



Quanser Engineering Trainer (QET) Series:

USB QICii Laboratory Workbook

DC Motor Control Trainer (DCMCT)



Karl Johan Åström

And

Jacob Apkarian, Hervé Lacheray

Student Workbook

Table of Contents

1. Introduction.....	1
1.1. Introductory Control Laboratories.....	1
1.2. The Laptop Process.....	3
1.3. The Method.....	4
1.3.1. The Experiments.....	5
1.3.1.1. Modelling.....	5
1.3.1.2. Speed Control.....	5
1.3.1.3. Robustness.....	6
1.3.1.4. Position Control.....	6
1.3.1.5. Haptic Interaction.....	6
1.3.2. CAD Files.....	7
1.3.3. Chapter Structure.....	8
1.3.3.1. Pre-Laboratory Assignments.....	8
1.3.3.2. Solutions And Typical Results.....	8
1.3.3.3. Marking Scheme.....	8
1.3.3.4. Results Summary Tables.....	8
1.4. Curriculum Summary And Scheduling.....	9
1.5. System Requirements.....	11
1.6. References.....	11
2. Modelling.....	12
2.1. Laboratory Objectives.....	12
2.2. Preparation And Pre-Requisites.....	12
2.3. Introduction.....	12
2.4. Nomenclature.....	15
2.5. Pre-Laboratory Assignments: First Principles Modelling.....	16
2.5.1. Motor First Principles.....	16
2.5.2. Static Relations.....	21
2.5.3. Dynamic Models: Open-Loop Transfer Functions.....	27
2.5.4. Pre-Laboratory Results Summary Table.....	34
2.6. In-Laboratory Session.....	35
2.6.1. QICii Modelling Module.....	35
2.6.1.1. Module Description.....	35
2.6.1.2. Module Startup.....	37
2.6.2. Static Relations.....	38
2.6.2.1. Initial Experimental Tests.....	38
Objectives.....	38
Experimental Procedure.....	38
2.6.2.2. Estimate The Motor Resistance.....	40
2.6.2.3. Estimate The Motor Torque Constant.....	41

2.6.2.4. Obtain The Motor Transfer Function.....	44
2.6.2.5. Estimate The Measurement Noise.....	44
2.6.3. Dynamic Models: Experimental Determination Of System Dynamics.....	46
2.6.3.1. The Bump Test.....	46
Preamble.....	46
Experimental Procedure.....	47
2.6.3.2. Model Validation.....	50
Preamble.....	50
Experimental Procedure.....	50
2.6.4. Concluding Remarks.....	53
2.6.4.1. Load Disturbances and Measurement Noise.....	53
2.6.4.2. Automating The Tests.....	53
2.6.4.3. Nonlinearities.....	53
2.6.4.4. Unmodeled Dynamics.....	54
2.6.5. In-Laboratory Results Summary Table.....	56
3. Speed Control.....	57
3.1. Laboratory Objectives.....	57
3.2. Preparation And Pre-Requisites.....	57
3.3. Introduction: The PI Controller.....	59
3.3.1. PI Control Law.....	59
3.3.2. The Magic Of Integral Action.....	59
3.4. Nomenclature.....	61
3.5. Pre-Laboratory Assignments.....	62
3.5.1. PI Controller Design To Given Specifications.....	62
3.5.2. Integrator Windup.....	71
3.5.2.1. Definition.....	71
3.5.2.2. Windup Protection.....	71
3.5.3. Tracking Triangular Signals.....	73
3.5.4. Response To Load Disturbances.....	76
3.5.5. Pre-Laboratory Results Summary Table.....	79
3.6. In-Laboratory Session.....	80
3.6.1. QICii Speed Control Module.....	80
3.6.1.1. Module Description.....	80
3.6.1.2. Module Startup.....	82
3.6.2. Qualitative Properties Of Proportional And Integral Control.....	83
3.6.2.1. Pure Proportional Control.....	83
3.6.2.2. Pure Integral Control.....	86
3.6.2.3. Proportional And Integral Control.....	89
3.6.3. Manual Tuning: Ziegler-Nichols.....	91
3.6.3.1. Preamble: Ziegler-Nichols Method.....	91

3.6.3.2. Experimental Procedure: Ziegler-Nichols Tuning.....	91
3.6.4. Set-Point Weighting.....	95
3.6.4.1. Preamble.....	95
3.6.4.2. Experimental Procedure.....	95
3.6.5. Design To Given Specifications.....	97
3.6.5.1. PI Control With No Set-Point Weighting.....	97
3.6.5.2. PI Control With Set-Point Weighting.....	99
3.6.6. Integrator Windup.....	101
3.6.6.1. Preamble.....	101
3.6.6.2. Windup Protection.....	103
3.6.7. Tracking Triangular Signals.....	105
3.6.8. Response To Load Disturbances.....	108
3.6.8.1. Manual Load Disturbances.....	108
3.6.8.2. Simulated Load Disturbances: Disturbance Response With PI Control. .	109
3.6.9. In-Laboratory Results Summary Table.....	113
4. Robustness.....	113
4.1. Laboratory Objectives.....	113
4.2. Preparation And Pre-Requisites.....	113
4.3. Introduction.....	114
4.3.1. Control Systems Design.....	114
4.3.2. The Gang Of Six.....	114
4.4. Nomenclature.....	116
4.5. Pre-Laboratory Assignments.....	117
4.5.1. MATLAB Example Script.....	117
4.5.2. Robustness And Sensitivity.....	118
4.5.2.1. Small Process Variations – The Sensitivity Function.....	118
4.5.2.2. Large Process Variations – The Complementary Sensitivity Function....	126
4.5.3. Stability Margins.....	129
4.5.3.1. Preamble.....	129
4.5.3.2. Assignment Questions.....	131
4.5.4. Pre-Laboratory Results Summary Table.....	135
4.6. In-Laboratory Session.....	136
4.6.1. QICii Robustness Module.....	136
4.6.1.1. Module Description.....	136
4.6.1.2. Module Startup.....	139
4.6.2. Stability Margins Evaluation.....	140
4.6.2.1. Preamble #1: Performance-Related Parameters.....	140
4.6.2.2. Preamble #2: Actual Closed-Loop Implementation.....	141
4.6.2.3. Stability Margins Evaluation.....	142
4.6.3. In-Laboratory Results Summary Table.....	150

5. Position Control.....	151
5.1. Laboratory Objectives.....	151
5.2. Preparation And Pre-Requisites.....	151
5.3. Introduction: The PID Controller.....	153
5.3.1. PID Control Law.....	153
5.3.2. The Magic Of Integral Action.....	153
5.3.3. Controllers With Two Degrees Of Freedom.....	154
5.4. Nomenclature.....	155
5.5. Pre-Laboratory Assignments.....	156
5.5.1. Comparison Between PD Position And PI Speed Controls.....	156
5.5.2. Fundamental Limitations And Achievable Performance.....	159
5.5.3. PD Controller Design To Given Specifications.....	162
5.5.4. Tracking Triangular Signals.....	167
5.5.5. Response To Load Disturbances.....	170
5.5.6. Pre-Laboratory Results Summary Table.....	177
5.6. In-Laboratory Session.....	178
5.6.1. QICii Position Control Module.....	178
5.6.1.1. Module Description.....	178
5.6.1.2. Module Startup.....	180
5.6.2. Qualitative Properties Of Proportional And Derivative Control.....	181
5.6.2.1. Pure Proportional (P) Control.....	181
5.6.2.2. Proportional And Derivative (PD) Control.....	184
5.6.3. PD Controller Design To Given Specifications.....	186
5.6.4. Tracking Triangular Signals.....	188
5.6.5. Response To Load Disturbances.....	192
5.6.5.1. Preamble: Simulated Load Disturbances.....	192
5.6.5.2. PD And PID Controllers.....	192
5.6.6. In-Laboratory Results Summary Table.....	196
6. Haptic Interaction.....	197
6.1. Background And Laboratory Objectives.....	197
6.2. Preparation And Pre-Requisites.....	198
6.3. Nomenclature.....	199
6.4. Pre-Laboratory Assignment.....	200
6.4.1. Impedance Control.....	200
6.4.2. Pre-Laboratory Results Summary Table.....	204
6.5. In-Laboratory Session.....	205
6.5.1. QICii Modules Description.....	205
6.5.1.1. Haptic Knob Module.....	205
6.5.1.2. Ball and Beam Module.....	207
6.5.2. Module Startup.....	210

6.5.3. Impedance Control.....	211
6.5.4. Haptic Knob.....	213
6.5.4.1. Preamble: Haptic Knob Implementation.....	213
6.5.4.2. Experimental Procedure: Haptic Knob Exploration.....	214
6.5.5. Haptic Ball And Beam.....	215
6.5.5.1. Preamble: Haptic Ball And Beam Implementation.....	215
6.5.5.2. Experimental Procedure: Ball And Beam Exploration.....	216
6.5.6. In-Laboratory Results Summary Table.....	218
Appendix A. DCMCT/USB QICii Hardware Guide.....	219
A.1. DCMCT System Capabilities.....	219
A.2. General Overview.....	220
A.2.1. System Nomenclature.....	220
A.2.2. System Schematic.....	223
A.2.3. Component Description.....	224
A.2.3.1. Maxon DC Motor.....	224
A.2.3.2. Linear Power Amplifier.....	224
A.2.3.3. QIC Compatible Socket.....	224
A.2.3.4. QIC Processor Core Board.....	224
A.2.3.5. Analog Current Measurement: Current Sense Resistor.....	225
A.2.3.6. Digital Position Measurement: Optical Encoder.....	225
A.2.3.7. Analog Speed Measurement: Tachometer.....	225
A.2.3.8. Analog Position Measurement: Potentiometer.....	225
A.2.3.9. A Wall Transformer.....	226
A.2.3.10. Built-In Power Supply.....	226
A.2.3.11. A12-Bit Digital-To-Analog Converter (D/A)	226
A.2.3.12. 24-Bit Encoder Counter.....	227
A.2.3.13. Secondary Encoder Input To QIC.....	227
A.2.3.14. External Analog Input To QIC.....	227
A.2.3.15. Analog Signals Header: J11.....	227
A.3. System Parameters.....	228
A.4. Hardware Configuration.....	230
A.4.1. DCMCT Configuration For QICii Use.....	230
A.4.2. DCMCT Configuration For HIL Board Use.....	234
A.5. System Overview Relevant To QICii Control.....	236
Appendix B. DCMCT/USB QICii Software Guide.....	238
B.1. Introduction.....	238
B.2. Software Installation.....	238
B.2.1. System Requirements.....	238
B.2.2. Installation.....	239
B.2.2.1. USB QICii Components.....	239

B.2.2.2. If Upgrading USB QICii.....	239
B.2.2.3. Installing USB QICii.....	240
B.2.3. Getting Started.....	245
B.2.4. Plots.....	247
B.2.4.1. Common Features.....	248
B.2.4.2. Individual Plot Functions.....	249
B.2.4.3. Saving Data.....	250
B.2.4.4. Taking Measurements On A Plot (Ctrl Zoom).....	251
B.2.5. Signal Generator.....	253
B.2.5.1. Square Wave.....	254
B.2.5.2. Triangular Wave.....	255
B.2.5.3. Sinusoidal Wave.....	256
B.3. Troubleshooting.....	257
B.3.1. Setting Up The QIC.....	257
B.3.1.1. Does Your QIC Have the USB QICii Firmware Program In It?.....	257
B.3.2. Problem Viewing Three-Dimensional Graphics.....	257

1. Introduction

1.1. Introductory Control Laboratories

Control is a very rich field with continuously increasing areas of applications. One reason for this is the beneficial properties of feedback. Feedback makes it possible to change the dynamic behaviour of a system, stabilization of an unstable system is a typical example. Feedback makes it possible to reduce the effect of disturbances. The major drawback is that feedback may create instabilities. The ubiquity of control makes it necessary to spread the knowledge of control to wider audiences. One of the conclusions in a recent panel [4] on control is the following recommendation: *Invest in new approach to education and outreach for the dissemination of control concepts and tools to nontraditional audiences.* The panel report goes on to say:

As a first step toward implementing this recommendation, new courses and textbooks should be developed both for experts and nonexperts. Control should also be made a required part of engineering and science curricula at most universities including not only mechanical, electrical, chemical, and aerospace engineering, but also computer science, applied physics, and bioengineering. It is also important that these courses emphasize the principles of control rather than simply providing tools that can be used in a given domain. An important element of education and outreach is the continued use of experiments and the development of new laboratories and software tools. This is much easier to do than ever before and also more important. Laboratories and software tools should be integrated into the curriculum.

The experiments described in this booklet are inspired by the recommendations by the panel report. A control engineer should master theory and have a good understanding of practical control problems. The skill base includes tasks such as modelling, control design, simulation, implementation, commissioning, tuning, and operation of a control system [1], [2]. These skills are becoming more important today when control is ubiquitous [3]. Many tasks can be learned from books and computer simulations but laboratory experiments are necessary to obtain the full range of skills.

The typical setup for control experiments consists of a physical process with sensors, actuators and power supply, a PC equipped with interfaces, and sometimes a DSP board. Control is performed using the DSP or the PC. The controller is either hand-coded (good luck!) or designed using commercially available design tools such as Simulink, SystemBuild, or LabVIEW. Once the design is performed, realtime code is generated and run on the PC using high performance realtime software such as WinCon, xPC Target, or LabVIEW RT.

This workbook is designed for an introductory course in controls. A first control course does not normally focus on practical issues. First time exposure to control typically focuses on the theoretical aspects. Special laboratory courses are offered as a complement to the theoretically-oriented courses but many students do not take such courses. This is unfortunate because good experiments can also be a strong motivation to pursue a career in controls.

Introductory courses in control with integrated labs are offered in most universities. Although integration of a lab has many advantages [5], [6], there is a difference between lectures and labs. A student can pick up a book or do a computer simulation at any time and at any place but experiments are heavily restricted in time and space.

This workbook focuses on a novel **portable** process that can be used with a laptop computer to investigate control system performance and evaluation. The system can be signed out by the student and taken home, library, or café, and thus eliminates the need for laboratory space. The system makes it possible to integrate theory and practice of control. The experiments can be done concurrently with studies of theory and computer simulation. This also makes it very suitable for practicing engineers who would like to brush up the knowledge of control.

The system is completely self-contained. The process consists of a DC motor and a PIC microcontroller that can be easily programmed to perform a series of control experiments of varying complexity. The process can be controlled using a laptop with no other software than that supplied with the system.

When designing the process and the experiments, we were guided by utility. Any user of control should have a good grasp of the fundamental ideas and concepts. It is therefore natural to include modelling, controller tuning, design, and robustness. The PID controller is by far the most common controller. We therefore made the decision to focus on PI, PD, and PID control. This gives opportunities to get a good grasp of the principles of control and the skills required to design simple control loops. To have a full understanding of PID control it is necessary to consider both linear and nonlinear phenomena.

The experiments were designed to maximize system use and expose the user to important industrial and theoretical control issues. A graphical user interface allows the user to download pre-compiled controllers and to plot and tune parameters on the fly. The system also exposes students to haptics and Virtual Reality (VR) which augments the system features with a "coolness" factor which, we hope, will arouse curiosity and stimulate students to pursue a career in controls. This workbook gives a brief description of the system, the rationale for its design, and some views on the pedagogy.

1.2. The Laptop Process

A photograph of the system (DCMCT) is shown in Figure 1.1.

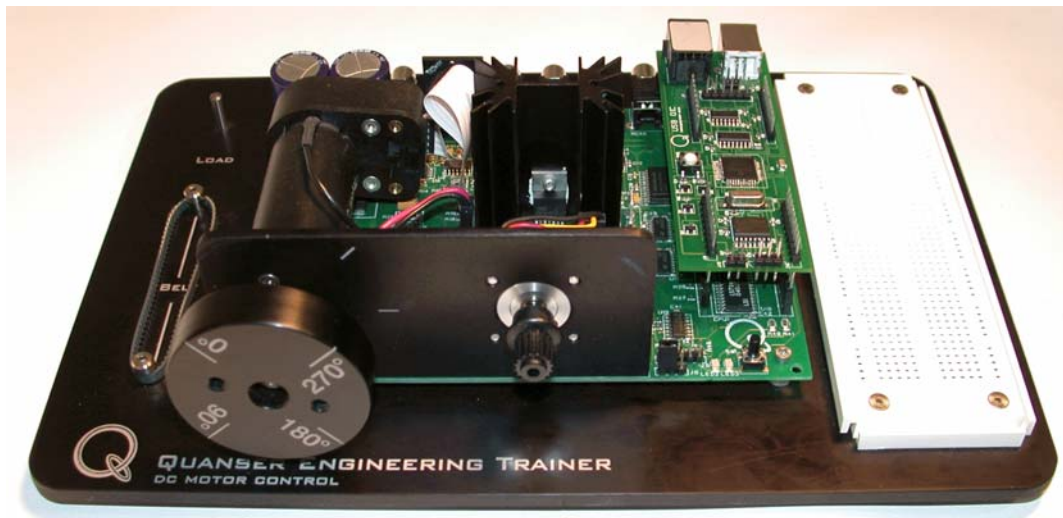


Figure 1.1 Photograph Of The QET DC Motor Control Trainer (DCMCT)

A complete description of the DCMCT is provided in Appendix A. The system consists of a motor instrumented with an encoder. The motor is driven using a linear power amplifier. The power to the system is delivered using a wall transformer. Signals to and from the system are available on a header as well as on standard connectors for control via a Hardware-In-the-Loop (HIL) board. The system may be controlled using an external PC equipped with a HIL board. Alternatively analog controllers can be implemented on the breadboard.

More to the point, a socket, which accommodates a PIC microcontroller, is also available. The PIC can measure the encoder, apply voltages to the motor amplifier, and communicate with a laptop using a USB cable.

Introduction

In the context of this workbook, this system is used as a portable embedded control system which can readily be configured to perform control experiments using a laptop computer that communicates with the PIC microcontroller. The PIC microcontroller module, named QIC, plugs into a custom socket on the DCMCT board.

A software package, called USB QICii (please refer to Appendix B), that runs on the laptop allows one to download pre-compiled code to the PIC which performs the actual real-time control. USB QICii communicates with the PIC in real-time allowing for parameter tuning on the fly and data collection and plotting. In the example illustrated in Figure 1.2, the system is running a PID position controller.

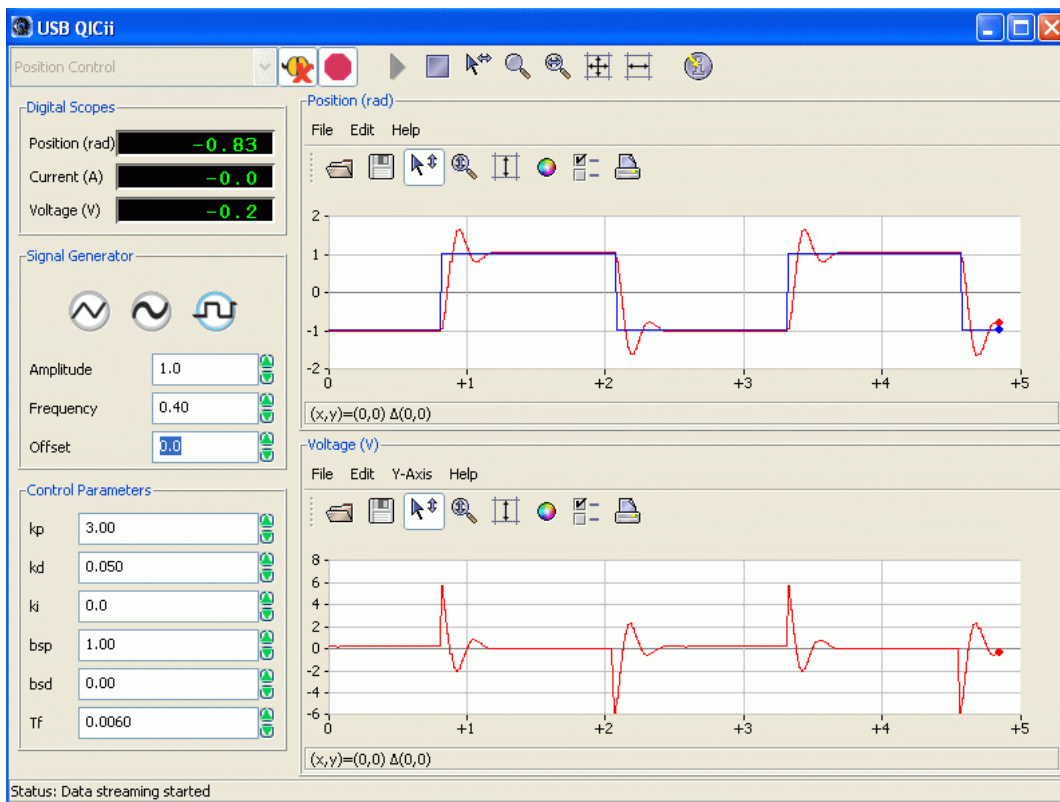


Figure 1.2 Screen Capture Of The QICii Software

1.3. The Method

The system can be used in many different ways. A detailed curriculum has been developed to guide students and teachers. The curriculum demonstrates the relevant characteristics of each control topic.

A systematic approach to performing the laboratories was designed. Each laboratory has a pre-lab preparation section in which the student performs all the theoretical developments required for the session and performs calculations for parameters which are subsequently used during the experiments. This ensures that the student is ready for the lab.

The pre-lab activity is followed by an in-lab activity where students do the actual experiments in a lab (or at home or in a coffee house). To do the lab the student simply launches the USB QICii application. A screen capture of a typical USB QICii session is shown in Figure 1.2. The manuals direct the student to perform specific experiments using the interactive software. Data is collected by the student for very specific activities and entered into pre-formatted tables. The tables facilitate the comparison of results obtained from theoretical derivations and actual performance. The student is then asked to discuss the results.

1.3.1. The Experiments

Many different experiments can be performed with the system. The following experiments were designed to entice the student into further examining control system design and to consider it as part of their future engineering expeditions.

1.3.1.1. Modelling

Although practicing industrial control engineers do not typically derive models of the system, they are controlling (the authors have seen heuristic manual tuning performed in some of the most demanding applications). This experiment stresses the importance of "**knowing the system before you control it**". This is also necessary to have a broader understanding of control. The students derive the theoretical open-loop model of the system and assess its performance limitations. The system is designed in such a way that a good model can be derived from first principles. The physical parameters can all be determined by simple experiments. Using QICii and the QET, the students perform experiments with its inputs and observe its outputs. Real-time open-loop tests are performed and system parameters are estimated using static and dynamic measurements. A first-order simulation of the derived model is run in real-time in parallel with the actual system and a bump test is performed to assess the validity of the estimated model.

1.3.1.2. Speed Control

The PI controller is perhaps the most commonly used controller. Both students and practitioners of control should be well familiar with it. Speed control of a motor is a good way to learn PI control. Students are asked to investigate the qualitative properties of proportional and integral action to develop a good intuitive feel for PI control. Controllers are tuned both by empirical methods of the Ziegler-Nichols type and by design to given specifications. The

student analyzes and tests the effect of set-point weighting. This is unfortunately often ignored in educational settings but relied on heavily in industrial control. The effect of integrator windup is examined and an integrator anti-windup scheme is tuned and evaluated. This is also a good way to demonstrate that performance can be drastically improved by introducing nonlinearities. Disturbance effects, simulated via a direct manual interaction or by a user switch activated by the QIC, are examined and steady-state errors due to triangular inputs are assessed. Tracking of square wave, sinusoidal, and triangular signals can be discussed.

1.3.1.3. Robustness

Robustness to modelling errors is an essential property of a good control system. Following the speed control experiment, the student is introduced to sensitivity analysis and stability margins. Sensitivity and complementary sensitivity functions for the speed control system are derived and the student is guided in designing a more robust controller than the previous one. Sampling delays and filtering effects are taken into account and the stability gain and phase margins are derived. The margins are then measured using the actual system. The QI-Cii software allows the user to introduce sample delays in the loop as well as alter the loop gain. Using these features, the system can be driven to instability and the actual phase and gain margins can be obtained and compared with the theoretically derived values. Disturbance response is also assessed in light of the robustness concepts.

1.3.1.4. Position Control

Control of motor position is a natural way to introduce the benefits of derivative action. The student is asked to design a PID controller to specifications and analyze its response to step inputs, triangular inputs, and disturbances. The controller is implemented in the QIC module and the user assesses the effects of the three gains on system performance. With derivative action, the effect of measurement noise is also clearly visible. This gives a nice way to introduce noise filtering. Disturbance response is evaluated with and without integral control. Response to triangular inputs is also assessed.

1.3.1.5. Haptic Interaction

To illustrate that control is much more than the servo and regulation problem, we have also included some elementary haptics experiments. The student is introduced to impedance control using a feedback system. The joint stiffness and damping are derived using a position PD controller. It is shown that a haptic knob can be simulated by combining a PID position controller with a finite-state machine. The student can define detents and step sizes on the motor shaft. The motor shaft behaves as though it is a notched knob using software running on the QIC only.

Introduction

The effectiveness of haptics in manual control is also illustrated by a second haptics experiment. The motor shaft is used as an input device to control a virtual ball and beam setup. The virtual ball and beam system is graphically animated on the laptop computer in real-time as shown in Figure 1.3.

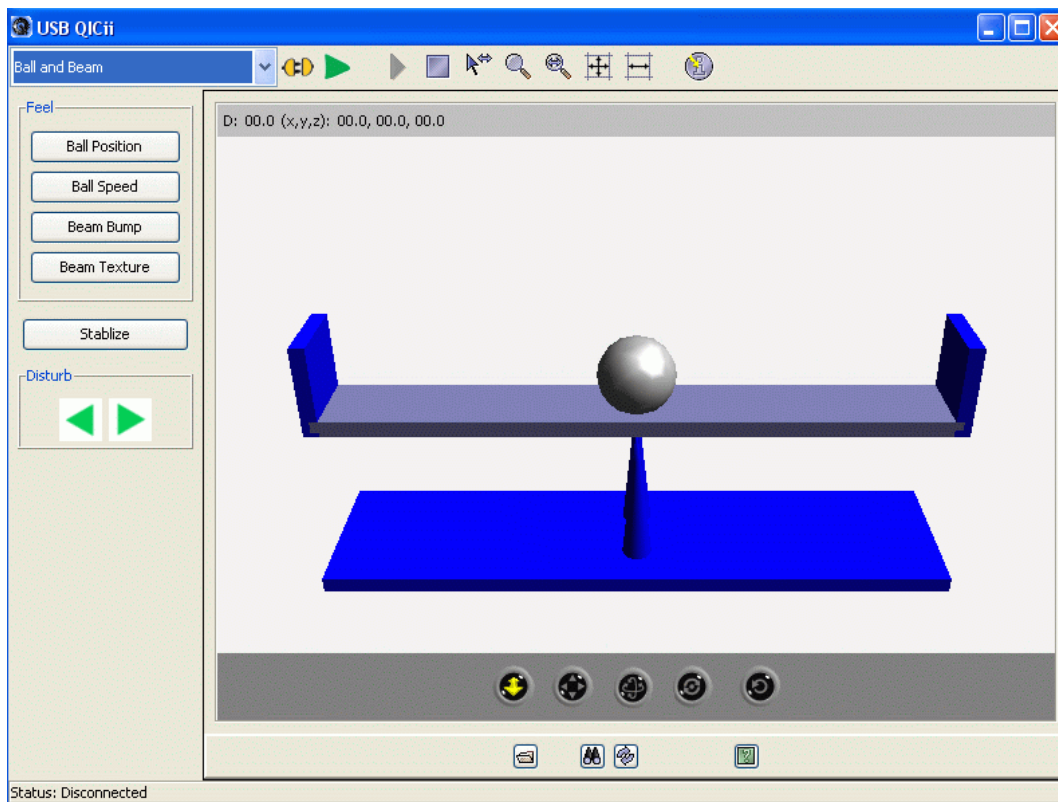


Figure 1.3 Screen Capture Of The Haptic Ball And Beam System

In the virtual ball and beam experiment, the DCMCT motor shaft is used to command the beam angle. Ball dynamics are simulated in real-time on the laptop. Force feedback is used to feed different signals back to the shaft. The user can select to feel a variety of effects such as ball speed, ball position, and beam texture from the simulation via the motor shaft. The graphics representation is in three Dimensions (3D) and runs in real-time. The student is asked to assess whether it is easier to balance the ball on the beam using haptic feedback. Analysis on the potential pitfalls is requested.

1.3.2. CAD Files

All the calculations and equations derived in this workbook, and more particularly in the pre-lab assignments, are carried out in Maple worksheets supplied on the accompanying

CD. Using Maple 8 or later, a worksheet can be edited or re-configured by the instructor and the equations automatically re-derived, accordingly, by Maple.

Some of the pre-lab assignments in the Robustness Chapter require the writing of MATLAB scripts. The solution files to these assignments are also supplied on the accompanying CD.

1.3.3. Chapter Structure

1.3.3.1. Pre-Laboratory Assignments



The pre-laboratory assignments must be performed by every student before they go to the laboratory session and run the actual laboratory.

1.3.3.2. Solutions And Typical Results



Regarding the Gray Boxes:

The gray boxes present in the **instructor manual** are not intended for the students as they provide solutions to the pre-lab assignments and contain typical experimental results from the laboratory procedure.

1.3.3.3. Marking Scheme

A marking scale is used at the end of each question to evaluate the student performance. The evaluation scale is described in Table 1.1 and should be ticked by the instructor when marking.

<i>Marking Scale</i>	<i>Designation</i>	<i>Description</i>
0	Poor	Most answers and/or experimental results are wrong.
1	Average	About half of the answers and/or results are correct.
2	Excellent	Most answers and/or experimental results are correct.



Table 1.1 Marking Scale Description

1.3.3.4. Results Summary Tables

Every laboratory contains two results summary tables that should be completed by the students.

The first table is called the *Pre-Laboratory Assignment Results* table. It should be

completed after all the pre-lab assignments are done. Please refer to the table of interest to resolve the pre-laboratory Section pertinent to the results.



Note:

The Teaching Assistant or Laboratory Supervisor should ensure that the table has been properly and fully completed before the student is allowed to perform the actual experiment. If the table is not completed, then the student cannot perform the experiment successfully. Information from this table is required to perform the experiment.

The second table is called the *In-Laboratory Results* table. It should be completed **during the in-laboratory session**. This table will assist the student in keeping track of their results in a concise manner. The table is used to compare theoretical parameters and results with experimentally obtained values.



Note:

The Teaching Assistant or Laboratory Supervisor should ensure that the table has been properly and fully completed before the student leaves the in-laboratory session.

Both tables are useful for quick and easy assessment of the work performed by the student. Of course honesty is assumed.

1.4. Curriculum Summary And Scheduling

The DCMCT system does not require the space and expense typically required for undergraduate control laboratories. The experiment can be signed out by a student and taken anywhere he or she wishes to perform the required experiment, at anytime (even on an evening).

The teaching material presented in this workbook can be divided into manageable two- to three- hour work periods, each of which covering one of the pre-laboratory or experimental sessions. One possible teaching outline is described in Table 1.2.

<i>Session Name</i>	<i>Pre-Lab Section(s)</i>	<i>In-Lab Section(s)</i>	<i>Laboratory Topics</i>
Modelling 1	2.5.1 - 2.5.2	2.6.2	Motor Static Relations Motor Parameter Estimation
Modelling 2	2.5.3	2.6.3	Dynamic Modelling: Bump test Model Fitting
Speed 1	3.5.1	3.6.1 – 3.6.5	Qualitative Properties Of PI Control Ziegler-Nichols Tuning Method Set-Point Weighting PI Controller Design To Specifications
Speed 2	3.5.2 – 3.5.4	3.6.6 – 3.6.8	Integrator Windup Protection Tracking Ramp Signals Response To Load Disturbances
Robustness	4.5.2 – 4.5.3	4.6.2	Sensitivity Complementary Sensitivity Nyquist Diagram Stability Margins
Position 1	5.5.1 – 5.5.3	5.6.2 – 5.6.3	PD Position vs. PI Speed Controls System Achievable Performance Qualitative Properties Of PD Control PD Controller Design To Specifications
Position 2	5.5.4 – 5.5.5	5.6.4 – 5.6.5	Tracking Ramp Signals Response To Load Disturbances
Haptics	6.4.1	6.5.3 – 6.5.5	Impedance Control Haptic Knob Haptic Ball And Beam

Table 1.2 Laboratory Curriculum

As far as scheduling is concerned, you may either assign in-laboratory sessions, or alternatively, let the student sign out the system for a 24-hour period. This should suffice to complete the work described in one row of Table 1.2. In doing so, one QET-DCMCT system can be used by one student every day. Therefore, to run one row of Table 1.2 per week with a class of 40 students, the instructor will need $40/5 = 8$ QET-DCMCT units.

Alternatively, if the instructor leaves one lab open for 8 hours every day and schedules two-hour sessions for each in-lab exercise, one DCMCT system can be used by $4 \times 5 = 20$ students (or groups) per week. Having 8 QET-DCMCT units set up in the lab results then in 160 students per week.

1.5. System Requirements

The laboratories described in this workbook are performed using the QET DCMCT module equipped with a USB QIC board and the USB QICii (QIC interactive interface) software. A full description of the system is provided in Appendices A and B.

1.6. References

- [1] Bristol, E.H. (1986) An industrial point of view on control teaching and theory. *IEEE Control Systems Magazine*, 1986, 6:1: pp 24—27.
- [2] Kheir, N.A., Åström, K.J., Auslander, D., Cheok, K.C., Franklin G.F., Masten, M., and Rabins, M. (1996) Control Systems Engineering Education, *Automatica*, 1996, 32:2, pp 147—166.
- [3] Murray, R.M., Åström, K.J., Boyd, S.P., Brockett, R.W., and Stein, G. Future directions in control in an information rich world. *IEEE Control Systems Magazine*, 2003, 23:2: pp 20—33.
- [4] Murray, R.W. (editor) Control in an Information Rich World. Report of the Panel on Future Directions in Control, Dynamics and Systems. SIAM 2003.
- [5] Åström, K.J. and Östberg, A.-B. (1986) A teaching laboratory for process control, *IEEE Control Systems Magazine*, 1986, 6:5: pp 37—42.
- [6] Åström, K.J. and Lundh, M. (1992) Lund Control Program Combines Theory with Hands-On Experience, *IEEE Control Systems Magazine*, 12:3, pp 22—30.

2. Modelling

2.1. Laboratory Objectives

The objective of this laboratory is to develop your understanding of modelling for control design and to practice modelling skills using the DC Motor Control Trainer (DCMCT).

In particular you will:

- Practice first principles modelling.
- Determine parameters by experimental tests.
- Develop insight into linear and nonlinear aspects of the model.
- Practice modelling from experimental tests.
- Understand the effects of model uncertainty and disturbances.

2.2. Preparation And Pre-Requisites

Before the proceeding you should review modelling from your text book. A brief summary is given in Section 2.3. You should also review the following material from courses in mechanics and physics:

- Calculation of moments of inertia.
- Basic Newtonian mechanics.
- Forces on a conductor in a magnetic field.
- Voltage generated in a conductor that moves in a magnetic field.

You should also familiarize yourself with the system by reading Appendices A and B.

2.3. Introduction

Modelling is an essential aspect of control. The key elements of modelling are:

- Get an overview of the system and its components.
- Understand the system and how the system works.

- First principles component modelling.
- Modelling from experimental tests – Parameter Estimation.
- Disturbances.
- Model Uncertainty.
- Model Validation.

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 work is guided by focusing on the variables that are of primary interest for control.

The block diagram gives a natural partition of the system. Mathematical descriptions of the behaviour of the subsystems representing each block is necessary to have 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 from first principles by applying the basic physical laws that describe the subsystems or by experiments on a real system. First principles modelling 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 first principles modelling. This can also be used when the knowledge required for first principles modelling is not available. 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. Frequency response has the advantage that very accurate measurements can be obtained by using correlation techniques. A model can in principle be obtained by applying any type of input to the system and analyzing the corresponding response. There is abundant literature and good software available to do this. In this lab we will emphasize simple methods.

Even if linear models are often adequate for control it is important to have an assessment of the major nonlinearities of a system. Saturation of actuators is one nonlinearity that is important for all control systems.

Disturbances are often an important issue in control system design. They may appear both as load disturbances which drive the system away from its desired behaviour and measurement noise that distorts the information about the system obtained from the sensors.

An assessment of the disturbances is an important part of modelling for control.

When determining parameters experimentally it is customary to also provide estimates of the accuracy of the parameters. Similarly when modelling dynamical systems for the purpose of controlling them, it is also essential to provide some measure of model accuracy. When modelling dynamical systems the precision of a model can be expressed in terms of the accuracy of the parameters. This is however not sufficient because there may be errors due to neglected dynamics. It is important for design of control systems to make some assessment of this unmodeled dynamics. An assessment of the unmodeled dynamics is therefore an important aspect of modelling for control.

When a model is obtained, it is 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.

2.4. Nomenclature

The following nomenclature, as described in Table 2.1, is used for the system open-loop modelling.

<i>Symbol</i>	<i>Description</i>	<i>Unit</i>
ω_m	Motor speed which can be computed from the motor angle	rad/s
V_m	Voltage from the amplifier which drives the motor	V
T_d	Disturbance torque externally applied to the inertial load	N.m
T_m	Torque generated by the motor	N.m
I_m	Motor Armature 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
h	Sampling Interval	s
s	Laplace Operator	rad/s
t	Continuous Time	s

Table 2.1 Open-Loop System Nomenclature

2.5. Pre-Laboratory Assignments: First Principles Modelling

The DCMCT motor, inertial load, power amplifier, encoder, and the filter to obtain the velocity are modelled by the *Motor And Inertial Load* subsystem, as represented in Figure 2.1. The block has one input the voltage to the motor V_m and one output the angular velocity of the motor ω_m . Additionally, a second input is also considered: the disturbance torque, T_d , applied to the inertial load.

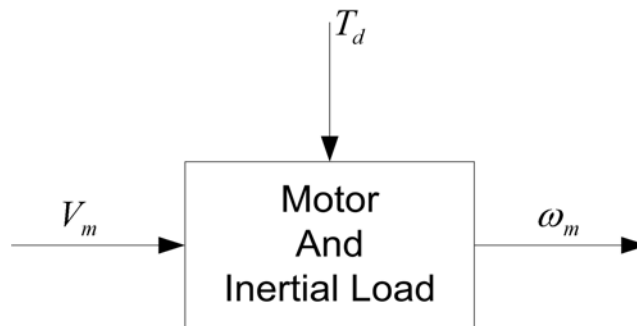


Figure 2.1 The *Motor And Inertial Load* Subsystem

In the following, the mathematical model for the *Motor And Inertial Load* subsystem is derived through first principles.

2.5.1. Motor First Principles

Please answer the following questions.

1. When using SI units, the motor torque constant, k_t , is numerically equal to the back-electro-motive-force constant, k_m . In other words, we have:

$$k_t = k_m \quad [2.1]$$

This workbook uses the SI (International System) units throughout and the motor parameter named k_m represents both the torque constant and the back-electro-motive-force constant.

Considering a single current-carrying conductor moving in a magnetic field, derive an expression for the torque generated by the motor as a function of current and an

expression for the back-electro-motive-force voltage produced as a function of the shaft speed. Show that both expressions are affected by the same constant, as implied in relation [2.1].

Solution:

0	1	2
---	---	---

2. Figure 2.2 represents the classic schematic of the armature circuit of a standard DC motor.

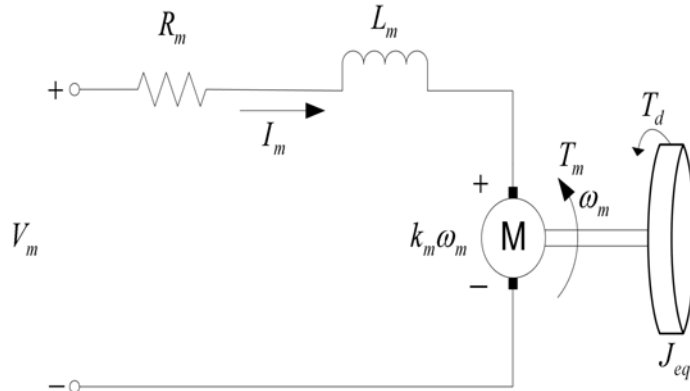


Figure 2.2 DC Motor Electric Circuit

Determine the electrical relationship characterizing a standard DC motor.

Solution:

0 1 2

- Determine and evaluate the motor electrical time constant τ_e . This is done by assuming that the shaft is stationary. You can find the parameters of the motor in Table A.2.

Solution:

0	1	2
---	---	---

- Assume τ_e is negligible and simplify the motor electrical relationship previously determined. What is the simplified electrical equation?

Solution:

0	1	2
---	---	---

5. Neglecting the friction in the system, derive from first dynamic principles the mechanical equation of motion of a DC motor. Determine the total (or equivalent) moment of inertia of motor rotor and the load.

Solution:

0	1	2
---	---	---

6. Calculate the moment of inertia of the inertial load which is made of aluminum. Also, evaluate the motor total moment of inertia. The motor data sheet is given in Table A.2.

Solution:

0	1	2
---	---	---

2.5.2. Static Relations

Modelling by experimental tests on the process is a complement to first principles modelling. In this section we will illustrate this by static modelling. Determining the static relations between system variables is very useful even if it is often neglected in control systems studies. It is useful to start with a simple exploration of the system.

Please answer the following questions.

1. Assuming no disturbance and zero friction, determine the motor maximum velocity: ω_{\max} .

Solution:

0	1	2
---	---	---

2. Determine the motor maximum current, I_{\max} , and maximum generated torque, T_{\max} .

Solution:

0	1	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 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.

Solution:

0	1	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 and measuring both corresponding steady-state current and speed (in radians per second). Assuming that the motor resistance is known, derive an expression that will allow you to solve for k_m . What is the effect of the inertia of the inertial load on the determination of the motor constant?

Solution:

0	1	2
---	---	---

5. Applying first principles, find an estimate of the measurement noise, e_ω , present in the velocity signal ω_m . Note that the encoder resolution is 4096 counts per revolution and the sampling period h is 0.01 s.

Hints:

Consider the position measurement error e_θ .

You can also use e_h the error in the sampling interval (a.k.a. jitter).

The following series expansion can also be relevant:

$$\frac{1}{1+x} = 1 - x + O(x^2) \quad [2.2]$$

Solution:

Solution (continued):



0	1	2
---	---	---

2.5.3. Dynamic Models: Open-Loop Transfer Functions

Please answer the following questions.

1. Determine the transfer function, $G_{\omega,V}(s)$, of the motor from voltage applied to the motor to motor speed.

Hint:

The motor armature inductance L_m should be neglected.

Solution:

0	1	2
---	---	---

2. Express $G_{\omega, v}(s)$ as a function of the parameters a and b , defined such as:

$$G_{\omega, v}(s) = \frac{b}{s + a} \quad [2.3]$$

Solution:

0	1	2
---	---	---

3. Express $G_{\omega, v}(s)$ as a function of the parameters K and τ , defined such as:

$$G_{\omega, v}(s) = \frac{K}{\tau s + 1} \quad [2.4]$$

Solution:

0	1	2
---	---	---

4. Determine the transfer function, $G_{\omega,T}(s)$, from disturbance torque applied to the inertial load to motor speed. Express $G_{\omega,T}(s)$ as a function of the parameters K_{Td} and τ_{Td} , as defined below:

$$G_{\omega,T}(s) = \frac{K_{Td}}{\tau_{Td}s + 1} \quad [2.5]$$

Show that $\tau_{Td} = \tau$.

Solution:

0	1	2
---	---	---

5. From your previous calculations and the motor parameters given in Table A.2, summarize the system parameter values relevant to evaluate the transfer functions derived so far.

Solution:

0	1	2
---	---	---

6. Evaluate both transfer functions $G_{\omega,v}(s)$ and $G_{\omega,T}(s)$, as well as their parameters a , b , K , τ , and K_{Td} .

Solution:

0	1	2
---	---	---

7. Derive using basic blocks (gain, integrator, summer) the plant open-loop block diagram.

Solution:

0	1	2
---	---	---

8. Simplify the obtained open-loop block diagram so that it has the block structure depicted in Figure 2.3. Fill up the empty blocks.

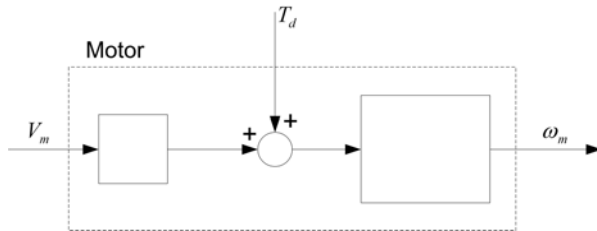


Figure 2.3 Simplified Open-Loop Block Diagram Template

Solution:

0	1	2
---	---	---

9. The transfer function $G_{\omega,v}(s)$ previously derived is only an approximation since the inductance of the motor has been neglected. Considering the motor electrical time constant τ_e previously evaluated, justify the approximation.

