the screen dimensions, but you can scroll up or down using the arrow keys on the controller to investigate all options)
- Execute a *Hard Home* by:
    1. In the *Main* menu press "4" to enter *"Motion Control"*
    2. In the *Motion Control* menu press "4" to choose *"Hard Home"*
    3. Press *Enter* to execute the hard home command.
- Press *"Teach"* menu key to enter the *Teach* mode
- Select *"L"* for **linear** (Cartesian) control or *"A"* for **Angular** (Joint-by-joint) control mode.
- *Manual* mode is now activated

Now you can control the manipulator using the hand-held terminal. Depending on which option you chose (Linear or Angular) you can move the robot either by a joint-by-joint basis or in the 3-D Cartesian space along *x,y* or *z* axis.

Using Figure 4.6 and Table 4.1 you can familiarize yourself with the hand-held terminal keyboard in order to move the robot in the desired direction. (It would be useful in this stage for the operator to try and pick up a wooden block)

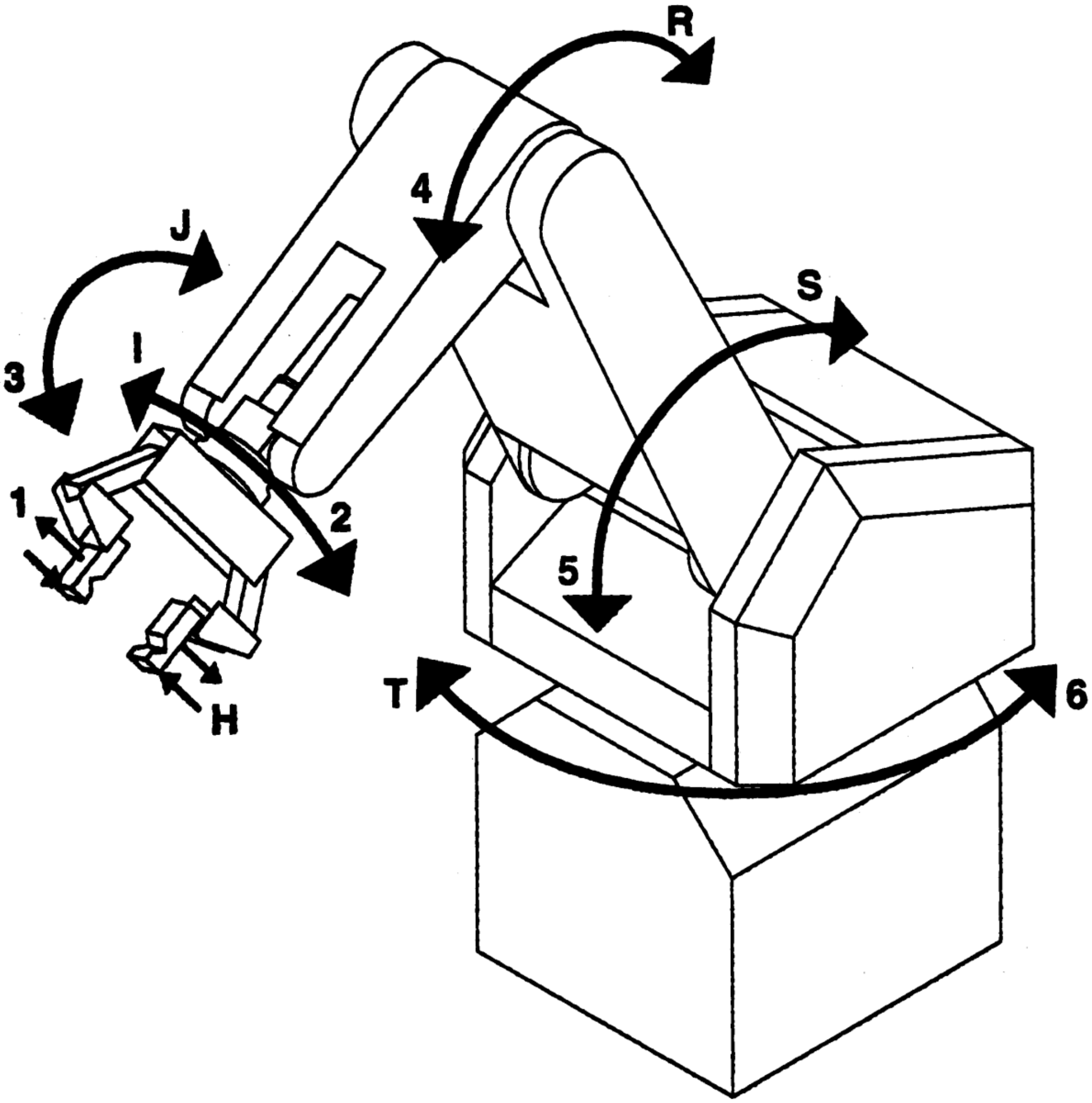


Figure 4.6 - Angular movement commands

Each point can be saved using the *"."* key on the hand-held terminal when in the *Teaching* mode.

