

**The Design and Development of an Educational Controls Laboratory:
Electromechanical Module, Control Software, and Graphical User Interface**

by

Richard Dick Wang

Bachelor of Science in Aeronautics and Astronautics

University of Washington (1993)

Submitted to the

Department of Mechanical Engineering

in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 1995

© 1995 Massachusetts Institute of Technology

All Rights Reserved

Signature of Author _____

Department of Mechanical Engineering

May 19, 1995

Certified by _____

Steven Dubowsky

Thesis Supervisor

Accepted by _____

Ain A. Sonin

Chairman, Departmental Graduate Committee

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

AUG 31 1995

LIBRARIES ARCHIVES

**The Design and Development of an Educational Controls Laboratory:
Electromechanical Module, Control Software, and Graphical User Interface**

Submitted to the

Department of Mechanical Engineering

on May 22, 1995

in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

by

Richard Dick Wang

Abstract

Hands-on experience plays an important role in an educational System Dynamics and Controls curriculum. At the Massachusetts Institute of Technology, the Department of Mechanical Engineering offers a series of undergraduate courses in the area of system dynamics and controls, including one that contains a laboratory component. The current Controls System Laboratory has been in service for ten years with little upgrading or modernization. The worn equipment and out dated technology has created a need for a new laboratory.

The Department of Mechanical Engineering at the Massachusetts Institute of Technology launched a study into redeveloping the controls system laboratory. A team of two masters students was supervised by a faculty member in the department to carry out this project.

As the result of this 18-month study, a modular and flexible laboratory prototype set-up was designed, fabricated, and tested. The prototype system uses commercially available hardware and software, plus in-house modifications and tailoring, to build a user friendly and educationally effective System Dynamics and Controls Laboratory workstation.

This thesis focuses on the design and fabrication of an electromechanical experiment module, and the creation of the fundamental software programs for the new laboratory. The overall software structure design, real-time control, and the graphical user interface are also documented.

Thesis Supervisor:

Dr. Steven Dubowsky

Professor of Mechanical Engineering

*To my Parents
Alice and David*

Acknowledgments

I would like to express my gratitude to my advisor, Professor Steven Dubowsky, for giving me this wonderful opportunity of my life. I deeply appreciate his invaluable direction and advice throughout this project.

I am also grateful to Professor Nam Suh, Professor Boris Mikic, and the Department of Mechanical Engineering at MIT for their support of this project.

To my project partners: Michele Tesciuba, Jason Hintersteiner, and German Soto; it has been my pleasure working with you in achieving this success. I would especially like to thank Michele for making this project truly a learning experience for me, and Jason for his encouragement and witty ideas.

I thank my groupmates Craig Sunada, Dinos Mavroidis, Peng-yun Gu, Tom Corrigan, Nathan Rutman, Jeff Cole, Bob Beretta, Kazuya Yoshida, and my officemates Ssu-Hsin Yu and Rienk Bakker, for their company and words of advice.

My special thanks to Norman Berube, who taught me everything I know about machining.

I hereby also recognize Thomson Industries, Inc.'s generous donation of a linear slide table, model 2EB-08-FTB-F L30.

To the following people who made my life here in MIT so much more fulfilling.....

I would like to thank my co-Co-Co's in the Hong Kong Student Society of MIT: Julian Lee, Preston Li, Ernie Yeh, Calvin Yuen, and Kayan Wong, for working together with me in making HKSS 94-95 such as beautiful year to remember. Also, thanks to all the members who's enthusiasm is immeasurable. Without them, it would have been impossible for Intercollegiate Hong Kong Week '95 to take flight.

My heartfelt thanks to the brothers and sisters in Hong Kong Student Bible Studies, who's love, care, and prayer support carried me through the toughest times in my two year stay in MIT. Especially to my cell groupmates, Kai-Pak Chan and Kinhong Kan: your open ears and understanding hearts are priceless.

To my two apartmentmates, Doug Chan and Ellen Hazen: Thank you for being so wonderful. And, how can I forget, Courtney the most lovely cat I know.

Thanks to all my fellow DJ's in the radio show *Touch of Hong Kong*, who's dedication to the show gave our imaginations a chance to become

realities. I simply just cannot describe how much fun and excitement I had in the studio.

To all my fellow mid-night snackers I thank - Jerome Lui for organizing the events; Albert Wong for providing transportation; and Tony Wong for constantly challenging my "Black Hole" title.

I must thank my "souters", Phoebe Fu, Vinci Leung, Jenny Lee, and Vivian Cheung, who provided me, the "soupee", the essential food group - soups - to survive the countless number of all-nighters in MIT.

To my many friends in MIT, Jocelyn Chen, Andy Shum, Felix Lo, Elaine Yiu, King-Chung Yu, Zachary Lee, and Levina Ha: Thanks for sharing this memorable time with me.

To my two best friends in MIT - Bernard Wong and Patrick Leung. Knowing you is definitely one of the best things that happened to me. The long talks at night, the fun we have in HKGSS, and trips we took together, keep us so close even when we are thousands of miles apart.

My love and thanks go to my Parents, for their unconditional love and consistent support, unmatched by any persons in the world. I would also like to thank Derek, for teaching me how to be an older brother, and making my life so much more interesting.

Most importantly, I would like to thank the almighty God, who has revealed Himself to me and given me the greatest love of all.

Table of Contents

CHAPTER 1 INTRODUCTION.....	11
BACKGROUND.....	11
RESULT OF THIS PROJECT.....	12
PURPOSE OF THIS THESIS.....	12
OUTLINE OF THIS THESIS.....	13
CHAPTER 2 INTELLECTUAL REQUIREMENTS AND DESIGN OBJECTIVES.....	15
INTRODUCTION.....	15
PREPARATORY WORK.....	16
<i>Interviews with the Faculty</i>	16
<i>Literature Review</i>	19
<i>Student Interaction</i>	20
INTELLECTUAL REQUIREMENTS FOR THE LABORATORY.....	22
DESIGN OBJECTIVES.....	27
SUMMARY.....	28
CHAPTER 3 OVERALL DESIGN OF THE NEW SYSTEM DYNAMICS AND CONTROLS LABORATORY.....	29
INTRODUCTION.....	29
OVERALL DESIGN SCHEME.....	29
ORGANIZATION OF THE COURSE.....	30
<i>Class Schedule</i>	30
<i>Class Size and Number of Stations</i>	30
<i>Teaching Assistant Schedule</i>	31
<i>Method of Communication</i>	31
PHYSICAL SET-UP DESIGN.....	32
<i>Overall Picture</i>	32
<i>Modular Design Approach</i>	32
<i>Computer Module</i>	35
<i>Electrical Module</i>	36
<i>Physical Modules</i>	37
SOFTWARE DESIGN AND SELECTION.....	38
WORKSTATION DESIGN.....	39
LABORATORY FLOOR PLAN.....	40
SUMMARY.....	40
CHAPTER 4 ELECTROMECHANICAL MODULE - THE LINEAR SLIDE TABLE...42	42
INTRODUCTION.....	42
SELECTION OF THE ELECTROMECHANICAL MODULE.....	42
DYNAMICS OF THE LINEAR SLIDE TABLE.....	43
DESIGN AND FABRICATION OF THE ELECTROMECHANICAL MODULE.....	44
COMPATIBILITY WITH THE ELECTRICAL AND COMPUTER MODULE.....	52
<i>Electrical Module Calibration</i>	52
<i>Computer Module Modification</i>	53
PERFORMANCE SPECIFICATIONS OF THE LINEAR SLIDE MODULE.....	53
MAINTENANCE FOR THE LINEAR SLIDE MODULE.....	54
<i>Lubrication of Ball Screw Assembly</i>	54
<i>Calibration of Servo Amplifier</i>	54
COST BREAKDOWN OF THE LINEAR SLIDE SYSTEM.....	56
SUMMARY.....	56
CHAPTER 5 DEVELOPMENT OF REAL-TIME CONTROL SOFTWARE.....	57
INTRODUCTION.....	57
RTC SOFTWARE DESIGN CONCEPT.....	58
<i>The Building Block Approach</i>	58

<i>The RTC Big Picture</i>	58
PROGRAMMING THE I/O BOARDS	60
<i>General Programming for DATEL PC-412A</i>	60
<i>General Programming for US Digital PC7166</i>	62
<i>General Programming for Video Blaster SE100</i>	63
RTC BUILDING BLOCKS	64
<i>Data Acquisition</i>	65
<i>Controllers</i>	68
<i>Command Signal Output</i>	69
<i>Timer</i>	70
<i>Reference Signal</i>	75
<i>Input</i>	77
<i>Output</i>	78
SUMMARY	79
CHAPTER 6 DEVELOPMENT OF GRAPHICAL USER INTERFACE	80
INTRODUCTION	80
OVERVIEW OF GUI FUNCTIONS	81
THE GUI ARCHITECTURE	82
MOSAIC AND THE WORLD WIDE WEB	83
MATLAB NOTEBOOK SUITE / MICROSOFT WORD	84
<i>Laboratory Report Template</i>	84
<i>How to Create an Laboratory Report Template</i>	85
MATLAB GUI	85
<i>MATLAB GUI Architecture</i>	85
<i>How is a MATLAB GUI Page Written?</i>	87
<i>Parameter Input Group</i>	88
<i>Analysis Group</i>	90
<i>Real-Time Control Group</i>	94
DEVELOPING NEW MATLAB GUI SETS	96
SUMMARY	96
CHAPTER 7 STUDENT TESTING	98
INTRODUCTION	98
STUDENT TESTING	98
<i>Demographics of students</i>	98
<i>Test Method</i>	99
<i>Survey</i>	99
<i>Results</i>	100
SUGGESTED IMPROVEMENTS	102
<i>Graphical User Interface</i>	102
<i>Electromechanical Module</i>	102
SUMMARY	103
CHAPTER 8 CONCLUSION	104
CONTRIBUTION OF THIS WORK	104
IMPACT OF THIS PROJECT ON FUTURE LABORATORY COURSE DEVELOPMENTS	105
BIBLIOGRAPHY	106
APPENDIX A INTERVIEW CONTACTS	109
APPENDIX B CLASS ENROLLMENT HISTORY	111
APPENDIX C PROPOSED COURSE SCHEDULE	112
APPENDIX D PROPOSED TEACHING ASSISTANT SCHEDULE	114

APPENDIX E COST BREAKDOWN OF COMPUTER MODULE.....116
APPENDIX F DYNAMICS OF LINEAR SLIDE TABLE SYSTEM.....117
APPENDIX G A SAMPLE OF LABORATORY HANDOUT AND REPORT121
APPENDIX H STUDENT TESTING SURVEY FORM.....132

List of Figures

Figure 1. A Current Controls System Laboratory Workstation.....	14
Figure 2. Primary Purpose of the Laboratory.....	23
Figure 3. Overall Physical Set-up Design.....	33
Figure 4. Schematics of Physical Set-up - Student Interaction.....	34
Figure 5. Workstation Layout Design.....	40
Figure 6. Laboratory Floor Plan.....	41
Figure 7. Photograph of Electromechanical Module.....	45
Figure 8. Overview of the Linear Slide Table Plant.....	46
Figure 9. Rigid Coupling and Inertial Load.....	48
Figure 10. Friction Device.....	49
Figure 11. CCD Camera and Mount.....	50
Figure 12. Electrical Switch Circuit Diagram.....	51
Figure 13. Circuit Diagram for Servo Amplifier Calibration.....	55
Figure 14. Schematic of Software Communication Architecture.....	57
Figure 15. The RTC "Big Picture".....	59
Figure 16. Schematic of A/D Building Block.....	66
Figure 17. Internal Structure of A/D Building Block.....	67
Figure 18. Schematic of Encoder Block.....	68
Figure 19. Schematic of the Open Loop Controller Block.....	68
Figure 20. Schematic of PID-Controller Building Block.....	69
Figure 21. Schematic of D/A Building Block.....	70
Figure 22. Schematic of the Timer Building Block.....	73
Figure 23. Schematic of the Vision System Building Block.....	76
Figure 24. Schematic of the Configuration Block.....	77
Figure 25. Schematic of the Graphical User Interface Block.....	78
Figure 26. Schematic of the Output Block.....	79
Figure 27. Schematic of Software Communication Architecture.....	80
Figure 28. Functions of the Graphical User Interface Software.....	81
Figure 29. GUI Architecture of the Laboratory.....	83
Figure 30. MATLAB GUI Architecture.....	86
Figure 31. Typical MATLAB GUI Menu Page.....	86
Figure 32. A Typical MATLAB GUI Script.....	87
Figure 33. Parameter Input Group.....	88
Figure 34. Analysis Group.....	90
Figure 35. Sample Output from Control Analysis Function.....	91
Figure 36. Typical Simulink Block Diagram.....	92
Figure 37. Typical Frame of Animation.....	92
Figure 38. Typical Simulation Data Display.....	93
Figure 39. Real-Time Control Group.....	94
Figure 40. Typical Experimental Data Display.....	95
Figure 41. Volunteer Student Demographics by Year.....	98
Figure 42. Survey Result.....	100
Figure 43. Survey Result.....	100
Figure 44. Survey Result.....	101
Figure 45. Schematic of the Linear Slide Table.....	117
Figure 46. Typical Velocity Response to Step Input.....	118
Figure 47. Typical Experimental Open Loop Step Velocity Response.....	118
Figure 48. Typical Friction Value Determination Plot (without friction device).....	119
Figure 49. Velocity Response to Step Input (with additional inertial load).....	119
Figure 50. Typical Frictional Value Determination Plot (with friction device).....	120
Figure 51. Typical Velocity Response to Step Input (with friction device).....	120

LIST OF TABLES

Table 1. Cost Breakdown of the Electromechanical Module.....	56
Table 2. Memory Map for PC-412A.....	61
Table 3. Memory Map for US Digital PC7166.....	63
Table 4. Functions of SE100 API.....	64
Table 5. RTC Building Blocks.....	65
Table 6. Summary of the Counter Functions and Mode Settings.....	72
Table 7. Reference Input Parameters.....	89
Table 8. Class Enrollment History for Course 2.14 (Control System Principles).....	111
Table 9. Cost Breakdown of the Computer Module	116

Chapter 1 Introduction

Background

System dynamics and controls is an integral part of a modern undergraduate education curriculum in the mechanical engineering discipline. In 1990, 72.5% of the mechanical engineering departments in the United States have at least one required course with a significant focus on controls [Feliachi 1991].