Solution:

0	1	2
---	---	---

2.5.4. Pre-Laboratory Results Summary Table

Table 2.2 should be completed before you come to the in-laboratory session to perform the experiment.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Value</i>	<i>Unit</i>
2.5.1. Motor First Principles				
3.	Motor Electrical Time Constant	τ_e		s
6.	Moment Of Inertia Of The Disc Load	J_l		kg.m ²
6.	Total Moment Of Inertia	J_{eq}		kg.m ²
2.5.2. Static Relations				
1.	Motor Maximum Velocity	ω_{max}		rad/s
2.	Motor Maximum Current	I_{max}		A
2.	Motor Maximum Torque	T_{max}		N.m
2.5.3. Dynamic Models				
5.	Motor Torque Constant	k_m		N.m/A
5.	Motor Armature Resistance	R_m		Ω
6.	Open-Loop Model Parameter a	a		kg.m/(W.s ⁴)
6.	Open-Loop Model Parameter b	b		1/(V.s ²)
6.	Open-Loop Steady-State Gain	K		rad/(V.s)
6.	Open-Loop Time Constant	τ		s
6.	Open-Loop Torque Disturbance Gain	K_{Td}		rad/(N.m.s)

Table 2.2 Modelling Pre-Laboratory Assignment Results

2.6. In-Laboratory Session

2.6.1. QICii Modelling Module

2.6.1.1. Module Description

The main tool for this lab is the front panel of the module entitled *Modelling* in the QICii software, which should be similar to the one shown in Figure 2.4.

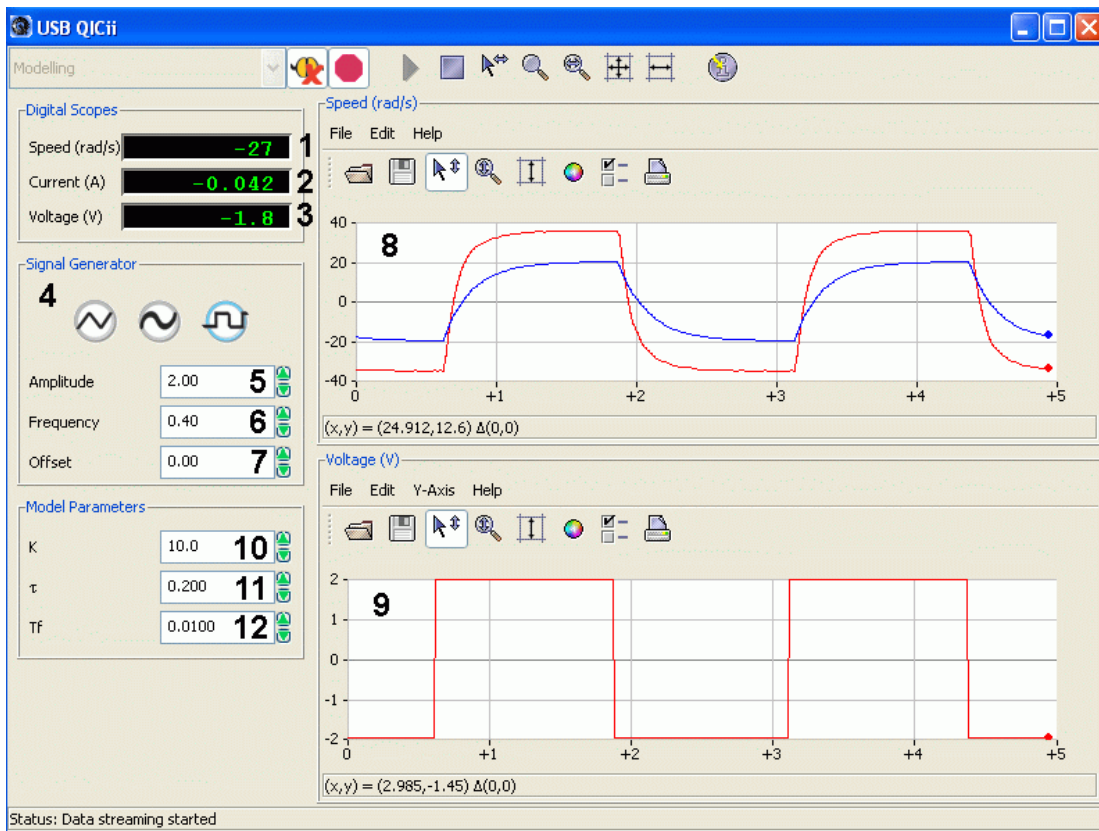


Figure 2.4 *Modelling* Module Of The QICii Software

As a quick module description, Table 2.3 lists and describes the main elements composing the QICii *Modelling* module user interface. Every element is uniquely identified through an ID number and located in Figure 2.4.

<i>ID #</i>	<i>Label</i>	<i>Parameter</i>	<i>Description</i>	<i>Unit</i>
1	Speed	ω_m	Motor Output Speed Numeric Display	rad/s

<i>ID #</i>	<i>Label</i>	<i>Parameter</i>	<i>Description</i>	<i>Unit</i>
2	Current	I_m	Motor Armature Current Numeric Display	A
3	Voltage	V_m	Motor Input Voltage Numeric Display	V
4	Signal Generator		Type of Generator For The Input Voltage Signal	
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) Motor Speeds	rad/s
9	Voltage	V_m	Scope With Applied Motor Voltage (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 Measured Signal	s

Table 2.3 QICii *Modelling* Module Nomenclature

The *Modelling* module program runs the process in open-loop using the motor voltage given by the signal generator. There are two windows that show the time histories of motor speed and motor voltage.

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 window 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:

$$\omega_s(s) = \frac{K V_m(s)}{\tau s + 1}$$

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

$$h = 0.01 [s]$$

The actual speed is obtained by filtering the position signal using the following filter:

$$\omega_m = \frac{s \theta_m}{T_f s + 1}$$

where θ_m is the position of the motor shaft measured by the encoder.

2.6.1.2. Module Startup

To start and use the *Modelling* module, follow the steps described below:

- Step 1. Power up the DCMCT. LED2 should light up while LED3 should flash on and off repeatedly.
- Step 2. Press the *Reset* button on the QIC. Again, LED2 should remain on while LED3 should flash on and off repeatedly.
- Step 3. Press the DCMCT User Switch (i.e. pushbutton next to the two flashing LED's). LED2 and LED3 should both turn off.
- Step 4. Launch the USB QICii software and select *Modelling* in the drop-down menu.
- Step 5. Select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller. LED2 should light up while LED3 should still be off, and the controller should start running.

Note:

The drop-down menu will be disabled (i.e. is unavailable) while the controller is running. You must select the *Stop controller* button on top of the QICii window to stop the controller (this should also turn off LED2, and enable the drop-down menu again). Once the drop down menu is enabled again and the controller is stopped, you can select any one of the other controller experiments, if you want. After selecting some other controller experiment, you can once again select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller of your choice.

The default module parameters loaded after download are given in Table 2.4.

<i>Signal Type</i>	<i>Amplitude [V]</i>	<i>Frequency [Hz]</i>	<i>Offset [V]</i>	<i>K [rad/(V.s)]</i>	<i>τ [s]</i>	<i>T_f [s]</i>
Square Wave	2.0	0.4	0.0	10.0	0.2	0.01

Table 2.4 Default Parameters For The *Modelling* Module

2.6.2. Static Relations

2.6.2.1. Initial Experimental Tests

Objectives

- Determine the maximum velocity and compare with calculations.
- Determine the Coulomb friction.
- Explain your findings and summarize them briefly.

Experimental Procedure

A procedure of this type is very useful to make sure that a system functions properly. Please follow the steps described below.

Step 1. Run the system open-loop by changing the voltage to the motor. The motor voltage is set by the signal generator. With zero signal amplitude, change the signal offset to generate a constant voltage. Sweep the voltage gently over the full signal range and observe the steady-state speed, current, and velocity. What happens to the variables as you change the offset?

Solution:

0	1	2
---	---	---

Step 2. Determine the maximum velocity and compare with calculations.

Note:

Although the motor maximum input voltage is 15 V, the *Offset* numeric input is limited to 5 V.

Solution:

0	1	2
---	---	---

Step 3. Start with zero voltage on the motor and increase the voltage gradually until the motor starts to move. Determine the voltage when this occurs. Repeat the test with negative voltages.

Solution:

0	1	2
---	---	---

2.6.2.2. 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 different from zero then the motor will spin in one direction, since a constant voltage is 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 the following table (i.e. Table 2.5). For each measurement **hold the motor shaft stationary** by grasping the inertial load to stall the motor. 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 before noting it down.

Sample: i	$V_m(i)$ [V]	Offset in Measured Current: I_{bias} [A]
0	0	


Sample: i	$V_m(i)$ [V]	Measured Current: $I_{meas}(i)$ [A]	Corrected for Bias: $I_m(i)$ [A]	Resistance: $R_m(i)$ [Ω]
1	-5			
2	-4			
3	-3			
4	-2			
5	-1			
6	1			
7	2			
8	3			
9	4			
10	5			
Average Resistance: R_{avg} [Ω]				

Table 2.5 Motor Resistance Experimental Results

0	1	2
---	---	---

Step 2. From Table 2.5, above, calculate for each iteration the motor resistance $R_m(i)$ and obtain an average value for it, R_{avg} . Explain the procedure you used to estimate the resistance R_m .

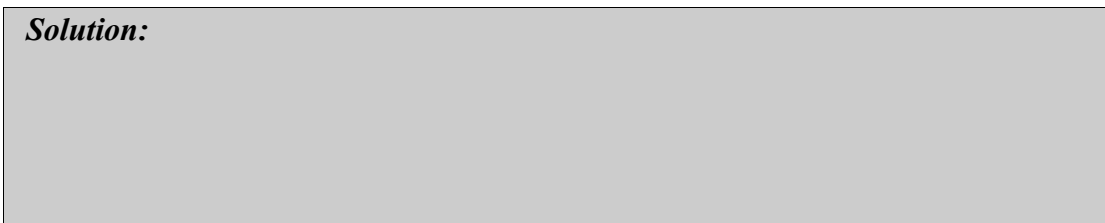
Solution:



0	1	2
---	---	---

Step 3. The system parameters are given in Table A.2. Compare the estimated value for R_m (i.e. R_{avg}) with the specified value and discuss your results.

Solution:



0	1	2
---	---	---

2.6.2.3. Estimate The Motor Torque Constant

To experimentally estimate the motor back-EMF constant, follow the steps described below:

Step 1. *With the motor free to spin*, apply the same procedure as above and fill the

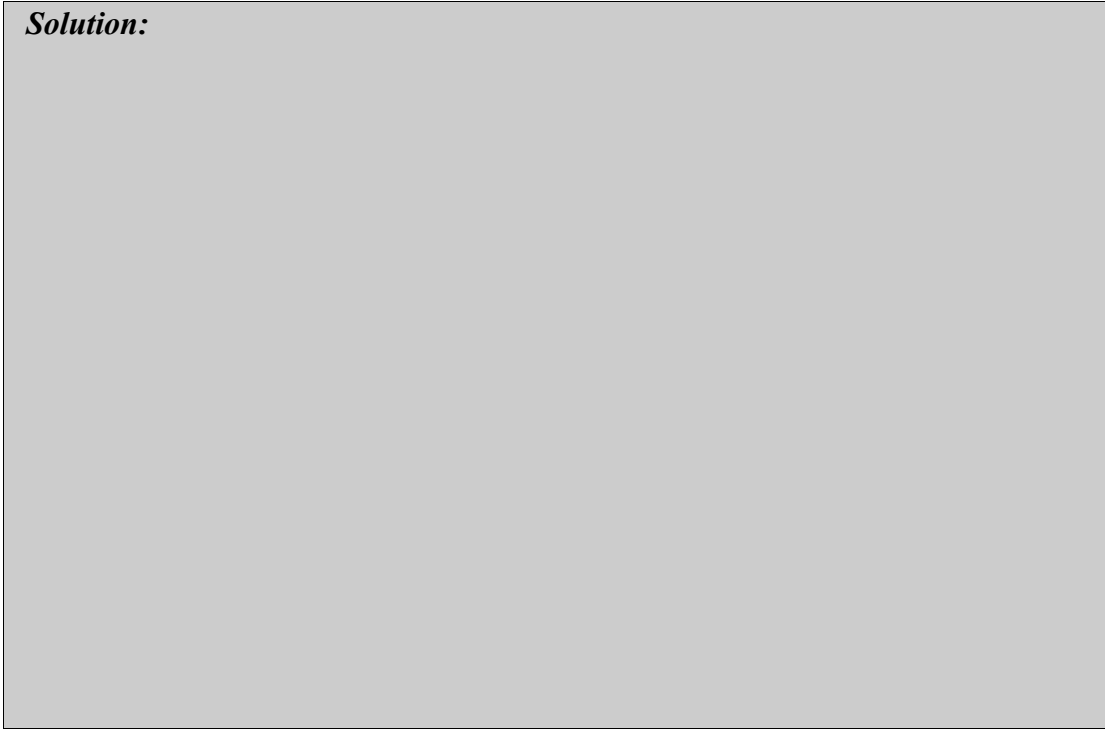
following table (i.e. Table 2.6). You can read a value for the motor angular speed from the digital display. Wait a few seconds after you enter a new voltage value as the displayed speed values are low-pass filtered. The angular speed value is in radians per seconds. The current measurement may have an offset which you will need to account for. The speed measurement will have a very small offset which will need to be compensated for. Calculate the motor back-EMF constant for each measurement iteration and then calculate an average for the 10 measurements. You should use the value of R_m that you estimated in the previous section.

<i>Sample:</i> <i>i</i>	$V_m(i)$ [V]	I_{bias} [A]	R_{avg} [Ω]		
0	0				
<i>Sample:</i> <i>i</i>	$V_m(i)$ [V]	<i>Measured Speed:</i> $\omega_m(i)$ [rad/s]	$I_{meas}(i)$ [A]	$I_m(i)$ [A]	$k_m(i)$ [V.s/rad]
1	-5				
2	-4				
3	-3				
4	-2				
5	-1				
6	1				
7	2				
8	3				
9	4				
10	5				
Average Back EMF-Constant: k_{m_avg} [V.s/rad]					

Table 2.6 Back-EMF Constant Experimental Results

Step 2. Explain the procedure you used to estimate k_m .

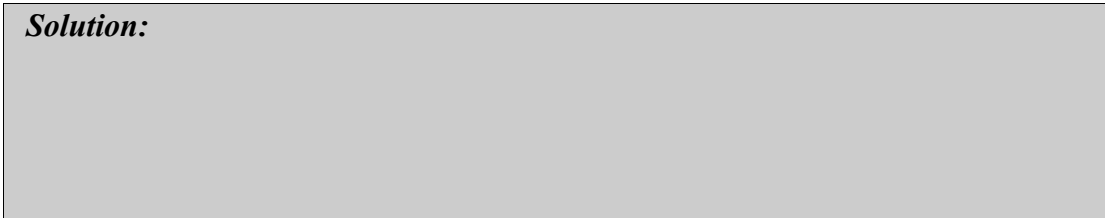
Solution:



0	1	2
---	---	---

Step 3. The system parameters are given in Table A.2. Compare the estimated value for k_m (i.e. k_{m_avg}) with the specified value and discuss your results.

Solution:



0	1	2
---	---	---

2.6.2.4. Obtain The Motor Transfer Function

From the above estimates, obtain a numerical expression for the motor open-loop transfer function $G_{\omega,V}$. What are the estimated open-loop steady-state gain and time constant? How does this compare with the open-loop transfer function you obtained in Section 2.5.3 Question 5?

Solution:

0	1	2
---	---	---

2.6.2.5. Estimate The Measurement Noise

The measurement noise can be determined experimentally. Please follow the steps described below:

Step 1. Determine the measurement noise for speed control by running the motor with a constant voltage and observing the fluctuations in the velocity.

Step 2. Does the noise level depend on the velocity? Compare your previous estimate of the variations from first principles derived in Section 2.5.3 Question 6 with the experimental results. Do you also observe any repeatable fluctuations in your velocity signal? Can you find their cause?

Hint:

Can the fluctuations in your velocity signal be related to the motor position?

Solution:



0	1	2
---	---	---

2.6.3. Dynamic Models: Experimental Determination Of System Dynamics

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

2.6.3.1. The Bump Test

Preamble

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

$$G_{\omega, \nu}(s) = \frac{K}{\tau s + 1} \quad [2.6]$$

can easily be fitted to the data. A bump test is illustrated in Figure 2.5.

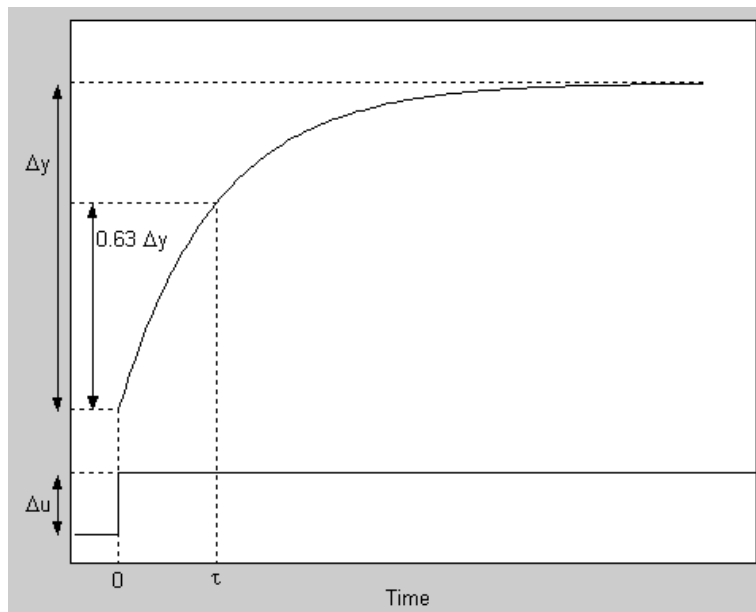


Figure 2.5 Bump test Input And Output

Assume that the input changes with Δu and that the corresponding change in the steady-state output is Δy . An estimate of the steady-state gain is then given by:

$$K = \frac{\Delta y}{\Delta u} \quad [2.7]$$

The quantity τ is approximately given by the time the output has reached 63% of its total change.

Experimental Procedure

Please read Appendix B which describes how to use the QICii plots to take measurements of the acquired data, to start and stop the plots, and to measure point coordinates on the plots. Please follow the steps below:

Step 1. Apply a series of step inputs to the open-loop system by setting the QICii module parameters as described in Table 2.7.

<i>Signal Type</i>	<i>Amplitude [V]</i>	<i>Frequency [Hz]</i>	<i>Offset [V]</i>	<i>K [rad/(V.s)]</i>	<i>τ [s]</i>
Square Wave	2	0.4	3	0	0.0

Table 2.7 Module Parameters For The Bumptest

Step 2. The open-loop controller now applies a constant-amplitude voltage square wave to the motor. Step voltages are applied to the motor from the signal generator with a period that is so long that the system well reaches steady-state at each step. The motor should run at the corresponding constant speeds. Determine the transfer function from voltage V_m to angular velocity ω_m using the obtained bump tests.

Solution:

0	1	2
---	---	---

Step 3. Determine the parameters K and τ of the model defined in [2.6] and compare them with the model obtained by first principles in Section 2.5.3 Question 6.

Step 4. Repeat the procedure for a few different signal amplitudes, as well as for rising and falling steps. Average your estimated parameters.

Solution:

0	1	2
---	---	---

2.6.3.2. Model Validation

Preamble

A simple form of model validation can be done because the software *Modelling* contains a first-order simulation, whose model expressed in Equation [2.6] 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 front panel. The output of the model is displayed together with the actual motor speed. You can explore this in the following procedure.

Experimental Procedure

Follow the steps described below:

Step 1. The fact that $K = 0$ means that the model output is zero. Activate the model by changing the simulation parameters K and τ .

Step 2. Set K and τ to the values you previously estimated from the bump test. Do you obtain a good fit between the estimated and the actual responses?

Solution:

0	1	2
---	---	---

Step 3. 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.

Step 4. Change the signal amplitude to see if you get better fit by changing the parameter values.

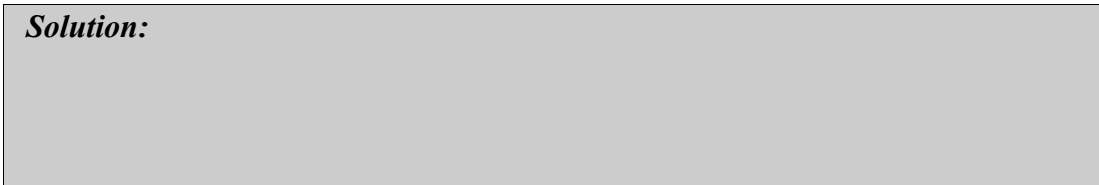
Solution:



0	1	2
---	---	---

Step 5. Compare with the results of first principles modelling in Section 2.5.3, Question 6.

Solution:



0	1	2
---	---	---

Step 6. Is your model valid? Summarize your observations briefly.

Solution:



0	1	2
---	---	---

2.6.4. Concluding Remarks

2.6.4.1. Load Disturbances and Measurement Noise

There are typically two types of disturbances in a control system. Load disturbances that drive the system away from its desired behaviour and measurement noise that corrupts the information obtained from the sensors.

Since the motor does not do any useful work there are no real load disturbances in this case. A load disturbance can be simulated by gently touching the inertial load with your finger. Load disturbances can also be simulated by injecting an extra voltage on the motor.

The major noise source for position control is due to the quantization of the angle measurements due to the encoder.

2.6.4.2. Automating The Tests

The experimental tests you have done can easily be automated. Measurement of motor resistance R_m and motor constant k_m can be done as follows:

- Resistance measurement:

Keep the wheel fixed with a clamp. Sweep the voltage slowly for a full cycle, measure the current, display curve, and present the linear fit and a measure of deviation from linearity.

- Current constant:

Free wheel. Sweep the voltage slowly for a full cycle, measure the speed, display curve, and present the linear fit and a measure of deviation from linearity.

The system parameter estimation procedures can also be automated by replacing manual search by an optimization algorithm. Automated test procedures of this type are essential to ensure quality in mass manufacturing.

2.6.4.3. Nonlinearities

Many aspects of control can be dealt with using linear models. There are however some nonlinear aspects that always have to be taken into account. The major nonlinearities are:

- Saturation of the motor amplifier.
- Friction in the motor.
- Quantization of the encoder.

It is very important to keep in mind that all physical variables are limited. The amplifier that drives the motor has a power supply that gives 15 V. This means that the voltage from the amplifier can never exceed $V_{\max} = 15$ V. A consequence is that the current through the

motor is also limited.

The limitation in signal ranges implies that the motor transfer functions $G_{\omega,V}$ and $G_{\omega,T}$ do not describe the system well for large signals.

The other main nonlinearities are due to Coulomb friction, approximately equivalent to 0.2-0.5 V, and quantization in the encoder $2\pi/4096 = 1.5 \cdot 10^{-3}$ rad.

2.6.4.4. Unmodeled Dynamics

When determining physical parameters it is customary to assign a precision to the values. There are uncertainties due to variations in component values, temperature variation of the armature resistance. It is therefore natural to give some measure of accuracy to the transfer functions $G_{\omega,V}$ and $G_{\omega,T}$. One way to do this is to give the accuracy of parameters such as K and τ . This does unfortunately not capture all relevant issues because the actual transfer function may be much more complicated than the simple first order system given by Equation [2.4], the system may even be nonlinear. This effect which is called unmodeled dynamics can be specified in many different ways. An estimate of the unmodeled dynamics is an essential aspect of modelling for control. It is equivalent to an error analysis in traditional measurements.

To have an indication of the accuracy of a model it is necessary both to have an estimate of the accuracy of its parameters and also an assessment of dynamics that has been neglected.

One obvious factor is that the controller and the computation of the velocity is implemented in a computer. The encoder gives values of the angle that are quantized with a resolution of $2\pi/4096 = 1.5 \cdot 10^{-3}$ rad. Since the controller is implemented on a computer there are also dynamic effects. A crude approximation is to assume that there is an extra time delay corresponding to half a sampling period.

The system has no sensor for velocity. The velocity is instead obtained by taking filtered differences of the position. A common rule of thumb is to approximate the effect of the computer by adding a delay of half a sampling interval or 0.005 s. Since the velocity is computed by taking differences of the angles between two sampling intervals there is an additional delay in the velocity signal of one sampling interval. Because of the extra sampling period required to compute velocity from the encoder position, the time delay will be approximately one and a half sampling interval. The signal is also filtered which introduces additional dynamics.

The inductance of the rotor has already been mentioned previously. The model we have obtained is an approximation because we have neglected the inductance in the motor rotor.

Taking the inductance into account, the motor electrical equation is given by Equation [2.s3]. Taking Laplace transforms of this and of Equation [2.s7], and eliminating the current we find:

$$(J_{eq} s (L_m s + R_m) + k_m^2) \Omega_m(s) = k_m V_m(s) + (L_m s + R_m) T_d(s)$$

The characteristic polynomial for both $G_{\omega,v}(s)$ and $G_{\omega,\tau}(s)$ is thus:

$$L_m s^2 J_{eq} + R_m J_{eq} s + k_m^2 = \left(J_{eq} s + \frac{k_m^2}{R_m} \right) (L_m s + R_m) - \frac{k_m^2 L_m s}{R_m}$$

The last term of the right-hand-side can be neglected if $k_m^2 L_m / R_m \ll J_{eq} R_m$ or $\tau_e \ll \tau$, where τ is the system mechanical time constant and τ_e the electrical time constant, both given by:

$$\tau = \frac{J_{eq} R_m}{k_m^2} \quad \text{and} \quad \tau_e = \frac{L_m}{R_m}$$

Therefore the right-hand-side can be approximated by:

$$\left(J_{eq} s + \frac{k_m^2}{R_m} \right) (L_m s + R_m)$$

A more accurate transfer function from voltage to motor speed is thus:

$$G_{\omega, v}(s) = \frac{k_m}{\left(J_{eq} s + \frac{k_m^2}{R_m} \right) (L_m s + R_m)}$$

which can also be expressed as:

$$G_{\omega, v}(s) = \frac{1}{k_m (\tau s + 1) (\tau_e s + 1)}$$

Introducing the numerical values we find $\tau = 0.0929$ s and $\tau_e = 0.0000774$ s, which means that the electrical time constant is much smaller than the time delay. The major contribution to the unmodeled dynamics is thus due to the effects of sampling. It is 0.005 s for the position signal and 0.015 s for the velocity signal (i.e. speed control).

2.6.5. In-Laboratory Results Summary Table

Table 2.8 should be completed using Table 2.2, which contains data from the pre-laboratory assignments, as well as experimental results obtained during the in-laboratory session.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Pre-Lab Value</i>	<i>In-Lab Result</i>	<i>Unit</i>
2.6.2.	<i>Static Relations</i>				
1.	Motor Maximum Velocity	ω_{\max}			rad/s
1.	Positive Coulomb Friction Voltage	V_{fp}	N/A		V
1.	Negative Coulomb Friction Voltage	V_{fn}	N/A		V
2.	Motor Armature Resistance	R_m			Ω
3.	Motor Torque Constant	k_m			N.m/A
4.	Open-Loop Steady-State Gain	K			rad/(V.s)
4.	Open-Loop Time Constant	τ			s
2.6.3.1.	<i>Dynamic Models: The Bump Test</i>				
	Open-Loop Steady-State Gain	K			rad/(V.s)
	Open-Loop Time Constant	τ			s
2.6.3.2.	<i>Dynamic Models: Model Validation</i>				
	Open-Loop Steady-State Gain	K			rad/(V.s)
	Open-Loop Time Constant	τ			s

Table 2.8 Modelling In-Laboratory Results

3. Speed Control

3.1. Laboratory Objectives

The objective of this laboratory is to develop an understanding of PI control (applied to speed), how it works, and how it can be tuned to meet required specifications.

In particular you will explore:

- Qualitative properties of proportional and integral action.
- Set-point weighting.
- Design of controllers for specifications on the set-point response.
- Integrator windup and windup protection.
- Tracking of triangular signals.
- Response to load disturbances.

3.2. Preparation And Pre-Requisites

A pre-requisite to this laboratory is to have successfully completed the modelling and model validation laboratory described in Chapter 2. Before the lab, you should also review Proportional-Integral (PI) control from your textbook.

From Chapter 2 and for the purpose of speed control, the system can be represented by the block diagram in Figure 3.1. This block diagram illustrates the parts of the system that are relevant for speed control.

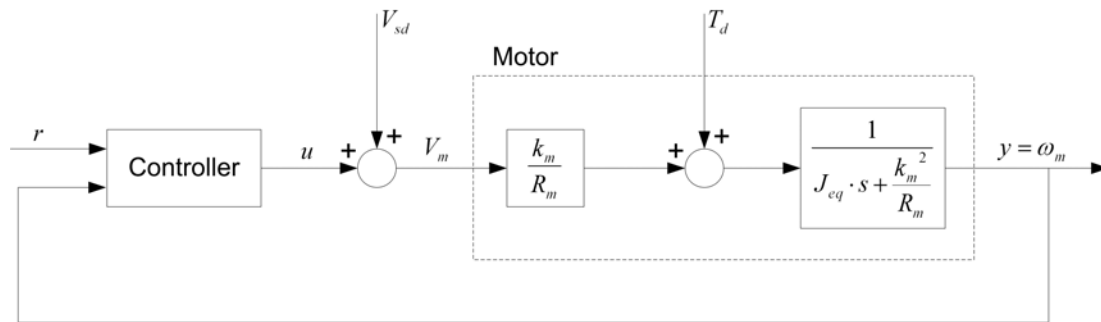


Figure 3.1 DCMCT Block Diagram For Speed Control

The process is represented by a block which has motor voltage V_m and torque T_d as inputs and motor speed ω_m as the output. The torque is typically a disturbance torque that you apply manually to the inertial load. The velocity is actually computed in the PIC by filtered differences of the motor angle θ_m using the following relationship:

$$\omega_m = \frac{s \theta_m}{T_f s + 1}$$

where T_f is the filter time constant and s the Laplace operator.

The controller block represents the control algorithm in the computer and the power amplifier. V_{sd} is a simulated external disturbance voltage. Make sure that you understand this fully.

The linear behaviour of the system is described by the transfer functions given in the block diagram. The major nonlinearities are the saturation of the motor amplifier at 15 V, Coulomb friction corresponding to 0.5 V, and the quantization of the encoder.

The major unmodeled dynamics is due to the effects of sampling and filtering. The former can be approximated by a time delay of one sampling period.

3.3. Introduction: The PI Controller

The PI controller is the most common control algorithm. It is used for a variety of purposes and it often works very well. For systems with simple dynamics it can give close to optimal performance and for processes with complicated dynamics, it can often give good performance provided that specifications are not too demanding. Better performance can, however, be obtained by using more complicated controllers.

3.3.1. PI Control Law

The linear behaviour of a PI controller can be described by:

$$u(t) = k_p (b_{sp} r(t) - y(t)) + k_i \int_0^t r(\tau) - y(\tau) d\tau \quad [3.1]$$

where $u(t)$ is the control signal, $r(t)$ the reference, and $y(t)$ the measured process output. The reference $r(t)$ is also called the set-point or the command. The linear behaviour of the controller is governed by three parameters:

- k_p : proportional gain
- k_i : integral gain
- b_{sp} : set-point weight

Further in this laboratory, we introduce a fourth parameter, a_w , which governs the nonlinear properties of the controller.

Sometimes the filtered measurement $y_f(t)$ is also used in the control loop. It is computed from T_f , the time constant of filter for measured signal. The filter time constant T_f is typically set to a constant value and it is often combined with the sensor. The filter provides roll-off at high frequencies. It is important to reduce the effects of sensor noise and it improves robustness. In this particular case the filtering is incorporated in the calculation of the velocity from the encoder signal.

3.3.2. The Magic Of Integral Action

A nice property of a controller with integral action is that it always gives the correct steady-state value provided that there is an equilibrium. This can be seen simply by assuming that there is a steady-state value with constant $u(t) = u_{ss}$, $r(t) = r_{ss}$, and $y(t) = y_{ss}$. Equation [3.1] can then be written as:

$$u_{ss} = k_p (b_{sp} r_{ss} - y_{ss}) + k_i (r_{ss} - y_{ss}) t$$

Since the left-hand-side is a constant, the right-hand-side must also be a constant. This requires that $y_{ss} = r_{ss}$. Notice that the only assumption that has been made about the process

is that there exists an asymptotic steady-state.

3.4. Nomenclature

The following nomenclature, as described in Table 3.1, is used for the system modelling and control design.

<i>Symbol</i>	<i>Description</i>	<i>Unit</i>
ω_m	Motor speed which can be computed from the motor angle	rad/s
V_m	Voltage from the amplifier which drives the motor	V
T_m	Torque generated by the 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 External 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
s	Laplace Operator	rad/s
h	Sampling Interval	s
t	Continuous Time	s
k_p	Proportional Gain	V.s/rad
k_i	Integral Gain	V/rad
b_{sp}	Set-Point Weight	
a_w	Windup Protection Parameter	
u	Control Signal	V
r	Reference Signal	rad/s
y	Measured Process Output	rad/s
T_f	Time Constant of filter for measured signal	s

Table 3.1 System Modelling Nomenclature

3.5. Pre-Laboratory Assignments

3.5.1. PI Controller Design To Given Specifications

In this section you will practice design of a controller for a specified response to reference signals. One of the key tasks in design is to assess fundamental limitations and assess the performance that can be achieved. The obtained design will then be verified by an experimental procedure in Section 3.6.5.

The following problem will be investigated: use the mathematical model of the process to design a PI controller that gives a step response with:

- i) no overshoot.
- ii) the following closed-loop undamped natural frequency:

$$\omega_0 = 16.0 \left[\frac{\text{rad}}{\text{s}} \right]$$

The relation between motor velocity and motor voltage can be modeled by the following transfer function:

$$G_{\omega, v}(s) = \frac{K}{\tau s + 1} \quad [3.2]$$

Please answer the following questions.

1. Using the PI control law [3.1] and the process transfer [3.2], determine the closed-loop block diagram used for speed control. Include the simulated disturbance voltage V_{sd} .

Solution:

0	1	2
---	---	---

2. Assuming no disturbance, determine the closed-loop transfer function, $G_{\omega,r}(s)$, from reference to output, as defined below:

$$G_{\omega,r}(s) = \frac{\Omega_m(s)}{R(s)}$$

Solution:

0	1	2
---	---	---

3. One possible way to design a controller is to choose controller gains which give a specified characteristic polynomial. One possibility is to choose gains which give the following quadratic characteristic equation:

$$s^2 + 2 \zeta \omega_0 s + \omega_0^2 \quad [3.3]$$

Determine the PI controller parameters k_p and k_i so that the closed-loop system satisfies the specified characteristic equation [3.3]. That is to say derive k_p and k_i as functions of ω_0 , ζ , K , and τ .

Solution:

0	1	2
---	---	---

Speed Control

4. Choose ζ to get the fastest possible response without overshoot (i.e. critically damped), as dictated by the design requirements.

Solution:

0	1	2
---	---	---

5. Using k_p , k_i , and ζ as previously determined and assuming a controller with no set-point weighting ($b_{sp} = 0$), express $G_{\omega,r}(s)$ under a fully-factored form. Determine the closed-loop system time response equation to a unit step input.

Solution:

0	1	2
---	---	---

6. Using the above closed-loop time response equation to a step, the 2% settling time, T_s , can be expressed as a function of ω_0 , as shown below:

$$T_s = \frac{5.8}{\omega_0}$$

Assuming that the closed-loop system meets the design specifications, evaluate its 2% settling time.

Solution:

0	1	2
---	---	---

7. Assume that the motor represents the velocity loop in a manufacturing process and that under normal operating conditions the maximum nominal motor speed ω_{nom} at which the motor operates is defined as follows:

$$\omega_{nom} = 150.0 \left[\frac{rad}{s} \right] \quad [3.4]$$

In order to derive the shortest achievable settling time T_s , calculate the maximum acceleration, a_{max} , achievable by the motor with the attached inertial load. To do so, assume that the maximum allowable constant current, I_{max} , is 0.6 A. Justify this assumption.

Solution:

0	1	2
---	---	---

8. The motor should switch speed between $-\omega_{\text{nom}}$ and $+\omega_{\text{nom}}$ as fast as possible. A simple estimate of the minimum achievable settling time is the transition time needed to make that change assuming full acceleration. Calculate the shortest time, $T_{s,\text{min}}$, to make the speed transition under the assumption that the motor uses maximum achievable acceleration.

Solution:

0	1	2
---	---	---

9. Does the designed closed-loop system respect the process physical limitations?

Solution:

0	1	2
---	---	---

10. Evaluate the PI controller parameters, k_p and k_i , satisfying the design requirements.

Hint:

Use the nominal values of the process parameters K and τ , as evaluated in Section 2.5.3.6.

Solution:

0	1	2
---	---	---

11. In a first design we have used the set-point weight $b_{sp} = 0$. An alternative design with faster response will now be investigated. It will be shown that faster response can be obtained by using a larger value of b_{sp} .

To do this, a PI controller that gives the following transfer function from reference to output:

$$G_{\omega, r}(s) = \frac{\omega_0}{s + \omega_0} \quad [3.6]$$

will be designed.

Using both proportional and integral gains previously designed, determine and calculate b_{sp} such that the system closed-loop transfer function is of the form [3.6].

Solution:



0	1	2
---	---	---

12. Determine the system time response equation to a unit step input. For an asymptotically stable system, the 2 % settling time, T_s , is defined as the time required for the amplitude of oscillation to decay permanently to within a 2% margin around the steady-state value. Using the system closed-loop time response equation to a step and considering that $e^{-4} \approx 0.02$, express T_s as a function of ω_0 .

Solution:

0	1	2
---	---	---

13. Assuming that the closed-loop system meets the design specifications, evaluate its 2 % settling time. Verify that the controller design with $b_{sp} \neq 0$ provides a faster response. Does it satisfy the system physical limitations?

Solution:

0	1	2
---	---	---

3.5.2. Integrator Windup

3.5.2.1. Definition

So far we have mainly been concerned with linear behaviour, as is often the case with analysis and design of control systems. There is, however, one nonlinear phenomena that we have to deal with namely saturation of the amplifier. Since the supply voltage is 15 V, it is not possible for the motor amplifier to deliver outputs that are larger than 15 V. The effect of saturation can be quite serious because the feedback loop is effectively broken when the amplifier saturates. In this case, saturation causes the loss of the important feedback information from the system output. The combination of a controller with integral action and a process with saturation is that the integrator may drift when the controller saturates. This is called integrator wind up.

3.5.2.2. Windup Protection

There are many ways to avoid integrator windup. One possibility is to arrange a feedback that resets the integral when the output saturates. This is illustrated in the block diagram in Figure 3.2.

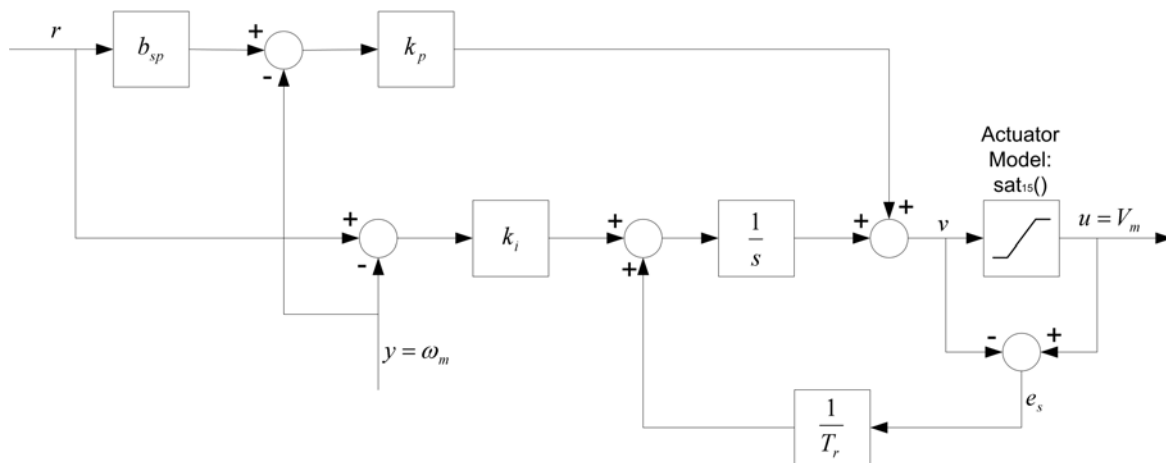


Figure 3.2 PI controller Block Diagram With Protection For Windup

For the DCMCT motor, the model of the actuator is simply as follows:

$$u(t) = sat_{15}(v(t))$$

with the saturation function defined as:

$$sat_{15}(x) = \begin{cases} -15 & x < -15 \\ x & -15 \leq x \text{ and } x \leq 15 \\ 15 & 15 < x \end{cases}$$

The extra feedback loop with the time constant T_r is inactive if the the control signal does not saturate because in this case we have $u = v$. When the controller output saturates, the extra feedback loop drives the saturation error e_s to zero which means that the integrator is driven towards a value just at the saturation limit. This means that the control signal will decrease from the saturation limit as soon as the control error goes negative.

The windup protection is governed by the parameter a_w which ranges between 0 and 1. There is no protection against windup if $a_w = 0$. If $a_w = 1$ the integrator is reset in one sampling period.

In discrete time, the PI control law with windup protection can be written as follows:

$$u(k) = \text{sat}_{15}(v(k))$$

with:

$$v(k) = k_p (b_{sp} r(k) - y(k-1)) + \left(\sum_{k=2}^N (k_i (r(k) - y(k-1)) h + a_w (u(k-1) - v(k-1))) \right)$$

where k is the sampling interval.

3.5.3. Tracking Triangular Signals

So far we have investigated tracking of reference signals in the form of square waves. We will now investigate tracking of triangular references.

Please answer the following questions.

1. Determine the transfer function, $G_{e,r}(s)$, from reference to control error for a PI control loop characterized by Equation [3.1]. $G_{e,r}(s)$ is defined below:

$$G_{e,r}(s) = \frac{E(s)}{R(s)} \quad [3.7]$$

where the velocity error is defined as follows:

$$E(s) = R(s) - \Omega_m(s) \quad [3.8]$$

Solution:

0	1	2
---	---	---

2. When a PI controller is used ($k_p \neq 0$ and $k_i \neq 0$), apply the final value theorem to calculate the steady-state error, e_{ss_PI} , in response to a ramp reference signal of slope R_0 .

Solution:

0	1	2
---	---	---

3. Using the PI tuning found in Sections 3.5.1 Question 10 and 3.5.1 Question 11, evaluate the steady-state error e_{ss_PI} for the following ramp slope R_0 :

$$R_0 = 160.0 \left[\frac{\text{rad}}{\text{s}^2} \right] \quad [3.9]$$

Solution:

0	1	2
---	---	---

Speed Control

4. For a system that does not have integral action ($k_i = 0$), apply the final value theorem to calculate the steady-state error, e_{ss_p} , in response to a ramp reference signal of slope R_0 . In this configuration, the closed-loop system has a pure proportional controller ($k_p \neq 0$, $k_i = 0$, and $b_{sp} = 1$).

Hint:

You can obtain e_{ss_p} by first applying the final value theorem for a step input of amplitude R_0 and then integrating.

Solution:

0	1	2
---	---	---

5. Using $k_p = 0.1$ V.s/rad, evaluate the steady-state error $e_{ss}(t)$ for the ramp slope R_0 , as defined in Equation [3.9]. Considering a 0.4 Hz triangular reference signal of slope R_0 , calculate the maximum steady-state error, $e_{ss_p_max}$.

Solution:

0	1	2
---	---	---

3.5.4. Response To Load Disturbances

Reduction of the effects of load disturbances is a key reason for using control. A torque on the motor axis is a typical example of a load disturbance for a speed control system. In this laboratory we will show the effects of controller tuning on load disturbance response.

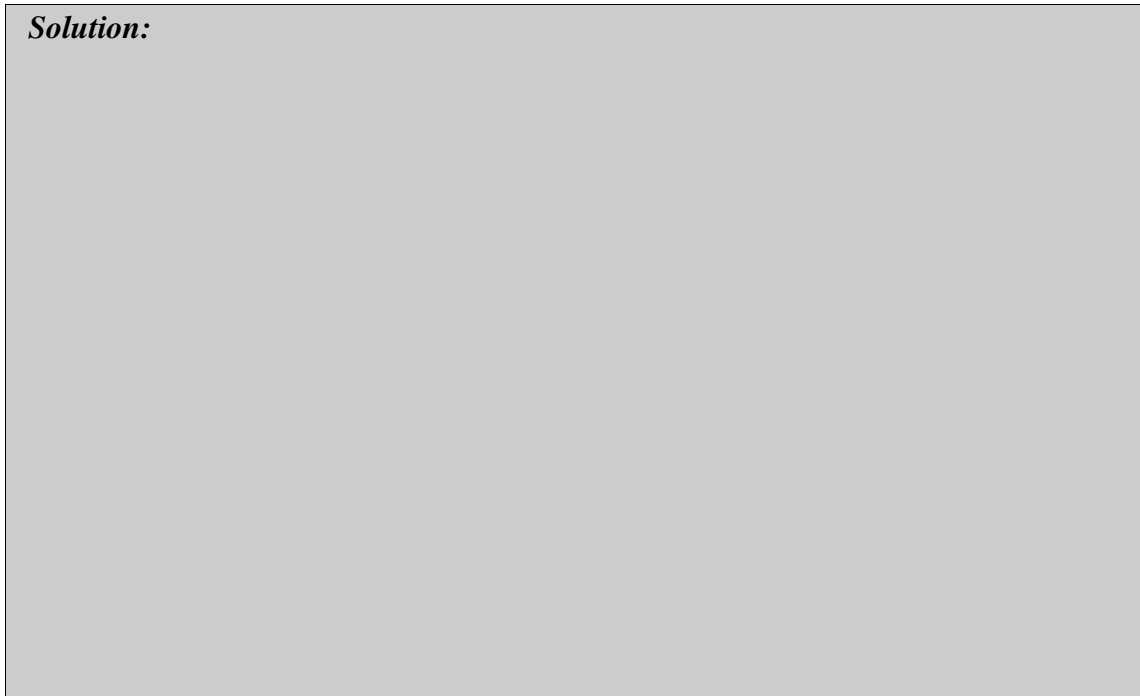
Please answer the following questions.