Speed Control: for controlling the manipulator's speed when in *Teaching* mode press *"0"* (as indicated in Table 4.1) and then a number (try 3 or 4 for 30% or 40% of the full speed at first). Speeds more than 50% are not recommended here.

Saving the position: you can press "." at any point to save the position of the robot into the hand-held memory. These positions will be used later for performing a complete task.

Table 4.1 - List of hand-held commands in *Angular* and *Linear* modes

Teach Menu Commands	Angular Mode	Linear Mode
Speed Control	0	0
Gripper Open	1	J
Gripper Close	H	H
Wrist CW	2	T
Wrist CCW	I	R
Wrist Pitch Down	3	S
Wrist Pitch Up	J	I
Elbow (Forearm) Pitch Down	4	
Elbow (Forearm) Pitch Up	R	
Shoulder (Upper Arm) Pitch Down	5	
Shoulder (Upper Arm) Pitch Up	S	
Base CCW (counter clockwise)	6	
Base CW (clockwise)	T	
Save Absolute Point	. (dot)	. (dot)
Save Relative Point	#	
Save Control	*	*
Z+		2
Z-		8
X+		7
X-		3
Y+		6
Y-		4

Running through all stored positions step-by-step: for running a step-by-step movement through all the positions you saved press the "*Main Menu"* key, then press *"2"* to go to the *Task Manager* menu, then press *"3"* for the *Trace Points* task and then press *"Enter"* to go to the next position. When the last instruction is executed the arm will return to the home position. You can press *"ESC"* at any time to abort the operation

Running through all the stored points: to *"Drive"* the robot to move through all the saved points one after another press the "*Main Menu"* key, then press *"2"* to go to the *Task Manager* menu, then press *"1"* for *Run Once* task. You can press *"ESC"* button to abort the operation at any time.

SAFETY NOTE: At any time, robot motion can be stopped by pressing the red <u>“EMERGENCY STOP”</u> button on the emergency stop module. It is mandatory that you keep one hand on the emergency stop button whenever the robot is moving and stop the robot when a collision is anticipated. You should inform your lab TA about the emergency stop and the release should be supervised by the lab TA.

3.2 Robotics application – Visual Interface

The *“Robotics”* application lets you control the manipulator from the external PC. The program can be started from the desktop. For using the visual interface application you need to follow these steps:

1. Turn on the controller
2. Double click on the program’s icon on desktop (as shown in Figure 1.7).

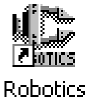
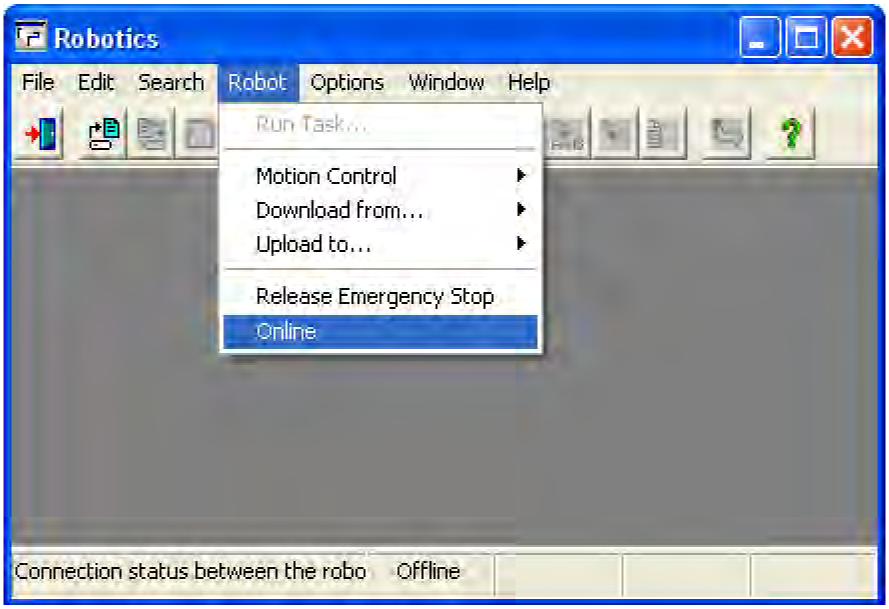