It is widely recognized that a practical laboratory course in system dynamics and controls is essential for an effective education program. Yet, only 35% of electrical and mechanical engineering departments in the United States offer undergraduate controls laboratories [Feliachi 1991].

At the Massachusetts Institute of Technology, the Department of Mechanical Engineering offers a series of undergraduate courses in the area of system dynamics and controls, including one that contains a laboratory component. The current Controls System Laboratory was developed in the mid-1980's and has been in service for ten years with little upgrading or modernization. With an average of approximately 160 students per year, the controls system laboratory has been under intensive usage and much of the equipment is worn out. Moreover, the technology used in the current laboratory is quite dated. For example, in the industrial world today, microprocessor-based real-time control is a standard. In the current laboratory, however, the students still build their controllers by "patching" an analog computer. With the lack of renovation, heavy wear and tear, and the employment of old technology, the current controls system laboratory has lost much of its educational value. Both the faculty and the students have expressed the need for substantial laboratory improvement.

In light of the existing situation, during the Fall Semester of 1993, the Department of Mechanical Engineering at the Massachusetts Institute of Technology launched a study into redeveloping the controls system laboratory. It is anticipated that this project will not only bring in state-of-the-

art technologies to the system dynamics and controls curriculum at MIT, but will also serve as a model for engineering schools in other universities as they develop similar laboratories. A team of two masters students was supervised by a faculty member in the department to carry out this project. The team surveyed the faculty and reviewed the intellectual needs of the laboratory. Literature search and site visits to other universities' controls system laboratories were conducted to investigate the approach of other engineering schools. In addition, the two graduate students became the laboratory teaching assistants to learn, first hand, about the strengths and short comings of the current laboratory. The team was later joined by two undergraduate thesis students from the Mechanical Engineering Department to assist the overall progress by specializing in developing certain areas of the project.

Result of this Project

As the result of this two year study, a modular and flexible laboratory prototype set-up was designed, fabricated, and tested. The prototype system uses commercially available hardware and software, plus in-house modifications and tailoring, to build a user friendly and educationally effective System Dynamics and Controls Laboratory workstation. It is hoped that this prototype can serve as a model for future development of similar types of laboratories at other engineering schools. Moreover, the modular design approach employed in the creation of this prototype is adaptable to develop educational laboratories in other engineering disciplines.

Purpose of this Thesis

This thesis, in conjunction with the master's thesis of M. Tesciuba [Tesciuba, 1995], and the bachelor's theses of J. Hintersteiner [Hintersteiner, 1995] and G. Soto [Soto, 1995], documents the design and development of a new System Dynamics and Controls Laboratory prototype set up. These four theses combined outline the important intellectual content of the system dynamics and controls courses taught in MIT and the overall design of the new laboratory. They cover the technical details regarding each component of the whole laboratory system, propose a laboratory schedule, and present guidelines for future development and expansion of the system. In

particular, the master's thesis for M. Tesciuba focuses on the electrical and the fluid tank module, as well as the physical layout of the new controls system laboratory floor plan and workstation set-up. The bachelor's thesis written by J. Hintersteiner concentrates on the development of the network interface of the laboratory computer system with the MIT school-wide computer network, Athena. It also covers some aspects of developing the real-time control algorithms for the physical plants. The bachelor's thesis by G. Soto specifies the design and technical descriptions for the fluid tank physical module.

An emphasis of this thesis is on the design and construction of a physical experiment module. Specifically, the design of a ball screw driven linear slide table system design, along with an evaluation of its performance, is documented in this thesis. Another emphasis of this thesis is the creation of the fundamental software programs for the project. The overall software structure design, the fundamental software building blocks for running real-time control, and the graphical user interface are described in detail. Some of the material contained in this thesis is taken from the companion theses [Hintersteiner, 1995] [Soto, 1995] [Tesciuba, 1995] and presented here for clarity.

Outline of this Thesis

This thesis is organized into eight chapters. This chapter introduces background and the motivation of the project, as well as a general overview of the thesis itself.

The second chapter describes the preparatory work that led to the formulation of the intellectual requirements and design objectives of the new system.

Chapter 3 presents the overall system design and general descriptions for each component, including the Computer Module, the Electrical Module, and the two Physical Modules. An overview of software configuration for the system is also introduced. Furthermore, logistics such as a proposed class schedule, course enrollment size, laboratory workstation layout, and a floor plan are included.

Chapter 4 documents the design of the linear slide module in detail. It also summarizes the performance specifications of the linear slide system, and presents maintenance procedures and a cost breakdown.

The main components of the real-time control software are detailed in Chapter 5. It illustrates the real-time control software architecture and the building block design approach which allows future developers to quickly set up the necessary supporting software for future physical plants.

Chapter 6 focuses on the development of user interface software. General descriptions on the user interface organizations as well as in depth discussions on creating graphical user interface applications are covered.

Chapter 7 summarizes the results of student testing of the new set up and improvement suggestions.

Chapter 8 is the conclusion chapter. It reviews the contributions made by this project and explores the potential impact on future laboratories development.

A series of comprehensive appendices provide coverage on the details of the work, such as logistics of the future course and component specifications for the hardware, as well as a cost breakdown of the prototype set-up modules and sample laboratory exercise handouts.

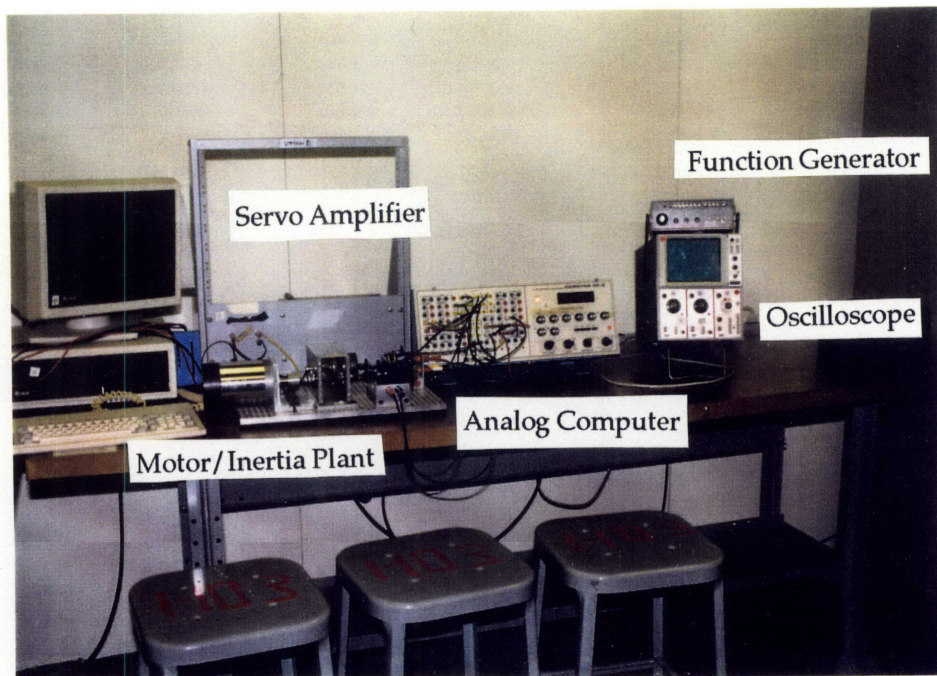


Figure 1. A Current Controls System Laboratory Workstation

Chapter 2 Intellectual Requirements and Design Objectives

Introduction

Prior to actually designing the new controls system laboratory, the project team had to investigate the needs of the clients. The two primary clients of this project are the faculty of the Mechanical Engineering Department at MIT and the students. The team started the project by carefully reviewing the materials taught in the course lecture and laboratory. At the same time, the team individually interviewed system dynamics and controls faculty members in Mechanical Engineering Department at MIT to obtain their views on the development of the new laboratory. Later, the team also had various opportunities to meet with faculty members outside the Mechanical Engineering Department and outside MIT to discuss contemporary issues in developing educational system dynamics and controls laboratories. An extensive literature search was also conducted to understand the current efforts of other engineering schools developing similar laboratories.

To better understand the need of the students, the two graduate members of the team were assigned the positions of laboratory teaching assistants for three consecutive regular semesters (Spring 94, Fall 94, and Spring 95). This teaching opportunity allowed the team to find out, first hand, the strengths and weaknesses of the current Controls Systems Laboratory from active interactions with the students.

The preparatory work led to formulating the intellectual requirements and defining the design objectives of the new System Dynamics and Controls Laboratory. This chapter documented the results of the faculty interviews, literature search, and student interactions. At the end of this chapter, the intellectual requirements and design objectives of the laboratory are presented.

Preparatory Work

Interviews with the Faculty

During the first stage of the project, the primary effort of the team was to gather information on the requirements of the new laboratory, which would define the scope of the project. The first step of achieving this task was done by interviewing professors knowledgeable in this area.

Altogether, a total of 23 professors were interviewed in person. Among these, 17 were MIT Mechanical Engineering faculty, one was an MIT professor from Ocean Engineering Department, and five of them were non-MIT professors who have expertise in system dynamics and controls education at other universities. In addition, three more professors from other universities were contacted by electronic mail and by telephone (see Appendix A).

The professors contacted had a common general view on what the new System Dynamics and Controls Laboratory should provide. In their opinion, the laboratory should be a place where the students can enhance their understanding in a physical sense such as by seeing and touching of the materials they have learned in the course lectures. However, their ideas on the means to achieve this goal were quite diverse. The following paragraphs summarize the main issues raised during the interviews. These main issues can be categorized into two groups: (1) Issues regarding the physical set-ups of the laboratory; (2) Issues regarding to the running of the laboratory course.

Summary of the Interviews

The following issues are in regard to the physical set-ups:

Digital Equipment vs. Analog Equipment

The choice of using digital equipment in the new laboratory or continue using analog equipment was discussed. The use of digital equipment will yield higher quality results. However, the digital aspects of control should be “masked”, i.e. the control experiments should appear as if they were operating in the continuous time domain. On the other hand, students can benefit from preserving analog computers in the laboratory curriculum since they are a good

way to implement block diagrams. Moreover, the importance of hands-on experiments in analog circuits, particularly, Op-Amp circuit building, was acknowledged. The addition of Op-Amp experiments allow students to be exposed to lead/lag compensator hardware building blocks.

Easy Interfacing

Professors recognized the need for easy interfacing with the laboratory set-ups (easy input and easy-to-see output) so that the students can concentrate their efforts in absorbing and understanding the important concepts instead of devoting excessive time getting the system to work. For example, the Macintosh style (use of windows, icons, click and select features with a mouse, etc.) graphical user interface was suggested.

Variety of Physical Set-ups

A popular idea among the professors was to include a number of different set-ups from various Mechanical Engineering disciplines, such as vibration, fluids, thermodynamics, and heat transfer, in addition to the traditional translation and rotation dynamics models in the laboratory. These physical systems can also be used for illustrating the modeling concepts presented in the introductory level system modeling course.

Reality in the Laboratory

Some professors mentioned that the material presented in the lectures is often too simplistic (for instance, noise and friction are ignored), and the laboratory should serve to give the students a taste of reality. Moreover, the use of actual engineering systems from industries as laboratory equipment was suggested to allow students becoming familiar with industry standards as well as keeping up with the latest technology.

Lego-Like System

Using a "Lego-Like" system design concept in the laboratory physical set-ups can enhance the hardware's modularity. For example, a laboratory station set-up which allows the students to place sensors of various qualities in different locations can give the students a feeling of both sensor noise and observability.

Up-to-date Equipment

The laboratory should be set up with the most up-to-date and widely used equipment, both in hardware and software.

The following issues are in regard to the running of the laboratory course:

Simulations

The idea of pre-laboratory simulations was discussed. The primary advantage for pre-laboratory simulations is to give the students the opportunity to exercise their modeling skills with the use of computer software. Simulation is also a good way for the students to predict the response and compare the theory to the practical results. Moreover, carrying out pre-laboratory simulations will allow the students to get familiar with the experiments prior to class time, and therefore to learn more when they carry out the laboratory experiments.

In-Class Demonstration

Once a week, ten-minute lecture demonstrations with robust and reliable physical systems was suggested. It was also suggested that set-ups in the new System Dynamics and Controls Laboratory be used for this purpose.

2 Hours-per-week versus "Live-in" Laboratory

The issue of how much time each laboratory session should take was considered. The "live-in" type of laboratory requires the students to spend about 8 hours per week, and it will be an "open-laboratory" format (i.e. the student can go in at any time of the day). This laboratory format would allow the students to get in-depth experience in controls, and more topics could be illustrated in the laboratory. The current 2-hour-per-week format requires the complexity of the experiments to be minimized so that the laboratory takes approximately 2 hours per week to finish.

Rotation Lab Set-ups

A new laboratory teaching philosophy, which involves having various groups of students work on different set-ups every week, was discussed. The benefit of this organization is to allow the students to work on a greater variety of systems without investing a large sum of

money to build several identical set-ups. On the other hand, giving the students too many systems to work on may prevent them from fully understanding any particular set-up.

Design Project

The issue of having a design project at the end of the term was raised. The design project could allow the students to sum up and apply the concepts they learned in class, but it could also possibly lead to a great deal of paper work and become a burden to the learning process.