1. Considering the regulation problem (for $r = 0$), determine the closed-loop system block diagram with a disturbance torque input T_d applied on the DCMCT inertial load. Assume a PI controller. The block diagram should contain the open-loop transfer function $G_{\omega,V}$ as formulated in [3.2], and be function of the following system parameters: k_p , k_i , K , τ , and J_{eq} .

Hint:

Assume $V_{sd} = 0$.

Solution:



0	1	2
---	---	---

2. Find the closed-loop transfer function, $G_{\omega,T}(s)$, from disturbance torque to motor speed, as a function of the following system parameters: k_p , k_i , K , τ , and J_{eq} .

Speed Control

$G_{\omega, T}(s)$ is defined below:

$$G_{\omega, T}(s) = \frac{\Omega_m(s)}{T_d(s)}$$

Solution:

0	1	2
---	---	---

3. When a pure integral controller is used ($k_p = 0$ and $k_i \neq 0$), apply the final value theorem to calculate the steady-state velocity, $\omega_{ss,1}$, in response to a step input disturbance torque of amplitude T_{d0} . Comment.

Solution:

0	1	2
---	---	---

4. When a pure proportional controller is used ($k_p \neq 0$ and $k_i = 0$), apply the final value theorem to calculate the steady-state velocity, $\omega_{ss,p}$, in response to a step input disturbance torque of amplitude T_{d0} . Comment.

Solution:

0	1	2
---	---	---

3.5.5. Pre-Laboratory Results Summary Table

Table 3.2 should be completed before you come to the in-laboratory session to perform the experiment.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Value</i>	<i>Unit</i>
2.5.3. Nominal Process Parameters				
6.	Open-Loop Steady-State Gain	K		rad/(V.s)
6.	Open-Loop Time Constant	τ		s
3.5.1. PI Controller Design				
4.	Desired Damping Ratio	ζ		
6.	2 % Settling Time: $b_{sp} = 0$	T_s		s
7.	Motor Maximum Acceleration	a_{max}		rad/s ²
8.	Shortest Achievable Settling Time	T_{s_min}		s
10.	Proportional Gain	k_p		V.s/rad
10.	Integral Gain	k_i		V/rad
11.	Set-Point Weight For Faster Response	b_{sp}		
12.	2 % Settling Time: $b_{sp} \neq 0$	T_s		s
3.5.3. Tracking Triangular Signals				
3.	Steady-State Error: PI Control	e_{ss_PI}		rad/s
5.	Maximum Steady-State Error: P Control	$e_{ss_P_max}$		rad/s
3.5.4. Response To Load Disturbances				
3.	Steady-State Velocity: I Control	ω_{ss_I}		rad/s

Table 3.2 Speed Control Pre-Laboratory Assignment Results

3.6. In-Laboratory Session

3.6.1. QICii Speed Control Module

3.6.1.1. Module Description

The main tool for this laboratory is the front panel of the module entitled *Speed Control* in the QICii software, which should be similar to the one shown in Figure 3.3.

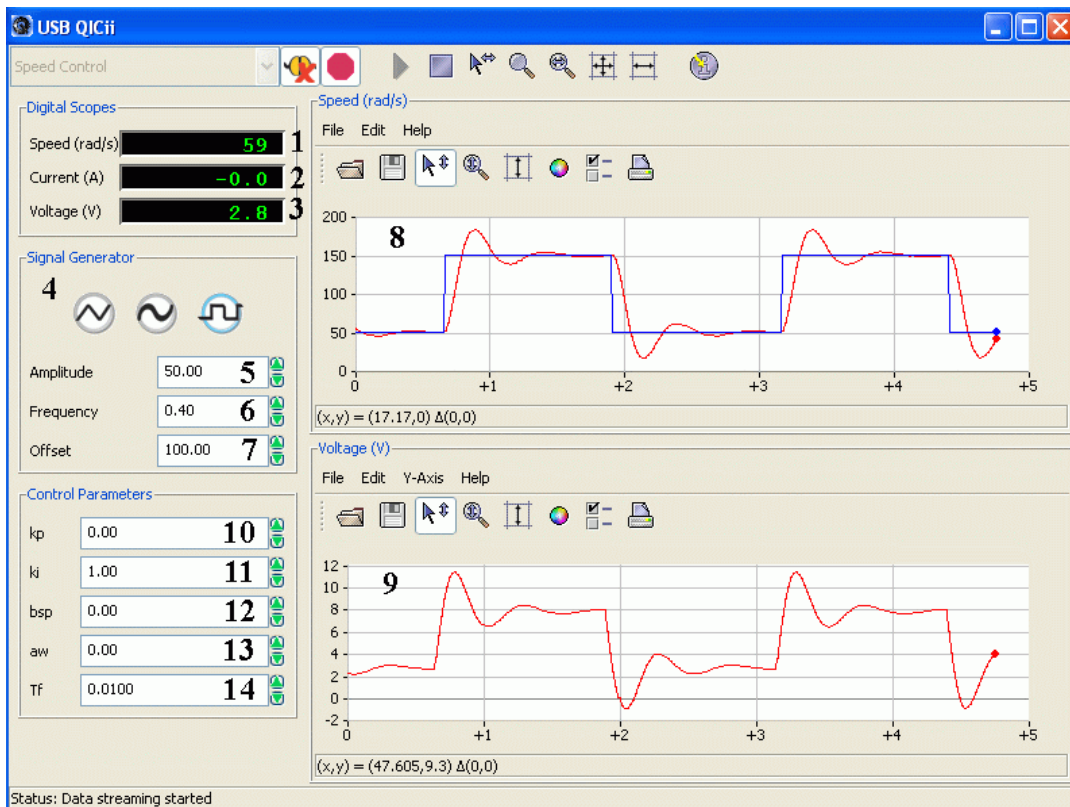


Figure 3.3 *Speed Control* Module Of The QICii Software

As a quick module description, Table 3.3 lists and describes the main elements composing the QICii *Speed Control* module user interface. Every element is uniquely identified through an ID number and located in Figure 3.3.

<i>ID #</i>	<i>Label</i>	<i>Parameter</i>	<i>Description</i>	<i>Unit</i>
1	Speed	ω_m	Motor Output Speed Numeric Display	rad/s
2	Current	I_m	Motor Armature Current Numeric Display	A
3	Voltage	V_m	Motor Input Voltage Numeric Display	V
4	Signal Generator		Type of Generator For The Speed Reference Signal	
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, r	Scope With Actual (in red) And Reference (in blue) Motor Speeds	rad/s
9	Voltage	V_m	Scope With Applied Motor Voltage (red)	V
10	kp	k_p	Controller Proportional Gain Input Box	V.s/rad
11	ki	k_i	Controller 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	Tf	T_f	Time Constant of Filter for Measured Signal	s

Table 3.3 QICii *Speed Control* Module Nomenclature

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

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

$$h = 0.01 [s]$$

The actual speed is obtained by filtering the position signal using the following filter:

$$\omega_m = \frac{s \theta_m}{T_f s + 1}$$

where θ_m is the position of the motor shaft measured by the encoder.

3.6.1.2. Module Startup

To start and use the *Speed Control* module, follow the steps described below:

- Step 1. Power up the DCMCT. LED2 should light up while LED3 should flash on and off repeatedly.
- Step 2. Press the *Reset* button on the QIC. Again, LED2 should remain on while LED3 should flash on and off repeatedly.
- Step 3. Press the DCMCT User Switch (i.e. pushbutton next to the two flashing LED's). LED2 and LED3 should both turn off.
- Step 4. Launch the USB QICii software and select *Speed Control* in the drop-down menu.
- Step 5. Select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller. LED2 should light up while LED3 should still be off, and the controller should start running.

Note:

The drop-down menu will be disabled (i.e. is unavailable) while the controller is running. You must select the *Stop controller* button on top of the QICii window to stop the controller (this should also turn off LED2, and enable the drop-down menu again). Once the drop down menu is enabled again and the controller is stopped, you can select any one of the other controller experiments, if you want. After selecting some other controller experiment, you can once again select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller of your choice.

The default module parameters loaded after download are given in Table 3.4.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	k_p <i>[V.s/rad]</i>	k_i <i>[V/rad]</i>	b_{sp}	a_w	T_f <i>[s]</i>
Square Wave	50.0	0.4	100.0	0.0	1.0	0	0	0.01

Table 3.4 Default Parameters For The *Speed Control* Module

3.6.2. Qualitative 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.

3.6.2.1. Pure 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, a reasonable amplitude is 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 sign of the velocity does not change. In this way you will avoid the effects of Coulomb friction. Set the integral gain to zero. Set the proportional gain to 0.02 V.s/rad to start with. Ensure that the following parameters of the *Speed Control* window, as displayed in Table 3.5, are set properly.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	k_i [V/rad]	k_p [V.s/rad]	b_{sp}
Square Wave	50	0.4	100	0	0.02	1

Table 3.5 Module Parameters For The Pure Proportional Control Test

Step 2. Change the proportional gain by incremental steps of 0.04 V.s/rad to investigate the closed-loop system for proportional controllers ($k_i = 0$) with different gains. What are your observations?

Solution:

0 1 2

Speed Control

Step 3. Determine the critical gain, k_{pc} , where the system becomes critically stable and a stable oscillation is achieved. Also determine the critical period T_{pc} of the corresponding oscillations.

Solution:

0	1	2
---	---	---

Step 4. Describe the steady-state error to a step input.

Solution:

0	1	2
---	---	---

Step 5. Repeat the previous observations. Change the *Amplitude* of the reference signal and observe under what conditions the control signal saturates.

Solution:

0	1	2
---	---	---

Speed Control

Step 6. Set the reference signal to a constant value so that the wheel spins with constant speed. Apply a torque on the inertial load by gently touching it with your finger. Observe the effect on the velocity.

Solution:

0	1	2
---	---	---

Step 7. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:

0	1	2
---	---	---

3.6.2.2. Pure Integral Control

In this procedure we will explore pure integral control. Please follow the steps below:

Step 1. Set the proportional gain to zero. Set the integral gain to 0.02 V/rad to start with. Ensure that the parameters of the QICii module window, as listed in Table 3.6, are set properly.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	<i>k_p [V.s/rad]</i>	<i>k_i [V/rad]</i>	<i>a_w</i>
Square Wave	50	0.4	100	0	0.02	0

Table 3.6 Module Parameters For The Pure Integral Control Test

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

Solution:

0 1 2

Step 3. Determine the critical gain, k_{ic} , where the system becomes critically stable and a stable oscillation is sustained. Also determine the critical period T_{ic} of the corresponding oscillations.

Solution:

0 1 2

Speed Control

Step 4. Determine a value of integral gain which gives a set-point response without overshoot. Determine the settling time for the closed loop system.

Solution:

0	1	2
---	---	---

Step 5. Repeat the previous observations. Change the *Amplitude* of the reference signal and observe under what conditions the control signal saturates.

Solution:

0	1	2
---	---	---

Step 6. Set the reference signal to a constant value so that the wheel spins with constant speed. To do this, set the *Amplitude* to zero and use the *Offset* to adjust the constant value. Apply a torque on the inertial load by gently touching it with your finger. Observe the effect on the velocity.

Solution:

0	1	2
---	---	---

Step 7. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:



0	1	2
---	---	---

3.6.2.3. 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 of the QICii module window as listed in Table 3.7.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	b_{sp}	a_w
Square Wave	50	0.4	100	1	0

Table 3.7 Module Parameters For The Proportional And Integral Control Test

Step 2. Set proportional gain to a constant value, like $k_p = 0.1$ V.s/rad, and change integral gain. Observe the tracking error and the control signal.

Solution:

0	1	2
---	---	---

Step 3. Set integral gain to a constant value, like $k_i = 0.5$ V.s/rad, and change proportional gain. Observe the tracking error and the control signal.

Solution:

0	1	2
---	---	---

Step 4. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:



0	1	2
---	---	---

3.6.3. Manual Tuning: Ziegler-Nichols

By performing the experimental procedures in Section 3.6.2, you should have obtained a good feel for the properties of proportional and integral control. This knowledge will now be used to develop manual tuning procedures. Manual tuning procedures are generally used when no mathematical model of the system is available to perform control system design. Typically in manual control, 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.

3.6.3.1. Preamble: Ziegler-Nichols Method

A good way to find ball park values of controller parameters is to look at pure proportional and integral controllers and to determine the gains where the system become critically stable (a.k.a. astable). The Ziegler-Nichols frequency response method is a classical tuning rule based on this idea. A proportional controller is adjusted so that the system reaches the stability limit. The critical gain k_{pc} where this occurs is observed together with the period of oscillation T_{pc} .

In the early forties J.G. Ziegler and N.B. Nichols experimentally developed PID tuning methods 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 most industrial processes.

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

$$\begin{aligned}
 k_p &= 0.4 k_{pc} \\
 T_i &= 0.8 T_{pc} \\
 k_i &= \frac{k_p}{T_i} \qquad \text{or} \qquad k_i = \frac{0.5 k_{pc}}{T_{pc}}
 \end{aligned}$$

3.6.3.2. Experimental Procedure: Ziegler-Nichols Tuning

Please follow the steps below:

Speed Control

Step 1. Using the Ziegler-Nichols method, determine the PI controller parameters.

Solution:

0 1 2

Step 2. Set the parameters of the QICii module window as listed in Table 3.8.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	<i>b_{sp}</i>	<i>a_w</i>
Square Wave	50	0.4	100	1	0

Table 3.8 Module Parameters For The Ziegler-Nichols-Tuned PI Control

Set both proportional and integral gains to their Ziegler-Nichols values as previously calculated. What is the resulting 2 % settling time? What are your observations?

Solution:

0 1 2

Step 3. Adjust proportional and integral gain manually to give a very slightly under-damped set-point response with no saturation.

Solution:

0 1 2

Step 4. Summarize your observations and your calculations in your report. Select some representative results, screen captures, and plots.

Solution:



0	1	2
---	---	---

Solution (continued):



0	1	2
---	---	---

3.6.4. Set-Point Weighting

3.6.4.1. Preamble

So far we have only investigated the effects of proportional gain k_p and integral gain k_i . The PI controller does however have more parameters. In this Section we will explore the effects of the set-point weight for proportional gain b_{sp} . This parameter which ranges from 0 to 1, was set to $b_{sp} = 1$ in the earlier experimental procedures. You will observe that the set-point weight has a significant effect on the response to command signals. However as expressed by Equation [3.1], it only affects the reference signal input but does not influence the response to load disturbances.

3.6.4.2. Experimental Procedure

Please follow the steps below:

Step 1. Use the controller parameters found by the Ziegler-Nichols method in Section 3.6.3.2 from the manual tuning. Choose a *Square Wave* signal from the QICii signal generator. Investigate the effects of the set-point weight on the step response and the control signal.

Solution:

0	1	2
---	---	---

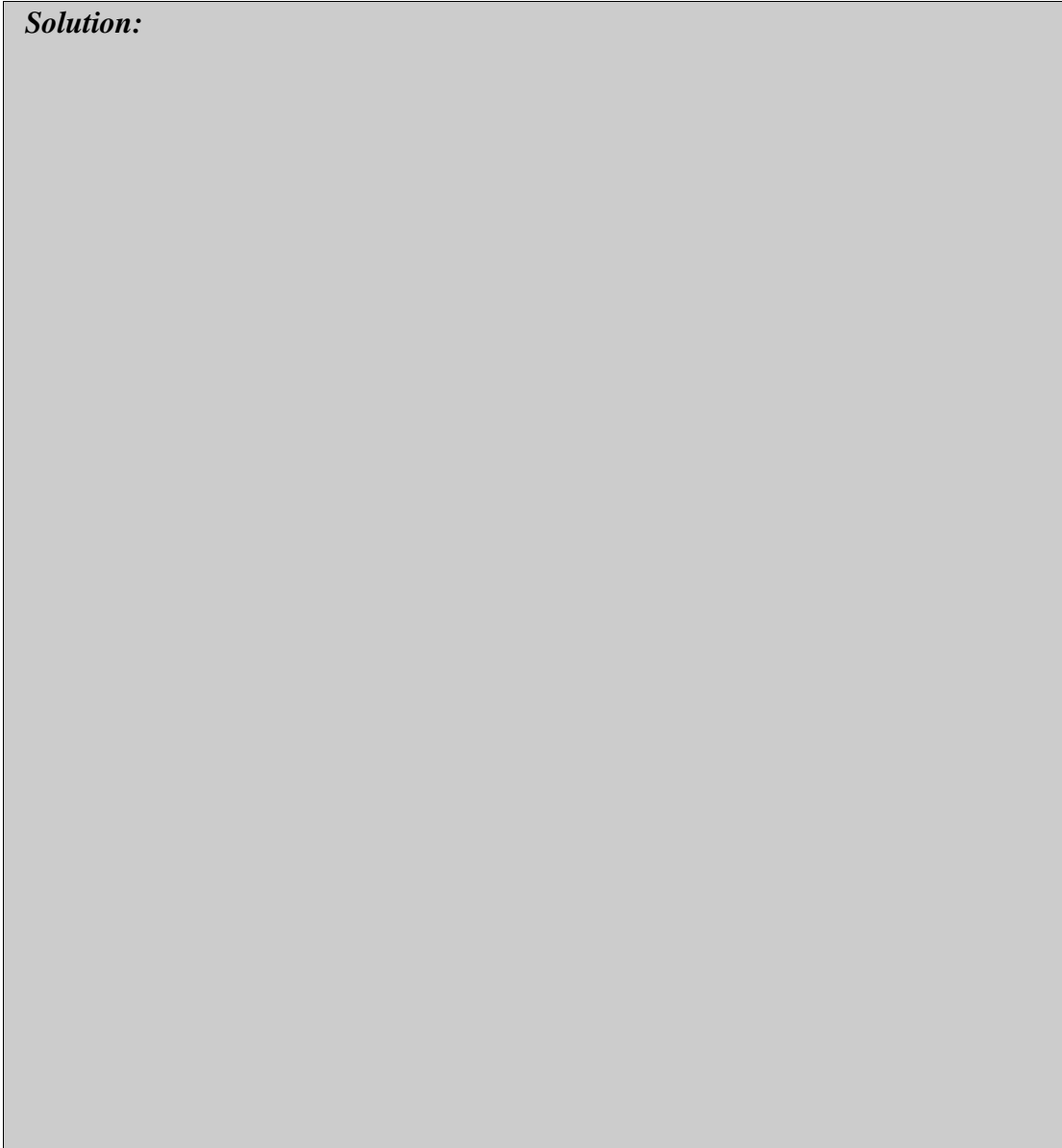
Step 2. Make some analysis that explains your observations.

Solution:

0	1	2
---	---	---

Step 3. Document in your report your results and your analysis and illustrate with typical responses.

Solution:



0	1	2
---	---	---

3.6.5. Design To Given Specifications

The controllers parameters can be determined by manual tuning or by design based on a process model. The latter is to use a mathematical model of the process and apply some method for control system design. In this laboratory, we are using a basic pole assignment method. This is simple to do in this particular case because the nominal process model is of first order.

This section provides the experimental verification of the two PI controller designs to given specifications, as carried out in the pre-lab assignments 3.5.1.

3.6.5.1. PI Control With No Set-Point Weighting

Please follow the steps below:

Step 1. Set the parameters of the QICii module window as described in Table 3.9.

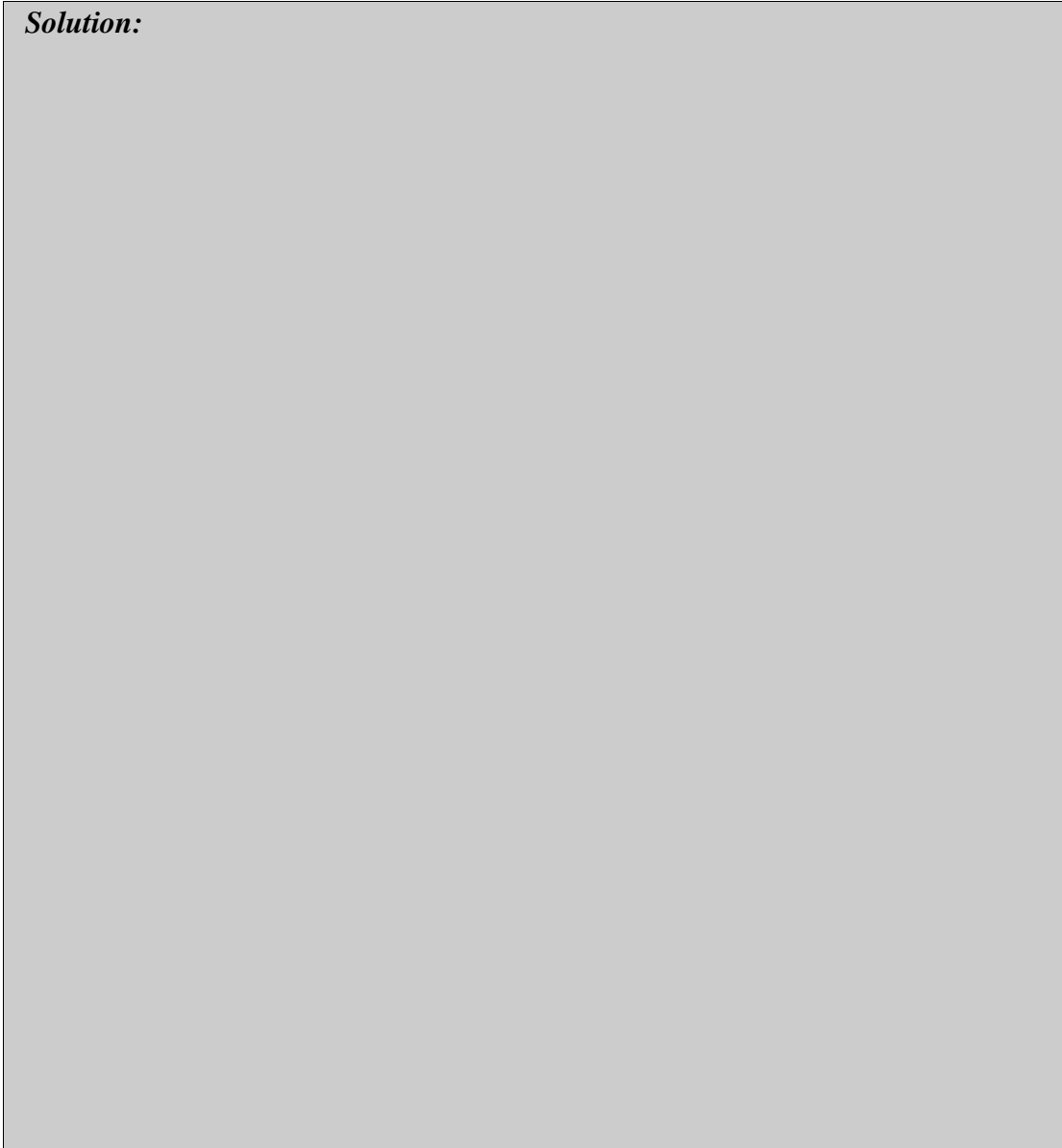
<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	b_{sp}	a_w
Square Wave	50	0.4	100	0	0

Table 3.9 Module Parameters For PI Control Design To Given Specifications

Step 2. Set both proportional and integral gains to the values you calculated in Section 3.5.1 Question 10. For the experimental verification of the first controller design, set $b_{sp} = 0$.

Step 3. Measure the obtained settling time. Does the system response meet the desired requirements? Summarize your observations and your calculations in your report. Select some representative results, screen captures, and plots.

Solution:



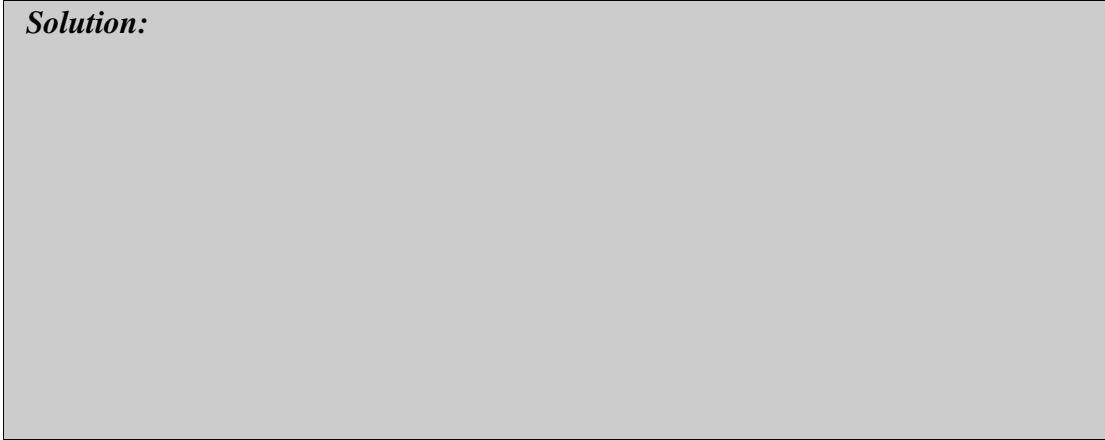
0	1	2
---	---	---

3.6.5.2. PI Control With Set-Point Weighting

Step 1. Verify now your second and alternative design obtained through pole-zero cancellation. Enter the corresponding value of b_{sp} , as you calculated in Section 3.5.1 Question 11.

Step 2. Compare the settling time with the one from the previous response with $b_{sp} = 0$. Is this closed-loop response faster? Are the design specifications met?

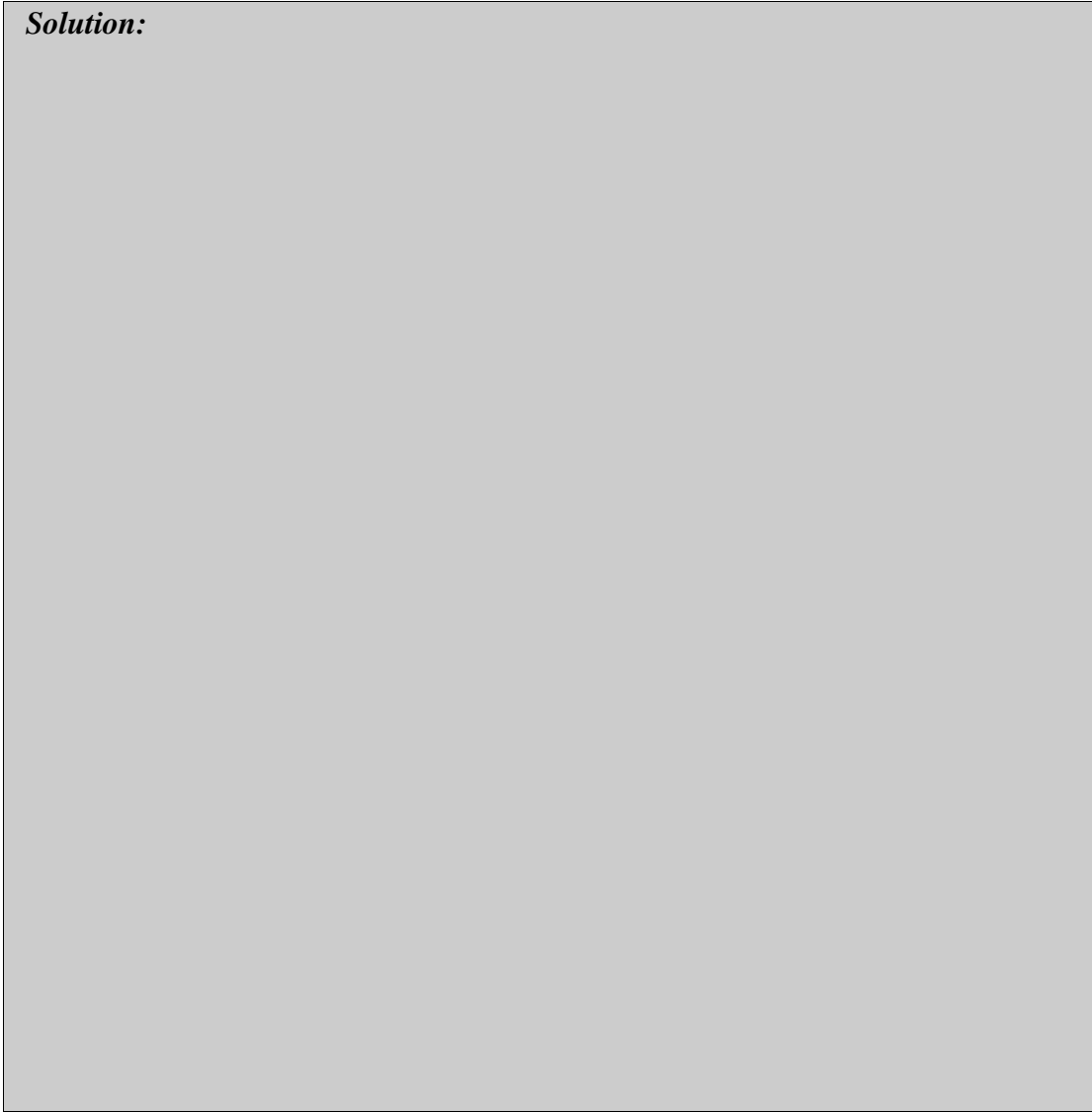
Solution:



0	1	2
---	---	---

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

Solution:



0	1	2
---	---	---

3.6.6. Integrator Windup

Please refer to pre-laboratory assignment 3.5.2 for the theoretical background and results required to successfully carry out this in-lab session.

3.6.6.1. Preamble

The behaviour of a control system can often be captured by linear models. Saturation is however an essential nonlinearity that is present in practically all systems. Saturation can cause windup in all controllers that have integral action.

Integrator windup is demonstrated in the following procedure. Please follow the steps below:

Step 1. Set the parameters of the QICii module window as described in Table 3.10.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	<i>a_w</i>
Square Wave	140	0.4	140	0

Table 3.10 QICii Module Parameters For Integrator Windup Test

Set the controller gains (k_p , k_i , and b_{sp}) to the values tuned to meet given design specifications, as calculated in Section 3.5.1 Questions 10 and 11.

Step 2. What are your observations? Compare your response to the one previously obtained with the same controller but for smaller reference steps.

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

Solution:



0	1	2
---	---	---

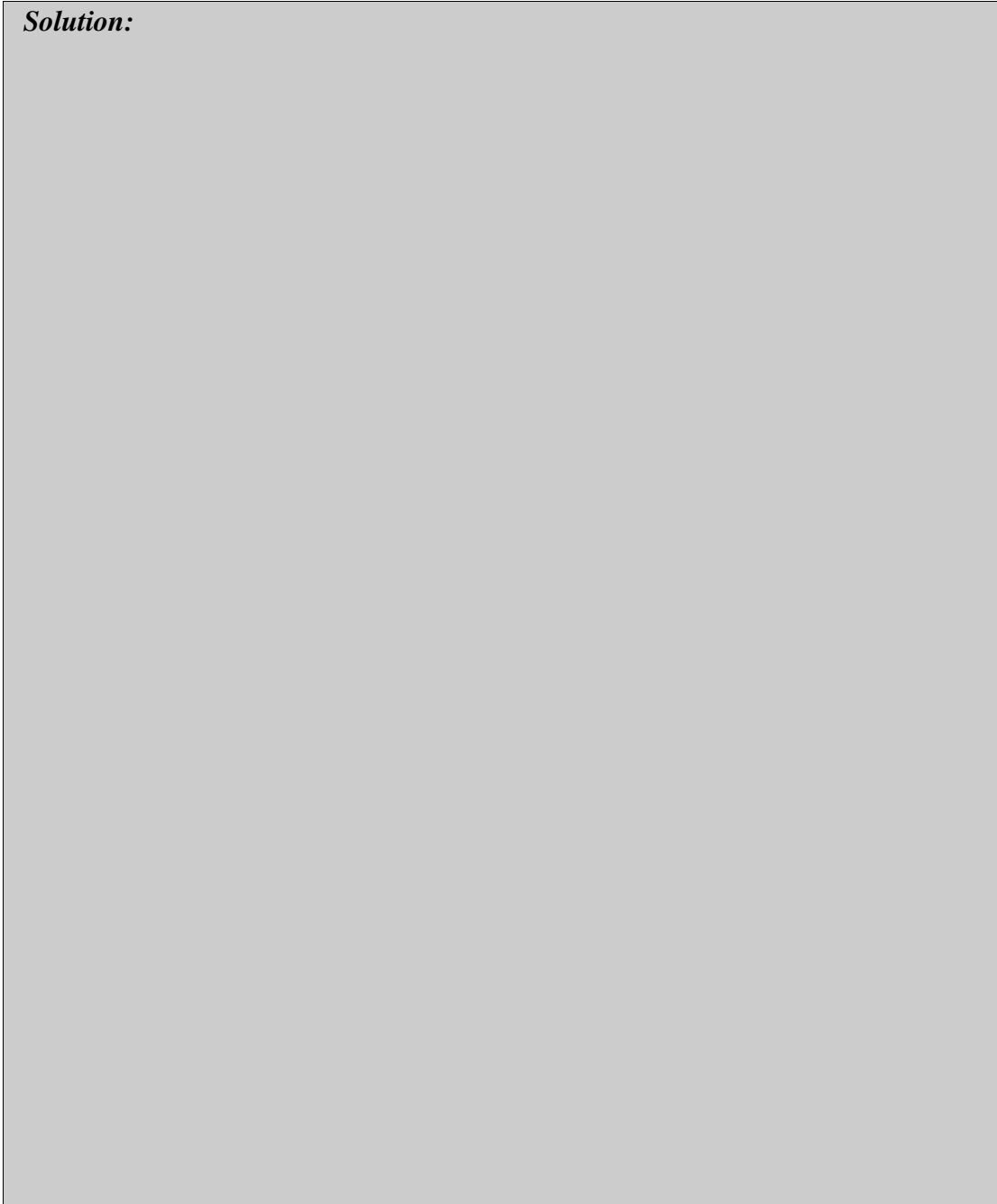
3.6.6.2. Windup Protection

Such windup can be avoided by adding a nonlinearity in the controller. The PI controller implemented in DCMCT uses a scheme for protection against windup that is governed by the parameter a_w . The windup protection design is detailed in Section 3.5.2.2. The parameter a_w , which ranges between 0 to 1, is available on the control panel. So far we have used the default value $a_w = 0$ which means that there is no protection against integrator windup.

Integrator windup protection is demonstrated in the following procedure. Please follow the steps below:

- Step 1. Keep the same reference signal and controller gains as the ones used in the previous sub-Section, where they result in integrator windup.
- Step 2. Increase the values of the parameter a_w from 0 to 1. Show the improvement in the response when windup protection is used. Investigate the effects of different values of the parameter a_w . Are there any drawbacks in using $a_w = 1$?
- Step 3. Summarize your observations and your calculations in your report. Select some representative results, screen captures, and plots.

Solution:



0	1	2
---	---	---

3.6.7. Tracking Triangular Signals

Please refer to pre-laboratory assignment 3.5.3 for the theoretical background and results required to successfully carry out this in-lab session. This session carries out an experimental investigation of tracking of triangular signals. Please follow the steps below:

Step 1. Set the parameters of the QICii module window as listed in Table 3.11. Note that a triangular (a.k.a. sawtooth) reference signal should be chosen.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	<i>a_w</i>
Triangular Wave	100	0.4	150	0

Table 3.11 QICii Module Parameters For Tracking Test

Also set the controller gains (k_p , k_i , and b_{sp}) to the values tuned to meet given design specifications, as calculated in Section 3.5.1 Questions 10 and 11.

Step 2. The motor turns to track the triangular wave command. What are your observations?

Solution:

0 1 2

Step 3. From the triangular wave specifications given in Table 3.11, calculate the slope R_0 of the obtained ramp signal.

Solution:

0 1 2

Step 4. Measure the actual asymptotic tracking error and compare with analytic estimates. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:

0	1	2
---	---	---

Step 5. Use a pure proportional controller with gain $k_p = 0.10 \text{ V.s/rad}$ and $b_{sp} = 1.0$ and determine the maximum tracking error. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:

0	1	2
---	---	---

3.6.8. Response To Load Disturbances

This session carries out an experimental investigation of the system response to load disturbances. Please refer to pre-laboratory assignment 3.5.4 as required for theoretical results.

3.6.8.1. Manual Load Disturbances

A load disturbance can be introduced by manually applying a torque to the inertial load. Please follow the steps below:

Step 1. The regulation problem, at $r = 100$ rad/s, is investigated. Set the QICii module parameters as described in Table 3.12.

<i>Amplitude [rad/s]</i>	<i>Offset [rad/s]</i>	<i>a_w</i>
0	100	0

Table 3.12 Module Parameters For The Manual Load Disturbance Test

Step 2. Choose a pure proportional controller ($k_i = 0$) with gain $k_p = 0.10$ V.s/rad and $b_{sp} = 1.0$.

Step 3. Apply a torque manually by gently touching the inertial load with your finger. Observe what happens when you change the gain of the controller.

Solution:

0	1	2
---	---	---

Step 4. 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.

Solution:

0	1	2
---	---	---

3.6.8.2. Simulated Load Disturbances: Disturbance Response With PI Control

A disturbance torque can be simulated by applying an additional voltage, V_{sd} , to the motor input. The DCMCT system uses this as a way to simulate a load disturbance in a repeatable manner, as illustrated in Figure 3.4. You can do this by depressing the User Switch on the DCMCT.

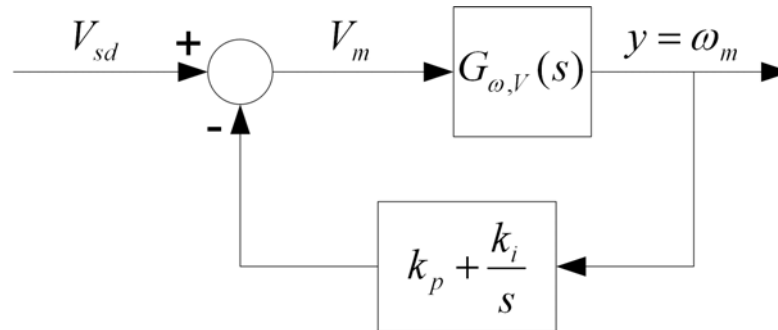


Figure 3.4 DCMCT Regulation Block Diagram With Simulated Load Disturbance

Please follow the steps below:

Step 1. The regulation problem at no velocity ($r = 0$ rad/s) is investigated. Set the QICii module parameters as described in Table 3.13.

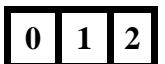
<i>Amplitude [rad/s]</i>	<i>Offset [rad/s]</i>	<i>a_w</i>
0	0	0

Table 3.13 Module Parameters For The Simulated Load Disturbance Test

Step 2. Emulate a load disturbance torque by pressing and holding or releasing the User Switch.

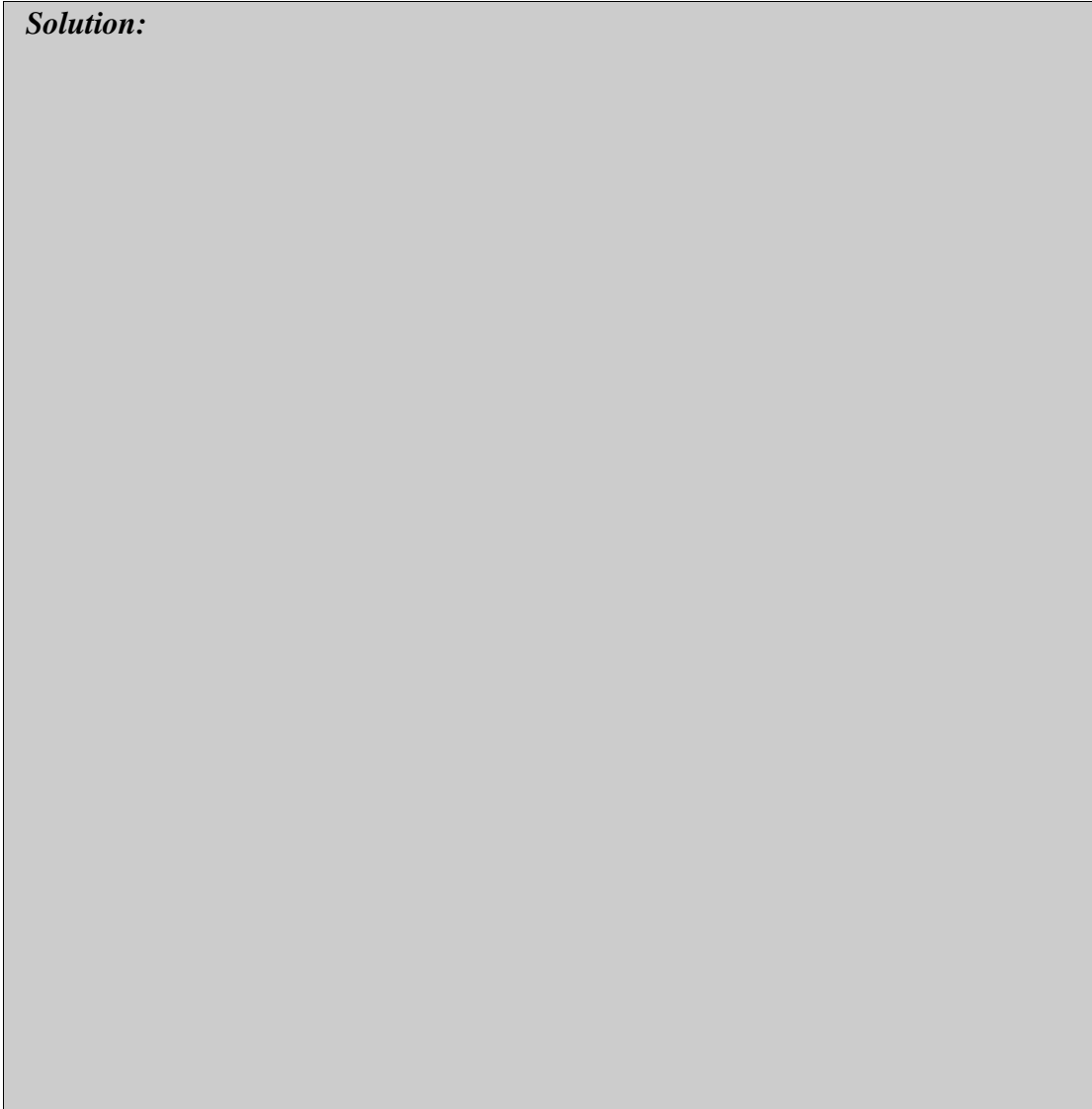
Step 3. Investigate the behaviour of the closed-loop system for pure proportional controllers ($k_i = 0$ and $b_{sp} = 1$) with different gains. Summarize your observations in your report. Determine the steady-state error ω_{ss_p} . Select some representative results, screen captures, and plots.

Solution:



Step 4. Investigate the behaviour of the closed-loop system for pure integrating controllers ($k_p = 0$ and $a_w = 0$) with different gains. Summarize your observations in your report. Select some representative results, screen captures, and plots.

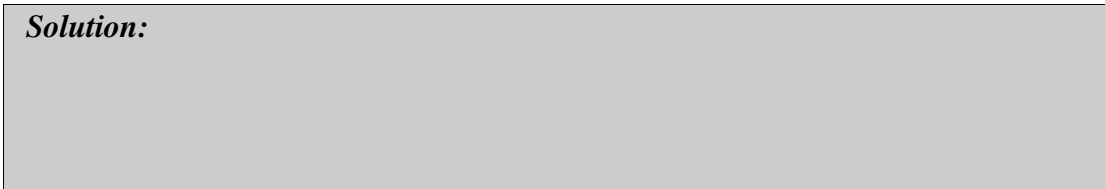
Solution:



0	1	2
---	---	---

Step 5. Can you estimate how much voltage disturbance, V_{sd} , is applied when you press the DCMCT User Switch?