Figure 1.7 - Program icon

3. Now the *“Robotics”* application will be running. It will look like Figure 4.8. You need to connect the application to the manipulator. To do so, click on the **Robot** menu and then click on **Online.**



Connection status between the robo Offline

Figure 4.8 - Robotics application

4. Now the application is connected to the robot and the *“Controller Status”* window can be seen, as shown in Figure 4.9.

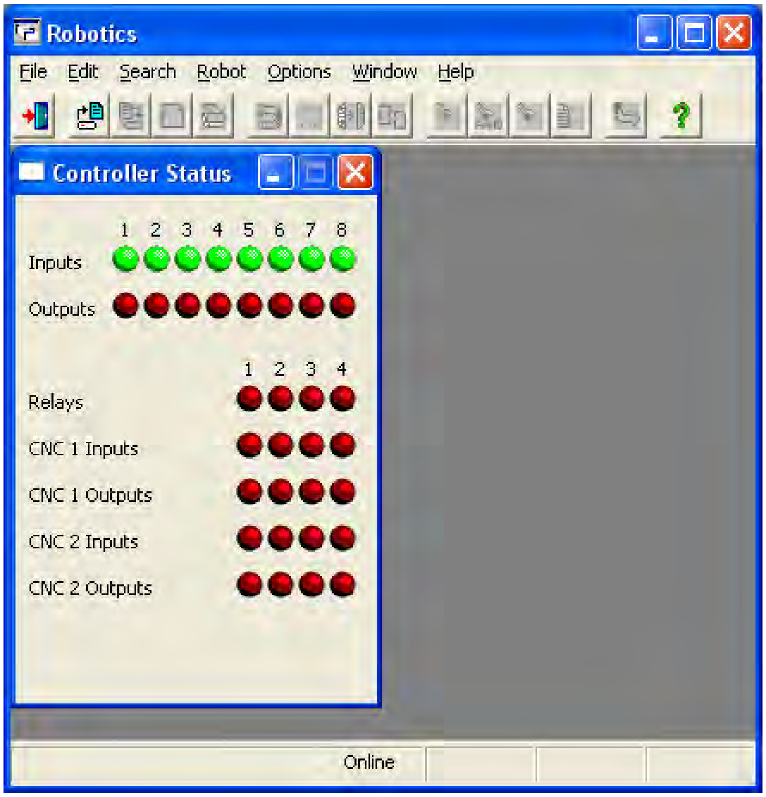


Figure 4.9 - Controller Status

5. To view the visual controller, click on **Window** menu and then click on **Point Editor**. The point editor window is shown in Figure 4.10.