Literature Review

Extensive literature searches on topics related to the development of educational system dynamics and controls laboratories was conducted in the early stage of the project. The primary sources of the literature were proceedings from American Control Conferences from 1988, 1991, and 1993, in which over two dozen papers were published on this topic. Other papers were obtained through private contacts with professors at other universities and engineers in the industry. The literature search provided the project team with a general overview of how other engineering schools approach the challenge of designing an effective educational controls laboratory. Some general observations were made from the literature search and they are listed as follows:

- Digital computers are commonly used for data acquisition, real time control, and simulations. They can also replace many devices such as oscilloscopes, function generators, etc. [Baish et al, 1988] [Huang et al, 1993] [Kopacek et al, 1988] [Kulakowski, 1988] [Vagners, 1993]
- Flexible laboratory set-up design are widely adopted. This makes the set-up capable of controlling a wide variety of processes and physical plants that could be of interest to students of many disciplines. [Baish et al, 1988] [Bonner et al, 1988] [Erickson, 1993] [Kulakowski, 1988] [Levine et al, 1993] [McDermott et al, 1988]
- The content of the laboratory should provide connection to reality, thus giving the curriculum relevance and reinforcement to the academic material by the need to actually apply it to achieve an experimental objective. [Auslander et al, 1988]

- A quote on defining the criteria for judging a physical system for a controls laboratory is found: “control objectives must be safe, reasonably fast but not too fast, have some ‘real world’ irregularities, but not so much that control theory never works, not break too often, and not cost too much.” [Auslander et al, 1988]
- Extensive variety of software (both commercial and customized) for simulation, analysis, and visualization are used. [Astrom, 1991] [Bodruzzaman, 1992] [Cheok et al, 1993] [Das et al, 1993] [Eaton, 1994] [Feliachi et al, 1991] [Franklin et al, 1988] [Huang et al, 1993] [Levine et al, 1993] [Schaufelberger, 1988] [Vagners,1993]
- The laboratory should place emphasis on hands-on experience. [Bonner et al, 1988]
- Students are encouraged to work in groups rather than individually. [Franklin et al, 1988]

Student Interaction

Since the ultimate beneficiary of this laboratory will be the students, their views on the current system dynamics and controls laboratory in the MIT Mechanical Engineering Department played an influential role in the team's subsequent design decisions on the new laboratory. Recognizing this fact, the two graduate students on the project team were assigned the positions of laboratory teaching assistants for the existing System Dynamics and Controls Laboratory. This teaching experience enabled the two graduate students to have frequent opportunities to interact with the students, through which the strengths and weaknesses of the current laboratory were better understood.

Student Demographics

During the three semesters of Spring 1994, Fall 1994, and Spring 1995, a total of 178 students were enrolled in the laboratory (Appendix B). The majority of the enrollments were juniors and seniors from the Mechanical Engineering Department. However, a handful of sophomores and graduate students were also part of the composition. In addition, small numbers of students from other departments, such as Electrical Engineering and Sloan

Business School, contributed to the total enrollment. Since the new laboratory is targeted mainly for upperclassmen in the Mechanical Engineering Department, the focus of this section will be on this group.

Observations from Student Interactions

From the interactions with the students in the current laboratory, the following strengths and weaknesses were observed.

Strengths

Robust Equipment

The current laboratory equipment can be qualified as robust, when one considers what the equipment has gone through in the past ten years - intensive usage by over a thousand students and relatively little maintenance - coupled with the fact that it is still in operational condition today.

Safety

Even under intensive usage of many students, no accident involving physical injury occurred in the laboratory in the past three terms (Spring 1994, Fall 1994, Spring 1995).

Provides Sufficient Hands-on Experience

The students seem to appreciate the hands-on experience provided by the laboratory set-ups. The students are able to employ their senses when they see the motor spin or when they use their hands to turn the motor and feel the spring like effect from a P-controller.

Compact Size

The physical plant has a relatively compact size - the whole system can be placed on a 8' by 2.5' work bench. This permits the accommodation of several workstations and over a dozen students to work in the laboratory room at the same time.

Weaknesses

Overhead in Learning Analog Computer Operations

Most upperclassmen, by the time they enroll in this laboratory, have theoretical and/or practical experience with analog circuits such as operational amplifier circuits. In general, however, their transition to the usage of the Comdyna analog computer does not prove to be a smooth one. Students spend significant amount of time learning to operate the analog computer. This overhead is a main cause of student frustration in the laboratory and consumes valuable laboratory time.

Lack of interesting experimental set-ups

Even though the current physical plant (motor - inertial load) serves the educational purpose, it lacks fun and excitement.

Too many components in the set-up

The existing set-up consists of five components per workstations: a Comdyna analog computer, a motor/inertial load plant, a voltage-to-current amplifier, an oscilloscope, and a function generator. The cables connecting these components can become unnecessarily complicated at times and thus diminish the educational value of the laboratory.

Equipment unable to perform the necessary tasks

In one of the experiments, the students are asked to apply a constant frictional torque to the system. However, there is no device in the physical set-up which allows the students to apply such a disturbance in a consistent manner. Another example is, in more than one instance, the analog computers are required to output a calculated control signal greater than its maximum voltage output, thus causing the system to be overloaded and sometimes become unstable.

The new laboratory design aims to preserve the strengths of the current laboratory set-up and combat the weaknesses.

Intellectual Requirements for the Laboratory

At the end of the first stage, the project team evaluated the results gathered from the preparatory work described above, and formulated the intellectual requirements for the new System Dynamics and Controls

Laboratory. The intellectual requirements served as the measurement standard in the team's design decisions as the project progressed. In other words, this intellectual requirements were the project's "constitution".

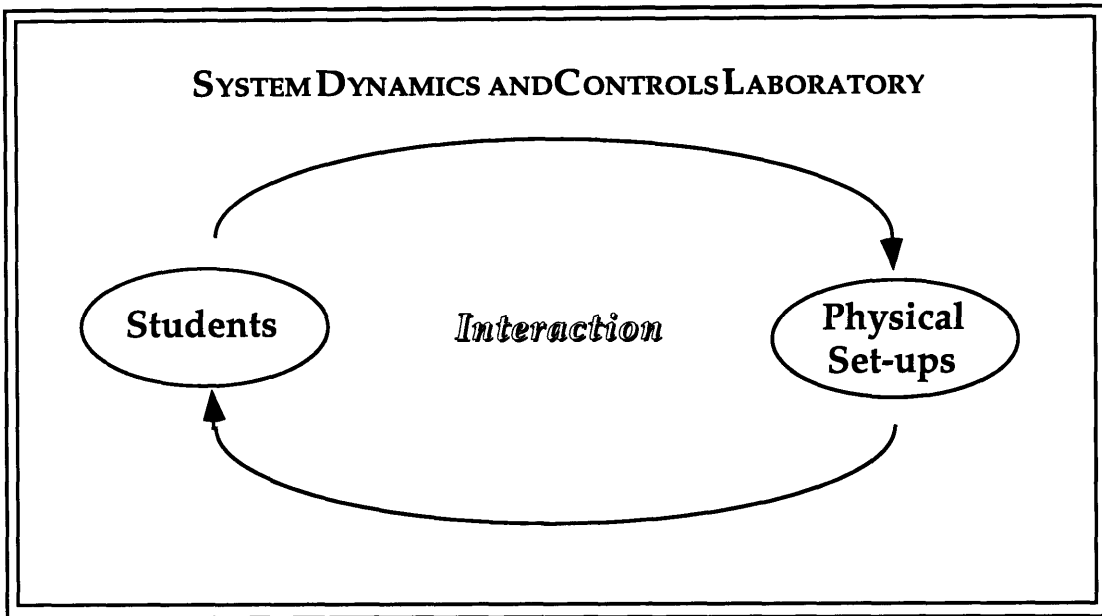


Figure 2. Primary Purpose of the Laboratory

The intellectual requirements are the concepts covered in the system dynamics and controls courses to be conveyed to the students through the laboratory. It is important to recognize that the purpose of the laboratory is not to teach the students those concepts, but rather to provide them with the means to feel, experience, and understand them. This will be achieved by the interaction between the student and the laboratory set-ups (see Figure 2).

The content of the intellectual requirements, and the proposed means to achieve them, are detailed as follows:

Modeling / System Identification

Intellectual Requirement:

Introduce students to the process of developing a model for a system. This includes applying physical laws and analytical methods to derive the dynamic equations and transfer functions of a system.

How to achieve:

The students are provided with a system in the laboratory and are asked to:

- derive equations of motion for the system.
- perform linearization to the equations of motion.

- perform Laplace transforms to convert from time-domain to s-domain.
- write out the transfer function for the system.
- present the system in the form of block diagrams.
- physically measure the value of system parameters to obtain a complete model.

Intellectual Requirement:

Illustrate to the students that many different systems, such as an electromechanical system and a fluid system, although different in nature, can have similar dynamics and can be modeled and controlled by similar methods.

How to achieve:

The new System Dynamics and Controls Laboratory should include various experimental set-ups which represent different areas of mechanical engineering, such as, electromechanical systems, fluid systems, thermo systems, and so forth.

Simulation

Intellectual Requirement:

By means of simulations, predict, compare, and verify the model derived earlier with the experimentally derived results.

How to achieve:

The students are asked to:

- use computers and software tools to simulate the model's dynamic behaviors.
- conduct experiments on the physical set-up and obtain experimental data.
- compare and verify the simulated data with the experimental data.

Dynamic Behaviors

Intellectual Requirement:

Observe and understand how changes in a system's parameters affect the response.

How to achieve:

The students are asked to:

- change the physical plant's parameter(s), such as its inertia, damping, stiffness, etc., or
- change the environment parameter(s), such as friction, external disturbances, etc.
- observe physically how the system's behaviors, such as the steady state error, rise time, settling time, overshoot, time constant, time delay, number of oscillations, etc., are affected after the changes are implemented.
- compare the system's stability versus its performance.

Intellectual Requirement:

Relate the observations in the experiment to the theories learned in the lecture.

How to achieve:

The students will attempt to explain these changes by Newton's Laws, from the system's transfer function, and by fundamental dynamics and controls theory learned in lecture.

Control Concepts / Feedback

Intellectual Requirement:

Compare the response of feedback control to open loop control in terms of accuracy, sensitivity, transient response, stability, disturbance rejection, and other performance criteria.

How to achieve:

The students are to:

- implement closed loop feedback on the system.
- perform experiments, and alter plant and environment parameters similar to the procedures described in the section "Dynamic Behaviors".
- observe the closed loop system response and compare it to the open loop response, particularly focusing on the improvements on accuracy, sensitivity, disturbance rejection, transient response and stability.

Controller Design

Intellectual Requirement:

Explore the effects of various controllers, explain them using theories learned in lectures, and describe how they can be applied to meet design specifications.

How to achieve:

The students will:

- implement various types of controllers.
- use analytical tools, such as root locus, Bode, Nyquist, Routh-Hurwitz, etc., to explain the effects of the controller to the systems damping, gain margin, phase margin, stability, etc.

Intellectual Requirement:

Design and implementation of controllers to make the system behave as desired.

How to achieve:

The students will:

- implement P, PD, PID, and lead/lag compensators on the laboratory physical systems.
- fine tune the controllers to meet a set of performance specifications.

Introduction to Discrete Time

Intellectual Requirement:

Introduce digital control and the concept of discrete-time, Zero-Order-Hold, and digital control hardware.

How to achieve:

The students will:

- alter sampling frequency of the controller and observe the effects on the system's response.
- open up a digital computer and get a glimpse of the necessary hardware, such as A/D, and D/A boards.

Advance Topics in Controls

Intellectual Requirement:

Introduce advance topics in control, for instance, MIMO systems, non-linear control, stochastic control, optimal control, adaptive control, etc.

How to achieve:

This topic is not intended to be implemented at this stage of the development. However, the new laboratory has a modular structure

that permits advance topics in system dynamics and controls to be incorporated relatively easily by adding appropriate experiments.

Design Objectives

The intellectual requirements of the System Dynamics and Controls Laboratory serve as the measurement criteria for the design of the laboratory. However, in order to practically implement these requirements in the new laboratory, a set of design objectives has to be defined.

Besides fulfilling the intellectual requirements, the design objectives should also include the framework for the new laboratory to become an effective educational tool, and to be a worthwhile investment for the Mechanical Engineering Department. The team reviewed their results from the literature search, interviews, and student interactions. After careful considerations, the team listed the following as the design objectives for the new system dynamics and controls laboratory, which are categorized into two groups: goals and constraints.

Goals:

1. Open architecture
Adaptable to adding new control experiments from various mechanical engineering disciplines;
2. Robustness of hardware and software
Physical hardware should be durable; software easily adaptable to tailor needs of physical hardware; computer hardware upgradable;
3. Minimize initial cost
4. Minimize running cost
Avoid annual fees, such as software licensing fees; low cost and easily available spare parts; laboratory handouts available on-line;
5. Minimize instructor workload
Automatic mentor by computer such as on-line help capabilities;
6. Optimize student exposure to the laboratory
Two students per laboratory station;
7. Interesting / fun
May include a small yet challenging and competitive design project.

Constraints:

1. Safety
2. Size of individual physical set-up

- The laboratory should be able to fit 5 to 8 stations;
3. The coherence between lecture and laboratory schedule
The materials taught in the laboratory should not precede the lecture materials;
 4. In class time limit for each laboratory session
Students can complete each laboratory session in 2 hours;
 5. Time to complete the project
The project should be completed within the 18-month time period.
With these intellectual requirements and secondary objectives clearly stated, the team began the design process of the new laboratory.

Summary

This chapter detailed the preparatory work conducted in the early stage of the project and presented a summary of the results. The preparatory work laid grounds for formulating the intellectual requirements and the design objective of the laboratory.

In the next chapter, the overall design of the new System Dynamics and Controls Laboratory will be presented. Discussions of the design approach, the physical set-ups, software selections, and logistics about the course will also be included.

Chapter 3 Overall Design of the New System Dynamics and Controls Laboratory

Introduction

In Chapter 2, the intellectual requirements (the “constitution”) of the laboratory as well as the design objectives were described in detail. This chapter will present the overall design of the new system dynamics and controls laboratory.

The chapter will begin with the logistics of the course, including a proposed course schedule, the course enrollment, and more. Then the three modules which make up the laboratory set-up - the Computer Module, the Electrical Module, and the Physical Module - will be introduced. Moreover, their individual functions, along with their interactions between each other, are described. Furthermore, a description of software for the laboratory is presented.

The last sections of this chapter are dedicated to the proposed workstation design and floor plan for the laboratory.

Overall Design Scheme

The overall design of the new system dynamics and controls laboratory covers a number of areas. The first area is the format of the class, or the organization of the course. This includes the number of laboratory sessions per semester, course schedule, number of students per section, number of students per station, teaching assistant time schedule, and method of communication between the instructor and the students. The second area is the design of the physical set-ups according to the class organization. Third, the layout of the workstations and the classroom floor plan needs to be designed in order for the design to be fully implementable.