Solution:



0	1	2
---	---	---

Step 6. Determine the parameters of a PI controller which gives a fast recovery of a load disturbance. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:

0	1	2
---	---	---

3.6.9. In-Laboratory Results Summary Table

Table 3.14 should be completed using Table 3.2, which contains data from the pre-laboratory assignments, as well as experimental results obtained during the in-laboratory session.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Pre-Lab Value</i>	<i>In-Lab Result</i>	<i>Unit</i>
3.6.2. Properties Of P And I Control					
1.	Critical Proportional Gain	k_{pc}	N/A		V.s/rad
1.	Critical Period For k_{pc}	T_{pc}	N/A		s
3.6.3. Ziegler-Nichols Design					
2.	Proportional Gain	k_p	N/A		V.s/rad
2.	Integral Gain	k_i	N/A		V/rad
2.	2 % Settling Time	T_s	N/A		s
3.6.5. Design To Given Specifications					
1.	2 % Settling Time: $b_{sp} = 0$	T_s			s
2.	2 % Settling Time: $b_{sp} \neq 0$	T_s			s
3.6.7. Tracking Triangular Signals					
	Ramp Slope	R_0			rad/s ²
	Steady-State Error: PI Control	e_{ss_PI}			rad/s
	Maximum Steady-State Error: P Control	$e_{ss_P_max}$			rad/s
3.6.8. Response To Load Disturbances					
2.	Steady-State Velocity: P Control	ω_{ss_P}	N/A		rad/s
2.	Steady-State Velocity: I Control	ω_{ss_I}			rad/s
2.	Simulated Disturbance Voltage	V_{sd}	N/A		V

Table 3.14 Speed Control In-Laboratory Results

4. Robustness

4.1. Laboratory Objectives

The objectives of this laboratory are to explore how sensitive the closed-loop system is to the parameters of the model and to the unmodeled dynamics and how the sensitivity depends on the specifications of the system. The important concepts of sensitivity, complementary sensitivity, and stability margins are also illustrated.

Particularly, the robustness of the PI controller design carried out in Chapter 3 for the motor speed is assessed and compared with a more robust design obtained here. The sensitivity and robustness of the closed-loop system are determined, as well as how they are affected by the design parameters ω_0 and ζ .

4.2. Preparation And Pre-Requisites

A pre-requisite to this laboratory is to have successfully completed the modelling and speed control laboratories, as described in Chapters 2 and 3, respectively. Before the lab, you should also review the concepts of robustness and sensitivity. The pre-laboratory assignments described hereafter require extensive use of MATLAB from The MathWorks, or equivalent.

4.3. Introduction

4.3.1. Control Systems Design

Many issues have to be considered when designing a control system, for example:

- Response to measurement noise
- Response to reference signals
- Robustness to process variations

Design typically involves trade-offs between all these factors.

4.3.2. The Gang Of Six

A standard closed-loop system is represented in Figure 4.1. It is influenced by three external signals: the reference $R(s)$, the load disturbance $D(s)$, and the measurement noise $N(s)$. Notice that in the case of pure error feedback, we have $F(s) = C(s)$. The process has the transfer function $P(s)$, and the controller is described by:

$$U(s) = F(s) R(s) - C(s) Y(s) \quad [4.1]$$

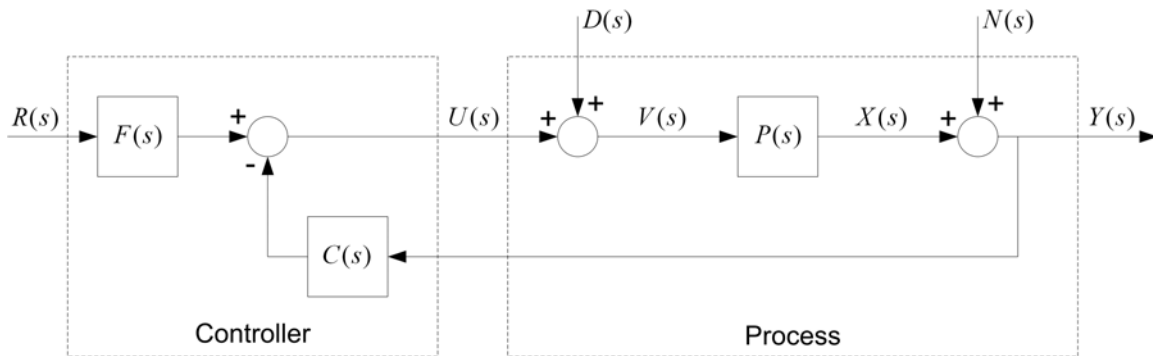


Figure 4.1 Block Diagram Of Standard Closed-Loop System

There are at least three signals $X(s)$, $Y(s)$, and $U(s)$ that are of great interest for control. The following relationships are obtained from the block diagram in Figure 4.1:

$$X = \frac{PFR}{1+PC} + \frac{PD}{1+PC} - \frac{PCN}{1+PC} \quad [4.2]$$

and:

$$Y = \frac{PFR}{1+PC} + \frac{PD}{1+PC} + \frac{N}{1+PC} \quad [4.3]$$

and:

$$U = \frac{FR}{1+PC} - \frac{PDC}{1+PC} - \frac{CN}{1+PC} \quad [4.4]$$

To simplify notations, the arguments of all Laplace transforms have been dropped. We can observe from the equations above that all nine relations between the input and the output signals are given by the following set of six transfer functions, which we call the "Gang Of Six":

$$\frac{PF}{1+PC} \qquad \qquad \frac{F}{1+PC} \quad [4.5]$$

and:

$$\frac{PC}{1+PC} \qquad \qquad \frac{C}{1+PC} \quad [4.6]$$

and:

$$\frac{P}{1+PC} \qquad \qquad \frac{1}{1+PC} \quad [4.7]$$

The two transfer functions in row [4.5] give the response of process variable and control signal to the set-point. The second row [4.6] gives the same signals in the case of pure error feedback, when $F = C$. The transfer functions $1/(1+PC)$ and $PC/(1+PC)$ also express the robustness properties of the system as will be discussed in the following. Notice that only four transfer functions are required to describe how the system reacts to load disturbance and measurement noise. Two additional transfer functions are required to describe how the system respond to set-point changes.

In the case of a system with (pure) error feedback (i.e. $F = C$), the linear properties of the closed-loop system can be completely captured by four transfer functions, called the "Gang of Four". They are expressed in rows [4.6] and [4.7]. Systems with two degrees of freedom are characterized by six transfer functions, the "Gang of Six".

To make an assessment of a control system, it is important to investigate all of the transfer functions expressed in [4.5], [4.6], and [4.7]. In a pole placement design, we have focused on two of the transfer functions. To have complete insight into the properties of the system, it is necessary to also investigate the remaining transfer functions.

4.4. Nomenclature

The following nomenclature, as described in Table 4.1, is used for the system modelling and control design.

<i>Symbol</i>	<i>Description</i>	<i>Unit</i>
ω_m	Motor speed which can be computed from the motor angle	rad/s
V_m	Voltage from the amplifier which drives the motor	V
V_{sd}	Simulated External Disturbance Voltage	V
K	Open-Loop Steady-State Gain	rad/(V.s)
τ	Open-Loop Time Constant	s
ω_0	Closed-Loop Undamped Natural Frequency	rad/s
ζ	Closed-Loop Damping Ratio	
k_p	Proportional Gain	V.s/rad
k_i	Integral Gain	V/rad
b_{sp}	Set-Point Weight On Proportional Part	
u	Control Signal	V
r	Reference Signal	rad
y	Measured Process Output	rad
ω_f	Cut-Off Frequency of Filter For Measured Signal	rad/s
M_s	Maximum Sensitivity	
M_t	Maximum Complementary Sensitivity	
g_m	Gain Margin	
ω_{gm}	Gain-Margin Frequency	rad/s
ϕ_m	Phase Margin	
ω_{gc}	Gain-Crossover Frequency	rad/s
s	Laplace Operator	rad/s
h	Sampling Interval	s

Table 4.1 System Modelling Nomenclature

4.5. Pre-Laboratory Assignments

4.5.1. MATLAB Example Script

In this laboratory, you will use MATLAB extensively to evaluate transfer functions in the Laplace domain. The following uses MATLAB to evaluate and plot an example Laplace transform over a given frequency range.

As an example, suppose you have a process $P(s)$ defined as follows:

$$P(s) = \frac{100}{s(s + 0.1)}$$

and a PD controller characterized by:

$$C(s) = (2.0 + 0.15 s) e^{(-s h)}$$

Assume $F(s) = C(s)$.

You can then evaluate the closed-loop transfer function $G(s)$ over a given frequency range using the MATLAB script illustrated below:

```
% define frequency range of interest: w between 1 and 100 [rad/s]
w = logspace( 0, 2, 1000 );
% sampling interval [s]
h = 0.01;
% define the Laplace operator
s = i * w;
% process transfer function
P = 100 ./ ( s + 0.1 ) ./ s;
% controller transfer function
C = ( 2.0 + 0.15 * s ) .* exp( - s * h );
% pure error feedback
F = C;
% closed-loop transfer function
G = F .* P ./ ( 1 + P .* C );    % note how easy this is!
% plot magnitude of G
figure(1)
loglog( w, abs(G) )
% plot phase angle of G [deg]
figure(2)
semilogx( w, 180*angle(G)/pi )
% find maximum magnitude of G and associated frequency
[xx, ii] = max( abs(G) );
fprintf('Maximum of G is %5.2f at w = %5.2f rad/s', xx, w(ii) )
```

Ensure to understand the above example well, as you will be using such a technique throughout this laboratory.

4.5.2. Robustness And Sensitivity

One of the amazing properties of feedback is that it is actually possible to design reliable control systems based on simple approximate models. This often leads to controversies between control engineers and specialists in modelling. There are two reasons why the simple models work, one is the amazing property of integral action, which implies that it is possible to maintain the correct steady-state. The other is that feedback can make the the closed-loop system quite insensitive to variations in the model. To see this we will consider the response to command signals. Let the process output be described by:

$$Y(s) = P(s) U(s) \quad [4.8]$$

where the control signal $U(s)$ is characterized by Equation [4.1].

Equations [4.1] and [4.8] give the following transfer function from reference $R(s)$ to output $Y(s)$:

$$G_{ry} = \frac{F(s) P(s)}{1 + P(s) C(s)} \quad [4.9]$$

4.5.2.1. Small Process Variations – The Sensitivity Function

To investigate how small variations in the process $P(s)$ influence the closed-loop transfer function G_{ry} , we consider the sensitivity function $S(s)$ defined as follows:

$$\frac{\Delta G_{ry}(s)}{G_{ry}(s)} = \frac{\Delta P(s)}{P(s)} S(s) \quad [4.10]$$

At the limit, Equation [4.10] becomes:

$$S(s) = \frac{dG_{ry} P(s)}{dP G_{ry}(s)} \quad [4.11]$$

Differentiating G_{ry} with respect to P results in:

$$\frac{dG_{ry}}{dP} = \frac{F}{(1 + P C)^2}$$

Using the above relation and Equation [4.9] into Equation [4.11] leads to the following expression for the sensitivity function:

$$S = \frac{1}{1 + P C} \quad [4.12]$$

Notice that the result [4.12] does not depend on F . Since the loop transfer function PC typically goes to zero for large s , it follows that the sensitivity function S goes to 1 as s goes to infinity. For controllers with integral action, the loop transfer function PC goes to infinity as s goes to zero, which means that the sensitivity function S goes to zero. Integral action thus ensures that the sensitivity function is small for low frequencies.

The largest value of the sensitivity function, M_s , is a measure of the sensitivity. M_s can be expressed as follows:

$$M_s = \max_{\omega} |S(i\omega)| \quad [4.13]$$

where \max is the maximum function. A typical requirement for a good control system is that the maximum sensitivity M_s is between 1 and 2. A reasonable standard value is:

$$M_s = 1.6 \quad [4.14]$$

Please answer the following questions.

1. Referring to Chapter 3, the transfer function from motor voltage to output velocity is expressed as follows:

$$P(s) = \frac{K}{\tau s + 1} \quad [4.15]$$

Also referring to Chapter 3, the PI controller of motor speed is implemented as follows:

$$U(s) = \left(k_p b_{sp} + \frac{k_i}{s} \right) R(s) + \left(-k_p - \frac{k_i}{s} \right) Y(s) \quad [4.16]$$

Identifying Equation [4.16] with Equation [4.1] results in the following transfer functions $F(s)$ and $C(s)$:

$$F(s) = k_p b_{sp} + \frac{k_i}{s} \quad C(s) = k_p + \frac{k_i}{s} \quad [4.17]$$

Notice that with the set-point weight $b_{sp} = 1$, we have $F = C$, which is characteristic of an error-feedback system.

Consider PI control of motor speed, as developed in Section 3.5.1 of Chapter 3, where the controller is designed to give a closed-loop system characterized by ζ and ω_0 . Derive the sensitivity function $S(s)$ corresponding to the PI control of the DCMCT motor speed. Express $S(s)$ as a function of τ , ω_0 , and ζ , **only**. Does $S(s)$ depend on b_{sp} ?

Hint:

According to Equation [4.9], $1+PC$ is the expected closed-loop characteristic equation.

Solution:

0 1 2

- Referring to the Modelling laboratory detailed in Chapter 2, report your results by filling Table 4.2. Consider the nominal parameter values from the system specifications. Note that $1/\tau$ is the motor open-loop bandwidth.

<i>Description</i>	<i>Symbol</i>	<i>Value</i>	<i>Unit</i>
Open-Loop Gain	K		rad/(V.s)
Open-Loop Time Constant	τ		s
Open-Loop Bandwidth	$1/\tau$		Hz

0 1 2

Table 4.2 DCMCT System Nominal Open-Loop Parameters

3. Referring to the Speed laboratory detailed in Chapter 3, re-compute the PI controller gains, k_p and k_i , resulting from the following pole placement design: $\zeta = 0.8$ and $\omega_0 = 20.0$ rad/s. Report the calculated gains by filling Table 4.3. The controller design corresponding to the parameters listed in Table 4.3 are referred as the **Speed Lab Design (SLD)**.

ω_0 [rad/s]	ζ	k_p [V.s/rad]	k_i [V/rad]
20.0	0.8		

0	1	2
---	---	---

Table 4.3 Speed Laboratory Design (SLD) Parameters

4. The controller implemented in the PIC microcontroller filters the speed measurement using the following transfer function:

$$G_f(s) = \frac{\omega_f}{s + \omega_f} \tag{4.18}$$

Furthermore, the digital implementation sampled at 100 Hz causes one sample delay of $h = 0.01$ s. The actual motor open-loop transfer function is then:

$$P(s) = \frac{K \omega_f e^{(-s h)}}{(\tau s + 1)(s + \omega_f)} \tag{4.19}$$

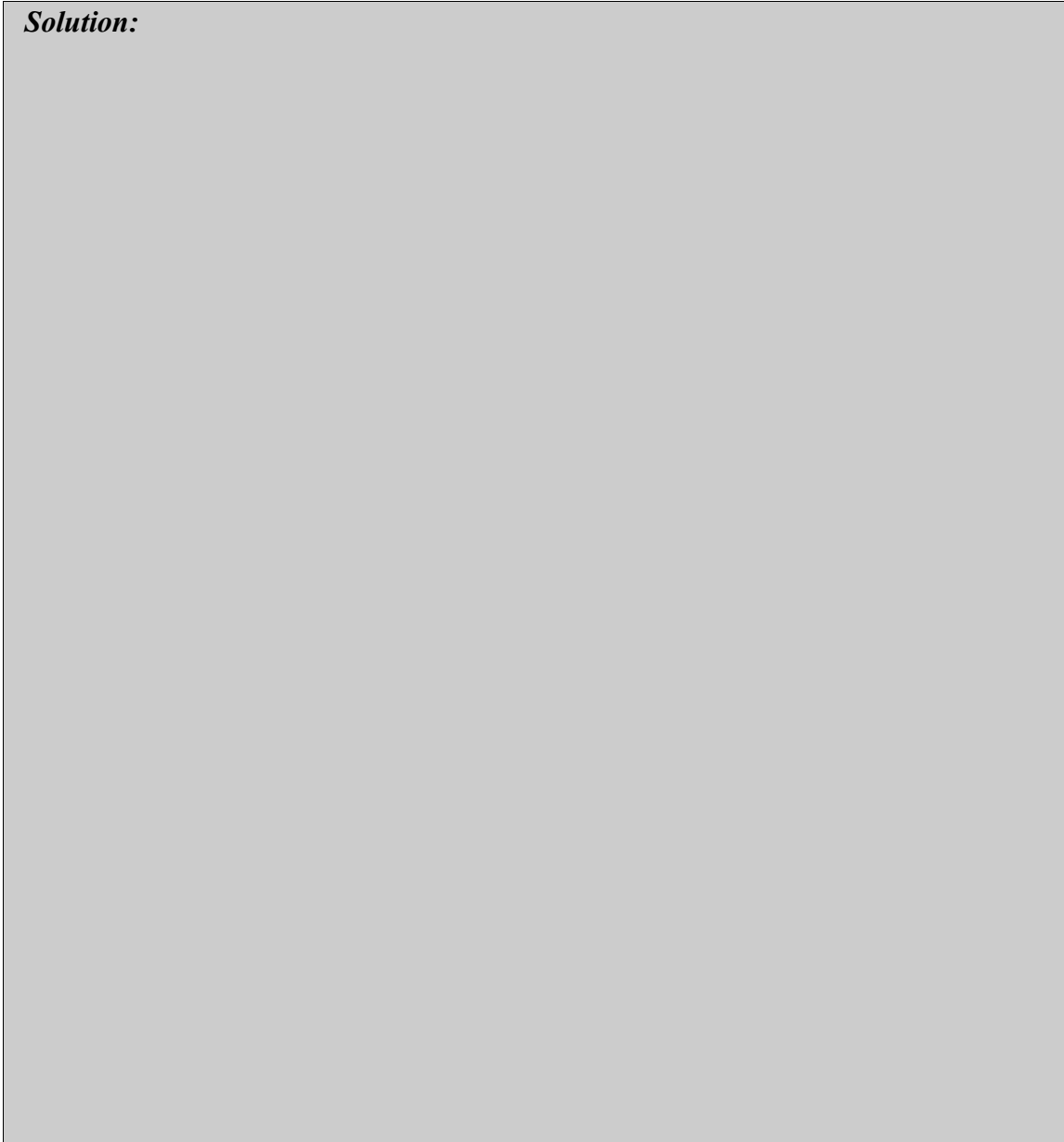
The motor open-loop transfer function as referred to in the rest of this laboratory includes both filter and computational delay effects, as characterized in Equation [4.19].

Using the SLD parameters, write a MATLAB script to plot the magnitude of the sensitivity function $S(s)$ of the complete system [4.19] over the frequency range from 1 to 1000 rad/s. Include the plot, on a log-log scale, in your report. What is the maximum sensitivity, M_s , of the system?

Hints:

Use the example MATLAB script provided in Section 4.5.1.
You do not need to derive any equations.

Solution:



0	1	2
---	---	---

5. Write a MATLAB script to plot the maximum values M_s of the magnitude of the sensitivity function $S(s)$ over a frequency range from 1 to 1000 rad/s for the three open-loop transfer functions, $P_1(s)$, $P_2(s)$, and $P_3(s)$, described underneath. In each case, consider $\zeta = 0.8$ and change ω_0 from 5 to 40 rad/s by increment of 1.5 rad/s.

The first process transfer function to consider is similar to Equation [4.15] and represents the basic voltage-to-speed system, as shown below:

$$P_1(s) = \frac{K}{\tau s + 1} \quad [4.20]$$

The second process transfer function to consider represents the basic system with a computational delay of one sample, as represented below:

$$P_2(s) = \frac{K e^{(-s h)}}{\tau s + 1} \quad [4.21]$$

The third process transfer function to consider represents the basic system with one-sample delay and the low-pass filter $G_f(s)$, as described below:

$$P_3(s) = \frac{K \omega_f e^{(-s h)}}{(\tau s + 1)(s + \omega_f)} \quad [4.22]$$

Include the graph containing the three plots in your report.

Discuss how the sample delay and the low-pass filter affect the sensitivity of the system.

Solution:



0	1	2
---	---	---

6. Determine the design value of ω_0 which gives the best robustness for the complete system characterized in Equation [4.22]. Justify your choice. What is the corresponding M_s ? Such a PI controller design is referred to as the **Minimum Sensitivity Design (MSD)**. Compare the robustness of both SLD and MSD systems. Which one is more robust?

Solution:

0	1	2
---	---	---

4.5.2.2. Large Process Variations – The Complementary Sensitivity Function

So far we have discussed small variations in the process transfer function $P(s)$. There are also useful results for large variations. Assume that the process transfer function changes from $P(s)$ to $P(s)+\Delta P(s)$, where $\Delta P(s)$ does not have any poles in the right half plane. Considering the Nyquist curve of a nominal loop transfer function PC and its uncertainty caused by process variations ΔP , the closed-loop system is found to be stable if the following condition is respected:

$$\frac{|\Delta P(s)|}{|P(s)|} < \frac{1}{|T(s)|} \quad [4.23]$$

where $T(s)$ is given by:

$$T(s) = \frac{P(s) C(s)}{1 + P(s) C(s)} \quad [4.24]$$

Considering Equations [4.12] and [4.24], it can be noted that:

$$S(s) + T(s) = 1 \quad [4.25]$$

$T(s)$ is called the complementary sensitivity function.

The inequality [4.23] gives important insight into the robustness issues. For example it tells that model precision is most critical at those frequencies where $|T(i\omega)|$ is large (considering $s = i\omega$). The value M_t , as defined below:

$$M_t = \max_{\omega} |T(i\omega)| \quad [4.26]$$

is therefore an important quantity. M_t is the largest value of the complementary sensitivity. If $M_t < 2$, it means that the process can be changed by up to 50% without making the system unstable. Also notice that very large errors are permitted for those frequencies where the complementary sensitivity function $T(i\omega)$ is small.

1. Modify the MATLAB script written in Step 5 of Section 4.5.2.1 in order to plot the maximum values M_t of the magnitude of the complementary sensitivity function $T(s)$ over a frequency range from 1 to 1000 rad/s for the three open-loop transfer functions, $P_1(s)$, $P_2(s)$, and $P_3(s)$, described in Equations [4.20], [4.21], and [4.22], respectively. In each case, consider $\zeta = 0.8$ and change ω_0 from 5 to 40 rad/s by increment of 1.5 rad/s. Discuss how the system is affected.

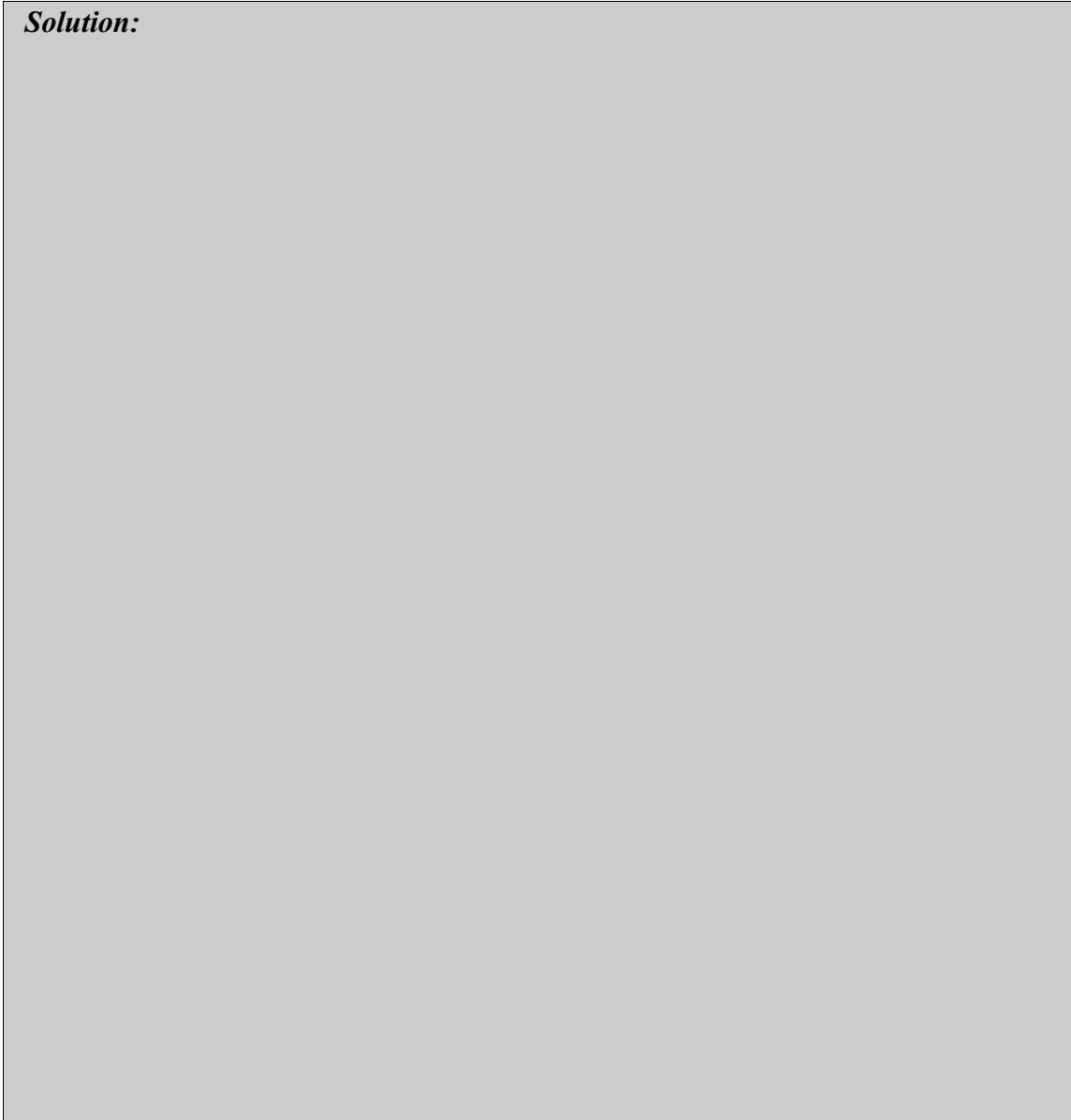
Hint:

Using Equation [4.25], the complementary sensitivity function T can be determined as follows:

$$T(i \omega) = 1 - S(i \omega)$$

[4.27]

Solution:



0	1	2
---	---	---

2. Calculate the largest value, M_t , of the complementary sensitivity function for the complete system characterized in Equation [4.22] resulting from the SLD controller parameters as well as from the MSD controller parameters. Which system is more robust to large changes in the open-loop process?

Solution:

0	1	2
---	---	---

4.5.3. Stability Margins

4.5.3.1. Preamble

Another way to investigate the robustness is to determine the changes in the process required to make the system unstable. A simple measure is the shortest distance, d_s , from the critical point -1 to the Nyquist curve of the loop transfer function PC . The value of d_s can be determined as follows:

$$d_s = \min_{\omega} |1 + P(i\omega)C(i\omega)| \quad [4.28]$$

Considering Equations [4.12] and [4.13], Equation [4.28] can also be written as:

$$d_s = \frac{1}{M_s} \quad [4.29]$$

This stability measure is sometimes preferable to the traditional stability measures of gain and phase margins because it suffices to give one number only: d_s . If we do not want to introduce an extra parameter, we can simply use the maximum sensitivity M_s . For a design with given M_s , the gain, g_m , and phase, ϕ_m , margins have the following properties:

$$\left(\frac{M_s}{M_s - 1} = \frac{1}{1 - d_s} \right) \leq g_m \quad [4.30]$$

and:

$$\left(2 \arcsin \left(\frac{1}{2} \frac{1}{M_s} \right) = 2 \arcsin \left(\frac{1}{2} d_s \right) \right) \leq \phi_m \quad [4.31]$$

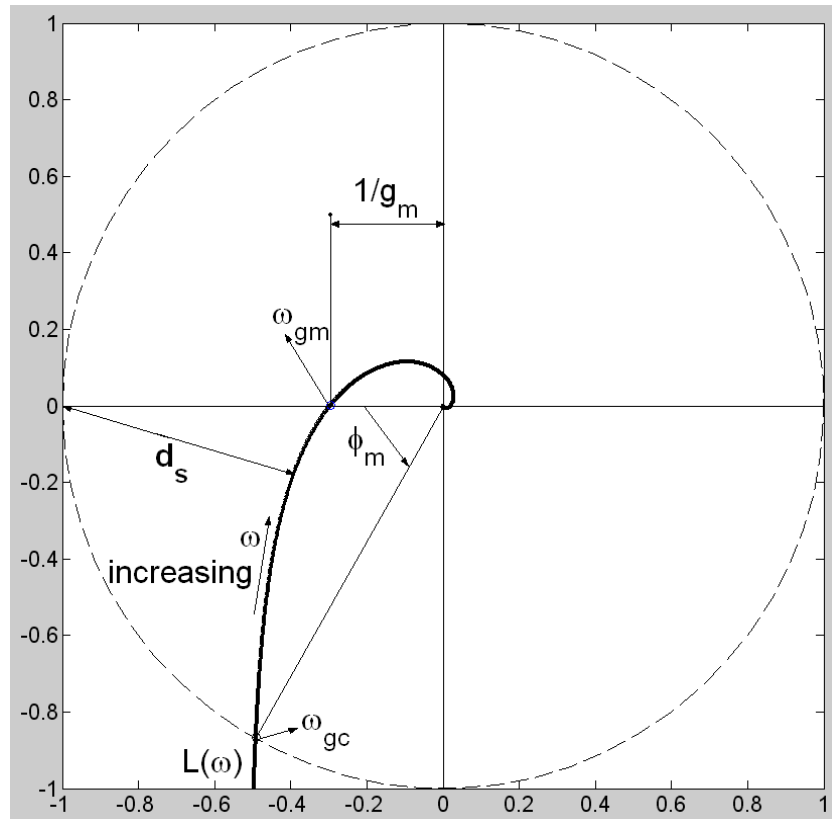
where g_m is the gain margin and ϕ_m is the phase margin.

The delay margin, T_{dm} , is another stability concept which is defined as the amount of time delay that brings the system to the stability boundary. For a simple system where the Nyquist curve is as shown in Figure 4.2, the delay margin is given by:

$$T_{dm} = \frac{\phi_m}{\omega_{gc}} \quad [4.32]$$

where ω_{gc} is the gain crossover frequency, that is to say the frequency where the loop gain equals one.

Figure 4.2 represents the Nyquist curve of the loop transfer function $L = PC$ with indication of gain and phase margins.

Figure 4.2 Nyquist Diagram Of The Loop Transfer Function $L=PC$

As illustrated in Figure 4.2, the gain crossover frequency ω_{gc} is the frequency at which the magnitude of the loop transfer function L equals one, that is to say: $|L(i\omega_{gc})| = 1$. At the gain crossover frequency, the phase margin is defined as the distance of the system phase angle above -180° .

The gain margin frequency ω_{gm} is the frequency at which the imaginary part of the loop transfer function L equals zero, that is to say where the system phase angle equals -180° . The gain margin equals one divided by the magnitude of the loop transfer function L At the gain margin frequency. In other words: $g_m = 1 / |L(i\omega_{gm})|$. In Figure 4.2, $1/g_m = |L(i\omega_{gm})|$.

4.5.3.2. Assignment Questions

Please answer the following questions.

1. For the complete system characterized by the open-loop transfer function [4.22], write a MATLAB script to plot the system gain margin g_m , phase margin ϕ_m , and delay margin $nDelay$, over a frequency range from 1 to 1000 rad/s. In each case, consider $\zeta = 0.8$ and change ω_0 from 7 to 40 rad/s by increment of 0.25 rad/s. Carry out your calculations using the Nyquist method as well as the maximum sensitivity parameter M_s . What are your observations? Do the inequalities [4.30] and [4.30] hold true?

Note:

$nDelay$ is defined as the maximum integer number of sampling interval delay that the closed-loop system can handle before becoming unstable. It is calculated as shown below:

$$nDelay = \text{floor}\left(\frac{T_{dm}}{h}\right) \quad [4.33]$$

where $\text{floor}(x)$ is the floor function, which truncates a number x to the greatest integer less than or equal to that number.

It follows that an extra time delay of $nDelay + 1$ sampling periods would make the closed-loop system unstable.

Solution:



0	1	2
---	---	---

Robustness

2. For the SLD system, calculate the gain, phase, and delay margins as well as $nDelay$ using both Nyquist and M_s methods. Fill up Table 4.4 below.

<i>Method</i>	ω_0 [rad/s]	ζ	g_m	ϕ_m [°]	T_{dm} [s]	$nDelay$
Nyquist		0.8				
M_s		0.8				

0	1	2
---	---	---

Table 4.4 Stability Margins For The Speed Laboratory Design (SLD) System

3. For the MSD system, calculate the gain, phase, and delay margins as well as $nDelay$ using both Nyquist and M_s methods. Fill up Table 4.5 below.

<i>Method</i>	ω_0 [rad/s]	ζ	g_m	ϕ_m [°]	T_{dm} [s]	$nDelay$
Nyquist		0.8				
M_s		0.8				

0	1	2
---	---	---

Table 4.5 Stability Margins For The Minimum Sensitivity Design (MSD) System

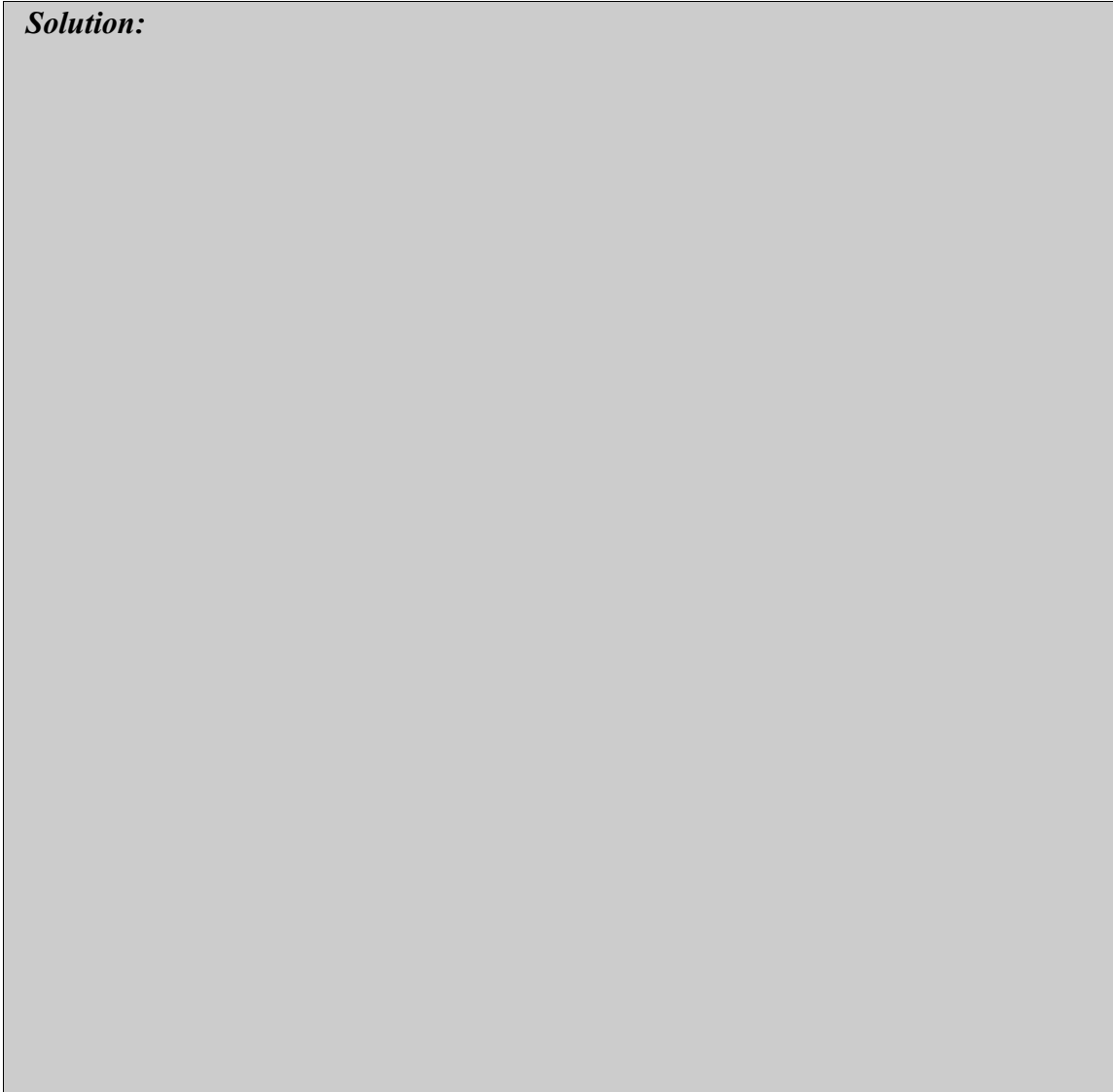
4. For the complete system characterized by the open-loop transfer function [4.22], write a MATLAB script to plot the Nyquist diagram over a frequency range from 1 to 1000 rad/s. Consider both SLD and MSD PI controllers. Identify the relevant crossover frequencies and fill up Table 4.6. What are your observations? Discuss.

<i>Design Type</i>	ω_0 [rad/s]	ζ	ω_{gc} [rad/s]	ω_{gm} [rad/s]
SLD		0.8		
MSD		0.8		

0	1	2
---	---	---

Table 4.6 Crossover Frequencies For Both Controller Designs

Solution:



0 1 2

4.5.4. Pre-Laboratory Results Summary Table

Table 4.7 should be completed before you come to the in-laboratory session to perform the experiment.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Value</i>	<i>Unit</i>
2.5.3. Nominal Process Parameters				
6.	Open-Loop Steady-State Gain	K		rad/(V.s)
6.	Open-Loop Time Constant	τ		s
4.5.2. Robustness And Sensitivity				
1.	SLD Proportional Gain	k_p		V.s/rad
1.	SLD Integral Gain	k_i		V/rad
1.	SLD Maximum Sensitivity	M_s		
1.	MSD Undamped Natural Frequency	ω_0		rad/s
1.	MSD Damping Ratio	ζ		
1.	MSD Maximum Sensitivity	M_s		
2.	SLD Maximum Complementary Sensitivity	M_t		
2.	MSD Maximum Complementary Sensitivity	M_t		
4.5.3. Stability Margins				
2.	SLD Gain Margin Using Nyquist	g_m		
2.	SLD Phase Margin Using Nyquist	ϕ_m		°
2.	SLD Delay Margin Using Nyquist	$nDelay$		
3.	MSD Gain Margin Using Nyquist	g_m		
3.	MSD Phase Margin Using Nyquist	ϕ_m		°
3.	MSD Delay Margin Using Nyquist	$nDelay$		
4.	SLD Gain Crossover Frequency	ω_{gc}		rad/s
4.	SLD Phase Crossover Frequency	ω_{gm}		rad/s
4.	MSD Gain Crossover Frequency	ω_{gc}		rad/s
4.	MSD Phase Crossover Frequency	ω_{gm}		rad/s

Table 4.7 Robustness Pre-Laboratory Assignment Results

4.6. In-Laboratory Session

4.6.1. QICii Robustness Module

4.6.1.1. Module Description

The main tool for this laboratory is the front panel of the module entitled *Robustness* in the QICii software, which should be similar to the one shown in Figure 4.3. The two additional *Parameters* tabs are shown in Figures 4.4 and 4.5.

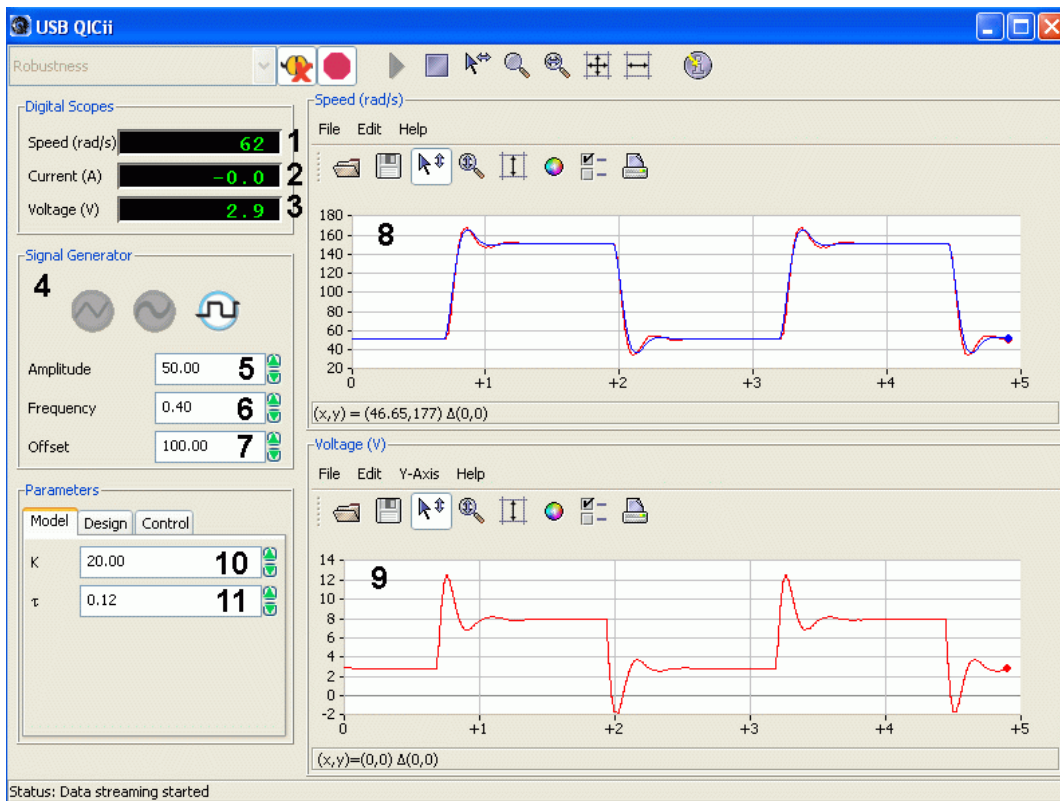


Figure 4.3 *Robustness* Module Of The QICii Software

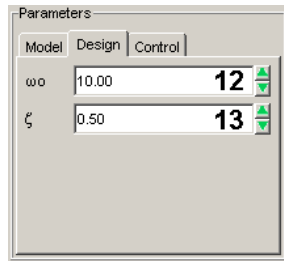


Figure 4.4 Module *Design* Tab

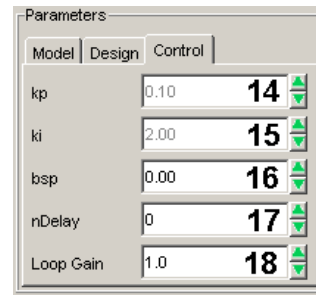


Figure 4.5 Module *Control* Tab

As a quick module description, Table 4.8 lists and describes the main elements composing the QICii *Robustness* module user interface. Every element is uniquely identified through an ID number and located in Figures 4.3, 4.4, and 4.5.

<i>ID #</i>	<i>Label</i>	<i>Parameter</i>	<i>Description</i>	<i>Unit</i>
1	Speed	ω_m	Motor Output Speed Numeric Display	rad/s
2	Current	I_m	Motor Armature Current Numeric Display	A
3	Voltage	V_m	Motor Input Voltage Numeric Display	V
4	Signal Generator		Square Wave Speed Reference Generator	rad/s
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 Simulated (in blue) Speeds	rad/s
9	Voltage	V_m	Scope With Applied Motor Voltage (in red)	V
10	K	K	<i>Model</i> Tab: Motor Model Steady-State Gain Input Box	rad/(V.s)
11	τ	τ	<i>Model</i> Tab: Motor Model Time Constant Input Box	s
12	ω_0	ω_0	<i>Design</i> Tab: Closed-Loop Undamped Natural Frequency	rad/s
13	ζ	ζ	<i>Design</i> Tab: Closed-Loop Damping Ratio	

<i>ID #</i>	<i>Label</i>	<i>Parameter</i>	<i>Description</i>	<i>Unit</i>
14	kp	k_p	<i>Control</i> Tab: Controller Proportional Gain Input Box	V.s/rad
15	ki	k_i	<i>Control</i> Tab: Controller Integral Gain Input Box	V/rad
16	bsp	b_{sp}	<i>Control</i> Tab: Controller Proportional Set-Point Weight Input Box	
17	nDelay	$nDelay$	<i>Control</i> Tab: Sampling Interval Number Delay Input Box	
18	Loop Gain		<i>Control</i> Tab: Loop Gain Input Box	

Table 4.8 QICii *Robustness* Module Nomenclature

The *Robustness* module program runs the process in closed-loop with the motor reference speed given by the signal generator. There are two windows that show the time histories of motor speed (control output) and motor voltage (control input). The top scope also shows a simulation of the closed-loop system, which runs in parallel with the hardware. The input of the simulation is equal to the generated reference speed and the output of the simulation is displayed (blue trace) in the same window as the actual motor speed (red trace). The simulation model parameters K and τ can be adjusted from the *Model* tab of the front panel.

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

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

The actual speed is obtained by filtering the position signal using the following filter:

$$\omega_m = \frac{s \theta_m}{T_f s + 1}$$

where θ_m is the position of the motor shaft measured by the encoder and T_f is the time constant of filter for measured signal.