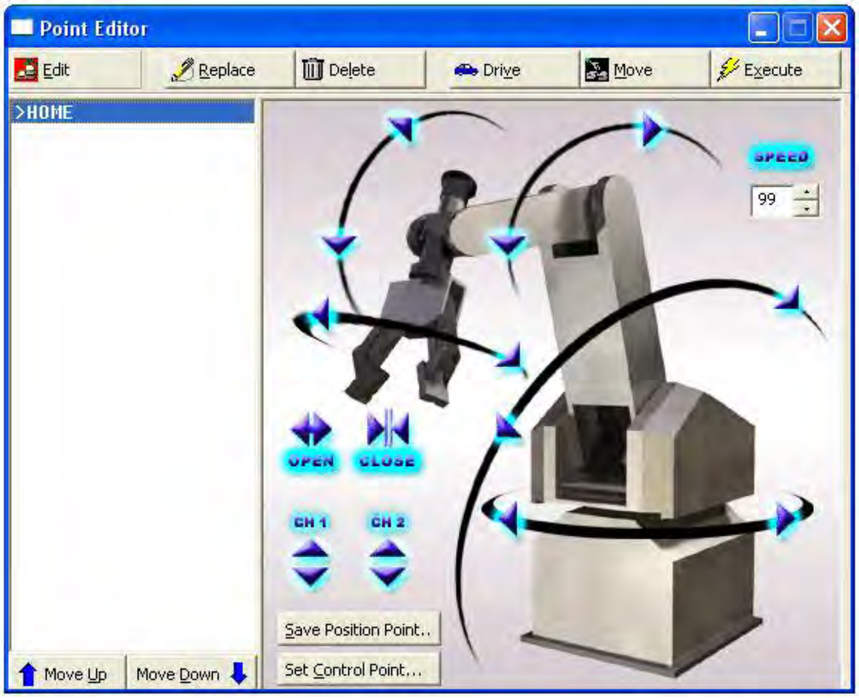


Figure 4.10 - Point Editor

6. Default speed is 99. As this speed is too fast for our tests first thing we need to do is to reduce this speed to no more than 50.
7. The green question mark (shown in Figure 4.11) in the menu is a helpful way to seek help about different menus and commands in the application.

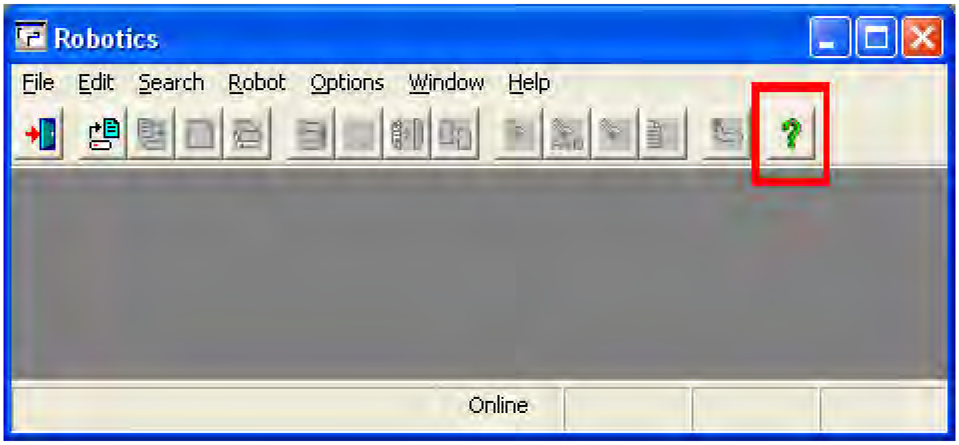
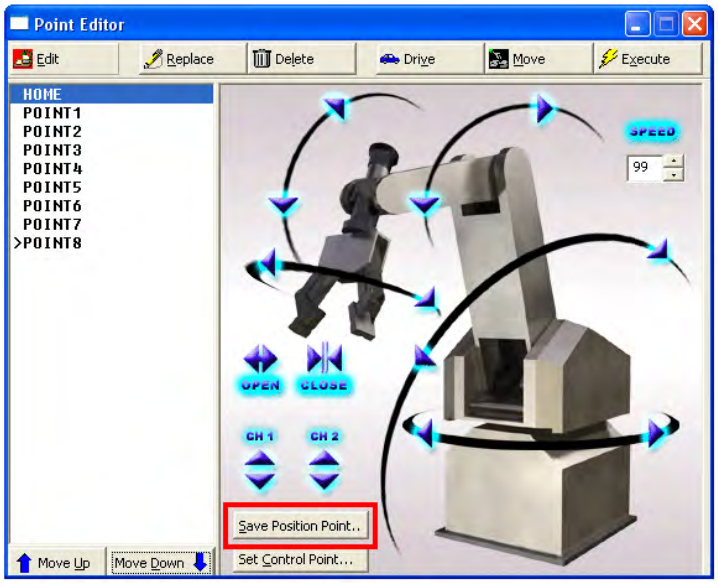


Figure 4.11 - Getting help

Now that we have set up the point editor, we can start to move the manipulator using the point editor. As it can be seen in the point editor, each arrow moves one joint in one direction. You can use the arrows to move the robot to your desired position.