Organization of the Course

Class Schedule

A typical semester at MIT spans a total of 13 weeks. Excluding the weeks with holidays, course quizzes, and the final examinations, there are nine weeks available for holding laboratory sessions. As a result, the intellectual requirements of this laboratory must be covered in these nine laboratory sessions. Furthermore, the design objectives stated the constraint on the coherence between the laboratory and lecture schedule, which ensures the materials covered in the lecture precedes the materials presented in the laboratory, has to be followed. Considering these issues stated above, the team arrived at the a proposed time schedule for the new system dynamics laboratory [Tesciuba, 1995] and is included in Appendix C. Under this proposed schedule, the new laboratory will have three weeks out of each semester available for other system dynamics and controls courses.

Class Size and Number of Stations

From the enrollment history (see Appendix B), the system dynamics and controls course in the Mechanical Engineering Department can be considered a relatively large class. To accommodate enrollment of this size, without compromising individual student exposure to the physical hardware in the laboratory, the laboratory should provide sufficient resources such that students can work in small groups.

From the team's experience in instructing the laboratory course, two observations were made:

1. It is found that groups of two students promote teamwork, foster academic discussions, and operates most efficiently in terms of time.
2. From the instructors point of view, a maximum class size of 15 to 16 students is reasonable. For a larger class size, students will not be able to receive the necessary attention during the laboratory session.

In order to accommodate all the students, as well as taking into consideration of the above observations, a total of six laboratory sections per semester, with eight set ups in the laboratory, is necessary. This will allow the maximum enrollment to be 96 students per semester.

Teaching Assistant Schedule

One of the design objectives is to minimize teaching assistant's workload. By accomplishing this goal, the teaching assistant will be able to spare extra time and possibly hold laboratory sessions for other courses related to the subject of system dynamics and controls.

Some methods to reduce the workload of the teaching assistant, such as developing an automatic mentoring system, are also mentioned in the secondary objective. The automatic mentoring system is aimed to reduce the time commitment for the teaching assistant to each individual student during the laboratory by providing the students with on-line help capabilities on the computer. The automatic mentoring system is implemented in the prototype design by the use of the World Wide Web (refer to Chapter 6 and [Hintersteiner, 1995]).

A proposed new teaching assistant schedule, along with the current one, is presented in Appendix D for reference.

Method of Communication

The area of information technology has been making great progress in the past few years, providing innovative and effective means of communication. The new system dynamics and controls laboratory design takes advantage of this latest technology and has tapped in a number of features from the information superhighway, keeping the laboratory technologically up-front and moving it a step closer to a paperless system.

The distribution of laboratory manuals and the submission of student reports is done by using MIT's school wide computer network, Athena. The laboratory reading materials and pre-laboratory assignments are available through the very popular World Wide Web [Hintersteiner, 1995]. Ethernet connections for the computers in the laboratory make it possible to tap into the resources in Athena from the laboratory. Moreover, the commercial software, such as MATLAB, used in the laboratory are Athena supported, thus enabling the students to perform simulations and pre-laboratory calculations at Athena before coming to the laboratory, and import these results to the laboratory computers during the experiment for comparison with the results of the physical system. By the same token, the data generated by the real-time control experiments in the laboratory can be sent to the student's Athena account for storage and future reference.

The adoption of such innovative communication methods will promote more efficient time usage and significantly reduce the department's photocopying costs.

Physical Set-up Design

After considering the logistics of the laboratory, the team went on to design the physical set-ups for the new laboratory, with both the intellectual requirements and the secondary objectives in mind. This portion of the chapter describes the overall picture of the physical set-up, then goes into more detail about the individual components of the set-up, why they were selected, as well as how these components are combined to form a stand alone laboratory workstation.

Overall Picture

Figure 3 shows the overall physical set-up design. It is primarily consisted of a Computer Module, an Electrical Module, and one or more Physical Modules. Figure 4 shows the schematic of the connections between these components.

Modular Design Approach

The modular design concept is geared towards the open architecture objective, which allows the laboratory to grow with time and to be less susceptible to becoming obsolete. The way to achieve this goal is to make each one of the modules flexible and compatible with other modules. For instance, the software can be easily developed to suit the needs of each physical plant, the electrical module can be modified to meet the specifications of new actuators or sensors, and the physical module can be replaced, provided that the modifications follow a set of simple guidelines.

The main advantage to this design approach is that it enables graceful modernization of software and hardware components as upgrades become available. Furthermore, it allows the adaptation of more sophisticated physical systems, or the addition of physical modules from other mechanical engineering disciplines.

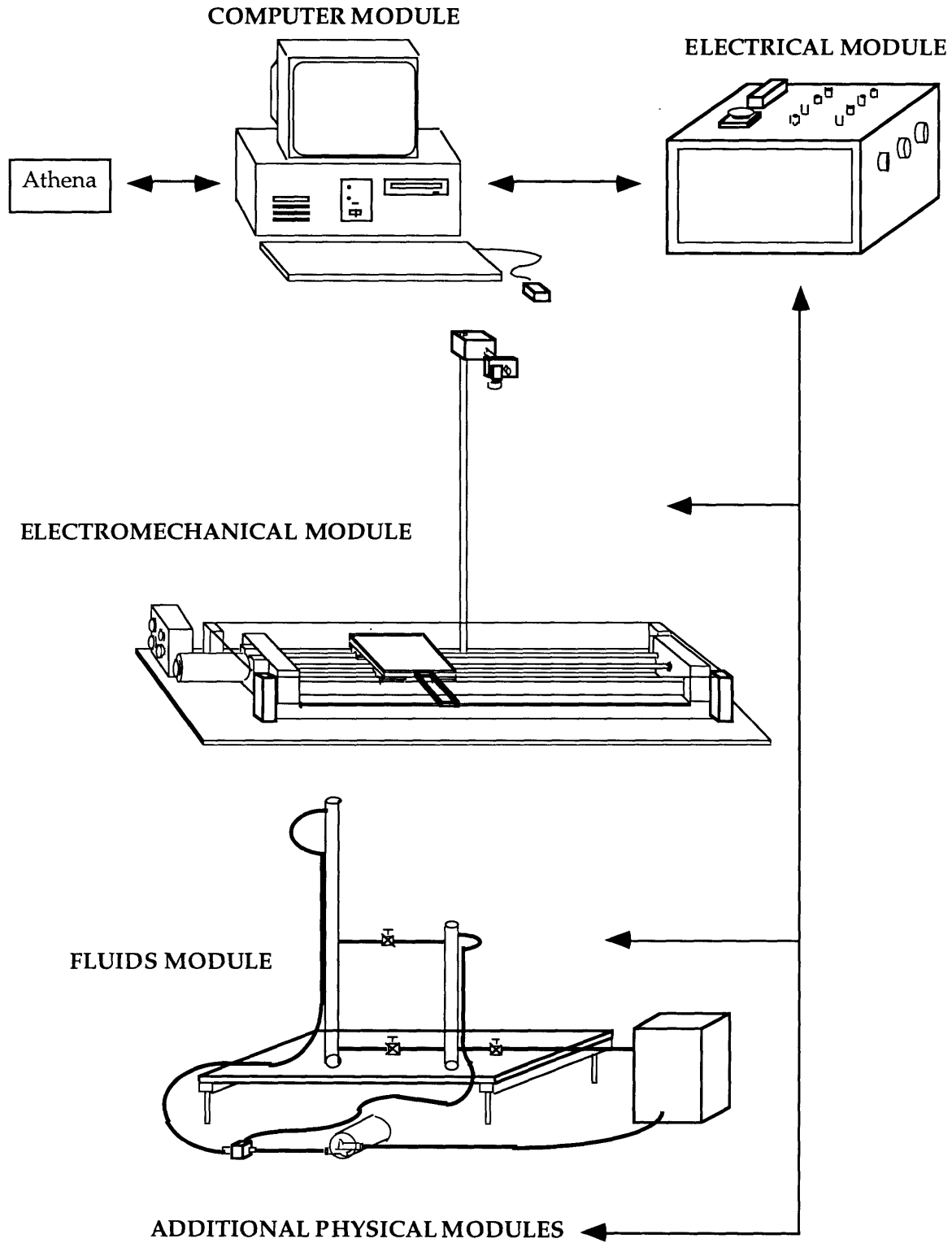


Figure 3. Overall Physical Set-up Design

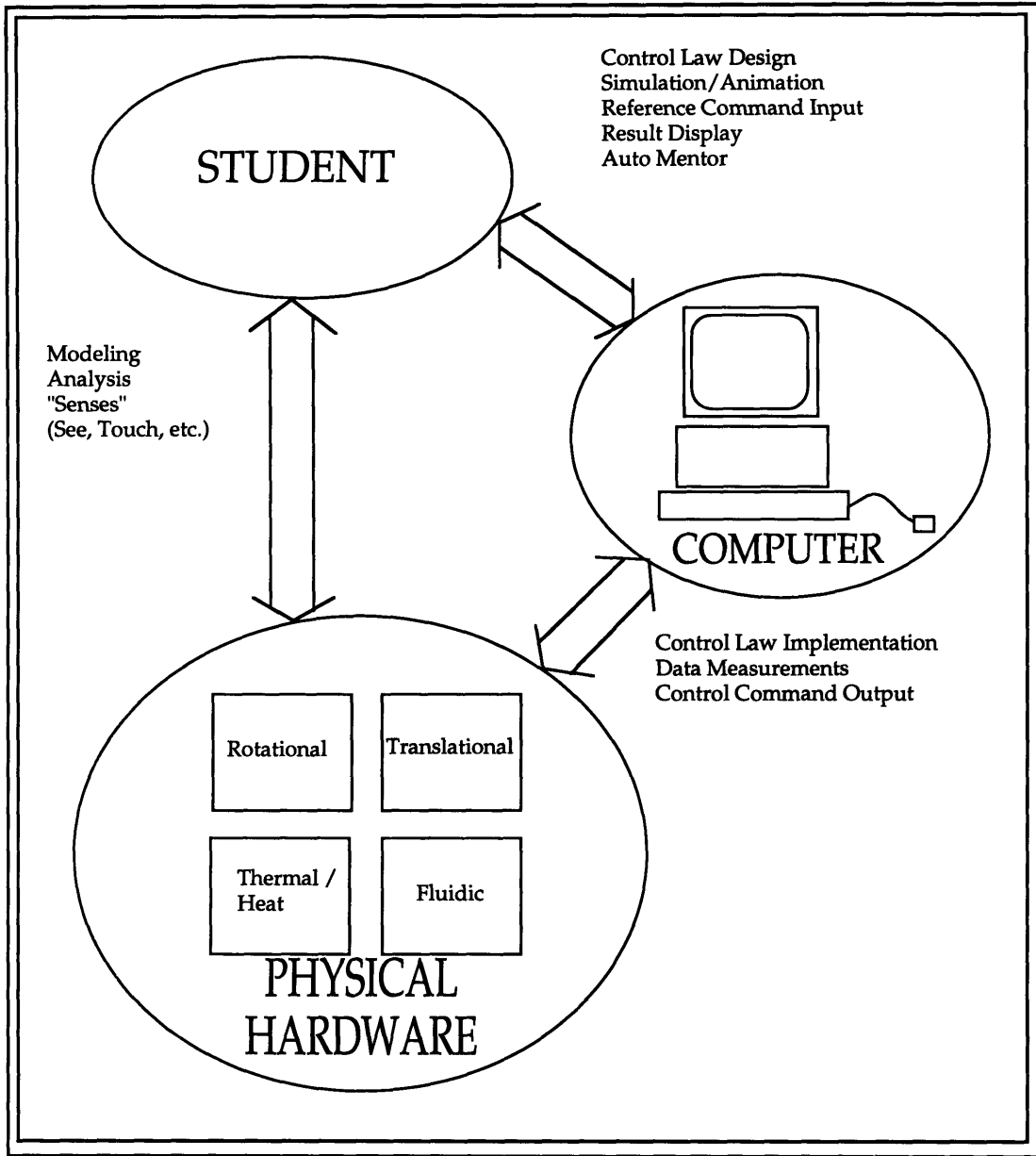


Figure 4. Schematics of Physical Set-up - Student Interaction

One of the intellectual requirements states that the students should have exposure to implementing the controls theory to more than one type of physical system. Therefore in the prototype set-up, two physical modules, a linear slide table and water tank level, which represent the electromechanical and fluid fields of mechanical engineering, have been included.

The following sections will present a more detailed descriptions of the three components that comprise the physical set-up.

Computer Module

Hardware

The heart of the laboratory station is a Dell 466 DX/2 personal computer, operating at 66 MHz with a 80486 processor and five 16 bit ISA expansion slots, a math co-processor, a 17" color monitor, 20MB of RAM and 210MB of hard disk. The clock speed of 66 MHz is selected because of the computational speed needed for performing real-time control algorithms. The math co-processor is necessary for the same reason. The ample RAM capacity is necessary to avoid "crashes" due to the number of applications running at the same time during the operation. A 17" monitor was chosen since the computer has to be viewed by two group members at the same time.

Inside the personal computer is a DATEL PC412A multifunction I/O board, which provides 16 single-ended or 8 differential analog input channels capable of 83.3kHz of sampling rate and 12 bit A/D resolution. In addition, the board has 4 analog output channels and 8 input, 8 output discrete digital I/O channels. In addition, a U.S. Digital PC7166 encoder interface board consisting of 4 encoder counters, is installed in one of the ISA slot in the PC. A Video Blaster SE100 video board from Creative Labs is included in the PC to enable the incorporation of a vision system. For better graphics capability, a Diamond Multimedia Window's accelerator (SpeedStar Pro) with 1MB of video RAM is connected to one of the ISA slot. Last but not least, an ethernet board will be installed in the PC for future connection to Athena. These 5 daughter boards occupies all the ISA slots offered by the host computer.

Software

The software interface between the various daughter boards and the computer is done through C programs developed in-house. These programs are documented in greater detail in Chapter 5.