4.6.1.2. Module Startup

To start and use the *Robustness* module, follow the steps described below:

- Step 1. Power up the DCMCT. LED2 should light up while LED3 should flash on and off repeatedly.
- Step 2. Press the *Reset* button on the QIC. Again, LED2 should remain on while LED3 should flash on and off repeatedly.
- Step 3. Press the DCMCT User Switch (i.e. pushbutton next to the two flashing LED's). LED2 and LED3 should both turn off.
- Step 4. Launch the USB QICii software and select *Robustness* in the drop-down menu.
- Step 5. Select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller. LED2 should light up while LED3 should still be off, and the controller should start running.

Note:

The drop-down menu will be disabled (i.e. is unavailable) while the controller is running. You must select the *Stop controller* button on top of the QICii window to stop the controller (this should also turn off LED2, and enable the drop-down menu again). Once the drop down menu is enabled again and the controller is stopped, you can select any one of the other controller experiments, if you want. After selecting some other controller experiment, you can once again select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller of your choice.

The default module parameters loaded after download are given in Table 4.9.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>	<i>K [rad/(V.s)]</i>	<i>τ [s]</i>	<i>ω_0 [rad/s]</i>
Square Wave	50.0	0.4	100.0	20.0	0.12	20.0
<i>ζ</i>	<i>k_p [V.s/rad]</i>	<i>k_i [V/rad]</i>	<i>b_{sp}</i>	<i>nDelay</i>	<i>Loop Gain</i>	
0.5	0.07	2.40	0	0	1.0	

Table 4.9 Default Parameters For The *Robustness* Module

4.6.2. Stability Margins Evaluation

This procedure provides experimental verification of the theoretical work and simulations carried out in Section 4.5.3.

4.6.2.1. Preamble #1: Performance-Related Parameters

Digital computers give great flexibility in implementing control systems. In particular there are many interesting possibilities to design systems that are much easier to use than traditional control systems. Take the PI controller as an example. Most PI controllers are implemented in such a way that the user can adjust proportional and/or integral gain(s). There are, however, many other possibilities. The controller used for this laboratory can be interpreted as a PI controller for a general first-order system. When the parameters of the process model, K and τ , are introduced the properties of the closed loop system are influenced simply by changing the parameters ζ and ω_0 which directly give the performance of the closed-loop. The parameters ζ and ω_0 can therefore be called *performance-related parameters*. The response speed is given by ω_0 and the shape of the response by ζ . It is much easier to change these parameters than to change proportional and integral gains, k_p and k_i .

The traditional approach to control which was used in Chapter 3 is to find suitable values of ζ and ω_0 from a given specification and to compute the controller parameters in order to achieve a desired pole placement and/or closed-loop transfer function. An alternative would be to consider ζ and ω_0 as the controller parameters which the operator can adjust. These parameters have very good physical interpretations. Parameter ζ determines the shape of the step response and ω_0 determines the time scale. The parameter ζ has a limited range, say $0.5 \leq \zeta \leq 1$. The value $\zeta = 0.5$ gives an oscillatory response with overshoot and $\zeta = 1$ gives a critically-damped response without overshoot. The parameter ω_0 also has limited range because it must be greater than a minimum value in order to have a positive gain k_p . Too large values of ω_0 lead to saturation of the control signal and possibly instabilities due to unmodeled dynamics.

In the *Control* tab, both proportional and integral gains, k_p and k_i , are automatically calculated by the software as functions of K and τ from the *Model* tab and of ω_0 and ζ from the *Design* tab. Please refer to Chapter 3 as required. It is reminded that this is done to achieve the desired closed-loop transfer function from reference to velocity expressed as follows (with a set-point weight $b_{sp} = 0$):

$$G_{\omega, r}(s) = \frac{\omega_0^2}{s^2 + 2 \zeta \omega_0 s + \omega_0^2} \quad [4.34]$$

4.6.2.2. Preamble #2: Actual Closed-Loop Implementation

The actual closed-loop implementation contains two additional parameters, both available in the *Control* tab. They are namely *Loop Gain* and *nDelay*. The corresponding block diagram is presented in Figure 4.6, where z^{-1} is the discrete time operator. The *Loop Gain* parameter is a pure gain and is used to estimate the system gain margin. The *nDelay* parameter is an integer number of sample periods and is used to determine the extra time delay needed to make the system unstable.

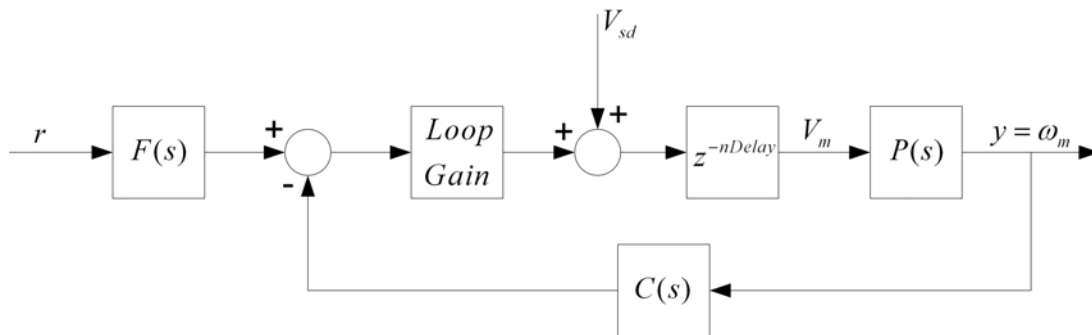


Figure 4.6 Actual Closed-Loop Block Diagram

A disturbance torque can also be simulated by applying an additional voltage, V_{sd} , to the motor input. The DCMCT system uses this as a way to simulate a load disturbance in a repeatable manner, as illustrated in Figure 4.6. You can do this by triggering the User Switch (next to the two LED's) on the DCMCT.

4.6.2.3. Stability Margins Evaluation

Please follow the steps below:

Step 1. Set the reference signal to a constant speed of 100 rad/s. When you change the reference signal level ensure that the control signal does not saturate. In the *Model* tab, set both parameters K and τ to their values given in Table 4.2. You will now experimentally assess the gain margin g_m , and associated frequency ω_{gm} as well as the phase margin represented by $nDelay$ and its associated frequency ω_{gc} and compare the obtained values with the theory. In the *Control* tab, the set-point weight b_{sp} should stay set to 0 throughout this procedure.

Step 2. Start by investigating the Speed Lab Design (SLD) closed-loop system. Do so by setting in the *Design* tab both tuning parameters ω_0 and ζ to their SLD values given in Table 4.3.

Step 3. At the gain-margin frequency ω_{gm} , the system phase angle is equal to -180° and the gain margin value g_m corresponds to the value of the extra pure gain in the closed-loop necessary to make the system on the verge of instability. Therefore to experimentally estimate g_m , increase the *Loop Gain* parameter, in the *Control* tab, starting from 1 by steps of 0.2. Ensure that $nDelay$ is set to 0. At every iteration, press the disturbance voltage User Switch on the DCMCT. The output speed responds to the disturbance with a transient. If the transient does not die off, or the speed becomes unstable, you have hit the margin. The corresponding frequency of the obtained oscillations is the gain-margin frequency ω_{gm} . Fill up Table 4.10.

Hint:

Average 20 speed oscillation periods to obtain a good estimate of ω_{gm} .

	<i>Theoretical</i>	<i>Experimental</i>
<i>Gain Margin: g_m</i>		
<i>Gain-Margin Frequency: ω_{gm} [rad/s]</i>		



Table 4.10 Assessment Of Gain Margin And Associated Frequency For The SLD System

Step 4. As expressed in Equations [4.32] and [4.33], $nDelay$ is a measure of the phase margin. $nDelay$ represents the extra sample time delay (equivalent to an extra phase lag) required to make the system on the verge of instability. The corresponding frequency of the obtained oscillations is the gain-crossover frequency ω_{gc} . Increase the $nDelay$ parameter, in the *Control* tab, starting from 0 by steps of 1. Ensure that *Loop Gain* is set to 1.0. At every iteration, press the disturbance voltage User Switch. The output speed responds to the disturbance with a transient. If the transient does not die

off, or the speed becomes unstable, you have hit the margin. Fill up Table 4.11.

Hint:

Average 10 speed oscillation periods to obtain a good estimate of ω_{gc} .

	<i>Theoretical</i>	<i>Experimental</i>
<i>nDelay</i>		
<i>Gain-Crossover Frequency: ω_{gc} [rad/s]</i>		

0	1	2
---	---	---

Table 4.11 Assessment Of *nDelay* And Associated Frequency For The SLD System

Step 5. You will now investigate the Minimum Sensitivity Design (MSD) closed-loop system. Do so by setting, in the *Design* tab, both tuning parameters ω_0 and ζ to their values found in Section 4.5.2.1 Question 6.

Step 6. Estimate g_m by increasing the *Loop Gain* parameter, in the *Control* tab, starting from 1 by steps of 0.2. Ensure that *nDelay* is set to 0. At every iteration, press the disturbance voltage User Switch. The output speed responds to the disturbance with a transient. If the transient does not die off, or the speed becomes unstable, you have hit the margin. The corresponding frequency of the obtained oscillations is the gain-margin frequency ω_{gm} . Fill up Table 4.12.

Hint: Average 20 speed oscillation periods to obtain a good estimate of ω_{gm} .

	<i>Theoretical</i>	<i>Experimental</i>
<i>Gain Margin: g_m</i>		
<i>Gain-Margin Frequency: ω_{gm} [rad/s]</i>		

0	1	2
---	---	---

Table 4.12 Assessment Of Gain Margin And Associated Frequency For The MSD System

Step 7. Increase the $nDelay$ parameter, in the *Control* tab, starting from 0 by steps of 1. Ensure that *Loop Gain* is set to 1.0. At every iteration, press the disturbance voltage User Switch. The output speed responds to the disturbance with a transient. If the transient does not die off, or the speed becomes unstable, you have hit the margin. Fill up Table 4.13.

Hint: Average 10 speed oscillation periods to obtain a good estimate of ω_{gc} .

	<i>Theoretical</i>	<i>Experimental</i>
<i>nDelay</i>		
<i>Gain-Crossover Frequency: ω_{gc} [rad/s]</i>		

0

1

2

Table 4.13 Assessment Of $nDelay$ And Associated Frequency For The MSD System

Step 8. Using your previous theoretical and experimental results, discuss the merits of both PI controller designs, namely: SLD and MSD.

Solution:

0

1

2

Step 9. Ensure that both tuning parameters ω_0 and ζ , located in the *Design* tab, are set to their SLD values, as given in Table 4.3. Change (i.e. increase or decrease) the value of the design specification ω_0 and see how gain and phase margins get affected. Does this comply with theory?

Solution:

0

1

2

Step 10. Ensure that both tuning parameters ω_0 and ζ , located in the *Design* tab, are set to their SLD values, as given in Table 4.3. Change (i.e. increase or decrease) the value of the design specification ζ and see how gain and phase margins get affected.

Solution:

0 1 2

Step 11. You will now investigate the sensitivity to model parameters for the Speed Lab Design (SLD) closed-loop system. The QICii module allows you to directly see the effects of variations (typically due to model uncertainty) in the model parameters on the closed-loop response. The module *Model* tab allows you to change the parameters K and τ of the process model. The PI controller gains are then automatically calculated by the software according to the pole-placement design carried out in Chapter 3, so that the desired closed-loop transfer function [4.34] is achieved. Set the reference signal parameters of the QICii module window as displayed in Table 4.14. Also set in the *Design* tab both tuning parameters ω_0 and ζ to their SLD values given in Table 4.3. In the *Model* tab, ensure that both parameters K and τ are set to their nominal values, as given in Table 4.2.

<i>Signal Type</i>	<i>Amplitude [rad/s]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad/s]</i>
Square Wave	50	0.5	100

Table 4.14 Signal Generator Parameters For The Sensitivity To Model Parameters Tests

Step 12. Change the open-loop steady-state gain K , in the *Model* tab, as per the first column of Table 4.15. Assess what the effect on the speed response is of designing for a process whose gain K is perturbed from its nominal value. Do so by comparing the actual output speed with the desired closed-loop response, as simulated by the software according to Equation [4.34] and plotted in the QICii *Speed* graph. Fill up Table 4.15 as required.

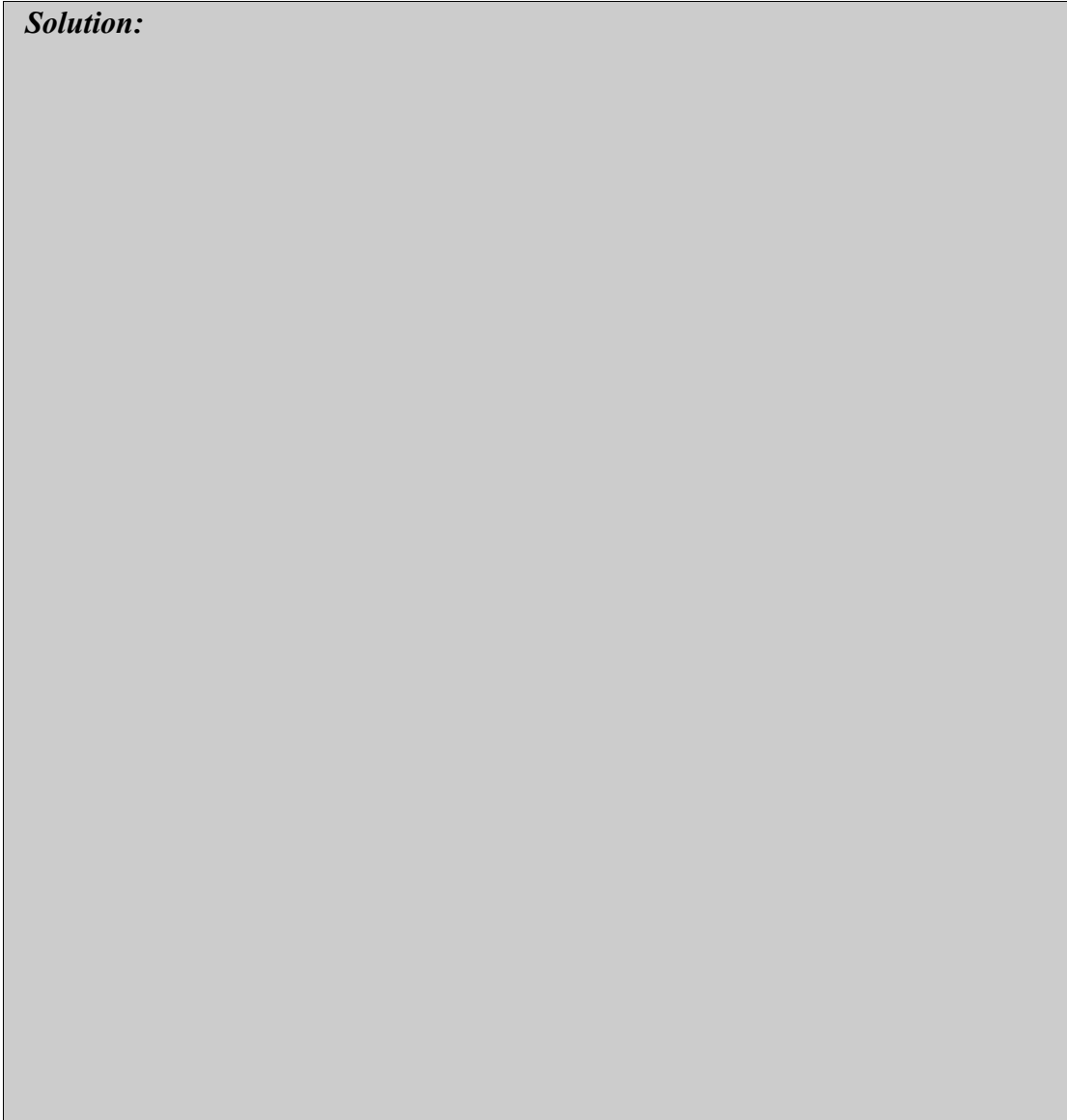
<i>Model Parameter: K [rad/(V.s)]</i>	<i>Response Speed [faster, slower, match]</i>	<i>Response Damping [more, less, match]</i>
10		
20		
30		

0	1	2
---	---	---

Table 4.15 Actual SLD Response Characteristics Compared To Desired Output With Variations On K

Step 13. Summarize your observations in your report. Select some representative results, screen captures, and plots. Discuss.

Solution:



0	1	2
---	---	---

Step 14. In the *Model* tab, reset the parameter K to its value given in Table 4.2. Change the open-loop time constant τ , in the *Model* tab, as per the first column of Table 4.16. Assess what the effect on the speed response is of designing for a process whose time constant τ is perturbed from its nominal value. Do so by comparing the actual output speed with the desired closed-loop response, as simulated by the software according to Equation [4.34] and plotted in the QICii *Speed* graph. Fill up Table 4.16 as required.

<i>Model Parameter:</i> τ [s]	<i>Response Speed</i> [faster, slower, match]	<i>Response Damping</i> [more, less, match]
0.04		
0.09		
0.14		



Table 4.16 Actual SLD Response Characteristics Compared To Desired Output With Variations On τ

Step 15. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:



0	1	2
---	---	---

4.6.3. In-Laboratory Results Summary Table

Table 4.17 should be completed using Table 4.7, which contains data from the pre-laboratory assignments, as well as experimental results obtained during the in-laboratory session.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Pre-Lab Value</i>	<i>In-Lab Result</i>	<i>Unit</i>
4.6.2.	<i>Stability Margins (Using Nyquist)</i>				
3.	SLD Gain Margin	g_m			
3.	SLD Gain-Margin Frequency	ω_{gm}			rad/s
3.	SLD Delay Margin	$nDelay$			
3.	SLD Gain-Crossover Frequency	ω_{gc}			rad/s
3.	MSD Gain Margin	g_m			
3.	MSD Gain-Margin Frequency	ω_{gm}			rad/s
3.	MSD Delay Margin	$nDelay$			
3.	MSD Gain-Crossover Frequency	ω_{gc}			rad/s

Table 4.17 Robustness In-Laboratory Results

5. Position Control

5.1. Laboratory Objectives

The objective of this laboratory is to develop an understanding of PID control of motor angle, how it works, and how it can be tuned to meet required specifications.

In particular you will explore:

- Qualitative properties of proportional, derivative, and integral action.
- Set-point weighting.
- Design of controllers for specifications on the set-point response.
- Tracking of triangular signals.
- Response to load disturbances.
- Comparison with PI control of motor speed.

5.2. Preparation And Pre-Requisites

A pre-requisite to this laboratory is to have successfully completed the modelling and model validation laboratory described in Chapter 2. Before the lab, you should also review Proportional-Integral (PI) control of motor speed, as carried out in Chapter 3.

For the purpose of position control, the system can be represented by the block diagram in Figure 5.1. This block diagram illustrates the parts of the system that are relevant for position control.

Position Control

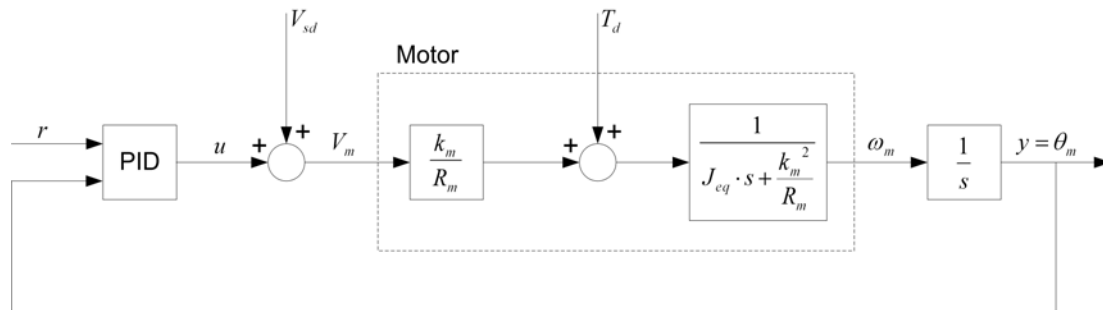


Figure 5.1 DCMCT Block Diagram For Position Control

The process is represented by a block which has motor voltage V_m and torque T_d as inputs and motor angle θ_m as the output. The torque is typically a disturbance torque that you apply manually to the inertial load. The PID controller block represents the control algorithm in the computer and the power amplifier. V_{sd} is a simulated external disturbance voltage.

The linear behaviour of the system is described by the transfer functions given in the block diagram. The major nonlinearities are the saturation of the motor amplifier at 15 V, Coulomb friction corresponding to 0.4 V, and the quantization of the encoder.

The major unmodeled dynamics is due to the effects of sampling. This can be approximated by a time delay of one sampling period.

5.3. Introduction: The PID Controller

The PID controller is a very common control algorithm. It is used for a variety of purposes and it often works very well. For systems with simple dynamics it can give close to optimal performance and for processes with complicated dynamics, it can often give good performance provided that specifications are not too demanding. Better performance can, however, be obtained by using more complicated controllers.

5.3.1. PID Control Law

The linear behaviour of a PID controller can be described by:

$$u(t) = k_p (b_{sp} r(t) - y(t)) + k_i \int_0^t r(\tau) - y(\tau) d\tau + k_d \left(b_{sd} \left(\frac{d}{dt} r(t) \right) - \left(\frac{d}{dt} y(t) \right) \right) \quad [5.1]$$

where $u(t)$ is the control signal, $r(t)$ the reference, and $y(t)$ the measured process output. The reference $r(t)$ is also called the set-point or the command. The linear behaviour of the PID controller is governed by the following parameters:

- k_p : proportional gain
- k_i : integral gain
- k_d : derivative gain
- b_{sp} : proportional set-point weight
- b_{sd} : derivative set-point weight
- T_f : time constant of filter for measured signal

Sometimes the filtered measurement $y_f(t)$ is also used in the control loop. It is computed from T_f , the time constant of filter for measured signal. The filter time constant T_f is typically set to a constant value and it is often combined with the sensor. The filter provides roll-off at high frequencies. It is important to reduce the effects of sensor noise and it improves robustness.

The PID controller is similar to the PI controller; the additional derivative term gives added flexibility. In particular it improves the possibilities of introducing damping because it is an approximate prediction of future measurements.

5.3.2. The Magic Of Integral Action

A nice property of a controller with integral action is that it always gives the correct steady-state value provided that there is an equilibrium. This can be seen simply by assuming that there is a steady-state value with constant $u(t) = u_{ss}$, $r(t) = r_{ss}$, and $y(t) = y_{ss}$. Equation [5.1] can then be written as:

$$u_{ss} = k_p (b_{sp} r_{ss} - y_{ss}) + k_i (r_{ss} - y_{ss}) t$$

Since the left-hand-side is a constant, the right-hand-side must also be a constant. This requires that $y_{ss} = r_{ss}$. Notice that the only assumption that has been made about the process is that there exists an asymptotic steady-state.

The load disturbance response of a PID controller has an interesting property which can be seen as follows. For load disturbances, the PID controller reduces to:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \left(\frac{d}{dt} e(t) \right)$$

where e is the control error, as defined below:

$$e(t) = r(t) - y(t)$$

Assume that the PID controller is connected to a process and that the closed loop system is stable. Apply a load disturbance in the form of a step. There will be a transient and the error will then go to zero because the system is stable. The above equation implies that:

$$\int_0^{\infty} e(\tau) d\tau = \frac{u(\infty)}{k_i} \quad [5.2]$$

The value $u(\infty)$ is the steady-state control signal required to eliminate the load disturbance. We can conclude from Equation [5.2] that the integral of the error due to a load disturbance is inversely proportional to the integral gain of the controller.

5.3.3. Controllers With Two Degrees Of Freedom

A PID controller with set-point weights $b_{sp} = 1$ and $b_{sd} = 1$ is a controller with error feedback because control actions are based on the error $e = r - y$. A controller where one of the set-point weights is different from one is said to have two degrees of freedom because the signal transmission from reference r to control u is different than from the signal transmission from measurement y to control u . A controller with two degrees of freedom has two inputs one output as shown in the block diagram in Figure 5.1. The set-point weights have no effect on load disturbance response or system dynamics but they can influence the response to command signals significantly. Both set-point weights range from 0 to 1.

5.4. Nomenclature

The nomenclature for system modelling and control design is given in Table 5.1.

<i>Symbol</i>	<i>Description</i>	<i>Unit</i>
θ_m	Motor angle	rad
ω_m	Motor speed which can be computed from the motor angle	rad/s
V_m	Voltage from the amplifier which drives the motor	V
T_d	Disturbance Torque externally applied to the inertial load	N.m
V_d	Disturbance Voltage corresponding to T_d	V
V_{sd}	Simulated External 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}	Equivalent 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
s	Laplace Operator	rad/s
h	Sampling Interval	s
t	Continuous Time	s
k_p	Proportional Gain	V/rad
k_i	Integral Gain	V/(rad.s)
k_d	Derivative Gain	V.s/rad
b_{sp}	Set-Point Weight On Proportional Part	
b_{sd}	Set-Point Weight On Derivative Part	
u	Control Signal	V
r	Reference Signal	rad
y	Measured Process Output	rad

Table 5.1 System Modelling Nomenclature

5.5. Pre-Laboratory Assignments

5.5.1. Comparison Between PD Position And PI Speed Controls

This Section shows the strong similarities that exist between PI control of motor speed and PD control of motor angle; their respective block diagrams are shown in Figures 5.2 and 5.3. For simplicity, Figures 5.2 and 5.3 represent closed-loop controllers with error feedback, i.e. $b_{sp} = 1$ and $b_{sd} = 1$, and where the disturbance torque T_d is neglected. The following analysis is made under the same assumptions.

Please answer the following questions.

1. Figure 5.2 describes the block diagram of the motor PI speed control.

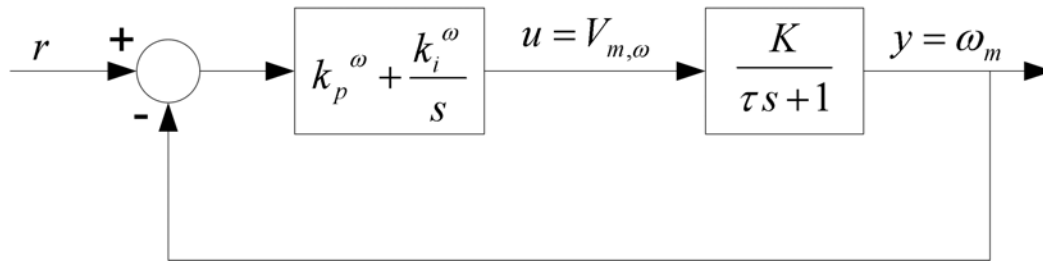


Figure 5.2 Block Diagram For PI Control Of Motor Speed

As characterized in Chapter 2, the process is defined by the following Laplace equation:

$$\Omega_m(s) = \frac{K V_{m,\omega}(s)}{\tau s + 1} \quad [5.3]$$

where the control signal $V_{m,\omega}$ is function of the two PI control gains k_p^ω and k_i^ω , as expressed below:

$$V_{m,\omega}(s) = \left(k_p^\omega + \frac{k_i^\omega}{s} \right) (R(s) - \Omega_m(s)) \quad [5.4]$$

Derive the closed-loop transfer function $\Omega_m(s)/R(s)$ for the PI controller of motor speed. Also determine the transfer function $V_{m,\omega}(s)/R(s)$ from reference to control signal.

Solution:

0 1 2

2. Figure 5.3 describes the block diagram of the motor PD position control.

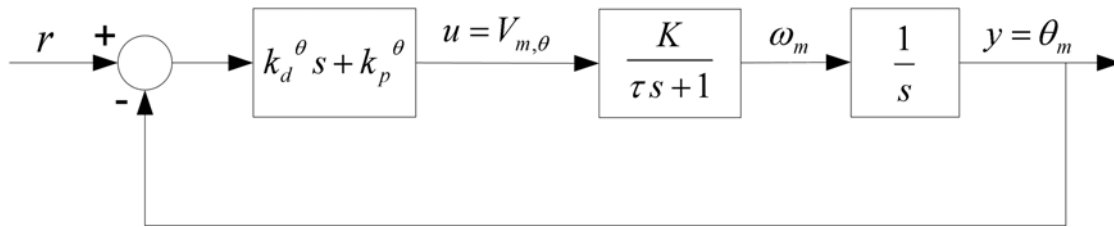


Figure 5.3 Block Diagram For PD Control Of Motor Angle

The process is defined by integrating Equation [5.3], as shown by the following Laplace equation:

$$\Theta_m(s) = \frac{K V_{m,\theta}(s)}{s(\tau s + 1)} \quad [5.5]$$

where the control signal $V_{m,\theta}$ is function of the two PD control gains k_p^θ and k_d^θ , as expressed below:

$$V_{m,\theta}(s) = (k_p^\theta + k_d^\theta s) (R(s) - \Theta_m(s)) \quad [5.6]$$

Derive the closed-loop transfer function $\Theta_m(s)/R(s)$ for the PD controller of motor position. Also determine the transfer function $V_{m,\theta}(s)/R(s)$ from reference to control signal.

Solution:

0	1	2
---	---	---

3. Compare the transfer functions between PI control of motor speed and PD control of motor angle. Considering the controller parameters, k_p^ω , k_i^ω , k_p^θ , and k_d^θ , what are their similarities and differences? Explain.

Solution:

0	1	2
---	---	---

5.5.2. Fundamental Limitations And Achievable Performance

One of the key tasks in controller design is to assess physical limitations and assess the performance that can be achieved by the actual system. This Section addresses this issue.

Assume that the motor represents the position loop in a manufacturing process and that the normal operation is a position change $\Delta\theta_{op}$. Since the motor is voltage-controlled, the minimum time to make that move is obtained by applying maximum input voltage, V_{max} , during the first half of the move and minimum input voltage, $-V_{max}$, during the second half in order to bring the motor to a stop. The shortest time to make the full transition is the peak time t_p . In other words, we assume full positive input voltage at V_{max} for $t_p/2$ and full negative input voltage at $-V_{max}$ for $t_p/2$.

For a voltage-controlled motor, the maximum velocity ω_{max} is obtained when V_{max} is applied. The velocity time-domain equation comes from taking the inverse Laplace transform of Equation [3.2], as shown below:

$$\omega_{max}(t) = K V_{max} \left(1 - e^{\left(-\frac{t}{\tau}\right)} \right) \quad [5.7]$$

A plot of ω_{max} is represented in Figure 5.4 during acceleration.

Differentiating Equation [5.7] gives the corresponding acceleration equation $a_{max}(t)$ in the time domain, as follows:

$$a_{max}(t) = \frac{K V_{max}}{\tau} e^{\left(-\frac{t}{\tau}\right)}$$

A plot of a_{max} is represented in Figure 5.5 during acceleration. It can be observed that with increasing velocity, the back-EMF voltage increases, thus reducing the achievable motor current and consequently acceleration.

Integrating Equation [5.7] with zero initial condition gives the maximum angular position equation θ_{max} during acceleration, as expressed below:

$$\theta_{max}(t) = K V_{max} \left(t + \tau e^{\left(-\frac{t}{\tau}\right)} - \tau \right)$$

A plot of θ_{max} is represented in Figure 5.6 during acceleration.

Considering Newton's second law of motion, the motor power P_{max} can be expressed as:

$$P_{max}(t) = J_{eq} \left(\frac{d}{dt} \omega_m(t) \right) \omega_m(t)$$

A plot of P_{max} is illustrated in Figure 5.7.

The peak power, as observed in Figure 5.7, occurs at the time t_{pP} , determined as follows:

$$t_{pP} = \ln(2) \tau$$

At peak power time, both velocity and acceleration have gone through half of their respective changes during full input voltage.

To make the most efficient use of the motor, we choose $t_p / 3 = t_{pP}$. The position response peak time is thus given by:

$$t_p = 3 \ln(2) \tau \quad \text{or} \quad t_p = 0.193 \text{ [s]}$$

Having obtained an estimate of achievable performance, you will now proceed with the controller design procedure.

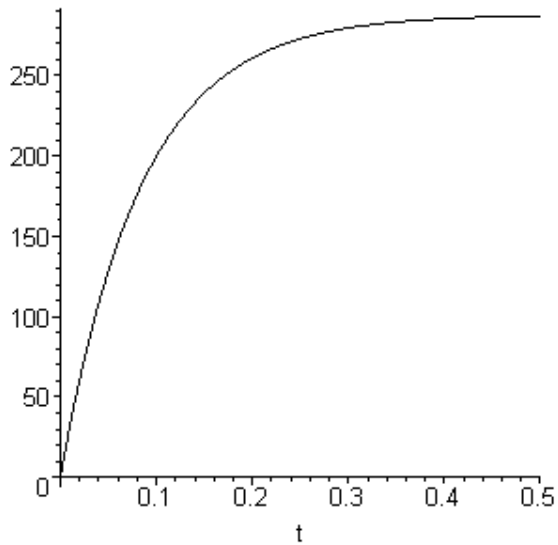


Figure 5.4 Velocity Plot For Maximum Input Voltage

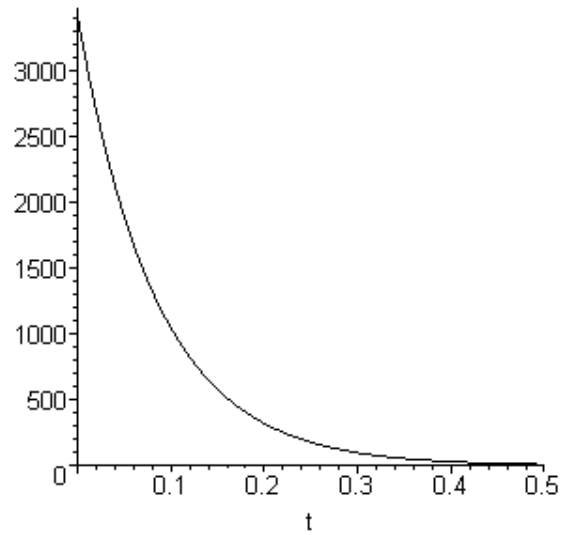


Figure 5.5 Acceleration Plot For Maximum Input Voltage

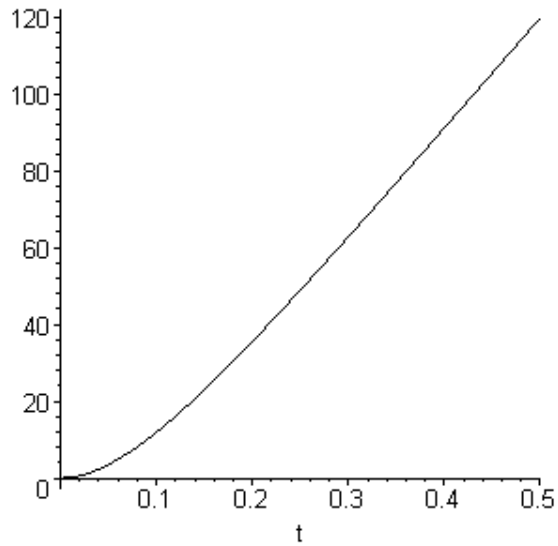


Figure 5.6 Angle Plot For Maximum Input Voltage

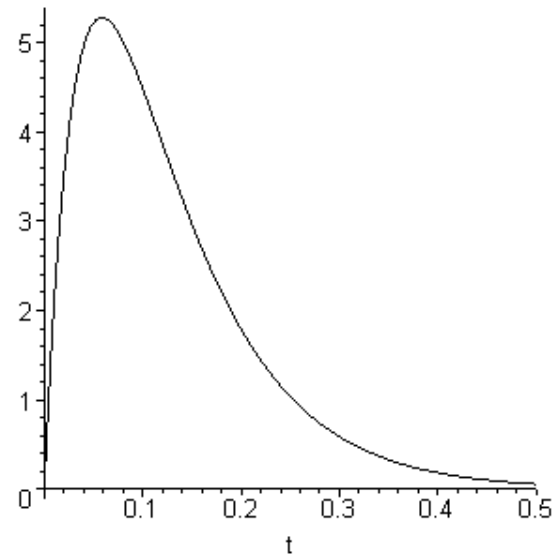


Figure 5.7 Power Plot For Maximum Input Voltage

5.5.3. PD Controller Design To Given Specifications

In this section you will practice design of a controller for a specified response to reference signals. The obtained design will then be verified by an experimental procedure in Section 5.6.3.

The following problem will be investigated: use the mathematical model of the process to design a Proportional-Plus-Derivative (PD) controller that gives:

i) a Percent Overshoot (PO) less than 18%:

$$PO \leq 18 \text{ ["\%"]} \quad [5.8]$$

ii) and a desired peak time t_p of:

$$t_p = 0.193 \text{ [s]} \quad [5.9]$$

The above design specifications should be achieved in response to a reference step input of amplitude $\Delta\theta_{op}$ such that the input voltage is close to its saturation level.

Integrating Equation [3.2] gives the transfer function from motor input voltage to motor output angle, as shown below:

$$G_{\theta, v}(s) = \frac{K}{s(\tau s + 1)} \quad [5.10]$$

Please answer the following questions.

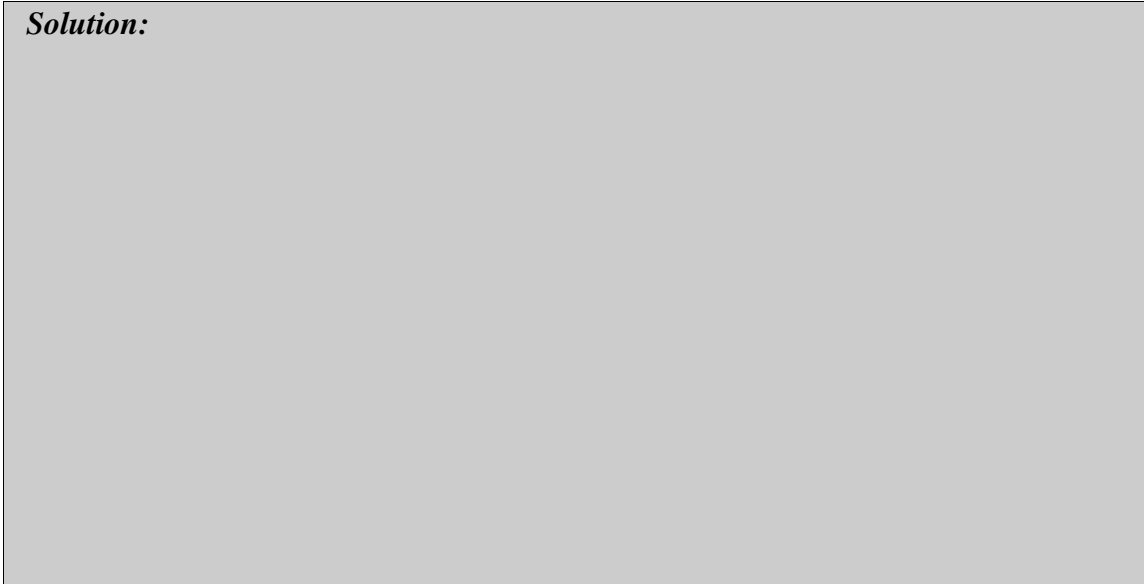
- Using the general PID control law expressed in Equation [5.1], derive the command equation in the Laplace domain, $U(s)$, for a PD controller ($k_i = 0$) with $b_{sp} = 1$.

Solution:

0	1	2
---	---	---

2. Considering the process transfer function [5.10], determine the closed-loop block diagram used for position control with the above PD controller. Include the simulated disturbance voltage V_{sd} .

Solution:

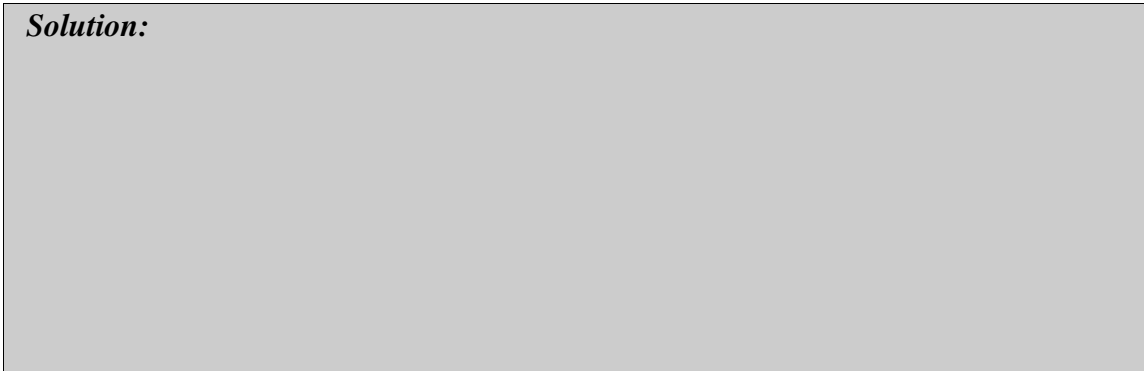


0	1	2
---	---	---

3. Assuming no disturbance, determine the closed-loop transfer function, $G_{\theta,r}(s)$, from reference to output, as defined below:

$$G_{\theta,r}(s) = \frac{\Theta_m(s)}{R(s)} \quad [5.11]$$

Solution:



0	1	2
---	---	---

4. 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 gives the following quadratic-lag transfer function:

$$G_{\theta,r}(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \quad [5.12]$$

In other words, derive k_p , k_d , and b_{sd} as functions of ω_0 , ζ , K , and τ .

Solution:

0	1	2
---	---	---

5. For a pure quadratic-lag system, as the one expressed in Equation [5.12], the maximum Percentage Overshoot, PO , over the steady-state response is given by the following relationship:

$$PO = 100 e^{\left(-\frac{\pi \zeta}{\sqrt{1-\zeta^2}}\right)} \quad [5.13]$$

You need to choose a damping ratio, ζ , that satisfies the design requirements. Is the value $\zeta = 0.5$ acceptable? Explain.

Solution:

0	1	2
---	---	---

6. Typically for quadratic-lag systems, the time to first peak t_p is expressed as follows:

$$t_p = \frac{\pi}{\omega_0 \sqrt{1-\zeta^2}} \quad [5.14]$$

Calculate ω_0 such that the peak time specification formulated in Equation [5.9] is satisfied.

Solution:

0	1	2
---	---	---

7. Evaluate the PD controller gains, k_p and k_d , satisfying the design requirements.

Hint:

Use the nominal values of the process parameters K and τ , as evaluated in Section 2.5.3 Question 6.

Solution:

0	1	2
---	---	---

8. Determine the maximum amplitude $\Delta\theta_{op}$ of a reference step input before reaching saturation of the command voltage. $\Delta\theta_{op}$ represents the motor move distance under desired operating conditions for which the design specifications apply.

Evaluate the maximum amplitude R_{max} of a square wave reference position signal such that the input voltage is close to and just below its saturation level.

Solution:

0	1	2
---	---	---

5.5.4. Tracking Triangular Signals

So far we have investigated tracking of reference signals in the form of square waves. We will now investigate tracking of triangular references.

Please answer the following questions.

- Using the complete PID control law expressed in Equation [5.1], write the commanded motor input voltage equation in the Laplace domain, $U(s)$.

Solution:

0	1	2
---	---	---

- Considering the complete PID controller for the open-loop transfer function [5.10] and assuming no disturbance torque, determine the closed-loop transfer function, $G_{\theta,r}(s)$, from reference to output angles, as defined in Equation [5.11].

Solution:

0	1	2
---	---	---

- Determine the transfer function, $G_{e,r}(s)$, from reference to control error for the above PID control loop. $G_{e,r}(s)$ is defined below:

$$G_{e,r}(s) = \frac{E(s)}{R(s)} \quad [5.15]$$

where the angular error is defined as follows:

$$E(s) = R(s) - \Theta_m(s) \quad [5.16]$$

Solution:

0	1	2
---	---	---

4. When a PD controller is used (i.e. no integral action: $k_i = 0$) with the proportional set-point weight $b_{sp} = 1$, apply the final value theorem to calculate the steady-state error, e_{ss_PD} , in response to a ramp reference signal of slope R_0 , defined such as:

$$R(s) = \frac{R_0}{s^2} \quad [5.17]$$

Comment.

Solution:

Solution (continued):

0	1	2
---	---	---

5. Evaluate the steady-state error e_{ss_PD} for the following ramp slope R_0 :

$$R_0 = 32.0 \left[\frac{\text{rad}}{s} \right] \quad [5.18]$$

Do so by using the PD tuning obtained in Section 5.5.3. Consider both cases where $b_{sd} = 0$ and where $b_{sd} = 1$. What can you infer?

Solution:

0	1	2
---	---	---

6. When a PID controller is used, apply the final value theorem to calculate the steady-state error, e_{ss_PID} , in response to a ramp reference signal of slope R_0 , as defined in Equation [5.17]. Comment.

Solution:

0	1	2
---	---	---

5.5.5. Response To Load Disturbances

Reduction of the effects of load disturbances is a key reason for using control. A torque on the motor axis is a typical example of a load disturbance for a position control system. In this laboratory we will show the effects of controller tuning on load disturbance response.

Please answer the following questions.

1. Considering the regulation problem (for $r = 0$), determine the closed-loop system block diagram with a disturbance torque input T_d applied on the DCMCT inertial load. Assume a PID controller with $b_{sp} = b_{sd} = 1$. The block diagram should contain the open-loop transfer function $G_{\theta,V}$ as formulated in [5.10], and be function of the following system parameters: k_p , k_i , k_d , K , τ , and J_{eq} .

Hint:

Assume $V_{sd} = 0$.