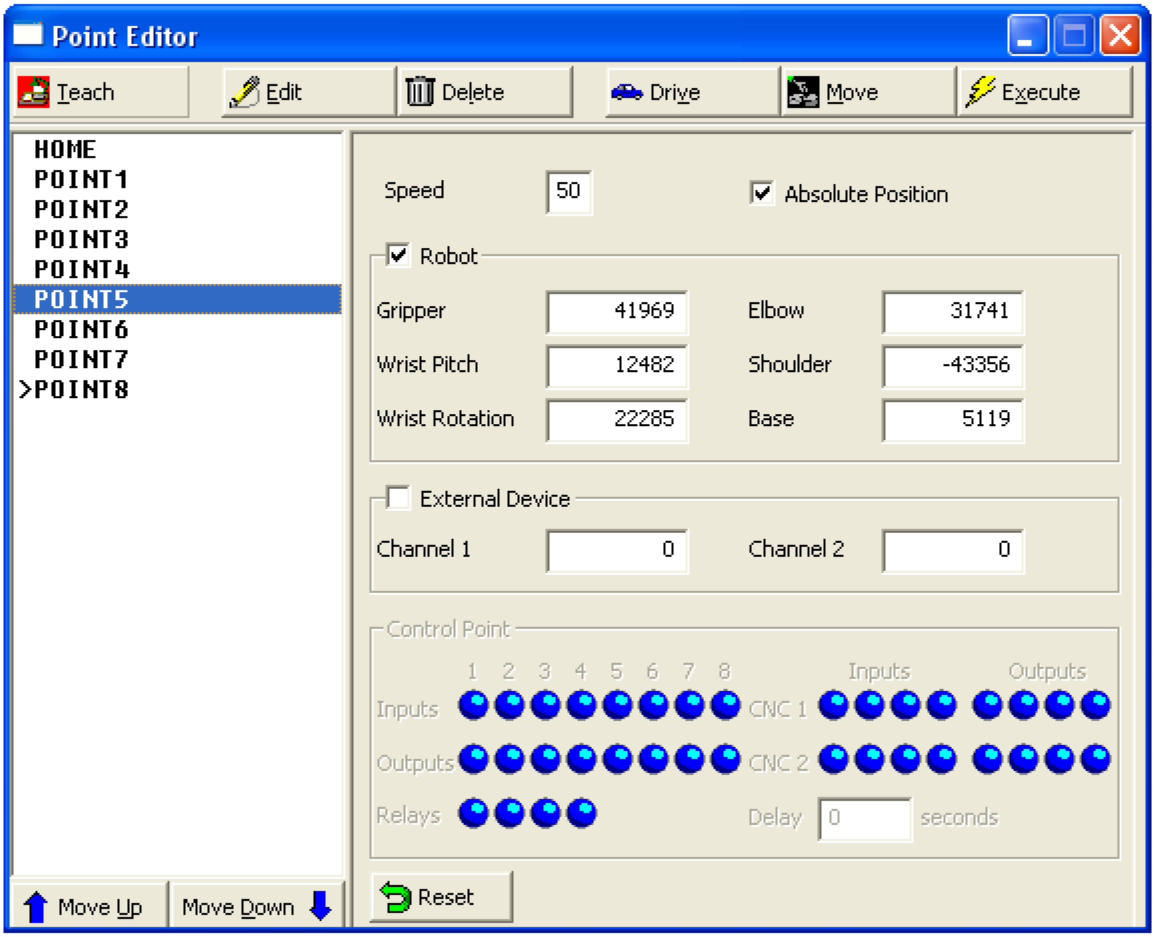
After reaching the desired position, you can save the position by clicking on **Save Position Point** button or by pressing *Alt+S*. The *"save position point"* window is shown in Figure 4.12. You can choose a name for your point and save it.



SPEED 99

Figure 4.12 - Saving position

For editing a point, click on it in the left hand list in the point editor window, click on **Edit** button on top of the list and then click **Edit** again. As Figure 4.13 shows, you can see the joint values for each joint of the robot and change it if you want. After you finish editing the point, you can press the **Teach** button to go back to the point editor main view.



HOME
POINT1
POINT2
POINT3
POINT4
POINT5
POINT6
POINT7
>POINT8

Speed 50

☑ Absolute Position

☑ Robot

Gripper	41969	Elbow	31741
Wrist Pitch	12482	Shoulder	-43356
Wrist Rotation	22285	Base	5119

□ External Device

Channel 1	0	Channel 2	0

Delay 0 seconds

Figure 4.13 - Edit a point window (The joint values represent the orientation of Figure 4.1)

You can move a point up or down in the list by clicking on the point and then clicking on **Move up** or **Move down** in the lower left part of the point editor window.