In addition to the programs developed in-house, the personal computer is loaded with MATLAB version 4.2c as it's primary software. Mathworks' Simulink and Control System Toolbox are also part of the software libraries which enable students to carry out system dynamics simulation, see an animation of the response, and control system analysis. Moreover, performing real-time control of physical systems directly from MATLAB is done by linking it with in-house developed C programs (refer to Chapter 5). Automatic report generation can be done by the MATLAB

Notebook Suite and Microsoft Word (refer to Chapter 6 and [Tesciuba, 1995]). Furthermore, Mosaic is used as the World Wide Web browser for the students to view the laboratory handouts and on-line help [Hintersteiner, 1995].

This assortment of software shall provide the students a full perspective of control system design from system analysis to simulation, and ultimately a physical implementation of controllers and experimental data analysis. Furthermore, these tasks will be carried out without exiting the environment of MATLAB. Under such a working environment, the students will spend less time learning complicated commands, and thus will be able to focus their attention on the intellectual requirements of the course.

Refer to Appendix E for the cost breakdown of the Computer Module.

Electrical Module

The function of the electrical module is to serve as the interface between the computer and the physical modules. It provides the system with the necessary power to drive the actuators and sensors, the safety features to interrupt the system's process in an emergency, and additional features for incorporating extra analog circuits, such as lead-lag compensators.

Two power sources are included in the electrical module: one for the actuators in the physical modules, one for the sensors and peripheral components. The power source for the actuators consists of a power supply and two pulse-width modulated servo amplifiers (one for the electromechanical system and one for the fluid system). A filter card is connected to an amplifier to provide additional load inductance. The power source for the sensors and peripheral components are provided by single power supply unit, which gives out +12V, -12V, +5V, and common. Both power supplies need to be connected to 120V AC at 60 Hz and are housed in a chassis purchased commercially with in-house modifications.

The electrical module is designed such that extra analog circuits can be easily incorporated to the whole system. An array of standard electrical connectors located on the top of the chassis allow students to introduce analog circuits between the D/A output channels from the computer module and the servo amplifiers in the electrical module. Lead-lag compensation can be incorporated by plugging in circuits boards with lead-lag operational amplifiers circuits to these sets of connectors. The power for the operational

amplifiers are supplied by the +/-12 volt power sources inside the Electrical Module.

For safety, an emergency shut-off button is located on the top of the electrical module. In case of an emergency, the users can hit this button and the power to the actuators in the physical module is cut off immediately.

Connections to the multifunction I/O board in the computer module are done via a multi-colored flat ribbon cable with 25-pin male-female D-sub connectors, and a multi-colored flat ribbon cable with 9-pin male-female D-sub connectors for the A/D and D/A respectively. Four flat ribbon cables with 10 pin connectors are used to connect the electrical module with the encoder board in the computer module. All connectors on this end of the Electrical Module are protected from the students by a Plexiglas cover. The electrical module provides seven connectors to the physical modules - one 9-pin connector for the actuator signal, four 10-pin connectors for encoder signals, one 10-pin connector for the sensor signal, and one 4-pin connector for the sensor and peripheral power source. All connectors on this side of the electrical module offer strain-relief, can tolerate high voltages and high currents, and are single direction oriented for fool-proof purposes.

For detail design descriptions, specifications, and a cost breakdown of the electrical module, refer to [Tesciuba, 1995].

Physical Modules

With the current overall system design, each laboratory station can support a variety of physical modules, namely thermal, fluid, electromechanical, heat transfer, etc. Two versions of physical modules are developed as prototypes: a fluid module and an electromechanical module.

Fluid Module

The fluid module is a water level control system. Two water tanks, one higher than the other, can be connected in coupled, uncoupled, and single configurations. The objective is to control the water level of the lower water tank.

During the project, two physical set-ups for the fluid module were developed. The first set-up was built to investigate the feasibility of the design. It includes two water tanks, a reservoir, a 12 Volt DC centrifugal pump, a pressure sensor, a 3-way ball valve, two needle valves, and the necessary connecting flexible tubing. All the tubes are made of clear plastic

materials to allow easy recognition of the water level by the students. As part of the fluid module, a simple analog circuit providing a sensor signal amplification gain of 100 is included to match the sensor's output voltage to the input voltage level of the A/D card. The details of the design, specifications, and results generated from this first generation fluid system set-up are documented in [Tesciuba, 1995].

The second generation fluid system refines the details of and made appropriate modifications from the original fluid system design. For detailed documentation of the second generation fluid system, refer to [Tesciuba, 1995] and G. Soto's bachelor's thesis [Soto, 1995].

Both set-ups for the fluid module were fabricated rather than purchased because similar systems were not commercially available for sale. In addition, no appropriate fluid system was revealed by our literature search (see Bibliography).

Electromechanical Module

The electromechanical module is a linear slide table driven by a servo motor. The control objectives here are the linear position and velocity of the sliding table.

In addition to the linear slide table, a number of accessories are included in the electromechanical module. A CCD camera is mounted onto this module as part of a vision system. Moreover, plant parameters can be adjusted by an inertial module which increases the rotational inertia of the system, and a friction module which applies extra friction to the plant. The additions of these accessories enable this physical system to illustrate more intellectual concepts to the students.

Detail design descriptions of this module are documented in Chapter 4.

Software Design and Selection

A combination of commercially available programs and programs developed in-house is used in the new laboratory. The selection of commercial software enables the students to use state-of-the-art programs to conduct simulations, view animations, perform analyses, etc., while the programs developed in-house allow for specific tailoring to suit the specific needs of the laboratory. Another reason for writing programs in-house was the lack of driver software available commercially for running real-time control with the hardware of choice.

The primary software is MATLAB from Mathworks, Inc. There are a number of reasons for this selection. First and foremost, MATLAB is a very powerful tool for control analysis and simulation. Secondly, it is a widely used software package and is popular among students. Many students will have exposure to MATLAB prior to attending the laboratory course. Students who have not worked with MATLAB before, will have the opportunity to learn how to use it. Thirdly, MATLAB is supported by MIT's school-wide computer network, Athena. Therefore, technical support is available to the students within the Institute. Moreover, Mathworks, Inc. has added features in MATLAB for developing graphical user interfaces (GUI) and a Notebook Suite for directly interfacing with Microsoft Word. Lastly, MATLAB has the capability of calling external programs in the DOS or Windows environment, which enables students to run real-time control experiments without exiting MATLAB.

Other commercial/public software used in the system are Microsoft Word 6.0 for report generation and Mosaic for World Wide Web browsing. Again, Mosaic is supported by Athena. The compiler used for in-house C program development was Borland C++.

Several C programs were developed in-house primarily for running real-time control algorithms and data collection. Numerous MATLAB ".m-files" were written for interfacing with the real-time control software and for creating GUI's, simulations, and animation.

Both the MATLAB ".m-files" and the C programs, as well as the design of the overall software architecture, are presented in greater detail in Chapters 5 and 6, and in J. Hintersteiner's bachelor's thesis [Hintersteiner, 1995].

Workstation Design

All of the components of the physical set-up described in previous sections need to be put together in a neat and compact fashion to form a workstation. The workstation design layout for the new laboratory set-up is shown in Figure 5. A new table top was designed [Tesciuba, 1995] and constructed by an outside vendor.

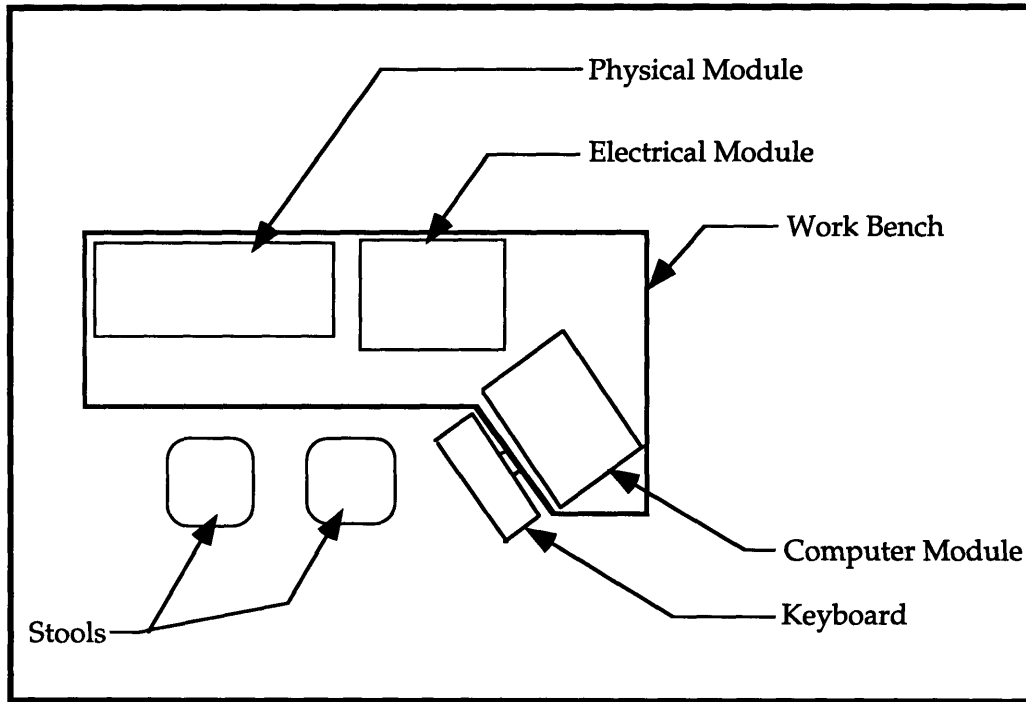


Figure 5. Workstation Layout Design

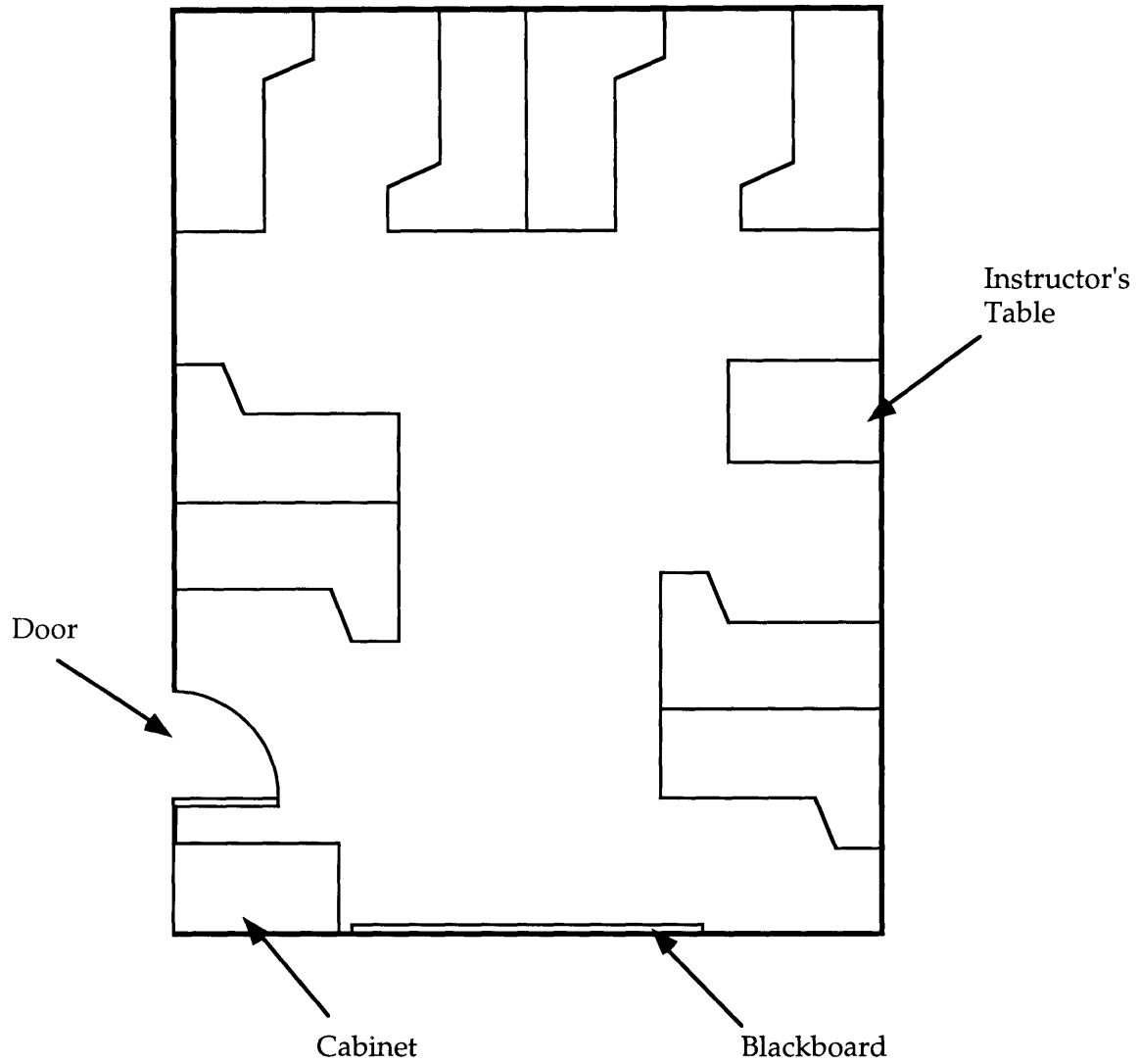
Laboratory Floor Plan

In order to accommodate eight workstations and 16 students in the system dynamics and controls laboratory, major rearrangement of the laboratory floor plan is necessary. Figure 6 shows the new floor plan design [Tesciuba, 1995].

Summary

This chapter presented the overall design of the new system dynamics and controls laboratory, including the logistics, physical set-ups, software, workstation design, and laboratory floor plan.

The next chapter will focus on the design, construction, and performance specifications of the electromechanical physical module. Maintenance procedures and a cost breakdown are included.



Room Dimensions: 26'8" X 23'2"

Figure 6. Laboratory Floor Plan

Chapter 4 Electromechanical Module - The Linear Slide Table

Introduction

Chapter 3 presented the overall design of the new system dynamics and controls laboratory, including a general description of the physical modules. This chapter focuses on the electromechanical module. A linear slide table has been selected in the prototype design to demonstrate applications of controls theory to electromechanical systems. In this chapter, the detailed designs of the linear slide table, the accompanying accessories, and the performance specifications are documented. After reading this chapter, one should be able to reproduce replicas of such a system. At the end of the chapter, maintenance procedures and a cost breakdown is included.