Solution:

0	1	2
---	---	---

Position Control

2. Find the closed-loop transfer function, $G_{\theta,T}(s)$, from disturbance torque to motor angle, as a function of the following system parameters: k_p , k_i , k_d , K , τ , and J_{eq} . $G_{\theta,T}(s)$ is defined below:

$$G_{\theta,T}(s) = \frac{\Theta_m(s)}{T_d(s)}$$

Solution:

0	1	2
---	---	---

3. Let us now consider a PD controller ($k_i = 0$). From your previous results, determine the closed-loop transfer function, $G_{\theta,T}(s)$, from disturbance torque to motor angle, as a function of the following system parameters: k_p , k_d , K , τ , and J_{eq} .

Solution:

0	1	2
---	---	---

4. When a PD controller is used (i.e. no integral action), apply the final value theorem to calculate the steady-state angle, θ_{ss_PD} , in response to a step input disturbance torque of amplitude T_{d0} , such as:

$$T_d(s) = \frac{T_{d0}}{s} \quad [5.19]$$

Comment.

Solution:

0	1	2
---	---	---

5. Assume that a constant simulated disturbance voltage $V_{sd} = V_{d0} = 3 \text{ V}$ is added to the motor voltage to simulate a disturbance torque $T_d = T_{d0}$. Using the system parameters, determine and evaluate the corresponding disturbance torque amplitude T_{d0} . Evaluate then the resulting steady-state angle θ_{ss_PD} when the PD controller derived in Section 5.5.3 is used.

Solution:

0	1	2
---	---	---

6. When a full PID controller is used, apply the final value theorem to calculate the steady-state angle, θ_{ss_PID} , in response to the step input disturbance torque defined in Equation [5.19].

Solution:

0	1	2
---	---	---

7. Determine the parameters of a PID controller for position control such that the closed-loop characteristic polynomial from disturbance torque to motor angle has the form:

$$(s^2 + 2 \zeta \omega_0 s + \omega_0^2) (s + p_0) = s^3 + (2 \zeta \omega_0 + p_0) s^2 + (\omega_0^2 + 2 \zeta \omega_0 p_0) s + \omega_0^2 p_0 \quad [5.20]$$

In other words, derive k_p , k_i , and k_d as functions of ω_0 , ζ , p_0 , K , and τ .

Solution:

0	1	2
---	---	---

8. We assume that the PID closed-loop system has a recovery response to a load disturbance step of the form of a first-order lag transfer function of time constant τ_i . Since the recovery to zero steady-state error is solely due to the integral part of the controller, the time constant τ_i can be expressed as a function of the pole p_0 , as shown below:

$$\tau_i = \frac{1}{p_0} \quad [5.21]$$

Design a PID controller so that recovery time constant τ_i is equal to (or less than) 0.5 seconds, as formulated below:

$$\tau_i = 0.5 \text{ [s]} \quad [5.22]$$

For such a system, keep the same closed-loop poles (defined by ζ and ω_0) as those determined in Section 5.5.3 during the design of the PD controller.

Evaluate the three PID gains k_p , k_i , and k_d respecting the above design requirements.

Solution:

0	1	2
---	---	---

5.5.6. Pre-Laboratory Results Summary Table

Table 5.2 should be completed before you come to the in-laboratory session to perform the experiment.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Value</i>	<i>Unit</i>
2.5.3. Nominal Process Parameters				
6.	Open-Loop Steady-State Gain	K		rad/(V.s)
6.	Open-Loop Time Constant	τ		s
5.5.3. PD Controller Design				
4.	Set-Point Weight On Derivative Part	b_{sd}		
5.	Desired Damping Ratio	ζ		
6.	Desired Undamped Natural Frequency	ω_0		rad/s
7.	Proportional Gain	k_p		V/rad
7.	Derivative Gain	k_d		V.s/rad
8.	Maximum Square Wave Amplitude	R_{max}		rad
5.5.4. Tracking Triangular Signals				
5.	Steady-State Error: PD Control, $b_{sd} = 0$	e_{ss_PD}		rad
5.	Steady-State Error: PD Control, $b_{sd} = 1$	e_{ss_PD}		rad
6.	Steady-State Error: PID Control	e_{ss_PID}		rad
5.5.5. Response To Load Disturbances				
5.	Equivalent Disturbance Torque Amplitude	T_{d0}		N.m
5.	Steady-State Angle: PD Control	θ_{ss_PD}		rad
6.	Steady-State Angle: PID Control	θ_{ss_PID}		rad
8.	Desired Closed-Loop Integrator Pole	p_0		rad/s
8.	Proportional Gain	k_p		V/rad
8.	Integral Gain	k_i		V/(rad.s)
8.	Derivative Gain	k_d		V.s/rad

Table 5.2 Position Control Pre-Laboratory Assignment Results

5.6. In-Laboratory Session

5.6.1. QICii Position Control Module

5.6.1.1. Module Description

The main tool for this laboratory is the front panel of the module entitled *Position Control* in the QICii software, which should be similar to the one shown in Figure 5.8.

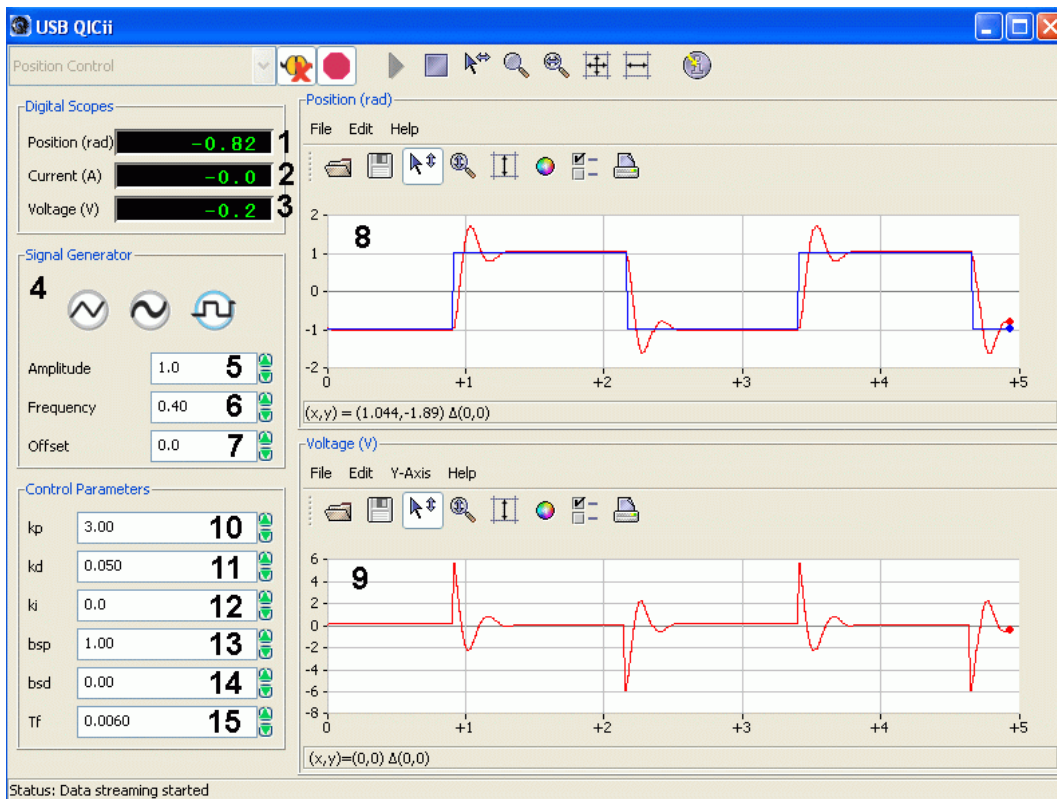


Figure 5.8 *Position Control* Module Of The QICii Software

As a quick module description, Table 5.3 lists and describes the main elements composing the QICii *Position Control* module user interface. Every element is uniquely identified through an ID number and located in Figure 5.8.

ID #	Label	Parameter	Description	Unit
1	Position	θ_m	Motor Output Position Numeric Display	rad

<i>ID #</i>	<i>Label</i>	<i>Parameter</i>	<i>Description</i>	<i>Unit</i>
2	Current	I_m	Motor Armature Current Numeric Display	A
3	Voltage	V_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	Position	θ_m	Scope With Actual (in red) And Reference (in blue) Angles	rad
9	Voltage	V_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 Set-Point Weight Input Box	
14	bsd	b_{sd}	Controller Derivative Set-Point Weight Input Box	
15	Tf	T_f	Time Constant of Filter for Measured Signal	s

Table 5.3 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).

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

$$h = 0.01 [s]$$

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

$$\omega_m = \frac{s \theta_m}{T_f s + 1}$$

where θ_m is the position of the motor shaft measured by the encoder.

5.6.1.2. Module Startup

To start and use the *Position Control* module, follow the steps described below:

- Step 1. Power up the DCMCT. LED2 should light up while LED3 should flash on and off repeatedly.
- Step 2. Press the *Reset* button on the QIC. Again, LED2 should remain on while LED3 should flash on and off repeatedly.
- Step 3. Press the DCMCT User Switch (i.e. pushbutton next to the two flashing LED's). LED2 and LED3 should both turn off.
- Step 4. Launch the USB QICii software and select *Position Control* in the drop-down menu.
- Step 5. Select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller. LED2 should light up while LED3 should still be off, and the controller should start running.

Note:

The drop-down menu will be disabled (i.e. is unavailable) while the controller is running. You must select the *Stop controller* button on top of the QICii window to stop the controller (this should also turn off LED2, and enable the drop-down menu again). Once the drop down menu is enabled again and the controller is stopped, you can select any one of the other controller experiments, if you want. After selecting some other controller experiment, you can once again select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller of your choice.

The default module parameters loaded after download are given in Table 5.4.

<i>Signal Type</i>	<i>Amplitude</i> [rad]	<i>Frequency</i> [Hz]	<i>Offset</i> [rad]	k_p [V/rad]	k_d [V.s/rad]	k_i [V/(rad.s)]
Square Wave	1.0	0.4	0.0	3.0	0.05	0.0
b_{sp}	b_{sd}	T_f [s]				
1.0	0.0	0.006				

Table 5.4 Default Parameters For The *Position Control* Module

5.6.2. Qualitative 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.

5.6.2.1. Pure Proportional (P) 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, 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 0.1 V/rad to start with. Ensure that the following parameters of the *Position Control* window, as displayed in Table 5.5, are set properly.

<i>Signal Type</i>	<i>Amplitude [rad]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad]</i>	<i>k_p [V/rad]</i>	<i>b_{sp}</i>
Square Wave	3	0.4	0	0.1	1

Table 5.5 Module Parameters For The Pure 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?

Solution:

0 1 2

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

Solution:

0 1 2

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

Solution:

0	1	2
---	---	---

Step 5. Apply a torque on the inertial load by gently touching and rotating it with your finger. Observe the effect on the position.

Solution:

0	1	2
---	---	---

Step 6. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:



0	1	2
---	---	---

5.6.2.2. Proportional And Derivative (PD) Control

The combination of proportional and derivative control will now be explored. Please 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. Set the parameters of the QICii module window as listed in Table 5.6. Set the integral gain to zero ($k_i = 0$).

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

Table 5.6 Module Parameters For The Proportional And Derivative Control Test

Step 2. Change the derivative gain by incremental steps of 0.01 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?

Solution:

0 1 2

Step 3. Determine a value of derivative gain which gives a set-point response without overshoot. Determine the settling time for the closed loop system.

Solution:

0 1 2

Step 4. Look up your notes from PI control of motor speed described in Chapter 3 and compare the process outputs and the control signals for speed and position control in response to a reference input. Use the assignment of Section 5.5.1 to explain your observations analytically.

Solution:

0	1	2
---	---	---

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

Solution:

0	1	2
---	---	---

5.6.3. 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 Section 5.5.3. In such a case, the controller parameters are determined by design based on a process model. This uses a mathematical model of the process and applies some method for control system design. Here, we are using a basic pole assignment method.

Please follow the steps below:

Step 1. Set the parameters of the QICii module window as described in Table 5.7.

<i>Signal Type</i>	<i>Frequency [Hz]</i>	<i>Offset [rad]</i>	<i>b_{sp}</i>	<i>T_f [s]</i>
Square Wave	0.4	0	1	0.006

Table 5.7 Module Parameters For PD Control Design To Given Specifications

Ensure that the integral gain is zero ($k_i = 0$) and set b_{sd} and both proportional and derivative gains to the values you calculated in Section 5.5.3 Questions 4 and 7.

Step 2. Adjust through iterative increments/decrements the square wave reference signal *Amplitude* so that the motor input voltage is close to and just below its saturation limit. Compare the obtained maximum square wave amplitude R_{max} with the one calculated in Section 5.5.3 Question 8.

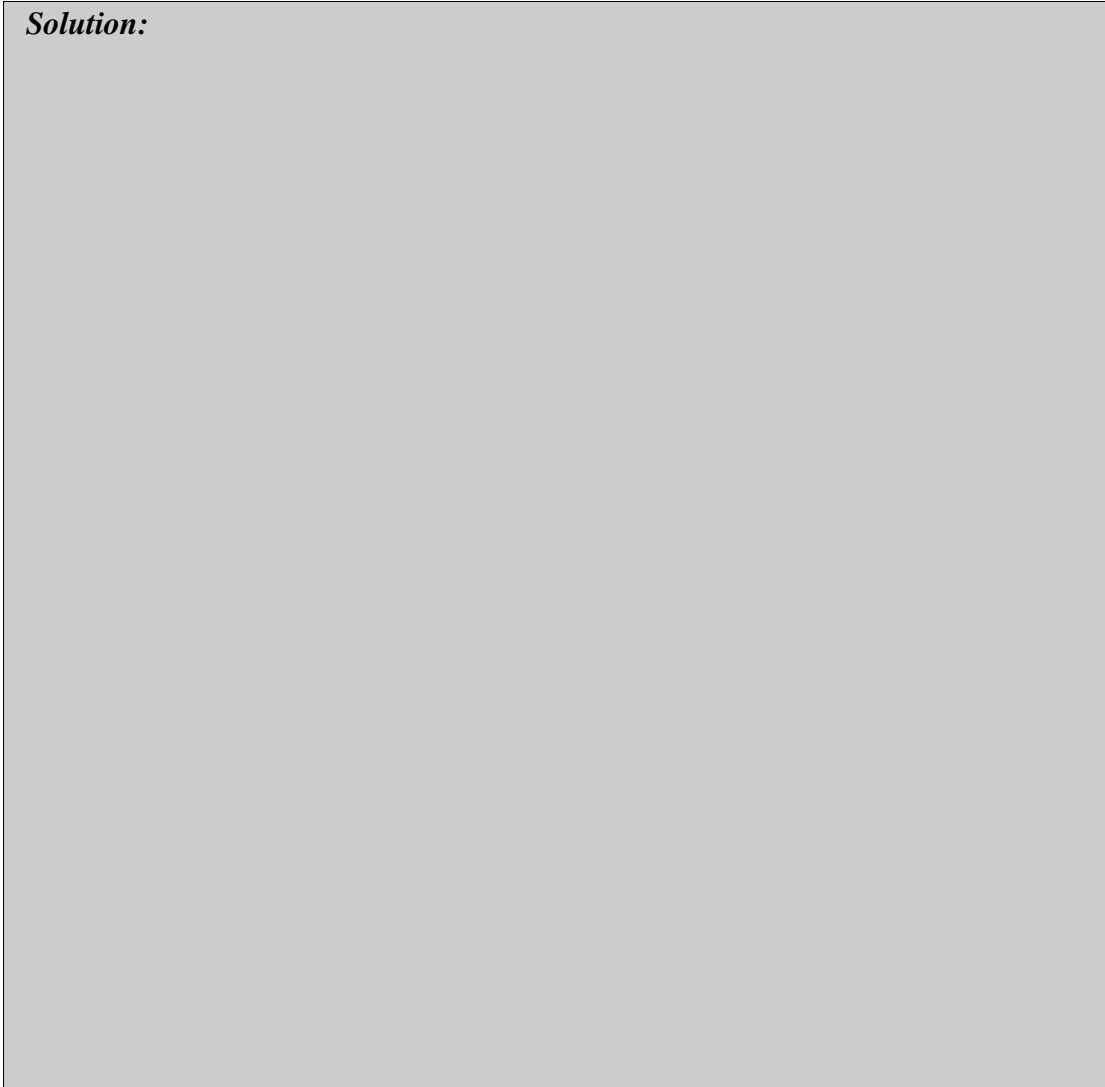
Measure the resulting Percent Overshoot (*PO*) and peak time t_p . Does the system's actual response meet the desired requirements? Do any adjustments need to be made to the values you calculated in Section 5.5.3 Questions 4 and 7?

Solution:

0 1 2

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

Solution:



0	1	2
---	---	---

5.6.4. Tracking Triangular Signals

Please refer to pre-laboratory assignment 5.5.4 for the theoretical background and results required to successfully carry out this in-lab procedure. This session carries out an experimental investigation of tracking of triangular signals. Please follow the steps below:

Step 1. Set the parameters of the QICii module window as listed in Table 5.8. Note that a triangular (a.k.a. sawtooth) reference signal should be selected.

<i>Signal Type</i>	<i>Amplitude [rad]</i>	<i>Frequency [Hz]</i>	<i>Offset [rad]</i>	<i>b_{sp}</i>	<i>T_f [s]</i>
Triangular Wave	20	0.4	0	1	0.006

Table 5.8 QICii Module Parameters For Tracking Test

Also set the controller gains to the PD controller parameters (i.e. k_p , k_d , and b_{sd}) determined in Section 5.5.3 and meeting given design specifications for a step reference input. Ensure that there is no integral action ($k_i = 0$).

Step 2. Observe how well the output tracks a triangular signal (in particular in terms of the tracking error)?

Solution:

0 1 2

Step 3. From the triangular wave specifications given in Table 5.8, calculate the slope R_0 of the obtained ramp signal.

Solution:

0 1 2

Step 4. Measure the actual asymptotic tracking error and compare with analytic estimate. Also change the controller proportional gain k_p by steps of ± 0.5 V/rad and explore its

effect on the tracking error. Summarize your observations in your report. Select some representative results, screen captures, and plots.

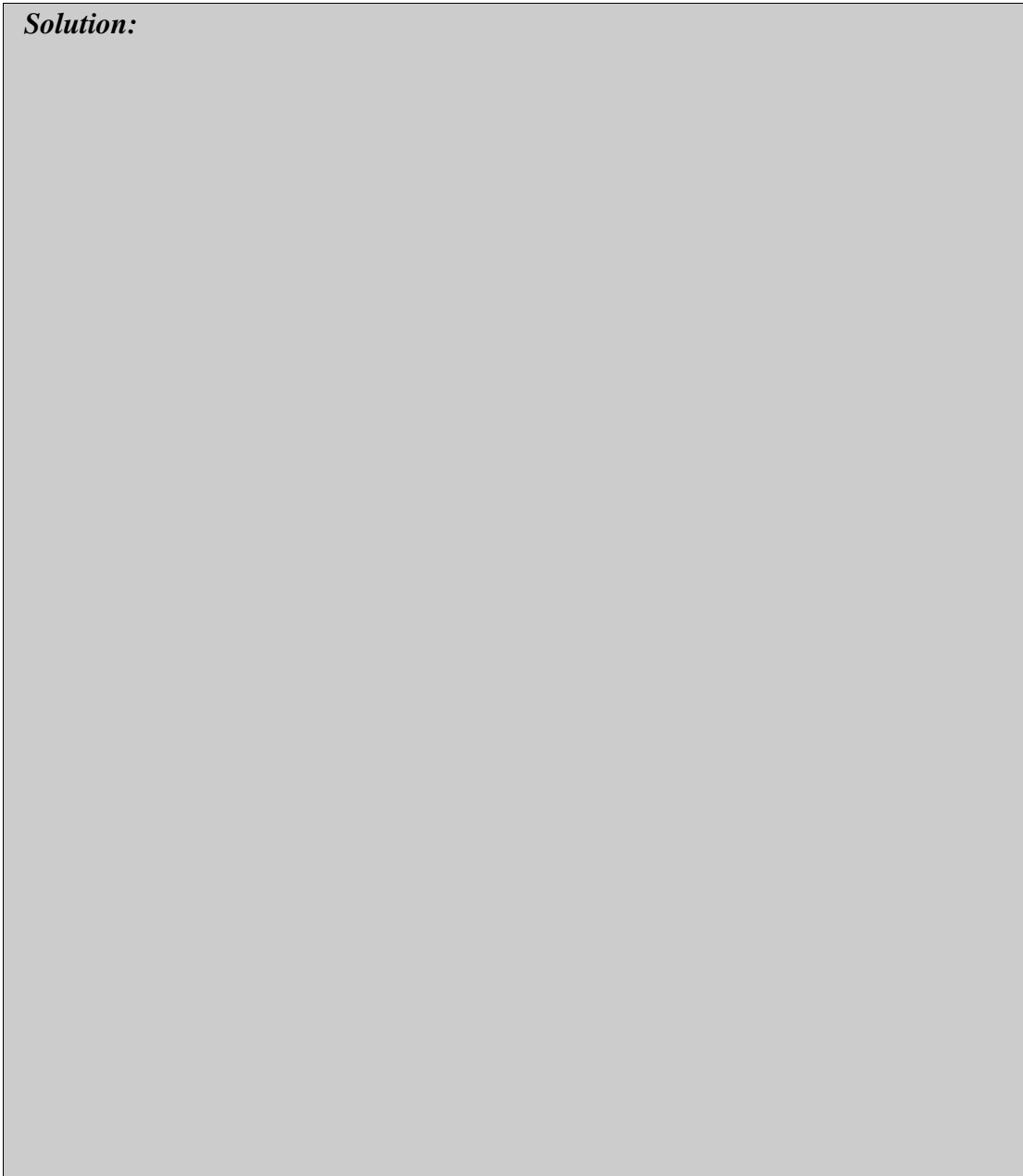
Solution:



0	1	2
---	---	---

Step 5. Keeping the same PD controller parameters, increase the set-point weight on derivative b_{sd} from 0 to 1. What is the effect on the position response and the tracking error? Summarize your observations in your report. Select some representative results, screen captures, and plots.

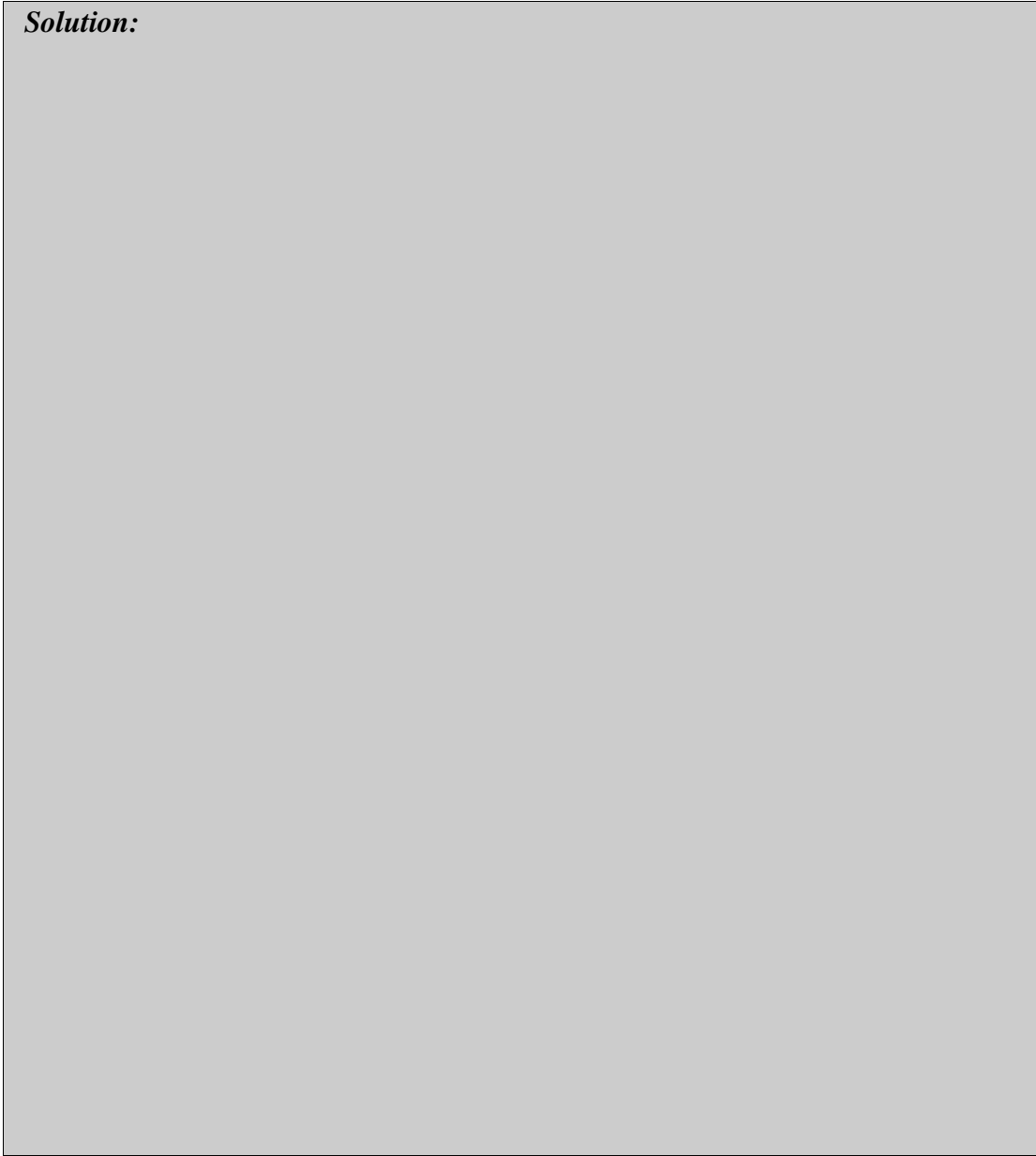
Solution:



0	1	2
---	---	---

Step 6. Use now the integral action of the controller by manually increasing the gain k_i by steps of 1.0 V/(rad.s). What is the effect on the tracking error in response to a ramp reference signal? Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:



0	1	2
---	---	---

5.6.5. Response To Load Disturbances

This session carries out an experimental investigation of the system response to load disturbances. Please refer to pre-laboratory assignment 5.5.5 as required for theoretical results.

5.6.5.1. Preamble: Simulated Load Disturbances

A disturbance torque can be simulated by applying an external voltage, V_{sd} , to the motor input. The DCMCT system uses this as a way to simulate a load disturbance in a repeatable manner, as illustrated in Figure 5.9. You can do this by triggering the User Switch on the DCMCT.

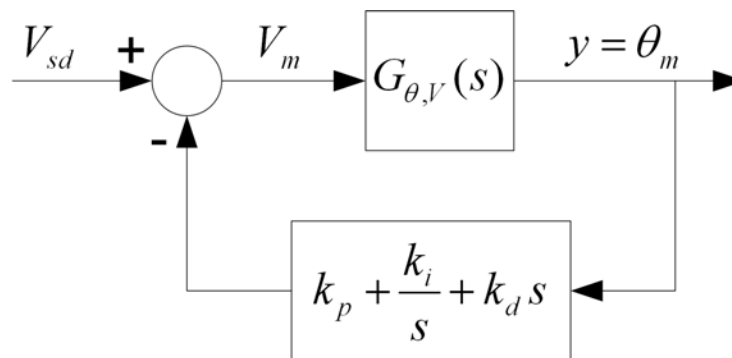


Figure 5.9 DCMCT Regulation Block Diagram With Simulated Load Disturbance

5.6.5.2. PD And PID Controllers

Please follow the steps below:

Step 1. The regulation problem at the starting position ($r = 0$ rad) is investigated. Set the QICii module parameters as described in Table 5.9.

<i>Amplitude [rad]</i>	<i>Offset [rad]</i>	b_{sp}	b_{sd}	$T_f [s]$
0	0	1	1	0.006

Table 5.9 Module Parameters For The Load Disturbance Test With PD Control

Also set the PD controller gains, k_p and k_d , to their values determined in Section 5.5.3. Ensure that there is no integral action ($k_i = 0$).

Step 2. Emulate a step in the load disturbance torque by pressing and holding (or releasing) the User Switch (next to the DCMCT two LED's). This applies the extra constant voltage $V_{sd} = V_{d0} = 3$ V to the motor input.

Step 3. What is the effect on the motor position? Measure the actual steady-state angle θ_{ss_PD} and compare with analytic estimate. Also change either the controller

proportional gain k_p or derivative gain k_d to investigate their effect on the system behaviour. Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:

0	1	2
---	---	---

Step 4. Keeping the same PD controller gains, introduce now the integral action of the controller by manually increasing the gain k_i by steps of 1.0 V/(rad.s). What is the effect on the load disturbance responses? Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:

0	1	2
---	---	---

Step 5. Set the PID controller gains, k_p , k_i , and k_d , to their design values as calculated in Section 5.5.5 Question 8. Measure the position response peak time t_p and peak overshoot θ_{p_PID} . Measure the actual time constant τ_i indicative of the 63 % decay from the angle peak overshoot. Are the design requirements satisfied? Summarize your observations in your report. Select some representative results, screen captures, and plots.

Solution:

0	1	2
---	---	---

5.6.6. In-Laboratory Results Summary Table

Table 5.10 should be completed using Table 5.2, which contains data from the pre-laboratory assignments, as well as experimental results obtained during the in-laboratory session.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Pre-Lab Value</i>	<i>In-Lab Result</i>	<i>Unit</i>
5.6.3. PD Controller Design					
	Maximum Square Wave Amplitude	R_{\max}			rad
	Percent Overshoot	PO			%
	Peak Time	t_p			s
5.6.4. Tracking Triangular Signals					
	Ramp Slope	R_0			rad/s
	Steady-State Error: PD Control, $b_{sd} = 0$	e_{ss_PD}			rad
	Steady-State Error: PD Control, $b_{sd} = 1$	e_{ss_PD}			rad
	Steady-State Error: PID Control	e_{ss_PID}			rad
5.6.5. Response To Load Disturbances					
2.	Steady-State Angle: PD Control	θ_{ss_PD}			rad
2.	Steady-State Angle: PID Control	θ_{ss_PID}			rad
2.	Overshoot Angle: PID Control	θ_{p_PID}	N/A		rad
2.	Peak Time: PID Control	t_p	N/A		s
2.	Integrator Time Constant	τ_i			s

Table 5.10 Position Control In-Laboratory Results

6. Haptic Interaction

6.1. Background And Laboratory Objectives

Control is much more than regulation and servoing. This laboratory illustrates some higher level control tasks by using haptics. The word haptics derives from the Greek *haptikos* which means to grasp or touch. According to Webster's dictionary, haptics is the science of sense and touch which has been an interesting research field in physiology for a long time. Lately, it has also been used in a different meaning in connection with computer interfaces.

The engineering interest in haptics has its origin in research on manual control in the aerospace industry in the fifties. A key problem was to investigate the behaviour of humans in tracking and steering tasks. With conventional rudders, the pilot gets useful feedback from the force he feels on the stick. This feedback was lost when the direct mechanical connection to the rudder surfaces was replaced by an electric wire *fly-by-wire* and the task became more difficult for the pilot. It was found that performance could be improved by introducing force feedback in the stick to provide sensory information to the pilot. The reason for this is that the signal pathway from touch is faster than the feedback from vision. When the forces are generated artificially, the question arises about what information should be fed back to the pilot. Another use of sensory feedback was made in flight simulators where it was found that motion cues could be generated with very limited motion. In this way, it was possible to imitate the seat-of-the-pants feeling which is useful for pilots.

Later force feedback was found to be very useful in robotics manipulators, particularly in teleoperation when manipulations are made over very long distances, space robotics is a nice example.

The word haptics started to be used in connection with human machine interfaces in computer science in the mid nineties where it was found to be useful to replace the mouse with other types of interfaces that reflected forces to the human. These interfaces are used in for hand-held tools for surgeons, robotic manipulators, computer games, and virtual

reality. The idea of using motion cues is also implemented extensively in amusement parks. Today, haptics is a rich research area by itself which attracts researchers from fields as different as control, computer graphics, neuroscience, psychology, robotics, advanced computer games, and virtual reality.

The objective of this experiment is to expose you to the potential use of haptics by two simple applications, namely:

- i. A Haptic Knob
- ii. Manual Control Using Haptics: Ball And Beam Application

6.2. Preparation And Pre-Requisites

You will also find that the understanding of PD control you gain in Chapter 5 is a useful ingredient for haptics. A pre-requisite to this laboratory is to have successfully completed the position laboratory described in Chapter 5. Before the lab, you may also be interested in reading about haptics at : <http://haptic.mech.nwu.edu/>

6.3. Nomenclature

The nomenclature for system modelling and control design is given in Table 6.1.

<i>Symbol</i>	<i>Description</i>	<i>Unit</i>
θ_m	Motor angle	rad
V_m	Voltage from the amplifier which drives the motor	V
T_d	Disturbance Torque externally applied to the inertial load	N.m
I_m	Motor Armature Current	A
k_m	Motor Torque Constant	N.m/A
R_m	Motor Armature Resistance	Ω
J_{eq}	Equivalent Moment Of Inertia Of Motor Rotor And The Load	kg.m ²
s	Laplace Operator	rad/s
h	Sampling Interval	s
t	Continuous Time	s
k_p	Proportional Gain	V/rad
k_d	Derivative Gain	V.s/rad
b_{sp}	Set-Point Weight On Proportional Part	
b_{sd}	Set-Point Weight On Derivative Part	
u	Control Signal	V
r	Reference Signal	rad
y	Measured Process Output	rad
δ	Haptic Knob transition threshold angle	rad

Table 6.1 System Modelling Nomenclature

6.4. Pre-Laboratory Assignment

6.4.1. Impedance Control

In Chapter 5, we have investigated position control problems where the task has been to bring the output to a desired steady-state (i.e. regulation problem) or to make the output follow a prescribed signal (i.e. servoing/tracking problem). There are many other control tasks that are interesting and useful. One example is impedance or also compliance control. According to Webster's dictionary, compliance is the ability of an object to yield elastically when a force is applied. Impedance control is used in many areas, robotics, manual control, and haptics. When robots are used for grinding and machining, it is often highly desirable to control the tool so that the tool is pushed with a constant force towards the work piece. In manual control for precision tasks, a joy stick is often provided with force feedback. This makes it possible to give direct feedback to the operator. Impedance control is also extensively used in haptics as is illustrated in this Chapter.

Simple aspects of impedance control can be illustrated using the inertial load of the motor as is illustrated by the following example. Please answer the following questions.

1. Applying mechanical and electrical first principles, derive in the Laplace domain a second-order differential equation in Θ_m , describing the open-loop system behaviour as a function of the input voltage V_m and the torque applied to the inertial load T_d .

Solution:

0	1	2
---	---	---

2. Considering a constant reference position signal R , the control voltage, V_m , due to Proportional-Plus-Derivative (PD) feedback can be expressed in the Laplace domain as follows (assuming $b_{sp} = b_{sd} = 1$):

$$V_m = k_p (R - \Theta_m) - k_d \Theta_m s \quad [6.1]$$

Taking Equation [6.1] into account, determine the resulting second-order differential equation in Θ_m of the obtained PD closed-loop system.

Solution:

0	1	2
---	---	---

3. Since the feedback system regulates a constant reference angle R , consider the applied torque T_d as the system input. It is reminded that a mechanical spring-dashpot-mass model describing the motion of the equivalent mass, or inertia, J when a torque T_d is applied can be described by the following second-order differential equation:

$$J \ddot{\theta} + C \dot{\theta} + k \theta = T_d \quad [6.2]$$

where C and k are the damping and stiffness coefficients, respectively, and θ is the displacement variable.

Using PD control, determine the DCMCT system equivalent damping and stiffness coefficients, C_s and k_s , respectively.

Solution:

0	1	2
---	---	---

4. Consider the following PD controller gains:

$$k_p = 0.025 \left[\frac{V}{deg} \right] \quad \text{and} \quad k_d = 0.0003 \left[\frac{V s}{deg} \right] \quad [6.3]$$

Using the system parameters given in Table A.2 of Appendix A, evaluate C_s and k_s .

Solution:

0	1	2
---	---	---

6.4.2. Pre-Laboratory Results Summary Table

Table 6.2 should be completed before you come to the in-laboratory session to perform the experiment.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Value</i>	<i>Unit</i>
6.4.1.	<i>Impedance Control</i>			
	Equivalent Damping Coefficient	C_s		N.m.s/rad
	Equivalent Stiffness Coefficient	k_s		N.m/rad

Table 6.2 Haptic Interaction Pre-Laboratory Assignment Results

6.5. In-Laboratory Session

6.5.1. QICii Modules Description

The motor position is measured by an encoder generating 4096 counts per revolution. The implemented digital controller in the QIC runs at 100 Hz. Thus the sampling interval is:

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

6.5.1.1. Haptic Knob Module

The main tool for the *Haptic Knob* laboratory is the front panel of the module entitled *Haptic Knob* in the QICii software, which should be similar to the one shown in Figure 6.1.

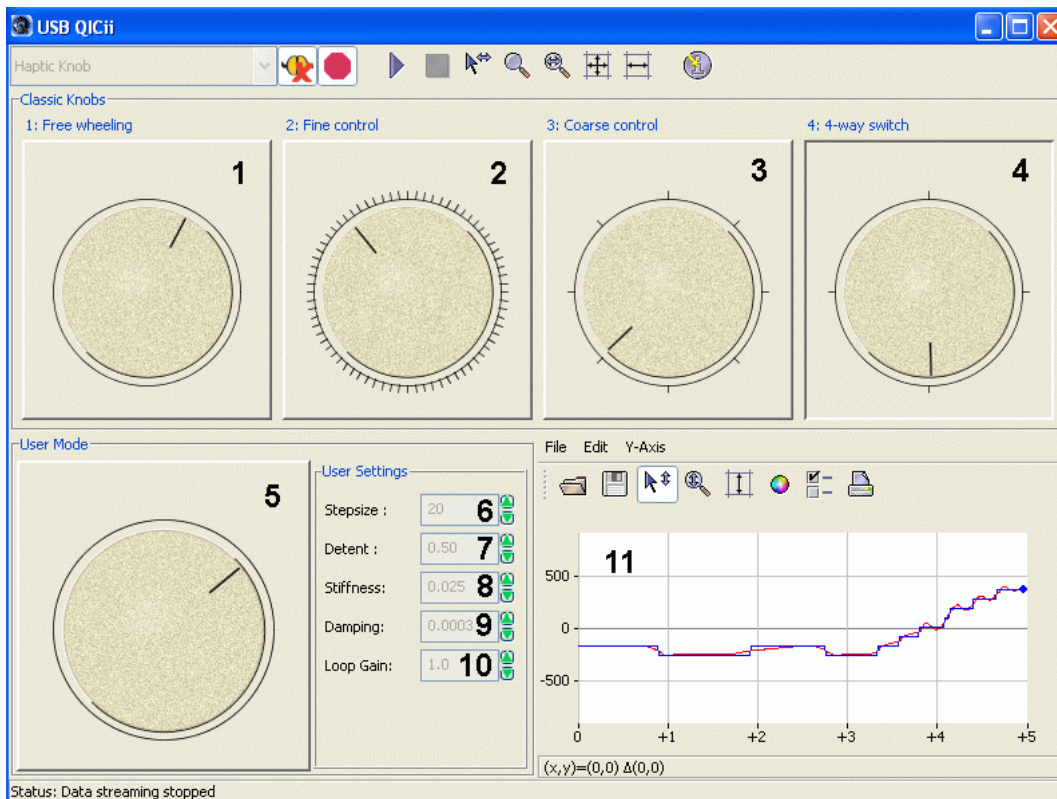


Figure 6.1 *Haptic Knob* Module Of The QICii Software

As a quick module description, Table 6.3 lists and describes the main elements composing the QICii *Haptic Knob* module user interface. Every element is uniquely identified through an ID number and located in Figure 6.1.

<i>ID #</i>	<i>Label</i>	<i>Description</i>
1	1: Free Wheeling	Classic Knob: Setting #1
2	2: Fine Control	Classic Knob: Setting #2
3	3: Coarse Control	Classic Knob: Setting #3
4	4: 4-Way Switch	Classic Knob: Setting #4
5	User Mode	User Mode Knob defined by the <i>User Settings</i> parameters
6	Stepsize	User Setting: <i>Stepsize</i>
7	Detent	User Setting: <i>Detent</i>
8	Stiffness	User Setting: <i>Stiffness</i>
9	Damping	User Setting: <i>Damping</i>
10	Loop Gain	User Setting: <i>Loop Gain</i>
11	Position	Scope With Actual (in red) And Reference (in blue) Angles

Table 6.3 QICii *Haptic Knob* Module Nomenclature

The *Haptic Knob* module makes the motor with the inertial load behave like a wheel with mechanical notches. If no force is applied, the wheel returns to the closest notch automatically. If sufficient force is applied the wheel moves to the next notch. The distance between the notches is called the *Stepsize*.

The *Haptic Knob* module program runs the process in closed-loop with the motor reference position angle generated on the QIC to reflect the desired knob feel.

6.5.1.2. Ball and Beam Module

The *Ball and Beam* module program runs the process in closed-loop with the motor reference position angle generated on the QIC to reflect the desired ball and beam feel.

The main tool for the *Ball and Beam* laboratory is the front panel of the module entitled *Ball and Beam* in the QICii software, which should be similar to the one shown in Figure 6.2.

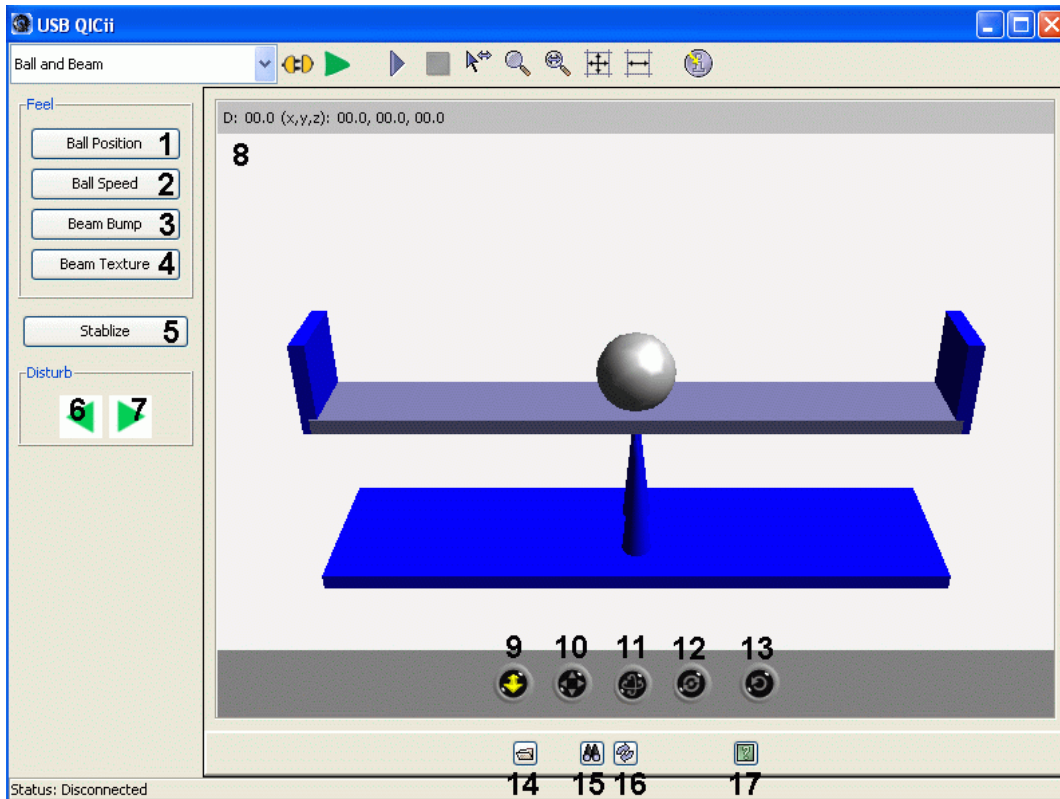


Figure 6.2 *Ball and Beam* Module Of The QICii Software

As a quick module description, Table 6.4 lists and describes the main elements composing the QICii *Ball and Beam* module user interface. Every element is uniquely identified through an ID number and located in Figure 6.2.

<i>ID #</i>	<i>Label</i>	<i>Description</i>
1	Ball Position	Triggerring this option gives the sensation of the ball position along the beam. The force on the inertial wheel is proportional to ball position.
2	Ball Speed	Triggerring this option gives the sensation of the ball speed along the beam. The force on the inertial wheel is proportional to the velocity of the ball. This option also lets you feel the ball hitting to a stop the beam end.
3	Beam Bump	Triggerring this option lets you feel the beam hitting the table
4	Beam Texture	Triggerring this option lets you feel the beam granularity
5	Stabilize	Triggerring this option implements a stable haptic feedback controller which automatically centers the ball
6, 7	Disturb Arrows	Pressing one of these two arrows causes a disturbance to the ball: just like giving it a push to one side
8	3D Viewer	<i>3D Viewer</i> display window. Please refer to the <i>3D Viewer</i> description provided in Table 6.5.