The robot can be moved to any of the positions in the list by clicking on the desired position and then clicking on **Move.**

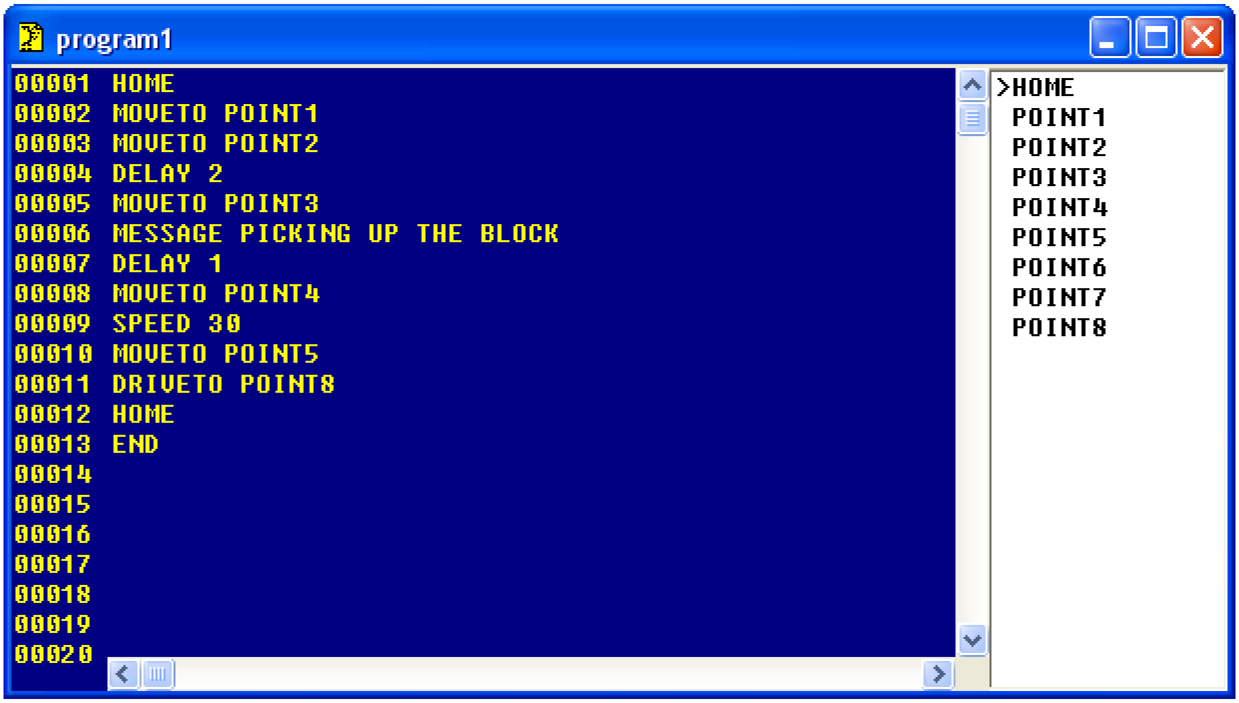
Clicking on a position and clicking on **Drive** steps the robot through the list up to the desired point.

The entire list can be run by clicking on **Execute.**

SAFETY NOTICE: always be prepared to press the <u>emergency button</u> when the robot is in motion.

3.3 Robotics application – Programming Interface

The point editor in the *Robotics* application is complemented with a programming environment for Labvolt 5250 Manipulator, the **Task Editor**. Figure 4.14 shows a sample program which is developed and ran in this environment.



```
HOME
MOVETO POINT1
MOVETO POINT2
DELAY 2
MOVETO POINT3
MESSAGE PICKING UP THE BLOCK
DELAY 1
MOVETO POINT4
SPEED 30
MOVETO POINT5
DRIVETO POINT8
HOME
END
```

```
>HOME
 POINT1
 POINT2
 POINT3
 POINT4
 POINT5
 POINT6
 POINT7
 POINT8
```

Figure 4.14 – A sample program in the Task Editor

To open the Task Editor, follow these steps:

1. Select **New** in the **File** menu and enter the required information in the *Program Information* menu. The *Task Editor* will open.
2. Copy all or some of the points saved in the point editor into right-side list of the *Task Editor*. You can use *Shift* or *Control* key to select a group of points and drag them into this list.
3. Write your code into the *Task Editor* window (the instructions follow).
4. You can run the task using **Run Task** option in the **Robot** menu.

Some of the commonly used commands are described in Table 4.2. The full commands list can be accessed by clicking on the **help button**, clicking on **Robotics Programming Basics** link, and then clicking on **Task Commands**.