Selection of the Electromechanical Module

One of the first questions when designing an educational laboratory is: "what physical system should be selected to demonstrate the concepts to the students?" Auslander answered it best by stating, "control objectives must be safe, reasonably fast but not too fast, have some 'real world' irregularities, but not so much that control theory never works, not break too often, and not cost too much." [Auslander et al, 1988]

Reviewing the fields covered by mechanical engineering, an electromechanical system seems to fit this purpose well. Typical electromechanical systems have response times on the order of a fraction of a second to seconds, which is observable and compatible with students' patience. Also, electromechanical systems are quite "clean" in terms of sensor noise, responses, and operation environment, as compared to other mechanical systems. Moreover, setting up an electromechanical system does not require installation of heavy and expensive supporting equipment such as hydraulic pumps needed for hydraulic actuators.

Considering the above advantages, the incorporation of an electromechanical system into the new system dynamics and controls laboratory appears to be appropriate.

So the next question is: “what specific electromechanical plant should be used for the module?”

The paramount consideration is that the plant chosen should be capable of covering the intellectual requirements of the laboratory stated in Chapter 2. Some additional considerations are pointed out by the professors during the interview stage of the project. For example, some professors mentioned the idea of using industrial equipment as physical plants for the new laboratory. The main purpose behind this idea was to raise the interest level for the students by giving them a sense of working on a real-world engineering equipment. Another reason was that the industrial quality of the equipment may allow for a longer life with little maintenance.

Cost should also be taken into account when choosing the physical plant. If the laboratory is to use industrial quality equipment, the cost could be high for full scale development of several workstations.

A ball screw driven linear slide table was selected since it fulfills the objectives stated above. It is capable of demonstrating the intellectual objectives of the laboratory. Moreover, it is commonly used in engineering applications such as manufacturing. Thompson Industries, Inc., a leading manufacturer of linear slide tables, donated an assembled linear slide table to this project, which has been integrated into the electromechanical module in the prototype set-up.

Dynamics of the Linear Slide Table

The dynamic equation describing the linear slide table is:

$$\left(J * p + \frac{M}{p} \right) \ddot{x} + (B * p) \dot{x} = T \quad \text{eq. (4.1)}$$

where

M Slide Table Mass [slug]

T Motor Torque [oz-in]

J	Ball Screw Rotational Inertia [slug/in ²]	B	Viscous Damping Coefficient [oz-in/rad/s]
p	Pitch of Ball Screw [rad/in]	x	Linear Slide Table Position [in]

which leads to the following transfer function:

$$\frac{X(s)}{T(s)} = \frac{K_{\text{plant}}}{s(\tau_{\text{plant}}s + 1)} \quad \text{eq. (4.2)}$$

where

$$K_{\text{plant}} = \frac{1}{B * p}$$

$$\tau_{\text{plant}} = \frac{J + M/p^2}{B}$$

From the characteristic equation, it is obvious that the linear slide table plant can be modeled as a first order system if the table's velocity is the output variable and a second order system if the position is the output variable. Therefore, the linear slide table is capable of demonstrating both first order and second order system dynamics for the laboratory course, thus satisfying one of the intellectual requirements. Refer to Appendix F for derivation of the dynamic model.

Design and Fabrication of the Electromechanical Module

The electromechanical module consists of the following components:

- Linear Slide Table
- Motor
- Encoder
- Flexible Coupling
- Inertial Load and Coupling
- Friction Device and Stand
- CCD Camera
- Electrical Switches
- Electrical and RCA Connectors
- Safety Cover
- Base Plate
- Screws and Wire Cables

Figures 7 and 8 show the overview of the electromechanical module.

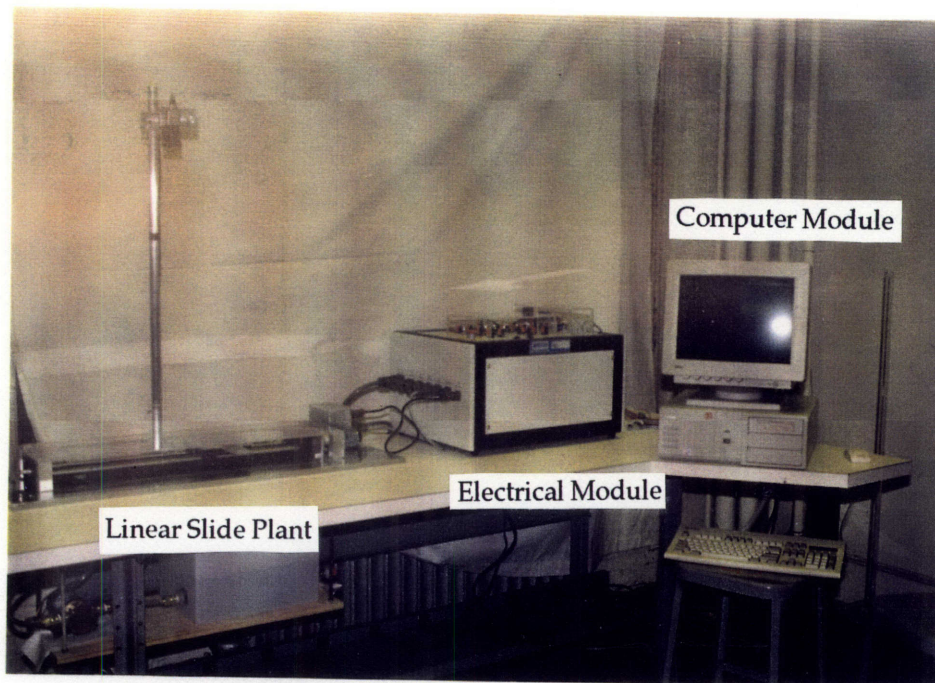


Figure 7. Photograph of Electromechanical Module

Linear Slide Table

Thomson Industries, Inc. generously donated a 30-inch ball screw driven linear slide table, model 2EB-08-FTB-F L30 for the development of the prototype laboratory set-up. The table has a maximum stroke length of 21.5 inches, screw pitch of 5 revolutions per inch, and a maximum acceptable rotational speed (which is set at 80 percent of critical speed) of 2400 rpm. Excerpts of the [Thomson Industries, Inc. Catalog] in the [Laboratory Manual] contains further specifications for the linear slide table.

Motor

The manufacturer of the linear slide table recommends the use of stepping motors for high accuracy performance. However, the prototype uses a permanent magnet DC motor as the actuator for the linear slide table because of its much lower cost and linearity.

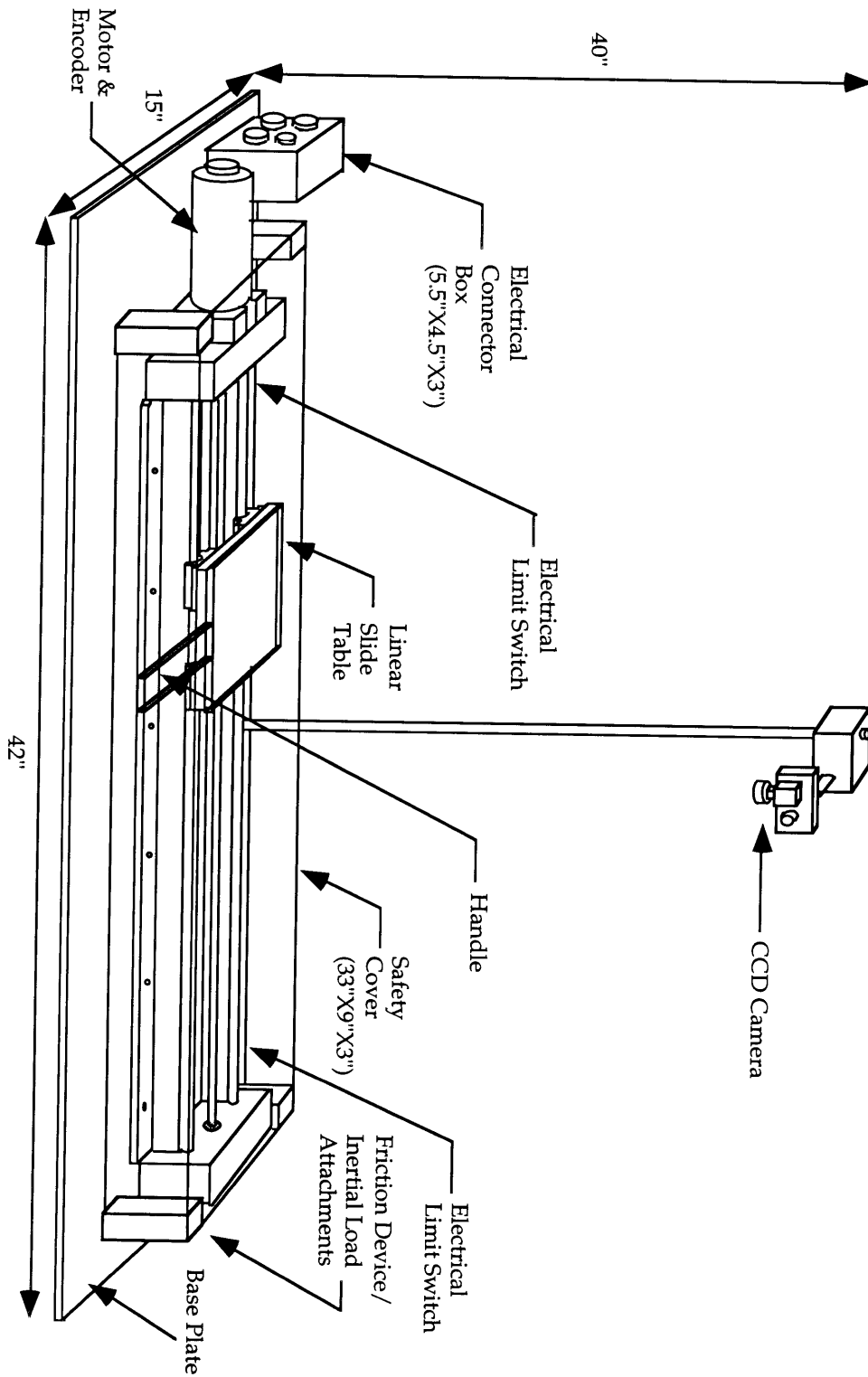


Figure 8. Overview of the Linear Slide Table Plant

A Clifton Precision Products motor, model JDH-2250-BX-1C (brushed permanent magnet DC servo motor), is chosen as the actuator for the plant. An aluminum motor support is used. The specification of the motor and the design drawing for the motor support are documented in the [Laboratory Manual].

Encoder

An optical incremental encoder is chosen as the sensor for this module because it generates “clean” signals and the signals’ digital nature are compatible with the digital computer, thus eliminates the need of analog-to-digital conversion.

The most important parameter to be considered during the selection of an encoder is the resolution. The degree of resolution for the linear slide table linear position should be high enough such that students would not detect errors of the system due to encoder resolution with the naked eye. The encoder selected is a BEI optical incremental encoder, model # 5540-A06. This model of encoder has a resolution of 500 counts per revolution (CPR). From the CPR of the encoder and the pitch of the ball screw in the linear slide table, the linear resolution of the sensor is determined as follows:

$$\text{Linear Resolution} = \frac{1/p}{\text{CPR}} = \frac{1/(5\text{rev/in})}{(500\text{counts/rev})} = 0.0004\text{ in/count}$$

The resolution of 0.0004 inches is beyond the sensitivity of human eyes. Therefore, the resolution of this encoder is suitable for this system.

The signal generated by the encoder is transistor-transistor logic (TTL) compatible, which can be read by the encoder board in the computer module (refer to Chapter 3). The 5 volt actuation voltage for the encoder is supplied by the encoder board in the computer module. The [Laboratory Manual] contains further specifications for the encoder.

Flexible Coupling

A Helical flexible coupling, model MCAC100-8-8, is used to transmit the motor torque to the ball screw shaft in the prototype set-up. Refer to the [Laboratory Manual] for the specifications of the flexible coupling.

Inertial Load and Coupling

One of the intellectual requirements of the new laboratory is to allow the students to investigate the effects of changes in the plant's physical parameters. In the electromechanical system, the plant's rotational inertia has been chosen to be the variable plant parameter. There are two ways to add inertia to the system: one is to add a mass on top of the sliding table, or add a rotational inertia at the end of the ball screw shaft. The latter method has been selected because the ball screw pitch for the current system is 5 revolution per inch. With this pitch value, the system becomes relatively insensitive to additional load or forces acting on the linear table. For example, if one wants to introduce an additional 30 percent rotational inertia to the system by putting extra mass on top of the linear slide table, the mass would have to weigh over 90 pounds!

To add inertia to the plant, an aluminum rigid coupling is attached to one end of the linear slide table (opposite end to the motor) by a set screw, which allows a rotational inertial load to be easily mounted and dismantled.

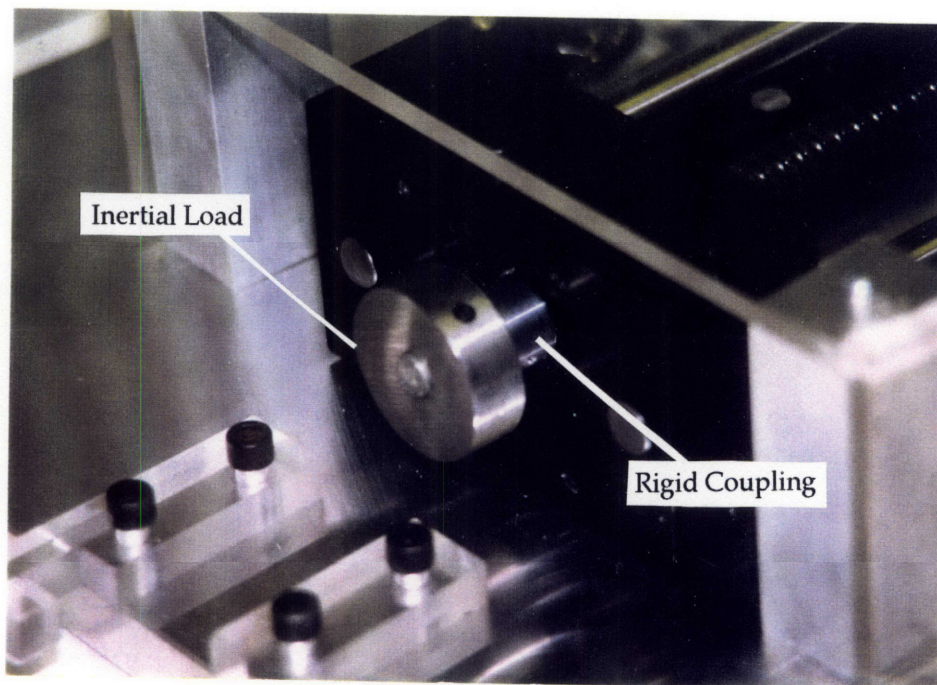


Figure 9. Rigid Coupling and Inertial Load

The rotational inertial load is a steel disk of 1.5 inch in diameter and 0.5 inch in thickness. It is mounted on the rigid coupling by a set screw. This steel disk adds approximately 30 percent to the linear slide table's original rotational inertia.