Table 6.4 QICii *Ball And Beam* Module Nomenclature

After controller download, all four *Feel* buttons are off, by default.

Table 6.5 lists and describes the main elements composing the *3D Viewer* of the *QICii Ball and Beam* module user interface. Every element is uniquely identified through an ID number and located in Figure 6.2.

<i>ID #</i>	<i>Button Name</i>	<i>Description</i>
9	Plan	Zoom in/out and horizontal displacement (along X) by clicking and dragging the mouse
10	Pan	Horizontal and vertical displacement (along X-Y) by clicking and dragging the mouse
11	Turn	Turn the orientation by clicking and dragging the mouse
12	Rotate	Rotate about the X, Y, and Z axes by clicking and dragging the mouse
13	Restore	Restore the default original 3D view
14	Open	Open VRML 97 (*.wrl), 3D Studio (*.3ds), 3D (*.q3d), and AC3D (*.ac) files
15	Enable Viewer	Enable/Disable the 3D Viewer
16	Undo	Undo change
17	Help	Help file

Table 6.5 QICii *Ball and Beam* Module: *3D Viewer* Nomenclature

6.5.2. Module Startup

To start and use the *Position Control*, *Haptic Knob*, or *Ball and Beam* module, follow the steps described below:

- Step 1. Power up the DCMCT. LED2 should light up while LED3 should flash on and off repeatedly.
- Step 2. Press the *Reset* button on the QIC. Again, LED2 should remain on while LED3 should flash on and off repeatedly.
- Step 3. Press the DCMCT User Switch (i.e. pushbutton next to the two flashing LED's). LED2 and LED3 should both turn off.
- Step 4. Launch the USB QICii software and select *Position Control*, *Haptic Knob*, or *Ball and Beam* in the drop-down menu.
- Step 5. Select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller. LED2 should light up while LED3 should still be off, and the controller should start running.

Note:

The drop-down menu will be disabled (i.e. is unavailable) while the controller is running. You must select the *Stop controller* button on top of the QICii window to stop the controller (this should also turn off LED2, and enable the drop-down menu again). Once the drop down menu is enabled again and the controller is stopped, you can select any one of the other controller experiments, if you want. After selecting some other controller experiment, you can once again select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller of your choice.

6.5.3. Impedance Control

This session carries out an experimental investigation of the system impedance with a PD controller. Please refer to the pre-laboratory assignment of Section 6.4.1, as required, for theoretical results.

Please follow the steps below.

- Step 1. Run the *Position Control* controller on the QIC and connect to it using the corresponding QICii module. You should have a window similar to the one displayed in Figure 5.10 of Chapter 5.
- Step 2. The regulation problem at the starting position ($r = 0$ rad) is investigated. Set the QICii module parameters as described in Table 6.6.

<i>Amplitude [rad]</i>	<i>Offset [rad]</i>	b_{sp}	b_{sd}	k_i [V/(rad.s)]	T_f [s]
0	0	1	1	0	0.006

Table 6.6 Module Parameters For The Impedance Control Test

- Step 3. Run the system open-loop by setting $k_p = k_d = 0$. Manually spin the inertial load slow and fast. Feel the reaction forces on the wheel. Document your observations and explain your findings analytically using the simple mathematical models derived in Section 6.4.1.

Solution:

0 1 2

- Step 4. Introduce a PD controller for the motor angle by setting the PD controller gains, k_p and k_d , to their values determined in Section 5.5.3 of Chapter 5. Ensure that there is no integral action ($k_i = 0$) and that the reference position is constant. Manually twist the inertial load and feel the torque applied to the wheel as you change the angle. Do you feel the "stiffness" in the motor shaft? What is the effect of changing the proportional feedback gain k_p ? What is the effect of changing derivative gain k_d ? And the reference r ? Document your observations and explain your findings analytically using the considerations carried out in Section 6.4.1.

Solution:



0	1	2
---	---	---

Step 5. You will find some interesting applications of this configuration properties in the following Sections.

6.5.4. Haptic Knob

This session gives an experimental introduction on how a DC motor can be made to behave like a mechanical knob or switch. The haptic knob motor controller consists of a PD position controller and a state machine to perform the logical operations.

6.5.4.1. Preamble: Haptic Knob Implementation

It is reminded that the *Haptic Knob* module makes the motor with the inertial load behave like a wheel with mechanical notches. The haptic knob is implemented by combining a PD controller for motor angle, θ_m , with a finite state machine which has three states labeled A (Anticlockwise), M (Middle), and C (Clockwise). The corresponding Petri net is represented in Figure 6.3.

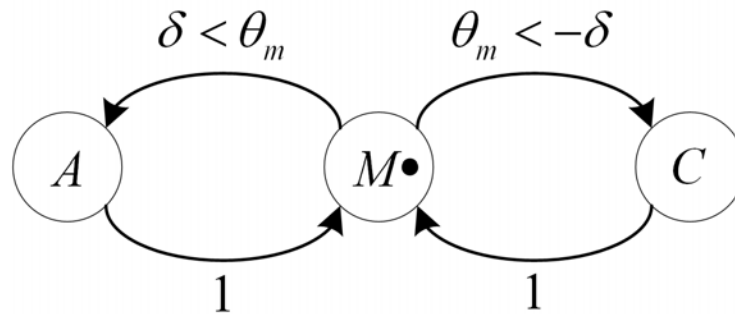


Figure 6.3 Finite State Machine Implementing The Haptic Knob

The meaning of the three states shown in Figure 6.3 is described in Table 6.7.

<i>State</i>	<i>Description</i>
<i>M</i>	In this state, the knob is in the Middle of a notch.
<i>A</i>	This state is reached when the knob is an angle δ Anticlockwise from the middle of a notch. Assuming that the positive sense of rotation is anticlockwise, the transition to this state can be expressed by: $\delta < \theta_m$. Once A is reached, the set-point for the motor angle is increased by <i>Stepsize</i> and the state is reset to <i>M</i> .
<i>C</i>	This state is reached when the knob is an angle δ Clockwise from the middle of a notch. The transition to this state can be expressed by the following relationship: $\theta_m < -\delta$. Once C is reached, the set-point for the motor angle is decreased by <i>Stepsize</i> and the state is reset to <i>M</i> .

Table 6.7 Haptic Knob State Description

In the the *Haptic Knob* module implementation, the angle transition threshold, δ , is defined as the product of the *Detent* and *Stepsize* parameters, as expressed below:

$$\delta = \textit{Detent} \textit{Stepsize} \quad [6.4]$$

6.5.4.2. Experimental Procedure: Haptic Knob Exploration

Please follow the steps below.

- Step 1. Run the *Haptic Knob* controller on the QIC and connect to it using the corresponding QICii module. You should have a window similar to the one displayed in Figure 6.1.
- Step 2. Rotate the dial (or inertial load) on the motor shaft. You feel clicks at a fixed increment. This is an implementation of a haptic knob. By moving the mouse cursor on top of one of the four *Classic Knobs*, displayed in the QICii window, and clicking on it, you can change the feel of the knob. Try all four *Classic Knobs* pre-defined settings.
- Step 3. In the QICii window, click on the *User Mode* dial and enter values for *Stepsize*, *Detent*, and *Loop Gain* setting parameters. By changing these variables, you can change the feel of the knob to meet a desired specification. Manipulate the three parameter values and get a feel for what they do.
- Step 4. Can you decipher the settings for the four pre-programmed *Classic Knobs*? Determine the values used for the *Classic Knobs* and fill up Table 6.8.

<i>Classic Knob:</i>	<i>Stepsize</i>	<i>Detent</i>	<i>Stiffness</i>	<i>Damping</i>	<i>Loop Gain</i>
1) <i>Free Wheeling</i>			0.025	0.0003	
2) <i>Fine Control</i>			0.025	0.0003	
3) <i>Coarse Control</i>			0.025	0.0003	
4) <i>4-Way Switch</i>			0.025	0.0003	

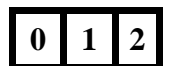


Table 6.8 *Classic Knob* Setting Parameters

6.5.5. Haptic Ball And Beam

This session gives an experimental introduction on how the DCMCT motor shaft can be used to command the virtual beam angle and balance the ball atop of it. Force feedback is used to feed different signals back to the inertial load. The graphical representation is in 3D and runs in real-time.

6.5.5.1. Preamble: Haptic Ball And Beam Implementation

In the *Ball and Beam* module, the voltage to the motor is given by the angle of the inertial load. There is a force feedback on the inertial wheel and you can select to feel a variety of effects from the simulation via the motor shaft and inertial load. The force can be chosen in different ways through the buttons on the interface. There are four buttons labeled *Ball Position*, *Ball Speed*, *Beam Bump*, and *Beam Texture*. The function of each one of the "force feedback" buttons is described in Table 6.4.

6.5.5.2. Experimental Procedure: Ball And Beam Exploration

Please follow the steps below.

Step 1. Run the *Ball and Beam* controller on the QIC and connect to it using the corresponding QICii module. You should have a window similar to the one displayed in Figure 6.2. The Three-Dimensional (3D) ball and beam view can be manipulated by clicking on the buttons at the bottom of the 3D Viewer and dragging the mouse cursor. The *3D Viewer* is described in Table 6.5.

Step 2. Manually rotate the inertial load on the DCMCT. You will see the beam rotating accordingly and the ball starting to roll. You can control the beam tilt using the DCMCT inertial load. You can also select to feel a variety of sensory feedback signals. The different *Feel* options are available through push buttons located on the left-hand side of the QICii window. Their individual action is described in Table 6.4.

Step 3. Click off all *Feel* options and apply a disturbance using the left and right arrows. Try to balance the ball on the beam and bring it back to the mid-stroke position. Is that difficult at all? Can you do it with your eyes closed?

Solution:

0	1	2
---	---	---

Step 4. Now turn on the option to feel the ball position. Try to balance it now. Does that help?

Solution:

0	1	2
---	---	---

Step 5. Try the *Stabilize* option. Can you comment on this option versus the others?

Solution:

0	1	2
---	---	---

Step 6. Turn off the *Stabilize* option. After some familiarization with the haptic ball and beam system, **disable the 3D Viewer**. Do so by clicking on the binoculars button which will turn off the 3D display. Turn on as many *Feel* options as necessary and try to balance the ball using haptic feedback only. Can you do it?

Solution:

0	1	2
---	---	---

Step 7. Comment on observed limitations, pitfalls, and suggest potential solutions.

Solution:

0	1	2
---	---	---

6.5.6. In-Laboratory Results Summary Table

Table 6.9 should be completed using the experimental results obtained during the in-laboratory session.

<i>Section</i>	<i>Description</i>	<i>Symbol</i>	<i>Pre-Lab Value</i>	<i>In-Lab Result</i>	<i>Unit</i>
6.5.4.	<i>Haptic Knob</i>				
	1) Free Wheeling Knob: <i>Stepsize</i>	<i>1-Stepsize</i>	N/A		°
	1) Free Wheeling Knob: <i>Detent</i>	<i>1-Detent</i>	N/A		
	1) Free Wheeling Knob: <i>Loop Gain</i>	<i>1-Loop Gain</i>	N/A		
	2) Fine Control Knob: <i>Stepsize</i>	<i>2-Stepsize</i>	N/A		°
	2) Fine Control Knob: <i>Detent</i>	<i>2-Detent</i>	N/A		
	2) Fine Control Knob: <i>Loop Gain</i>	<i>2-Loop Gain</i>	N/A		
	3) Coarse Control Knob: <i>Stepsize</i>	<i>3-Stepsize</i>	N/A		°
	3) Coarse Control Knob: <i>Detent</i>	<i>3-Detent</i>	N/A		
	3) Coarse Control Knob: <i>Loop Gain</i>	<i>3-Loop Gain</i>	N/A		
	4) 4-Way Switch Knob: <i>Stepsize</i>	<i>4-Stepsize</i>	N/A		°
	4) 4-Way Switch Knob: <i>Detent</i>	<i>4-Detent</i>	N/A		
	4) 4-Way Switch Knob: <i>Loop Gain</i>	<i>4-Loop Gain</i>	N/A		

Table 6.9 Haptic Interaction In-Laboratory Results

Appendix A. DCMCT/USB QICii Hardware Guide

A.1. DCMCT System Capabilities

The DC Motor Control Trainer (DCMCT) is a versatile unit designed to teach and demonstrate the fundamentals of motor servo control in a variety of ways. The system can readily be configured to control motor position and speed. In particular with respect to this workbook, the system can be used to teach PID control fundamentals. This is achieved when the DCMCT is used in conjunction with the Quanser QIC Processor Core and the QICii software. The QICii (i.e. QIC interactive interface) consists of pre-packaged ready-to-use experiments in Modelling, Speed Control, Robustness, Position Control, and Haptic Explorations. Course material is supplied allowing the instructor to teach these topics with minimal setup time. QICii also introduces the student to the concepts of haptic devices. Two haptic implementations are supplied:

i) **Haptic Knobs**

The haptic knob experiments can be performed through control of detent, stiffness, step size, loop gain, and damping.

ii) **Haptic Ball and Beam**

This is a Java component that simulates a ball and beam experiment dynamics on a PC. It also provides the user with a realtime graphic simulation such that the user can command the beam angle via the DCMCT and program the “feel” into the knob to sense system dynamics. This system demonstrates the positive effect that haptics has on operation in virtual or remote environments. One could balance the ball without looking at the simulation: simply feel.

Apart from the above mode of application, the system can easily be used to teach:

1) PC-Based Digital Control

This is done using a PC with realtime control capabilities and a Hardware-In-the-Loop (HIL) board. The user can either program the PC to perform the realtime control or use Simulink to generate code and run it in realtime on a PC. The DCMCT system is fully compatible with all of the Quanser's HIL boards as well as National Instruments' E-series boards and the dSPACE's DS1104 board (for DSP-based control). The system is fully controllable using Quanser's WinCon or SLX RT as well as The MathWorks' RTWT or xPC Target, NI's LabVIEW, and dSPACE's ControlDesk. In this case, the HIL board is wired to the RCA and DIN connectors on the DCMCT.

2) Embedded Microcontroller Control

The Quanser QIC Processor Core can readily be plugged into the DCMCT unit through the supplied socket. The signal connections are automatically made when the QIC is installed. The user may then write his or her own embedded code and program Microchip's PIC 18F4550 to control the motor without a PC. Alternatively, Quanser supplies QICii with pre-packaged experiments to perform control experimentation tuning and data collection.

3) Analog Feedback Control

This can be done using the Quanser Analog Plant Simulator (APS) or any other analog computer including OP-AMP circuits implemented on breadboards. A breadboard is available on the DCMCT system for students to implement their own analog controllers. In this case, the digital measurement from the encoder cannot be used.

For information on how to use the DCMCT in applications other than QICii, please refer to the *DCMCT Technical Manual*.

A.2. General Overview

A.2.1. System Nomenclature

The photograph in Figure A.1 shows an overview and the general layout of the DCMCT system with breadboard and QIC options.

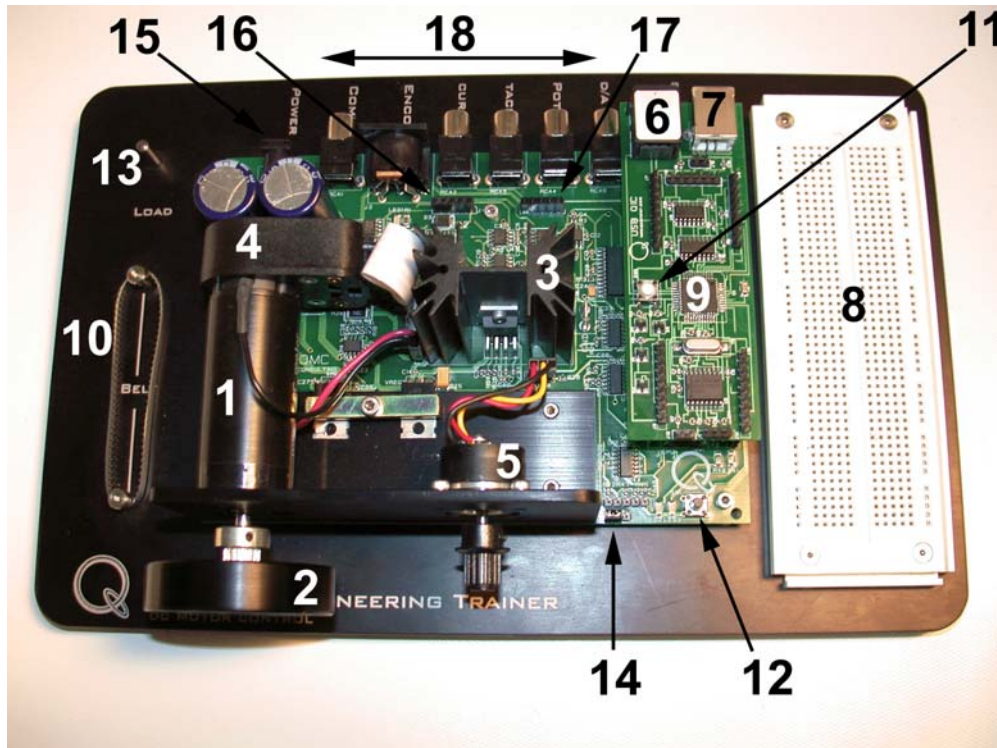


Figure A.1 DCMCT General Layout

The DCMCT components located and identified by a unique ID number in Figure A.1 are listed in Table A.1. Table A.1 also describes the integration of the various methods one could control the motor with, as previously mentioned in Section A.1.

<i>ID #</i>	<i>Description</i>
1	Maxon DC Motor
2	Removable Inertial Load
3	Linear Power Amplifier
4	High Resolution Optical Encoder
5	Ball Bearing Servo Potentiometer
6	RJ11 Port on QIC: for downloading firmware using a compatible programming device
7	USB Port on QIC: for online tuning and plotting (used by QICii)
8	Breadboard Option: to implement controllers with your own circuits

<i>ID #</i>	<i>Description</i>
9	Embedded/Portable Option: QIC installs in this socket to perform embedded control in place of PC-based control
10	Removable Belt: to drive the potentiometer
11	PIC Reset Switch
12	User Switch: Momentary Action Pushbutton Switch For Manual Interaction
13	Inertial Load Storage Pin
14	Jumper J6: to switch between HIL and QIC use
15	6-mm Power Jack
16	Power Supply Header: J4
17	Analog Signals Header: J11
18	<ul style="list-style-type: none"> i. PC Interface Option: this is implemented by using one of Quanser's HIL boards, NI's E-Series boards, or the dSPACE DS1104 board ii. Analog Controller Option: to implement controllers using Quanser's Analog Plant Simulator

Table A.1 DCMCT Component Nomenclature

A.2.2. System Schematic

A schematic of the DCMCT system is provided in Figure A.2.

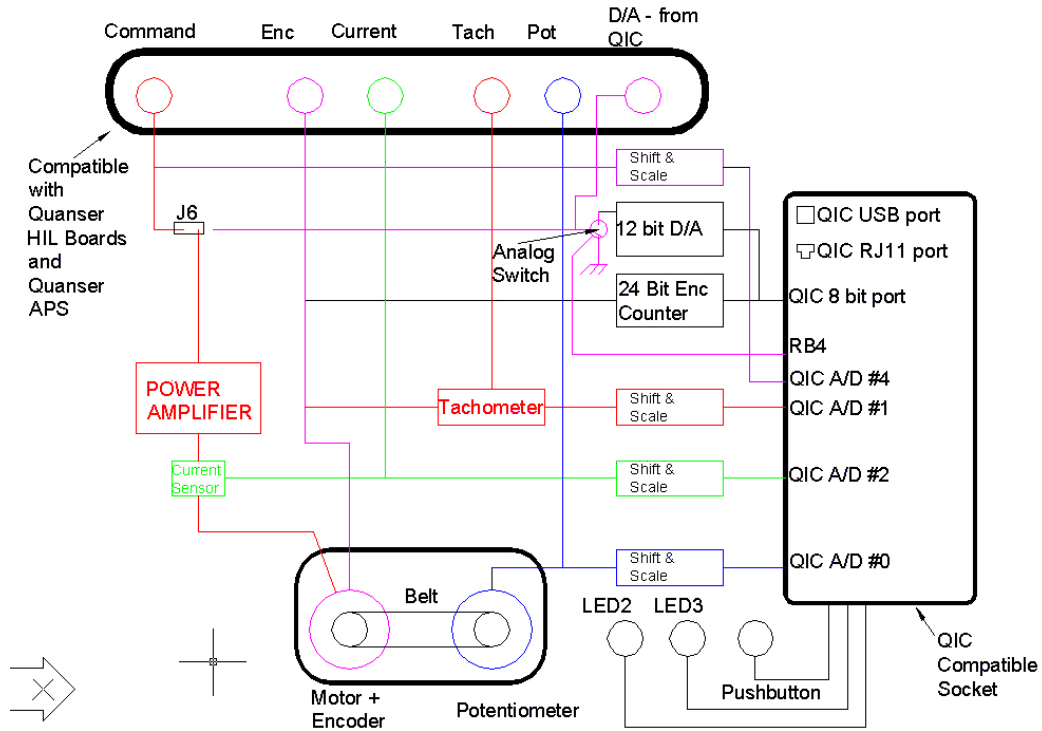


Figure A.2 DCMCT Schematic

A.2.3. Component Description

This Section provides a description of the individual elements comprising the full DCMCT system.

A.2.3.1. Maxon DC Motor

The DC motor is a high quality 18-Watt motor of Maxon brand. This is a graphite brush DC motor with a low inertia rotor. It has zero cogging and very low unloaded running friction.

A.2.3.2. Linear Power Amplifier

A linear 15-V 1.5-A power amplifier is used to drive the motor. The input to the amplifier can be configured to be either the voltage input at the RCA jack labelled *COMMAND* or the output of the built-in Digital-To-Analog converter (i.e. D/A). The built-in D/A can only be used if a QIC board is connected to the system and the appropriate jumper (i.e. J6) properly configured.

A.2.3.3. QIC Compatible Socket

The Quanser QIC processor core board can be plugged into the QIC compatible socket, which is depicted in Figure A.3. This is designed to enable one to perform closed-loop control using the QIC microcontroller.

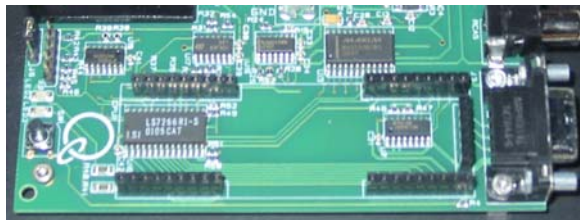


Figure A.3 QIC Compatible Socket

A.2.3.4. QIC Processor Core Board

The QIC processor core board is the Quanser board that contains Microchip's PIC 18F4550 microcontroller. It is shown in Figure A.4. For embedded microcontroller control applications, this QIC board needs to be installed in the corresponding QIC compatible socket on the DCMCT. The QIC can process the signals that are derived from the DCMCT sensors, command the power amplifier, drive two LED's, and read the state of the User Switch. The User Switch and two LED's are useful when performing certain types of experiments supplied with the QICii software. Note that the QIC *Reset* button is also shown in Figure A.4.

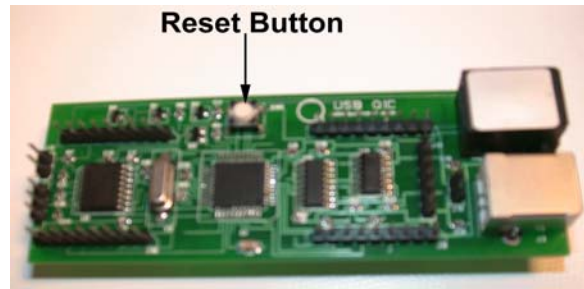


Figure A.4 QIC Processor Core Board

A.2.3.5. Analog Current Measurement: Current Sense Resistor

A series load resistor of 0.1Ω is connected to the output of the linear amplifier. The signal is amplified internally to result in a sensitivity of 1.8 V/A . The obtained current measurement signal is available at the RCA jack labeled *CURRENT*. Such a current measurement can be used to either monitor the current or in a feedback loop to control the current in the motor. Current control is an effective way of eliminating the effects of back-EMF as well as a means of achieving force and torque control. This signal is level-shifted and scaled, and then also made available at one of the QIC analog inputs (i.e. A/D's) so that it can be used by a QIC-based controller.

A.2.3.6. Digital Position Measurement: Optical Encoder

Digital position measurement is obtained by using a high resolution quadrature optical encoder. This optical encoder is directly mounted to the rear of the motor. The encoder signals are available at the 5-pin DIN plug, named *ENCODER*, and can be used by a HIL board. These encoder signals are also fed to an encoder counter integrated circuit that interfaces with the QIC in order to perform encoder feedback control using the QIC.

A.2.3.7. Analog Speed Measurement: Tachometer

An analog signal proportional to motor speed is available at the RCA jack labeled *TACH*. It is digitally derived from the encoder signal. It has the range of $\pm 5 \text{ V}$. Once scaled and level-shifted to the range of $0\text{-}5 \text{ Volts}$, the tachometer signal is also made available to one of the QIC A/D's so that it can be used by a QIC-based controller.

A.2.3.8. Analog Position Measurement: Potentiometer

The ball bearing servo potentiometer can be coupled via a toothed belt to the motor shaft in order to implement analog position feedback control. The potentiometer signal is available at the RCA jack labeled *POT* and can be used by an analog computer or a HIL board on a PC. This analog signal is in the range of $\pm 5 \text{ V}$. Once level-shifted and scaled to $0\text{-}5 \text{ V}$, the potentiometer signal is also made available to one of the QIC analog inputs so that it can be used by a QIC-based controller.

**CAUTION:**

The potentiometer belt should be removed in order to eliminate the effects of extra friction when running speed control and encoder position feedback experiments as well as to extend the life of the potentiometer. It is especially recommended that the potentiometer belt be removed while running speed control experiments. Although the potentiometer is rated for 10 million revolutions, its life will quickly diminish when running for long periods of time. Running the potentiometer at 2000 RPM, for example, would reduce its life expectancy to 83 hours! Potentiometers are typically used in position control systems and, as such, are not expected to continuously turn at high speeds. For QICii applications, the potentiometer is not used. Please remove the belt.

A.2.3.9. A Wall Transformer

The DCMCT system is provided with a wall transformer as a AC power supply. This is to deliver the required AC voltage to the DCMCT board. It connects to the 6-mm jack labeled *POWER*.

**CAUTION:**

Please make sure you use the right type of transformer or you will damage the system. It should supply 12 to 15 VAC and be rated for at least 2 A.

A.2.3.10. Built-In Power Supply

A built-in power supply converts the 15 VAC available at the *POWER* 6-mm jack to ± 20 VDC. The ± 20 VDC is regulated as well to supply ± 5 VDC and ± 12 VDC. These power supplies are available on the J4 header, whose location on the main board is shown in Figure A.1 by component #16. The power supply header J4 can be used for external circuitry. Its schematic is illustrated in Figure A.5.



Figure A.5 J4 Connector Schematic

A.2.3.11. A12-Bit Digital-To-Analog Converter (D/A)

One 12-bit Digital-To-Analog converter (D/A) is available and can be used only if the QIC board is installed. This will allow feedback controllers implemented on the QIC to drive the D/A to control the power amplifier instead of the external *COMMAND* input. The jumper J6 must be set to use the D/A output to drive the amplifier. This is illustrated in Figure A.6. The output of the D/A is also made available at the RCA jack labeled *D/A*.

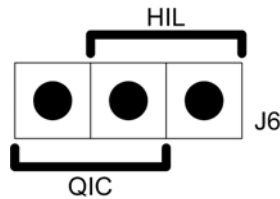


Figure A.6 Jumper J6 Configurations

A.2.3.12. 24-Bit Encoder Counter

One 24-bit encoder counter is connected to the encoder such that if a QIC board is installed, the encoder measurements can be read by the QIC.

A.2.3.13. Secondary Encoder Input To QIC

An additional external encoder can be attached to the system such that it can be read by the QIC. This may be used for the development of other experiments by the user.

A.2.3.14. External Analog Input To QIC

The analog input applied to the RCA jack labeled *COMMAND* is level-shifted and scaled such that a signal in the range of ± 5 V applied to it is made available as a signal in the range of 0-5 V at a QIC analog input. This is useful if you want to apply external command signals to a QIC-based controller.

A.2.3.15. Analog Signals Header: J11

The analog signals available at the RCA jacks, as previously discussed in this Section, are also accessible on the DCMCT J11 header, whose location on the main board is shown in Figure A.1 by component #17. The analog signals header J11 can be used for external circuitry. Its schematic is illustrated in Figure A.7.



Figure A.7 J11 Connector Schematic

A.3. System Parameters

The specifications on the DCMCT system model parameters are given in Table A.2.

<i>Symbol</i>	<i>Description</i>	<i>Value</i>	<i>Unit</i>
<i>Motor:</i>			
k_m	Motor Torque Constant	0.0502	N.m/A
R_m	Motor Armature Resistance	10.6	Ω
L_m	Motor Armature Inductance	0.82	mH
	Motor Maximum Continuous Torque	0.035	N.m
	Motor Power Rating	18	W
J_m	Moment Of Inertia Of Motor Rotor	1.16E-006	kg.m ²
τ_m	Motor Mechanical Time Constant	0.005	s
M_l	Inertial Load Disc Mass	0.068	kg
r_l	Inertial Load Disc Radius	0.0248	m
<i>Linear Amplifier:</i>			
V_{max}	Linear Amplifier Maximum Output Voltage	15	V
	Linear Amplifier Maximum Output Current	1.5	A
	Linear Amplifier Maximum Output Power	22	W
	Linear Amplifier Maximum Dissipated Power With Heat Sink, $R_{load} = 4 \Omega$	8	W
	Linear Amplifier Gain	3	V/V

Table A.2 DCMCT Model Parameter Specifications

The specifications on the DCMCT system sensors are given in Table A.3.

<i>Description</i>	<i>Value</i>	<i>Unit</i>
<i>Current Sense:</i>		
Current Calibration At $\pm 10\%$ At <i>CURRENT</i> RCA Jack	0.556	A/V
Current Calibration At $\pm 10\%$ At QIC A/D Input	1.112	A/V
Current Sense Resistor	0.1	Ω
<i>Encoder:</i>		
Encoder Line Count	1024	lines/rev
Encoder Resolution (In Quadrature)	0.0879	$^\circ$ /count
Encoder Type	TTL	
Encoder Signals	A, B, Index	
<i>Potentiometer:</i>		
Potentiometer Calibration At <i>POT</i> RCA Jack	39	$^\circ$ /V
Potentiometer Calibration At QIC A/D Input	78	$^\circ$ /V
Potentiometer Resistance	10	k Ω
Potentiometer Bias Voltage	± 4.7	V
Potentiometer Electrical Range	350	$^\circ$
<i>Tachometer:</i>		
Tachometer Calibration At <i>TACH</i> RCA Jack	667	RPM/V
Tachometer Calibration At QIC A/D Input	1333	RPM/V

Table A.3 DCMCT Sensor Parameter Specifications

It can be seen in Table A.3 that the analog sensors calibration constants are multiplied by a factor of two when measured by the QIC Analog-To-Digital converters (A/D's) as opposed to the RCA jacks. The reason is that the RCA outputs are in the ± 5 V range while the A/D inputs to the QIC are in the 0-5 V range.

A.4. Hardware Configuration

Although not used with QICii implementation, the DCMCT external connections are presented here. These are made via RCA and DIN jacks, as illustrated in Figure A.8.

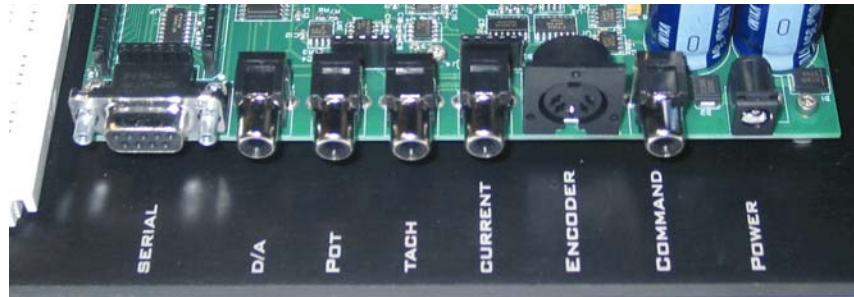


Figure A.8 DCMCT External Connectors (Not Used For QICii)

The DCMCT external connectors shown in Figure A.8 perform the signal connections described in Table A.4.

<i>Connector Label</i>	<i>Connector Type</i>	<i>Signal</i>	<i>Range</i>
<i>D/A</i>	RCA	QIC D/A Output	± 5 V
<i>POT</i>	RCA	Potentiometer Output	± 5 V
<i>TACH</i>	RCA	Tachometer Output	± 5 V
<i>CURRENT</i>	RCA	Current Measurement Output	± 5 V
<i>ENCODER</i>	5-Pin DIN	Encoder Output	A, B, Index
<i>COMMAND</i>	RCA	Command To Linear Amplifier	± 5 V
<i>POWER</i>	6-mm Jack	AC Power Input To DCMCT	15 VAC, 2.4 A

Table A.4 DCMCT External Connections

A.4.1. DCMCT Configuration For QICii Use

The DCMCT configuration and wiring required to use the QICii software is pictured in Figure A.9.

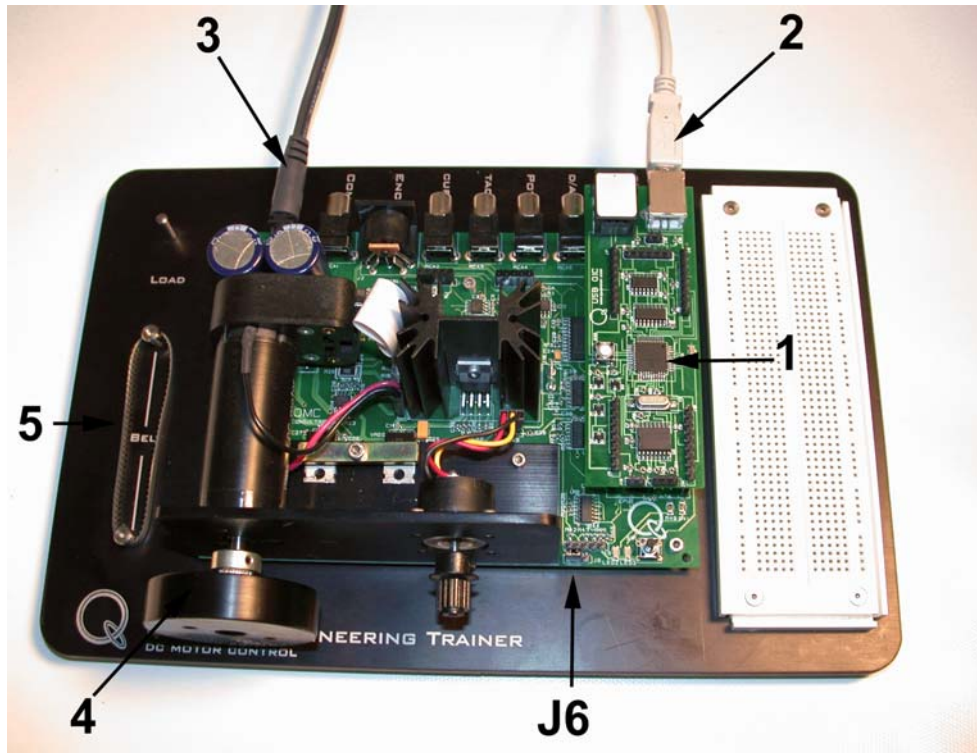


Figure A.9 DCMCT Configuration And Wiring For Use With QICii

Please configure and wire the DCMCT system for QICii use by following the steps below.

- Step 1. Ensure the QIC processor core board is properly installed into its corresponding socket on the DCMCT. This is illustrated by component #1 in Figure A.9. Be sure the QIC board is mounted correctly, with all pins properly inserted into the socket.
- Step 2. Ensure the inertial load is properly fastened to the DC motor shaft. This is depicted by component #4 in Figure A.9.
- Step 3. Ensure that the toothed belt does not drive the potentiometer. Remove it if necessary and place it on the corresponding storage pins, as shown by component #5 in Figure A.9.
- Step 4. When a QIC core board is inserted into the QIC Socket on the DCMCT, you can use the QIC to measure all the sensors and to control the power amplifier. In this case the jumper J6, located in Figure A.9, should be configured such that the onboard D/A output, controlled by the QIC, drives the linear amplifier. If not already done, you may need to physically move jumper J6 to the left to connect the output of the on-board D/A to the amplifier input. Enable the output of the D/A to drive the amplifier by setting jumper J6 to the left (i.e. QIC control) as shown in Figures A.10 and A.11.



Figure A.10 J6 Schematic: QIC Use

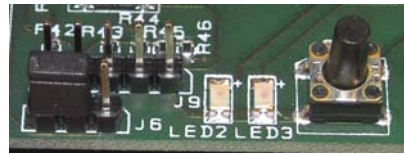


Figure A.11 J6 Setting: QIC Use

Step 5. Plug the provided AC power transformer into the wall outlet and insert the corresponding power connector into the DCMCT 6-mm jack labeled *POWER*. This QIC end of this power connection is represented by component #3 in Figure A.9.

Step 6. Connect the supplied USB cable from the USB connector on the QIC to a USB port on your PC. This QIC end of this USB connection is represented by component #2 in Figure A.9.

Note:

The USB QIC is compatible with both USB 1.1 as well as USB 2.0 PC hosts. If you are plugging the QIC to your PC for the first time, you must wait a few seconds for your PC to detect and configure the QIC. On a PC with Windows XP, the PC reports the detection and configuration of the QIC by presenting different message balloons near the clock display at the lower right-handed side of the screen. This process is similar on other versions of Windows that support USB. As an example, the following sequence of events occur on a PC with Windows XP.

1. After you plug in the QIC to your PC for the first time, you must wait a few seconds. After a few seconds elapse, the PC will report that it has detected the QIC as new hardware. The PC would typically report this fact by displaying a message balloon similar to the one shown in Figure A.12.
2. A few seconds after that, the PC would next display a message balloon similar to the one shown in Figure A.13. This second message balloon would indicate that your PC has recognised the QIC as an HID (Human Interface Device). An HID can be considered to be a member of a class of peripherals that can connect to PCs. The HID class includes many types of mice, keyboards, and joysticks, and other devices that typically allow a person to somehow physically interact with a PC. In some older version of Windows, Windows may display a message asking if it should try to find and install a driver for the QIC; you would respond *yes* in that case.
3. A few seconds after the second message balloon, a third message balloon similar to the one in Figure A.14 would be presented. This third and final message balloon would notify that the PC has installed the appropriate driver so that it can properly communicate with the QIC. A few seconds later, this message balloon should automatically disappear. By this point, the QIC will have been successfully installed and therefore should be ready to be used by your PC. In the future, if you plug in the QIC to the same USB port of the same PC as before, then this detection and configuration will not have to be done, so the QIC connected to the PC would be

ready for use right away.

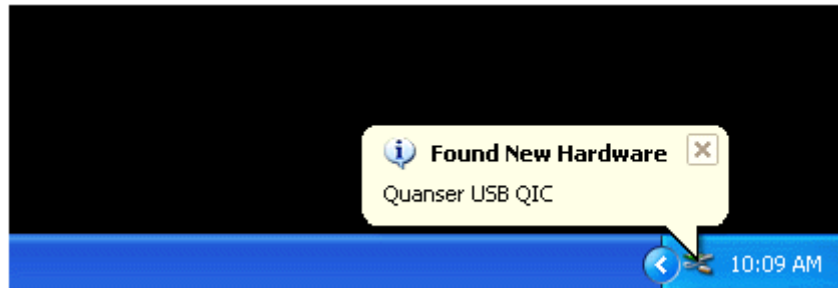


Figure A.12 Detection of the QIC on a PC with Windows XP.

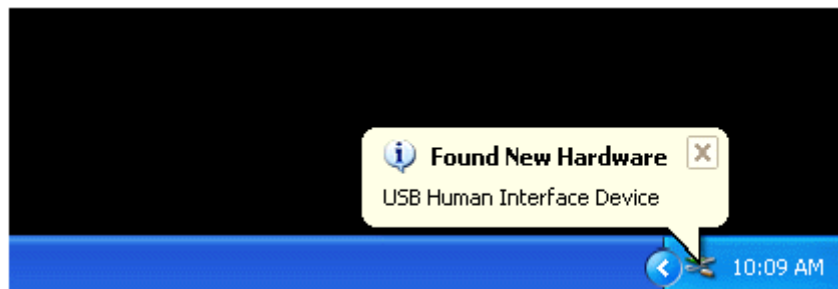


Figure A.13 Recognition of the QIC as an HID on a PC with Windows XP.



Figure A.14 Notification that the QIC is ready for use on a PC with Windows XP.

Step 7. A wiring summary is provided in Table A.5.

<i>From</i>	<i>To</i>	<i>Signal</i>
USB Port on PC	USB Port on QIC	Full Speed USB
DCMCT <i>D/A</i> Output		QIC <i>D/A</i> Output: ± 5 V
Wall Transformer	DCMCT <i>POWER</i> Input	Power Supply: 15 VAC, 2.4 A

Table A.5 DCMCT Wiring Summary When Using QICii

A.4.2. DCMCT Configuration For HIL Board Use

The DCMCT system is fully compatible with all Quanser's HIL (Hardware-In-the-Loop) terminal boards and as such can be used to perform control experiments on a PC. The real-time controller is typically designed in Simulink and runs on the PC in realtime using Win-Con.

Please configure and wire the DCMCT system for HIL board use by following the steps below.

- Step 1. In the case where a HIL board is used, a wiring summary is provided in Table A.6. Table A.6 also describes the connections performed by wiring the DCMCT system via the available RCA and DIN jacks.

<i>From</i>	<i>To</i>	<i>Signal</i>
DCMCT <i>POT</i> Output	HIL Board Analog Input #0	Potentiometer: ± 5 V
DCMCT <i>TACH</i> Output	HIL Board Analog Input #1	Tachometer: ± 5 V
DCMCT <i>CURRENT</i> Output	HIL Board Analog Input #2	Current: ± 5 V
DCMCT <i>ENCODER</i> Output	HIL Board Encoder Input #0	Encoder: TTL, A, B
HIL Board Analog Output #0	DCMCT <i>COMMAND</i> Input	Amplifier Command: ± 5 V
Wall Transformer	DCMCT <i>POWER</i> Input	AC Power Supply: 15 VAC, 2.4 A

Table A.6 Wiring Summary When Interfacing To A HIL Board

Step 2. It can be seen in Table A.6 that, when using a HIL board, the command to the DCMCT linear amplifier is generated by the PC and comes from one of the HIL analog outputs via the DCMCT RCA jack labeled *COMMAND*. However the jumper J6, located in Figure A.9, should be configured such that the voltage at the *COMMAND* RCA jack, controlled by the PC via the HIL board, drives the linear amplifier. If not already done, you may need to physically move jumper J6 to the right to enable the signal coming from the *COMMAND* RCA jack to drive the amplifier input. Do so by setting jumper J6 to the right (i.e. HIL board control) as shown in Figures A.15 and A.16.

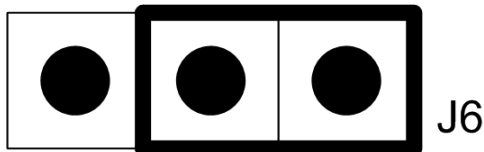


Figure A.15 J6 Schematic: HIL Board Use

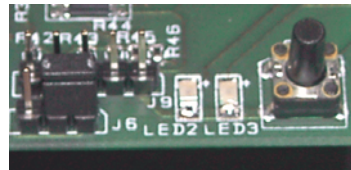


Figure A.16 J6 Setting: HIL Board Use

A.5. System Overview Relevant To QICii Control

Figure A.17 presents a picture of the DCMCT system which, in this particular example, is connected to a laptop. In this Section, Figure A.17 is used as a basis to give an overview of the DCMCT process modelling with respect to control systems.

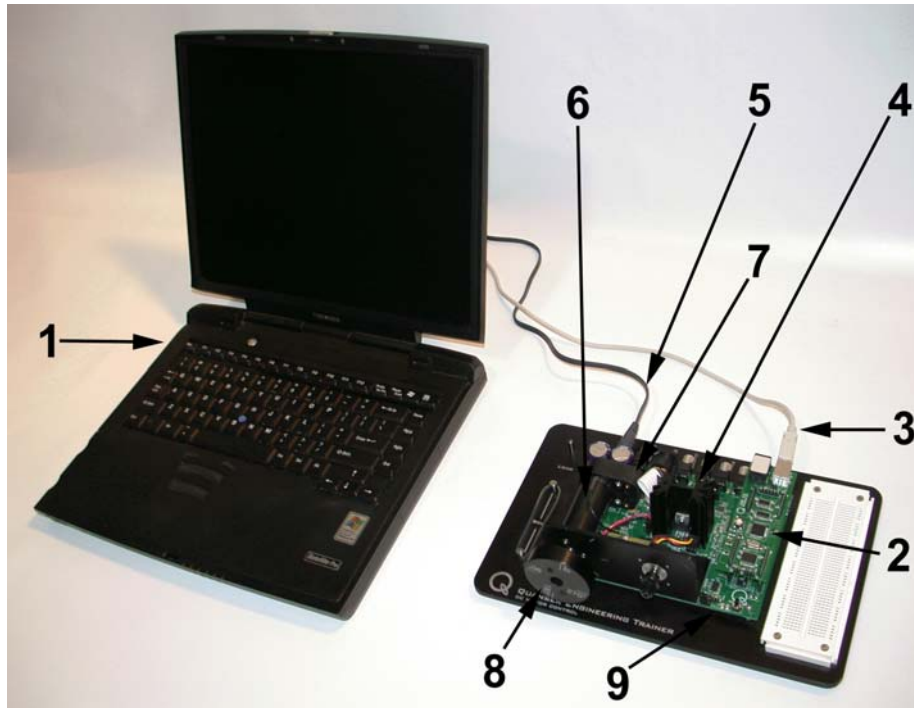


Figure A.17 System Overview Relevant To QICii Control

The DCMCT different components relevant to control are located and marked by an ID number in Figure A.17. They are listed and described in Table A.7.