A program developed using the *Robotics* application (such as the one in 4.14) can be saved in a file using the **Save As** option in the **File** menu and used later. The saved file can be opened using the windows *Notepad* application and the joint values of each position (as the ones shown in Fig 4.13) can be obtained.

Table 4.2 - Useful commands in the *Task Editor* and a brief description of each one

Command	Description
HOME	Tells the servo robot to move to the assigned "soft home" position
MOVETO	Causes the execution of a position point that follows
DELAY	Delays task execution for the value expressed in seconds
MESSAGE	Displays the string parameter data that follows the command
SPEED	Changes the execution speed with the value that follows the command
DRIVETO	Operates a shortcut instead of repeating multiple MOVETO commands, to execute a series of points from the list from the current active point to the point named in the command parameter
RUNTO	Tells the servo robot to move to the point named in the command parameter with maximum speed. The point speed will be ignored
END	Ends the execution

4. PRE LABORATORY QUESTIONS

Describe briefly the difference between *Angular* and *Linear* modes.

What is the *Hard Home* position and why is it being used.

5. IN-LABORATORY SESSION

5.1 PRECAUTIONS

- **Never touch the moving parts of the robot with your hands. Be very careful when placing tools and objects within the robot's workspace**
- **Never place your fingers in the claw of the gripper**
- **Make sure that the robot's home position has been established before attempting any experiment. You can run a *"Hard Home"* command before starting the experiment to make sure that the home position is set**

correctly. To execute a *“Hard Home”*, on the hand-held terminal press ESC → “4” → “4” → ENTER. Make sure that there is nothing in the claw of the gripper before executing the command.

- **Be aware that the robot can move to a position that will damage itself or the environment. Be ready to press the *“Emergency Stop”* button at all times.**
- **When the *“Emergency Stop”* button is pushed, depending on the environment you are using to control the robot, the following steps should be followed:**

  - **Hand-held terminal**
    1. **Inform the TA to release the button**
    2. **Press *“ESC”***
    3. **Press *“Clear Error”***
    4. **Press *“Enter”***

  - **Robotics application**
    1. **Cancel the currently running task**
    2. **Inform the TA to release the button**
    3. **From the Robot menu, press Release Emergency Stop**

5.2 PROGRAMMING A SIMPLE TASK IN MANUAL MODE

Program the robot arm using *Angular* and *Linear* modes to lift **one block**, move it and put it down. The procedure to be followed is:

- Use the Hand-Held Terminal to store the positions using *Angular* mode. Verify the position data using the option to run through each stored position step by step in the *Task Manager* menu.
- Make any corrections necessary in the position data.
- Demonstrate your program to your lab instructor by using the *Run Once* option in the *Task Manager* menu of the Hand-Held Terminal.

Repeat the above procedure using *Linear* mode and demonstrate it.

In your lab report discuss which mode, *Angular* or *Linear*, is more convenient? What did you observe regarding the linearity of the coordinate system in *Linear* mode?

5.3 PROGRAMMING A TASK IN PC MODE

Program the robot arm to move **3 blocks** which are initially stacked one over the other to the arrangement indicated in Fig. 4.4. Use the Robotics application running on the PC and demonstrate it to your lab instructor.

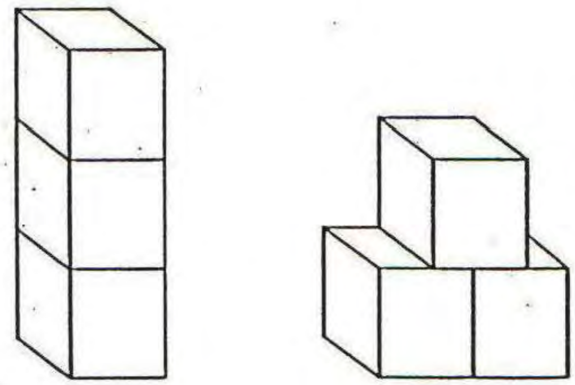


Figure 4.15. The two arrangements for the blocks

5.4 POWER OFF ROBOT

At the end of the lab execute a *"Hard Home"* using the Hand-Held Terminal.

Press "ESC" → then the *Main* menu key → Press *"9"* for *Shutdown.*

Switch off the main power at the back of the Robot Controller.

6. EQUIPMENT

Lab-Volt Servo Robot System Model 5250

External PC