Figure 9 shows a photograph of the rigid coupling and the rotational inertial load. Refer to the [Laboratory Manual] for the specifications of these two parts.

Friction Device and Stand

Another intellectual requirement of the new laboratory is to allow the students to alter the environment of the physical system. The design of the prototype electromechanical system consists of a friction module which enables students to apply extra friction to the linear slide table.

The friction device is located at one end of the linear slide table (same end as the rotational inertial mass). It consists of a spring loaded rubber stopper (which makes the adjustment of frictional torque applied to the plant possible) and a stand to hold up the friction device. See Figure 10 for a photograph of the friction device.

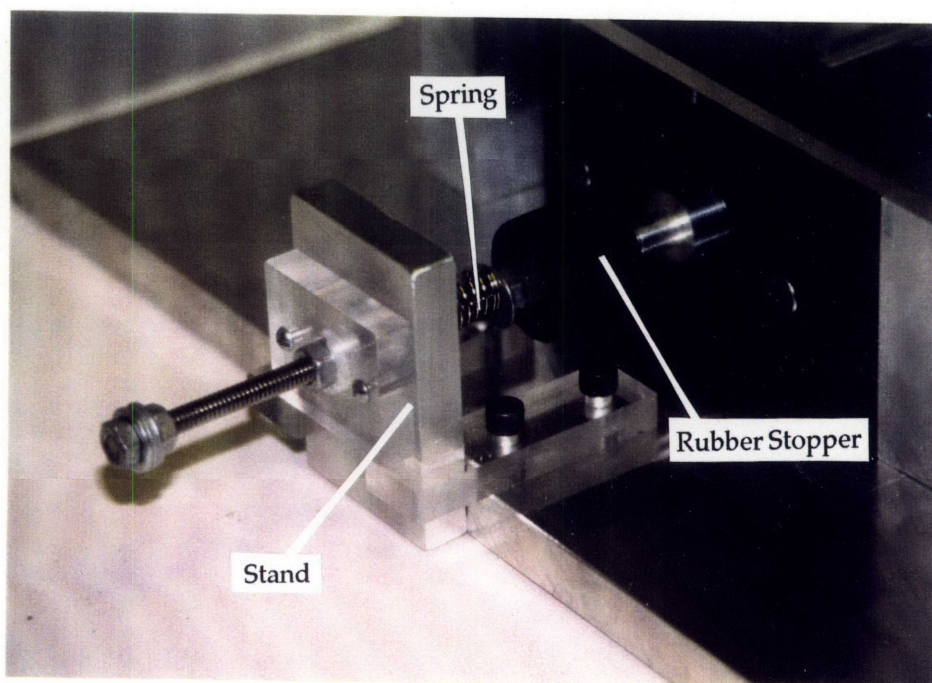


Figure 10. Friction Device

CCD Camera and Stand

A CCD camera (Chinon model CX-200/C) with a 8 mm lens (Rainbow T01-279J-990 by Sinsei Shoji Co, Ltd.) is included as part of the electromechanical module vision system. The camera sends out NTSC signals to the video board in the Computer Module for processing and determination of linear slide table position. The vision system provides an alternative sensor to the encoder for the electromechanical module.

An adjustable stand has been designed and fabricated to hold the CCD camera above the linear slide table. The stand has to be relatively rigid in order to minimize the vibration amplitude while the linear slide is in motion, so that the image captured by the camera is of good quality. Therefore, the stand is made of aluminum rods 1 inch in diameter and held by integral clamp blocks.

Figure 11 shows a photograph of the CCD camera and the camera mount. Refer to the [Laboratory Manual] for camera specifications and camera stand design drawings.

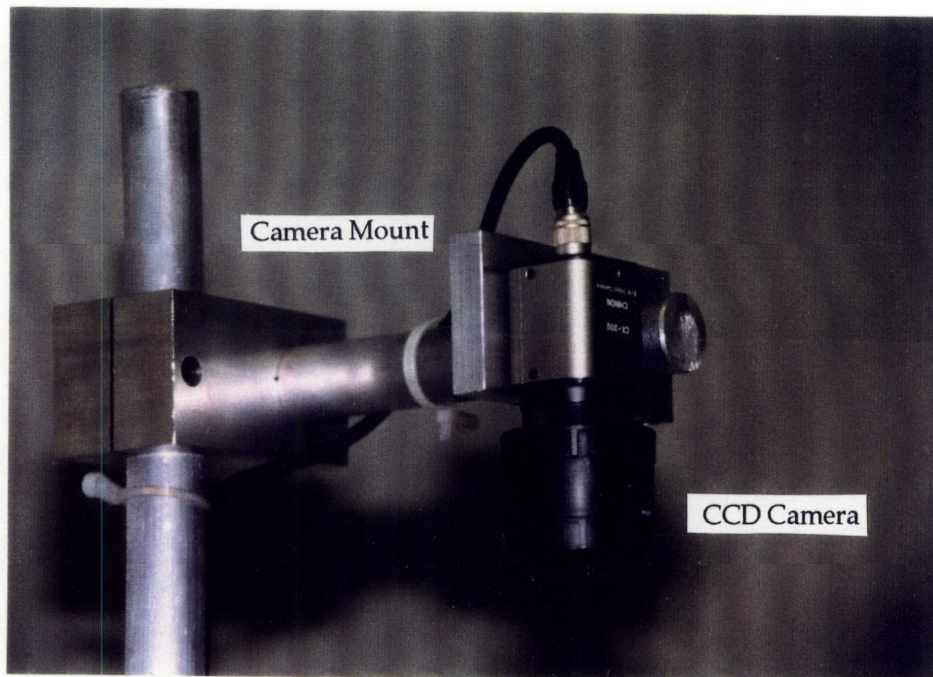


Figure 11. CCD Camera and Mount

The software for running the vision system on the linear slide module is described in Chapter 5.

Electrical Connector Box

The linear slide module connects to the electrical module by four standard connectors: one 9-pin connector for the actuator signal, one 10-pin connector for the encoder signals, one 4-pin connector for the sensor and peripheral power source, and one RCA female connector for the video signal output. Except for the RCA connector, all of the connectors have clamps for strain-relief, can tolerate high voltages and high currents, and are single-direction oriented for fool-proof purposes. These four connectors are housed in an electrical connector box. The wire cables are soldered to the connectors and shrink wrapped to prevent short circuits.

Electrical Limit Switch

There are two electrical snap switches, each located 3/4 inches from an end of the linear slide table. These electrical switches serve as part of the overall system safety design. The switches are electrically connected in series between the electrical connectors in the connector box and the motor. The switches are hooked up in the "closed circuit" mode. When the linear slide table travels to either end of the track, it actuates the switch to the "open circuit" mode, which cuts off the current to the motor. See Figure 12 for the circuit diagram of the electrical switches.

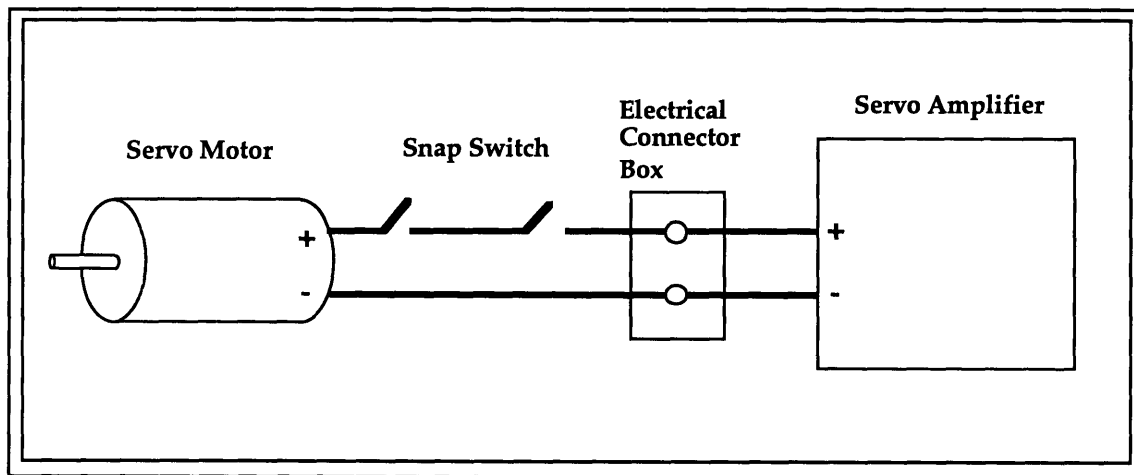


Figure 12. Electrical Switch Circuit Diagram

Two aluminum pieces have been fabricated to hold the snap switches in place. The specifications of the snap switches can be found in the [Laboratory Manual].

Safety Cover

Another safety feature for the linear slide system is the safety cover, which prevents hands from being caught in the pinch points at the ends of the linear slide travel. The safety cover is made of Plexiglas and aluminum supports. A horizontal handle mounted on top of the linear slide table allows the students access to adjust the location of the linear slide as well as to "feel" the effects of different controllers with their hands. For example, an increase in proportional gain for a P-controller in a position control system will increase the "springiness" of the system. With the handle, the students can push/pull the linear slide table to feel this effect.

Base Plate

All the different components described in the above are placed on a jig and tool aluminum base plate of dimension 42"X15"X5/8", which has a flatness tolerance of 0.015 inches as required by the linear slide specifications. The various components for the electromechanical module are screw mounted to this base plate.

Compatibility with the Electrical and Computer Module

In order for the linear slide module to be compatible with the Electrical and Computer Module, some tailoring of these two modules are necessary.

Electrical Module Calibration

Calibration of the servo amplifier in the electrical module is necessary in order for it to work with the linear slide module. The servo amplifier has to be set to voltage-to-current mode. Afterwards, the gain of the servo amplifier has to be calibrated to 4 amps/volt, which suits the requirement of the actuator of the linear slide system (refer to the section for calibration procedures in this chapter).

Since the linear slide module uses a transistor-transistor logic compatible encoder, no analog signal conditioning is necessary in the electrical module. The encoder signal is simply passed through the electrical module to the encoder board inside the computer module.

Computer Module Modification

The hardware in the computer module is unaffected by the linear slide module. However, some tailoring of the software was necessary. There are two main aspects in the software that need to be modified: the real-time control algorithm and the graphical user interface.

The real-time control software have a modular design which is based on several building blocks (refer to Chapter 5). In creating a real-time control program for the linear slide table system, the encoder block, PID-controller block, D/A block, timer block, and the input and output blocks were incorporated together to form a complete real-time control program [Hintersteiner, 1995].

In the System Dynamics and Controls Laboratory, each laboratory physical module has its own set of experiment procedures and handout materials. Therefore, each physical module will have a tailored graphical user interface that accompanies it. The graphical user interface development procedures for the linear slide table system follows the guideline described in Chapter 6, where the reader can obtain more information on creating graphical user interface.

Performance Specifications of the Linear Slide Module

The following specifications of the linear slide module were obtained experimentally:

- Time constant (τ_{plant}): 0.50 seconds
- DC gain of plant (K_{plant}): $0.76 \frac{\text{in/s}}{\text{oz-in}}$
- Static Friction: 5.4 oz-in

For description on the tests performed to obtain the above data, refer to Appendix F.

The combination of the above specifications and the ones provided by the motor and servo amplifier manufacturers (see [Laboratory Manual]), the transfer function of the linear slide system is determined to be:

$$\frac{X(s)}{T(s)} = \frac{44.8}{s(0.5s + 1)} \quad \text{eq. (4.3)}$$

Maintenance for the Linear Slide Module

The linear slide module is a relatively robust plant and should require little maintenance. However, two important maintenance procedures need to be performed regularly: lubrication of the ball screw and calibration of the servo amplifier in the electrical module.

Lubrication of Ball Screw Assembly

It is important to lubricate the ball screw assembly of the linear slide table at least once a semester, preferably at the beginning of the term, with lithium grease or a similar product. This will minimize wear and tear of the system as well as significantly reduce the amount of audible noise produced by the ball screw mechanism during its motion.

Calibration of Servo Amplifier

Over time, the PWM servo amplifier inside the electrical module may drift. It is recommended that the gain settings are calibrated every three month intervals. Since the output of the amplifier is current of relatively large magnitude (maximum current output is 20 Amps.), one method to calibrate the servo amplifier is proposed in the following paragraphs.

First, slide the linear table to one end. This essentially clamps the motor shaft and prevents rotation. Next, check to make sure the wire leads are properly connected to the positive and negative terminal of the motor and the servo amplifier. Connect, in series, a 3 Ω power resistor to the cable running between the motor and amplifier, as shown in Figure 13. The reason for including the motor in the calibration procedure is the PWM servo amplifier requires a minimum load inductance of 200 μH in order to operate correctly [Advance Motion Controls PWM Servo Amplifiers Catalog, 1992]. This amount of inductance is generated by the coils inside the DC permanent magnet motor.