<i>ID #</i>	<i>Description</i>	<i>ID #</i>	<i>Description</i>	<i>ID #</i>	<i>Description</i>
1	Laptop (or PC)	2	PIC Microcontroller	3	USB Cable
4	Linear Amplifier	5	AC Power Supply	6	DC Motor
7	Encoder	8	Inertial Load	9	User Switch And Two LED's

Table A.7 DCMCT System Components Relevant To Control

The major system components are the DCMCT and the PC (or laptop). Dissecting the system as carried out in Table A.7, we find that it has the following major components, as

marked in Figure A.17:

- Motor with Inertial Load
- Amplifier
- Encoder
- D/A Converter
- On-Board Computer (PIC)
- User Switch (i.e. Momentary Action Pushbutton)
- Two LED's
- PC or laptop

The corresponding schematic diagram showing how these components interact is given in Figure A.18.

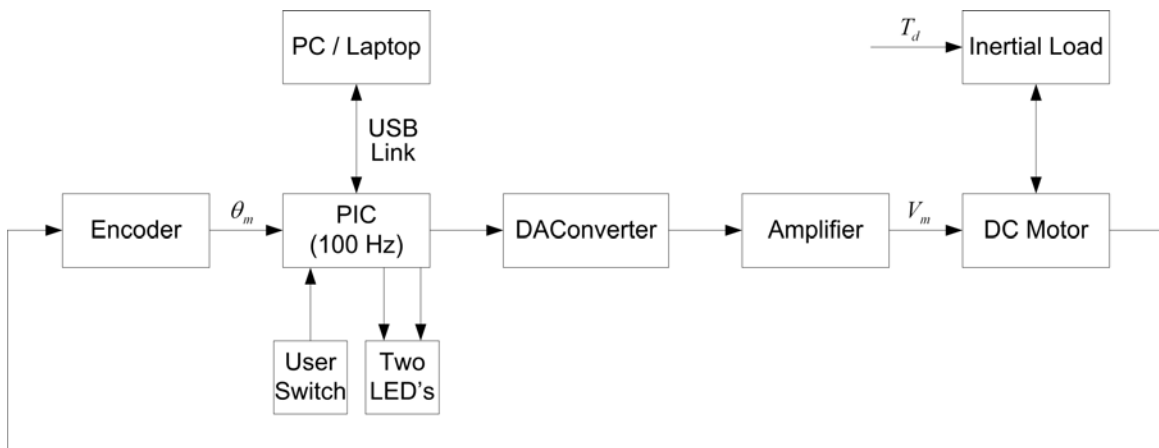


Figure A.18 A Schematic Diagram Of The System Outlining The Major Components And Their Interaction

The system variables used in Figure A.18 are described in the nomenclature given in Table A.8.

<i>Symbol</i>	<i>Description</i>	<i>Unit</i>
θ_m	Motor Angle	rad
V_m	Voltage from the amplifier which drives the motor	V
T_d	Disturbance Torque applied to the inertial load	N.m

Table A.8 System Variables

Appendix B. DCMCT/USB QICii Software Guide

B.1. Introduction

The QIC interactive interfaces, also known as QICii, consists of a series of pre-compiled QIC controllers and a PC-based Java interface. It allows you to:

- Connect to the QIC via the QICii Java module.
- Interactively change parameters in the QIC controller to affect system behaviour.
- Collect realtime data from the QIC for plotting and future analysis.

B.2. Software Installation

B.2.1. System Requirements

USB QICii runs on Microsoft Windows 2000, Windows XP, Windows Vista (32-bit and 64-bit), or Windows 7 (32-bit and 64-bit) and requires, at a minimum, the resources listed below to function properly.

The personal computer (i.e., PC or laptop) hardware requirements are:

- Minimum Pentium II 450 MHz processor.
- 128 MB RAM.
- 150 MB available hard drive space.
- External USB port.
- Graphics card compatible with OpenGL 1.2 or better.

The software requirements are:

- Windows 2000 with SP3 or later, Windows XP with SP2 or later, Windows Vista

- (32-bit and 64-bit), or Windows 7 (32-bit and 64-bit).
- Up-to-date OpenGL graphics card drivers (version 1.2 or later) fully installed for the operating system used. This is required to properly display the Java 3D graphics used by USB QICii. Note that not all graphics cards are completely supported by Microsoft Windows Vista.
- User account with administrative privileges to install USB QICii. Not required for running it.

Notes:

The minimum hard drive space of 150 MB is for the USB QICii software and other files that support it. Obviously, the PC must have greater than 150 MB of hard drive space for the Windows operating system and other applications that may be installed in the PC.

It is highly recommended that greater than 128 MB of RAM be installed in the PC.

B.2.2. Installation

B.2.2.1. USB QICii Components

The USB QICii installer installs and configures the following parts of the USB QICii software:

- Universal Serial Bus QIC interactive interfaces (USB QICii)**
This package contains a Java virtual machine as well as the USB QICii Java application.

B.2.2.2. If Upgrading USB QICii

CAUTION:



**Do not overwrite a previous installation of USB QICii.
Always uninstall the previous version using the uninstaller first!**

To remove a previous version of the USB QICii software, launch the Control Panel's *Add or Remove Programs* Windows utility. Select *QICii_USB* from the list and then click *Remove*. A program removal confirmation window should open up for the USB QICii, as displayed in Figure B.1. Click *Yes*.

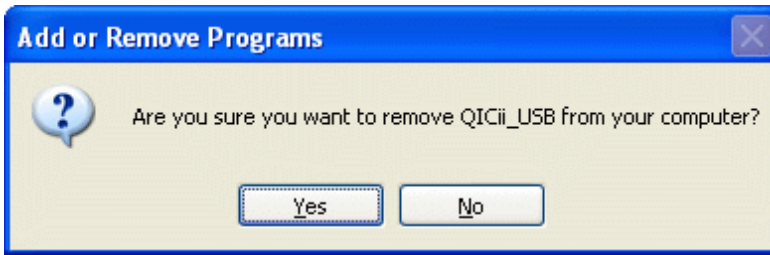


Figure B.1 USB QICii Removal Confirmation Window

The USB QICii uninstallation process should then start, as indicated by a progress bar similar to the one represented in Figure B.2.

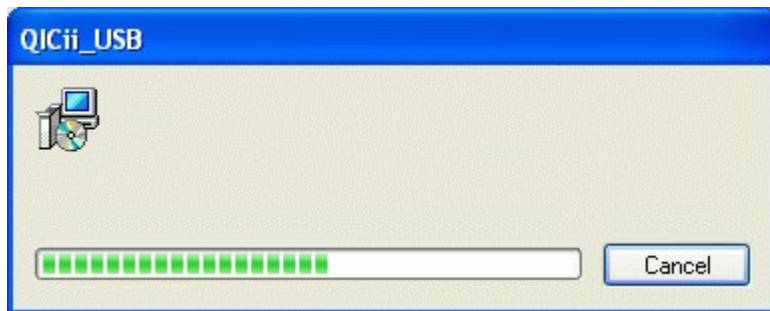


Figure B.2 USB QICii Removal Progress Bar

B.2.2.3. Installing USB QICii

Follow this procedure when installing USB QICii from the installation CD. When installing from an installation folder on your network, refer to this section for reference. However, you should also follow any additional instructions provided by your network administrator.

Please follow the instructions detailed below to install the USB QICii software.

Step 1. Insert the USB QICii CD supplied with the system into your CD-ROM drive.

Step 2. The QET-DCMCT welcome HTML page should open automatically. The USB QICii Windows installer can then be started by double-clicking on the *USB_QICii (Windows installer)* hyperlink in the first section of that page. If the QET-DCMCT welcome HTML page does not open automatically, navigate to your CD-ROM drive, and double-click on the *setup.exe* installation file located in the *Software\USB_QICii*

folder. The USB QICii installer screen should then appear as displayed in Figure B.3.

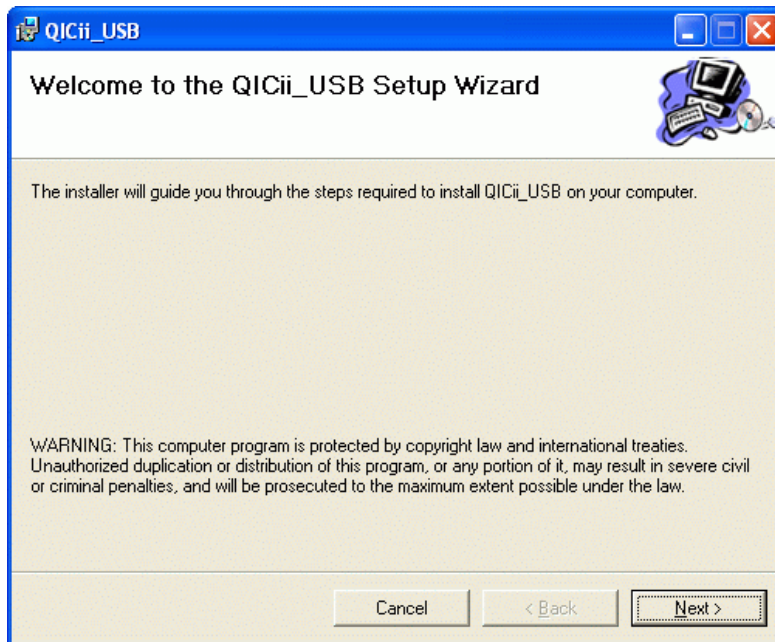


Figure B.3 USB QICii Installer Welcome Window

Step 3. Proceed the installation by clicking the *Next* Button. The End-User License Agreement (EULA) is then displayed in a dialog window, as illustrated in Figure B.4.

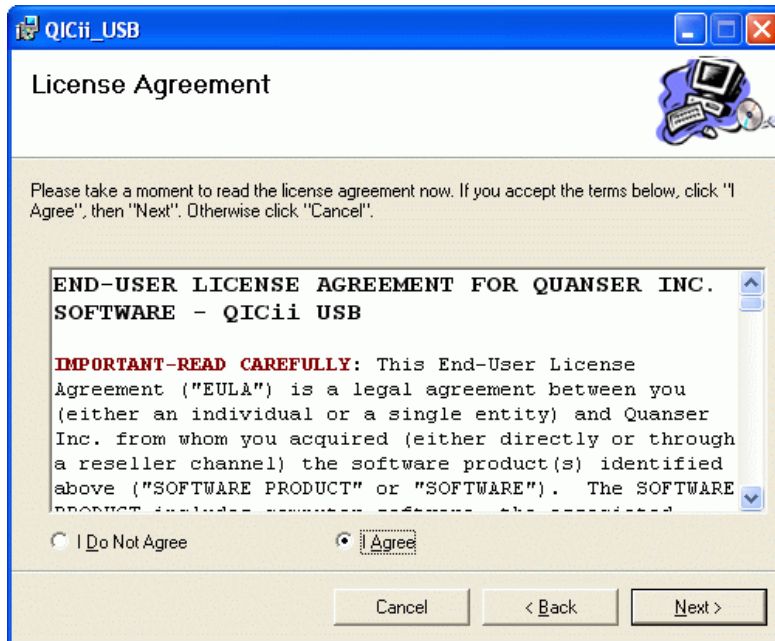


Figure B.4 USB QICii License Agreement Window

If you agree to it, select the *I Agree* radiobutton and click *Next* to pursue the installation. Otherwise, select the *I Do Not Agree* radiobutton and click *Next* to terminate the installation.

Step 4. The destination installation location page is then displayed, showing the default installation directory. This is illustrated in Figure B.5. To use the default directory, click *Next* for the following step. To install to a directory other than the default one, select the *Browse* button and enter a new pathname or browse through your file system to locate the desired installation directory. Once this is done, click *Next* to continue.

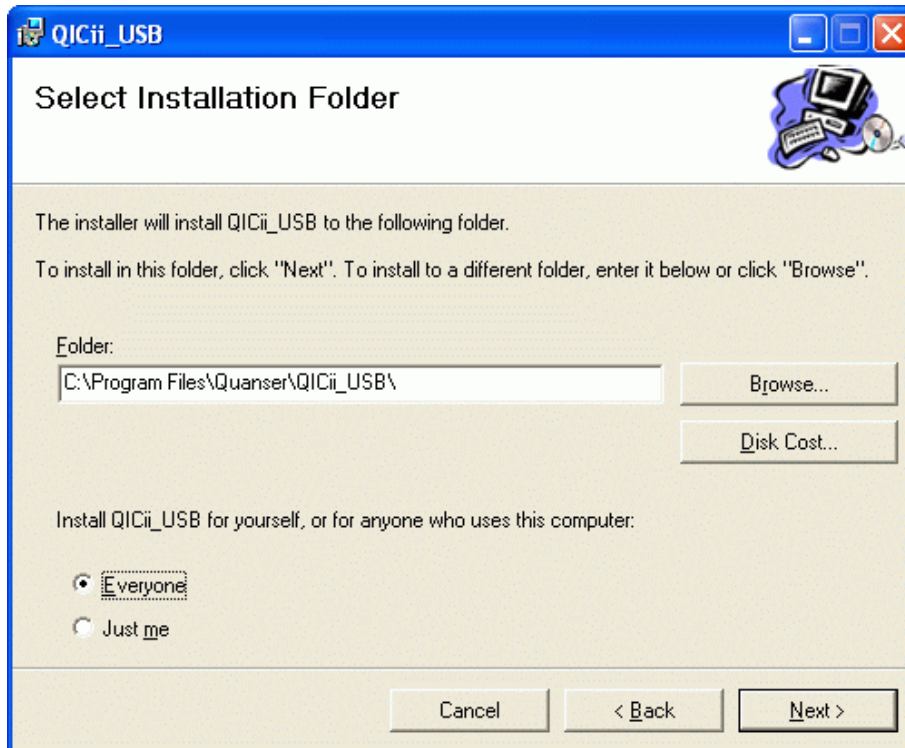


Figure B.5 USB QICii Installer Destination Directory Window

Step 5. If the installation directory does not already exist, a dialog box appears prompting you to create it. Click *Yes* to move on to the next installer window.

Step 6. When the installer is ready to install USB QICii, a *Confirm Installation* window then appears. Click *Next* to continue.

Step 7. The USB QICii installer wizard then displays a progress bar while it copies files to the destination location and configures your system. This is represented in Figure B.6.

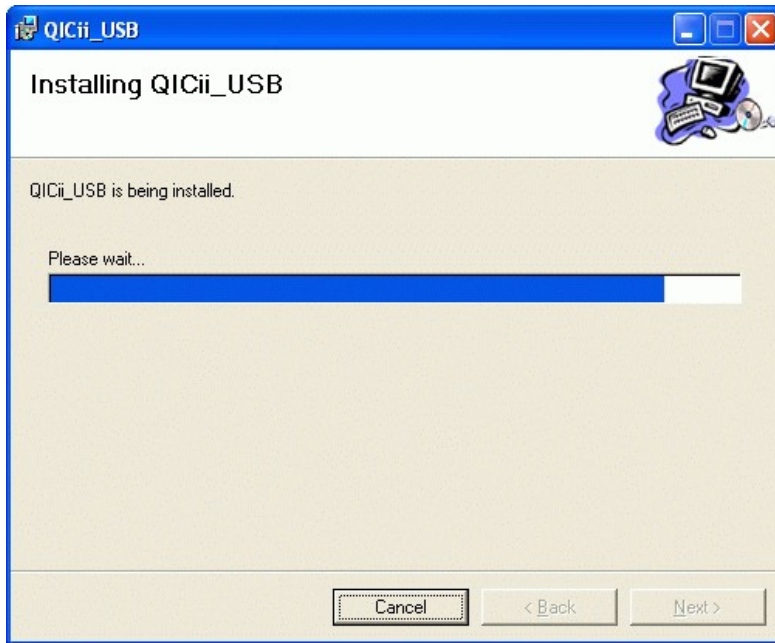


Figure B.6 USB QICii Installation Progress Bar

Step 8. When the installation is successfully finished, the installation complete page is displayed, as shown in Figure B.7. Click the *Close* button to exit the installer wizard.

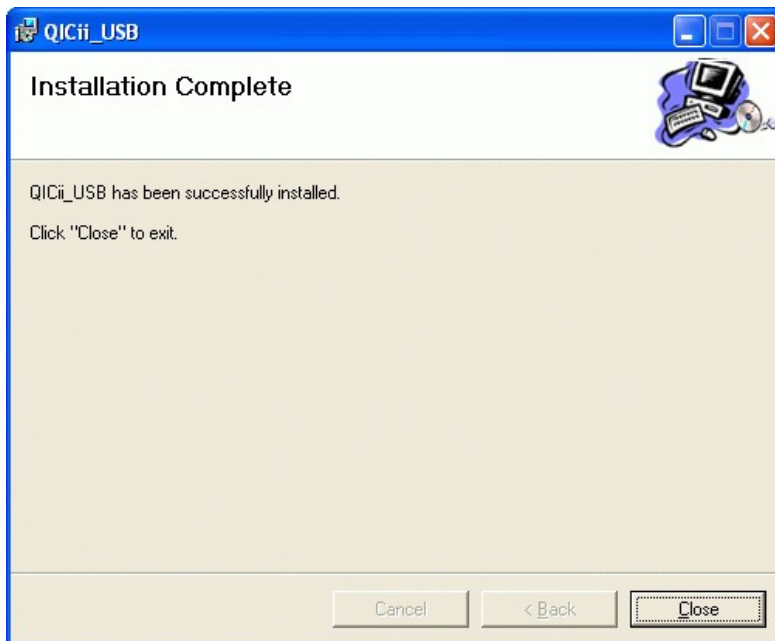


Figure B.7 USB QICii Installation Complete

B.2.3. Getting Started

Please follow the steps below to start and use the USB QICii software together with the DCMCT.

Step 1. To launch the USB QICii software, select *All Programs* | *Quanser* | *USB QICii* | *USB QICii* from the *Start* menu. A user interface window, similar to the one illustrated in Figure B.8, will be displayed.

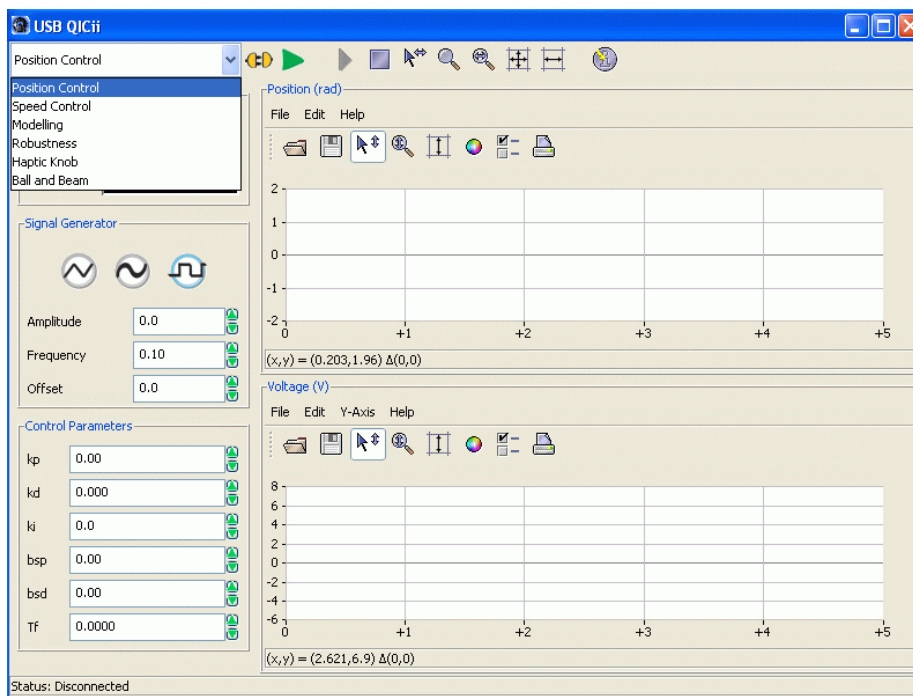


Figure B.8 QIC Interactive Interface Window

Step 2. As detailed in Section A.4.1 of Appendix A, ensure that the jumper J6 has been moved to the left in order to enable the QIC to drive the linear amplifier. This is illustrated in Figure B.9.

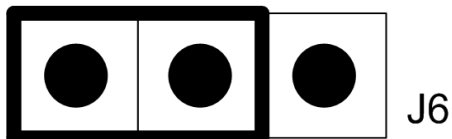


Figure B.9 J6 Schematic: QIC Use

Step 3. Ensure that the DCMCT is powered up.

Step 4. When the DCMCT is powered up, typically LED2 lights up while LED3 starts flashing, at which point you should press the *Reset* button on the QIC and then press the User Switch on the DCMCT. LED2 and LED3 should both turn off after those buttons are pressed in the above-mentioned specified order. As a reminder of Appendix A, the location of the *Reset* button on the QIC board is depicted in Figure B.10. The User Switch is defined as the momentary action pushbutton switch next to the two LED's on the DCMCT. It is represented in Figure B.11, which also shows jumper J6.

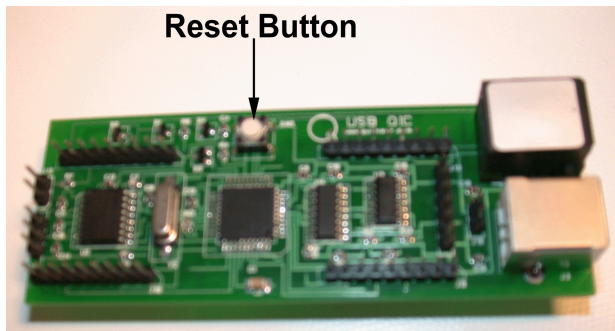


Figure B.10 *Reset* Button On QIC Processor Core Board

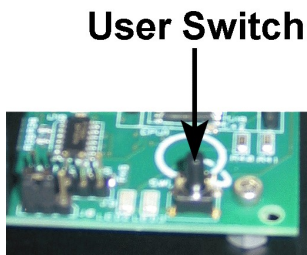


Figure B.11 User Switch

Step 5. Only when LED3 stops flashing can you connect to the QIC. Select which controller you want to run from the drop-down menu on the top left of the QICii window.

Step 6. Select the *Connect to data source* button on top of the QICii window.

Step 7. You are now able to receive/send data from/to your chosen controller. LED2 should light up while LED3 should remain off, and the controller should start running.

Note:

The drop-down menu is disabled while the controller is running. You must select the *Stop controller* button on top of the QICii window to stop the controller (this should also turn off LED2, and enable the drop-down menu again). Once the drop down menu is enabled again and the controller is stopped, you can select any one of the oth-

er controller experiments, if you want. After selecting some other controller experiment, you can once again select the *Connect to data source* button on top of the QICii window to be able to receive/send data from/to the controller of your choice.

Step 8. You can now follow the instructions for the corresponding laboratory.

The user interface toolbar buttons, which have been mentioned in the above procedure, are available on the left-hand side of the toolbar located on top row of the QICii main window. They are located and identified in Figure B.12 through the use of a unique ID number.

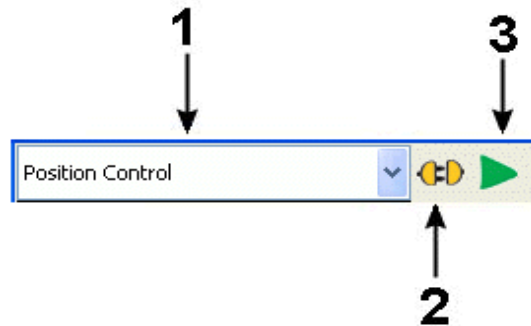


Figure B.12 Getting Started Features Toolbar

Table B.1 provides a description of each individual button of the "getting started" features. These functions correspond to the toolbar buttons represented in Figure B.12.

<i>Button ID #</i>	<i>Description</i>
1	Drop-down selection list: choose which controller you want to run.
2	Connect/Disconnect to/from the QIC.
3	Start/Stop the controller. When "stopped", the control loop keeps running in the QIC, the voltage to the motor however is set to zero Volts. Data can still be collected and viewed.

Table B.1 QICii Getting Started Toolbar Icon Description

The QICii window parameters for each individual controller are described in the Chapter associated with the corresponding laboratory. They are normally located on the left-hand side of the QICii window.

B.2.4. Plots

Most QICii modules have one or two plots displaying the parameters being controlled. As an example, this is illustrated in Figure B.13 in the case of the *Position Control* module.

Plots can be displayed in a variety of ways. Plots can be changed together or individually.

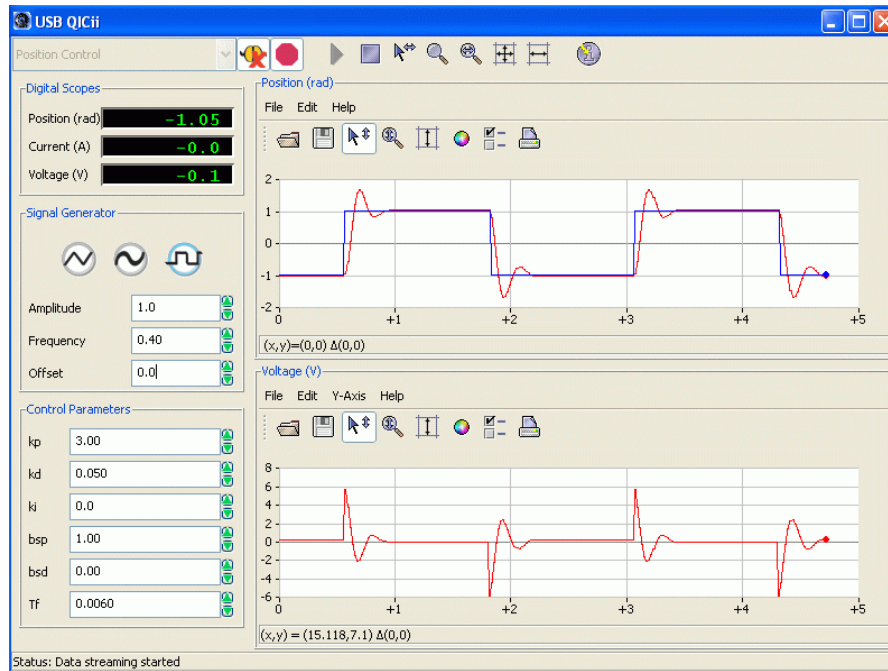


Figure B.13 QICii Position Control Plots

B.2.4.1. Common Features

In Figure B.13, changes to both plots can be applied simultaneously by using the toolbar located on the top row of the main window. That toolbar is detailed in Figure B.14 and Table B.2 through the use of a unique ID number identifying each individual button.

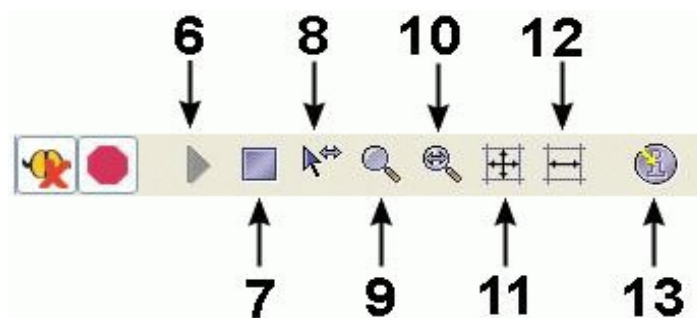


Figure B.14 Common Plot Features Toolbar

Table B.2 lists and describes the plot features applying to both plots. These functions correspond to the toolbar buttons identified and located in Figure B.14.

Button ID #	Description
6	Start Plotting.
7	Stop Plotting.
8	Enable pan and zoom of X-axis.
9	Zoom of X- and Y- axes. Select X-Y area to magnify. Hold <i>Ctrl</i> key to take measurements.
10	Zoom of X-axis. Select portion of X-axis to magnify. Hold <i>Ctrl</i> key to take measurements in X-axis.
11	Autoscale both axes.
12	Autoscale X-axis.
13	Display information about QICii.

Table B.2 Common Plot Features Button Description

B.2.4.2. Individual Plot Functions

Changes to each individual plot can also be applied independently. These separate changes can be done either by using the menu items or the buttons on top of the plot, or by right-clicking the mouse on the plot and using the pop-up context menu items. In Figure B.13, changes to each individual plot can be applied by using the toolbar located on top of the corresponding plot. That toolbar is detailed in Figure B.15 and Table B.3 through the use of a unique ID number identifying each individual button.

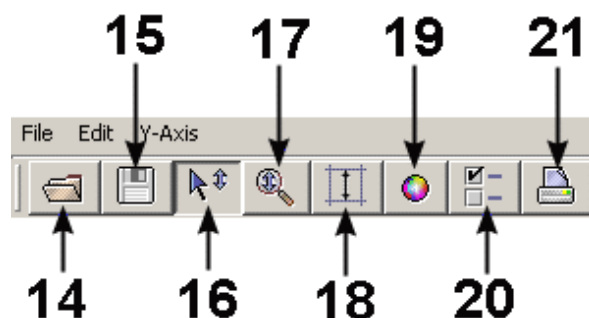


Figure B.15 Individual Plot Features Toolbar

Table B.3 lists and describes the plot features applying to an individual plot. These functions correspond to the toolbar buttons identified and located in Figure B.15.

<i>Button ID #</i>	<i>Description</i>
14	Open plot file (*.plt). Same as the <i>File Open...</i> menu item.
15	Save plot. Refer to Table B.4 for the supported file formats. Same as the <i>File Save As...</i> menu item.
16	Enable pan and zoom of Y-axis.
17	Zoom of Y-axis. Select portion of Y-axis to magnify. Hold <i>Ctrl</i> key to take measurements in Y-axis.
18	Autoscale Y-axis.
19	Choose plot colours. Click on the plot to select the colour to change. Same as the <i>Edit Colours...</i> menu item.
20	Change plot properties, e.g.: axes, grid, titles, labels. Same as the <i>Edit Properties...</i> menu item.
21	Print the plot. Same as the <i>File Print...</i> menu item.

Table B.3 Plot Individual Feature Toolbar Button Description

Additionally, the menu item *File | Exit* closes the corresponding plot definitely.

B.2.4.3. Saving Data

You can also save the data displayed in a plot to a file for future analysis. The types of file format to choose from are listed and described in Table B.4.

<i>File Extension</i>	<i>Type</i>	<i>How to use it</i>
.mat	MATLAB binary	Use the <i>load</i> command in MATLAB to load the data file into the MATLAB workspace. You can then process the data. Use <i>who</i> to discover the variable names.
.m	MATLAB script	Simply type in the name of the file you saved in the MATLAB prompt. This will load and plot the data you had collected. Use <i>who</i> to discover the variable names.
.csv	Microsoft Excel	Comma-separated-values data for importing into spreadsheets.
.plt	Plot file	Binary file readable by QICii.
.png	Image file	Take a screen capture of the plot and save it as a Portable Network Graphics (PNG) image file.

Table B.4 QICii-Supported File Formats To Save Plot Data

B.2.4.4. Taking Measurements On A Plot (Ctrl Zoom)

A utility is available that allows the user to take measurements (along both X- and Y-axes) off a plot. Consider the position plot in Figure B.13.

Measurement of the position overshoot value is depicted by the double arrow in Figure B.16. To take a measurement on the Y-axis, press the **Zoom-Y** button (illustrated by button #17 in Figure B.15) and hold the **Ctrl** key down. Start the measurement by moving the mouse cursor on top of the starting point to measure from, hold down the mouse left button, and move the cursor to the end measurement point. You can now release the mouse left button and the **Ctrl** key. The starting point Cartesian coordinates are displayed in the bottom left corner of the plot under the format (x,y) . The difference between the end and starting points Cartesian coordinates is also displayed in the bottom left corner of the plot under the format $\Delta(x,y)$. In the current example, we have: $\Delta(x,y) = \Delta(0, \text{Overshoot Value})$.

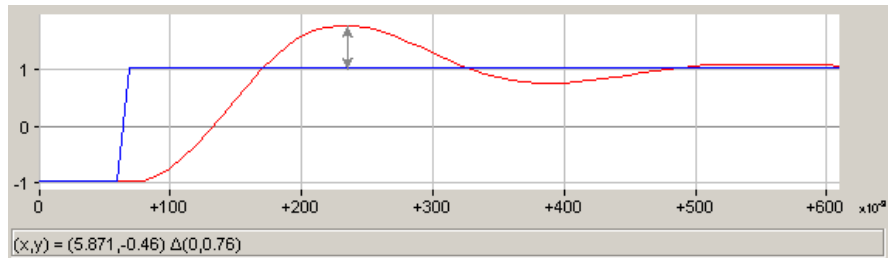


Figure B.16 Overshoot (Y-Axis) Measurement

Measurement of the position peak time value is depicted by the double arrow in Figure B.17. To take a measurement on the X-axis, press the **Zoom-X** button (illustrated by button #10 in Figure B.14) and hold the **Ctrl** key down. Start the measurement by moving the mouse cursor on top of the starting point to measure from, hold down the mouse left button, and move the cursor to the end measurement point. You can now release the mouse left button and the **Ctrl** key. The starting point Cartesian coordinates are displayed in the bottom left corner of the plot under the format (x,y) . The difference between the end and starting points Cartesian coordinates is also displayed in the bottom left corner of the plot under the format $\Delta(x,y)$. In the current example, we have: $\Delta(x,y) = \Delta(\text{Peak Time Value}, 0)$.

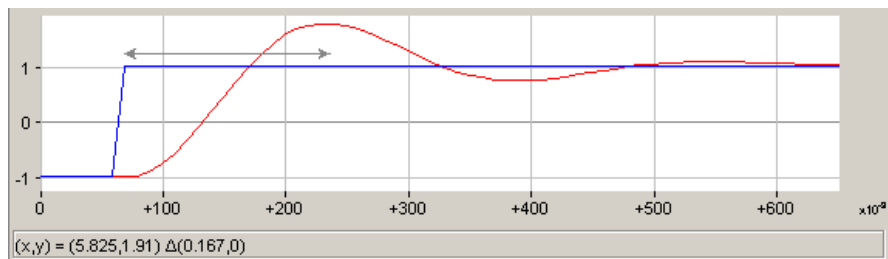


Figure B.17 Peak Time (X-Axis) Measurement

Similarly, measurements along both X and Y axes can also be taken simultaneously by using the **Zoom-XY** button (illustrated by button #9 in Figure B.14) and holding the **Ctrl** key down.

B.2.5. Signal Generator

Most QICii modules have a signal generation utility which can be used, for example, to generate the DCMCT motor voltage in open-loop (like in the *Modelling* laboratory) or the reference signal to a closed-loop system. If present, the signal generator input frame is located on the left-hand side of the QICii module window, as shown in Figure B.13. Furthermore, it is detailed in Figure B.18 where every input parameter defining the generated signal is identified through the use of an ID number.

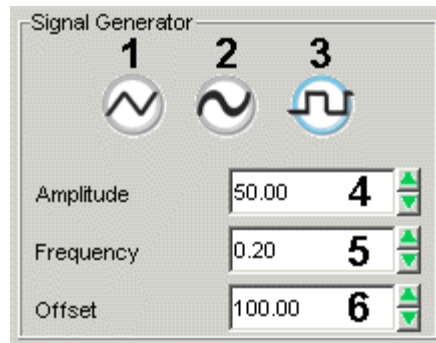


Figure B.18 QICii Signal Generator

Table B.5 provides a description of each input parameter defining the generated signal, as numbered in Figure B.18.

<i>ID #</i>	<i>Description</i>
1	Triangular Wave Generation Triggering Button
2	Sinusoidal Wave Generation Triggering Button
3	Square Wave Generation Triggering Button
4	Generated Signal Amplitude Input Box
5	Generated Signal Frequency Input Box
6	Generated Signal Offset Input Box

Table B.5 QICii Signal Generator Input Parameter Description

The actual implementation of the QICii signal generator is described in the following sub-sections.

B.2.5.1. Square Wave

When button #3 in Figure B.18 is pressed, the QICii signal generator outputs a square wave as a function of the *Amplitude*, *Frequency*, and *Offset* input parameters. This is illustrated in Figure B.19.

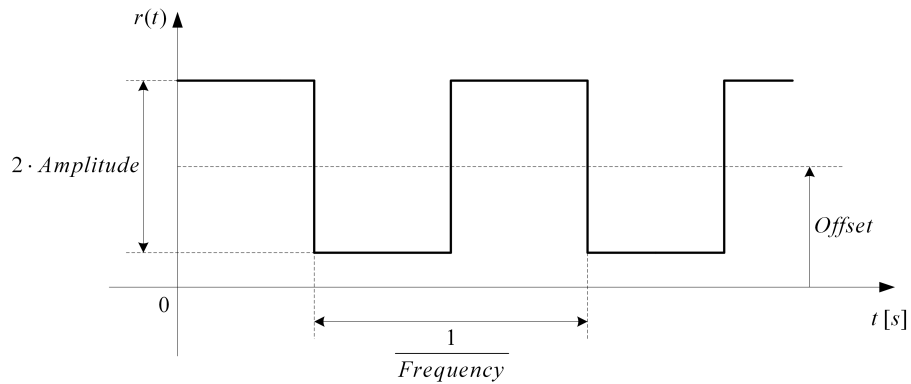


Figure B.19 QICii Square Wave Generation

Mathematically, the generated square wave signal, $r(t)$, can be characterized by the following piecewise-continuous function:

$$r(t) = \begin{cases} \text{Ampli} + \text{Offset} & 0 \leq t \text{ and } t < \frac{1}{2 \text{Freq}} \\ -\text{Ampli} + \text{Offset} & \frac{1}{2 \text{Freq}} \leq t \text{ and } t < \frac{1}{\text{Freq}} \\ r\left(t \left[\frac{1}{\text{Freq}} \right]\right) & \text{otherwise} \end{cases} \quad [\text{B.1}]$$

where t is the continuous time variable, *Ampli* the *Amplitude* input, *Freq* the *Frequency* input, *Offset* the *Offset* input, and $[]$ the modulo function.

B.2.5.2. Triangular Wave

When button #1 in Figure B.18 is pressed, the QICii signal generator outputs a triangular wave (a.k.a. sawtooth signal or ramp) as a function of the *Amplitude*, *Frequency*, and *Offset* input parameters. This is illustrated in Figure B.20.

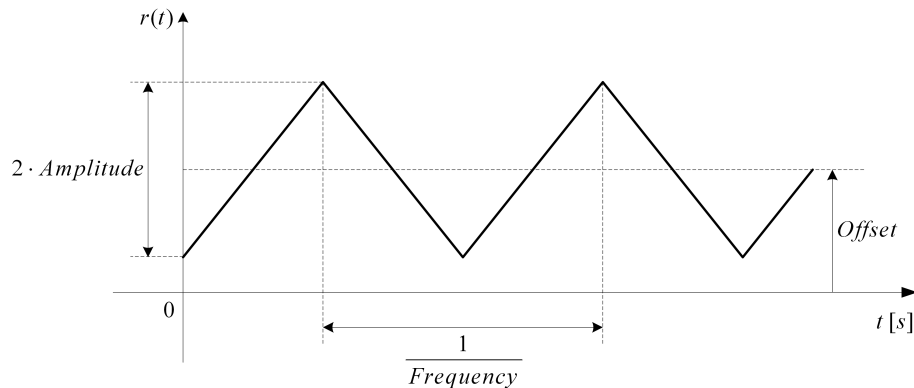


Figure B.20 QICii Triangular Wave Generation

Mathematically, the generated triangular wave signal, $r(t)$, can be characterized by the following piecewise-continuous function:

$$r(t) = \begin{cases} R_0 t - \text{Ampli} + \text{Offset} & 0 \leq t \text{ and } t < \frac{1}{2 \text{ Freq}} \\ -R_0 \left(t - \frac{1}{2 \text{ Freq}} \right) + \text{Ampli} + \text{Offset} & \frac{1}{2 \text{ Freq}} \leq t \text{ and } t < \frac{1}{\text{Freq}} \\ r\left(t \left[\frac{1}{\text{Freq}} \right]\right) & \text{otherwise} \end{cases} \quad [\text{B.2}]$$

where t is the continuous time variable, *Ampli* the *Amplitude* input, *Freq* the *Frequency* input, *Offset* the *Offset* input, and $[]$ the modulo function.

According to Figure B.20, the triangular wave positive slope, R_0 , is defined as:

$$R_0 = 4 \text{ Ampli } \text{ Freq} \quad [\text{B.3}]$$

where *Ampli* and *Freq* are the *Amplitude* and *Frequency* inputs, respectively.

Equation [B.3] can also be expressed as follows:

$$R_0 = 4 \text{ Amplitude } \text{ Frequency} \quad [\text{B.4}]$$

B.2.5.3. Sinusoidal Wave

When button #2 in Figure B.18 is pressed, the QICii signal generator outputs a sinusoidal wave as a function of the *Amplitude*, *Frequency*, and *Offset* input parameters.

Mathematically, the generated sinusoidal wave signal, $r(t)$, can be characterized by the following function:

$$r(t) = \textit{Ampli} \sin(2 \pi \textit{Freq} t) + \textit{Offset} \quad [\text{B.5}]$$

where t is the continuous time variable, *Ampli* the *Amplitude* input, *Freq* the *Frequency* input, *Offset* the *Offset* input, and *sin* is the sinusoidal function.

B.3. Troubleshooting

B.3.1. Setting Up The QIC

B.3.1.1. Does Your QIC Have the USB QICii Firmware Program In It?

If you purchased the complete DCMCT system then your QIC will have a firmware program in it that was installed using a programming device through the RJ11 port. This installed program inside the QIC contains all six controllers of the QICii's six experiments. The way to check if this program is inside the QIC is to power up the DCMCT with the QIC core installed and observe the two LED's next to the User Switch. If they alternate back and forth or if one of them flashes while the other one stays on, then the required firmware program is present. Otherwise if they do not flash (i.e. their state is constant and either on or off), the QIC does not have the firmware program in it.

B.3.2. Problem Viewing Three-Dimensional Graphics

For the *Haptic Knob* and *Ball and Beam* modules, QICii uses the Java 3D Viewer. To successfully run the Java 3D Viewer under Microsoft Windows 2000, DirectX 9.0 needs to be installed on your computer. This is readily available from the Microsoft website by running *Windows Update*.

The USB QICii installer should have automatically installed the necessary files to properly view 3D graphics. If the 3D graphics do not load properly then try the following:

- Step1. Go to your Windows desktop.
- Step2. Right click the mouse on an empty area of your Windows Desktop and choose Properties from the drop down menu. The Display Properties dialog should pop up.
- Step3. In the Display Properties dialog click on the *Settings* tab.
- Step4. Click on the *Advanced* button of this *Settings* tab. The Plug and Play Monitor dialog should pop up.
- Step5. In the Plug and Play Monitor dialog click on the *Troubleshoot* tab.
- Step6. For Hardware Acceleration move the slider to None and click the Apply button and then the *OK* button of the Plug and Play Monitor dialog.
- Step7. Click the *Apply* button and then the *OK* button of the Display Properties dialog.
- Step8. Reboot your PC.
- Step9. Start the USB QICii application and select the Ball and Beam module to see if the 3D graphics load properly.

Notes:

By following the above steps, the graphics might load properly as desired. However, the speed at which the various components are selected and activated on the USB QICii application might be noticeably slow and therefore take a noticeably long time to load. To improve the hardware acceleration, try moving the Hardware Acceleration slider to 1/5 in Step 6, above, and see if the 3D graphics load properly, as described in Steps 6 through 9 above. If they do then try increasing the hardware acceleration by increments of 1/5, such as 2/5, 3/5, etc. Each time you increase the hardware acceleration by a 1/5 increment, check to see if the 3D graphics load properly, as outlined in Steps 6 through 9 above. If they do then increase the hardware acceleration by another 1/5 increment. If the 3D graphics do not load properly then decrease the hardware acceleration by 1/5 increments until the 3D graphics load properly. The result of all this should be such that your PC will be configured to have the highest possible hardware acceleration that still allows 3D graphics to load properly for the USB QICii application.

If the hardware acceleration is altered, as outlined above, then remember to set the hardware acceleration back to its original setting after using the USB QICii software. By "original setting", we mean the setting that existed before modifying the hardware acceleration slider. By restoring the hardware acceleration back to its original setting, the operation and performance of other application software in the PC will not be affected. However, if the USB QICii is the only application that runs in the PC, then it is not necessary to restore the original hardware acceleration.