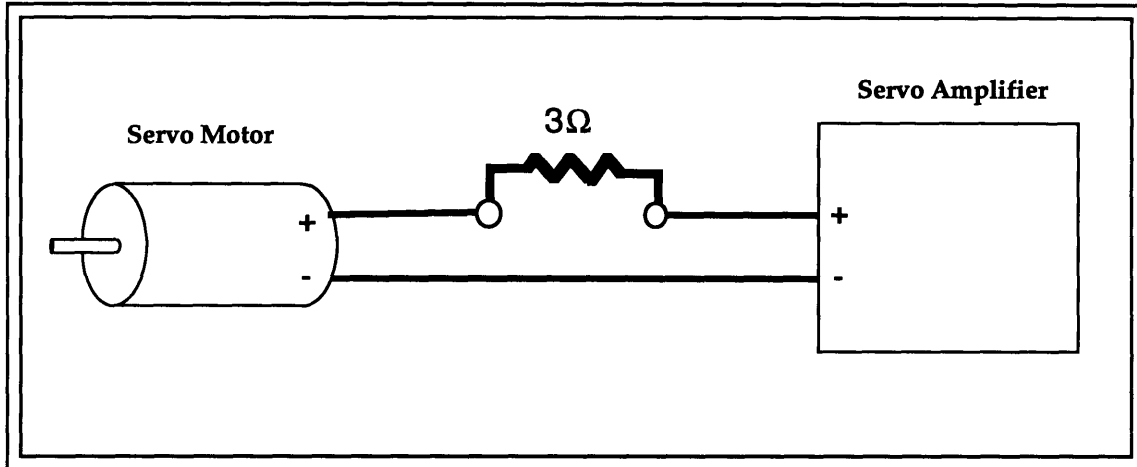


Figure 13. Circuit Diagram for Servo Amplifier Calibration

From the Computer Module, load up the Windows application D2A.EXE, which controls the analog signal output from the DATEL multifunction I/O board. The software will prompt the user to select a D/A channel. After selecting the D/A channel connected to the linear slide table, the user can command the D/A to output voltages in the range of -5.0 volts to +4.9976 volts through the program. To calibrate the servo amplifier gain, command the D/A to output a desired voltage, then measure the voltage across the 3Ω resistor. By Ohm's Law, the current supplied by the servo amplifier at a known input voltage can be obtained by dividing the measured voltage across the resistor by its resistance, as described by the following:

$$\begin{aligned} \text{Servo Amplifier Gain} &= \frac{\text{Current - from - Amplifier}}{\text{Voltage - from - D/A}} \\ &= \frac{\text{Voltage - across - resistor/Resistance}}{\text{Voltage - from - D/A}} \quad \text{eq. (4.4)} \end{aligned}$$

The voltage to current gain of the servo amplifier can be calibrated by trimming the potentiometers on the side of the amplifier. For details of adjusting the potentiometers, refer to [Advance Motion Controls PWM Servo Amplifiers Catalog, 1992].

Since the output of the servo amplifier is current with a magnitude of up to 20 amperes, care must be taken when performing calibration to avoid electrical shock or burn by the high temperatures due to power dissipation in the resistors.

Cost Breakdown of the Linear Slide System

Table 1 summarizes the cost breakdown of the linear slide table prototype.

Item	Description	Price
Linear Slide Table	Thompson Industries Model 2EB-08-FTB-F L30	\$1,950 [†]
Motor & Encoder	Motor: Clifton Precision Products Model JDH-2250-BX-1C Encoder: BEI Model 5640-A06	\$79
Flexible Coupling	Helical Model MCAC100-8-8	\$24
CCD Camera & Lens	Camera: Chinon Model CX-200/C	\$279
Camera Lens	Rainbow T01-279J-990	\$67 ^{††}
Electrical Accessories	Switches, connectors, cables	\$50 [*]
Machining Materials	Aluminum pieces, Plexiglas, rubber stopper, spring	\$250 [*]
	Total:	\$2,699

[†] Donated by Thompson Industries, Inc.

^{*} Estimates

^{††} Included in the camera price.

Table 1. Cost Breakdown of the Electromechanical Module

Summary

This chapter described the selection, design, and fabrication of the linear slide module. The necessary modifications to the Computer Module and Electrical Module, recommended maintenance, and cost breakdown were also presented.

Chapter 5 will focus on the real-time control software development aspects of the new System Dynamics and Controls Laboratory.

Chapter 5 Development of Real-Time Control Software

Introduction

The design decision to use a personal computer for the Computer Module was described in Chapter 3. In addition to the hardware needed inside the personal computer, appropriate software must be utilized to accomplish the job of the Computer Module.

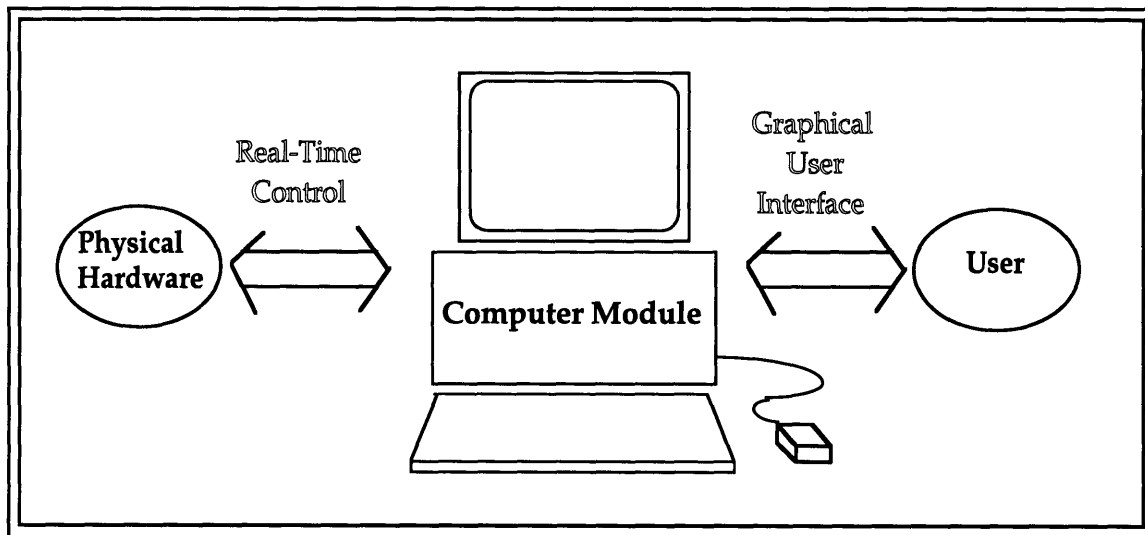


Figure 14. Schematic of Software Communication Architecture

Figure 14 shows a schematic of the software communication architecture design. The Computer Module serves as the liaison between the users and the physical system. On one side, it communicates with the physical system through the various daughter boards such as the multifunction I/O board and encoder board (refer to Chapter 3) via the real-time control software. On the other side, it communicates with the users through the keyboard, mouse, monitor, via the user interface software, such as MATLAB graphical user interface, Microsoft Word, and the World Wide Web browser.

In this chapter, the focus will be on the software that connects the Computer Module to the physical modules. The software performing this task is written in the C programming language, and is referred to as the real-time control (RTC) software. The RTC software performs three functions:

1. it operates the hardware inside the Computer Module, such as the DATEL PC-412A multifunction I/O board, the US Digital PC7166 encoder board, and the Creative Labs Video Blaster SE100 video board;
2. it computes the controller output;
3. it saves the experimental data to a file.

This chapter will begin with a discussion of the design concept for the RTC software, followed by a brief presentation on general programming scheme for the three daughter boards inside the personal computer. Afterwards, the components which make up the real-time control software for the new System Dynamics and Controls Laboratory are discussed.

RTC Software Design Concept

The RTC software for the new System Dynamics and Controls Laboratory is designed such that it can be quickly modified to handle new physical hardware. The method to achieving this objective is to use the modular design approach of building blocks.

The Building Block Approach

Under the building block approach, the RTC software is written as a collection of software blocks, each block performing a specific task. The program developer will select the blocks needed to control a certain plant from the building block repertoire and combine them to build a RTC application. The following sections will describe the function and internal logic of each building block. For more information on how to combine the blocks and build a RTC application, refer to [Hintersteiner, 1995].

The RTC Big Picture

Figure 15 show the “big picture” of the RTC software. Starting from the far left, sensor signals from the Physical Module will pass through the

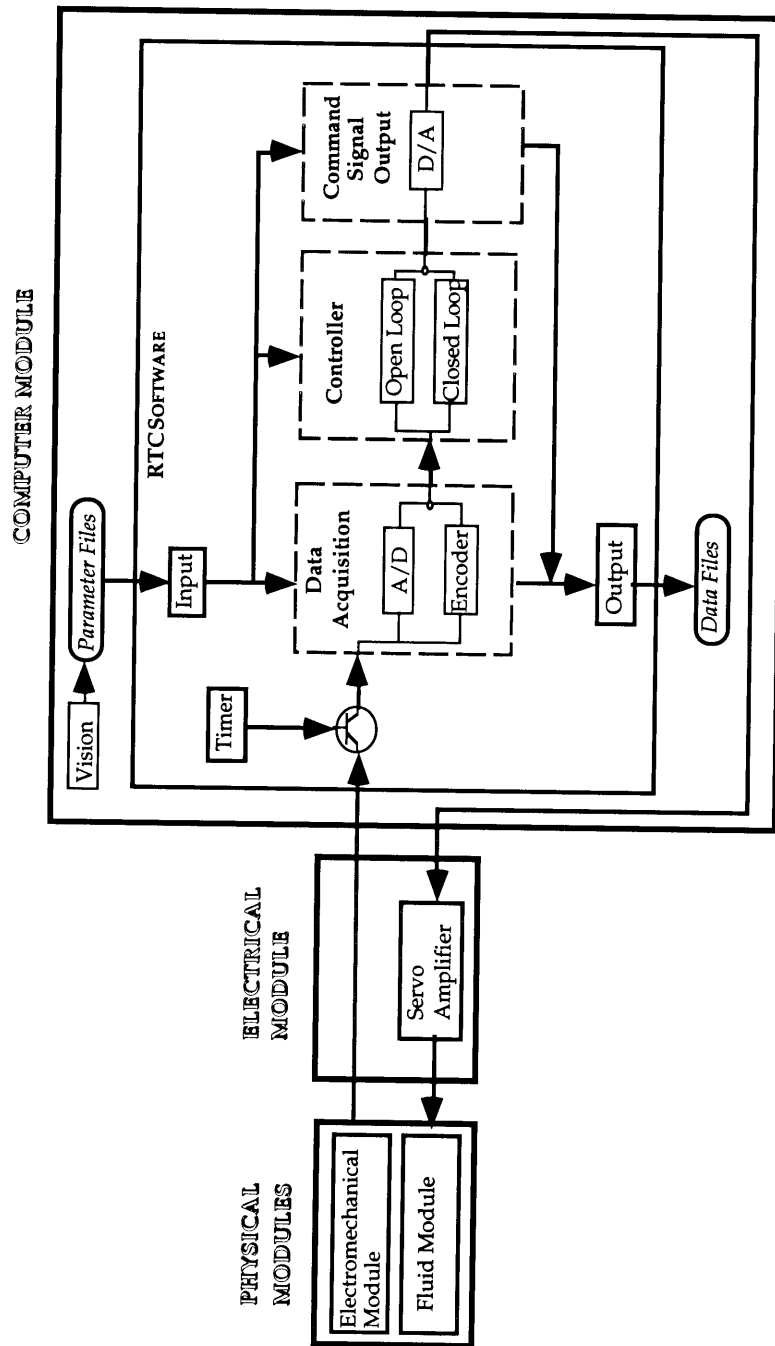


Figure 15. The RTC "Big Picture"

Electrical Module then enter the Computer Module. Inside the Computer Module, the Data Acquisition blocks will operate the data acquisition boards to retrieve the sensor data. The data is then passed on to the controller blocks for computing the command signal. The command signal is then passed on to the command signal output block, which convert the signal from digital form to analog form.

The timer block controls the timing of the real-time control loop. It serves as a gate to the RTC, permitting the start of a control cycle at regular time intervals. The timer block also checks whether the sampling frequency is too high for the RTC to handle.

There are two input/output (I/O) blocks inside the RTC. They serve to link the RTC software to other software inside the Computer Module, namely, MATLAB.

Programming the I/O Boards

Before the discussion of each building block, it is important to have some background knowledge of how to program the three input/output boards inside the Computer Module. The general programming scheme for the DATEL PC-412A board will be presented first, followed by the US Digital PC7166 and the Video Blaster SE100 video board.

General Programming for DATEL PC-412A

The general programming scheme for the DATEL PC-412A multifunction I/O board is similar to most other commercial I/O boards. Sixteen consecutive I/O addresses are required by the board. Each of these I/O addresses is accessed by the board for various purposes, such as registers, digital input/output ports, counters, and digital-to-analog channels.

Essentially, bits are written to the I/O register memory according to the memory map documented in Section 3 of the [DATEL PC-411/412 Hardware and Software Manual, 1991]. An excerpt of this map is shown in Table 2.

These command bits tell the I/O board exactly what to perform. For instance, to enable analog-to-digital conversion, one will write a command bit value of 1 to the first bit in the command register bit field, which is located at

the BASE + 0 address; to disable analog-to-digital conversion, a command bit value of 0 will be written to the same address instead.

I/O Address (hexadecimal)	Description
BASE + 0	Command Register
BASE + 0	Status Register
BASE + 1	Channel Address Register
BASE + 1	FIFO A/D Data Register
BASE + 2	Interrupt/DMA Register
BASE + 2	FIFO Reset Register
BASE + 3	Digital Output Port
BASE + 3	Digital Input Port
BASE + 4	Counter #0 (82C54)
BASE + 5	Counter #1 (82C54)
BASE + 6	Counter #2 (82C54)
BASE + 7	Control Word (82C54)
BASE + 8	DAC 0 low byte
BASE + 9	DAC 0 high byte
BASE + 10	DAC 1 low byte
BASE + 11	DAC 1 high byte
BASE + 12	DAC 2 low byte
BASE + 13	DAC 2 high byte
BASE + 14	DAC 3 low byte
BASE + 15	DAC 3 high byte

Table 2. Memory Map for PC-412A

The BASE address serves as the reference to all I/O addresses to the board. For instance, the I/O address of Counter #2 of the Intel 82C54 chip on the board is referred to as BASE + 6. This means that if the BASE address is set to 300 hexadecimal in the host computer, then the address for accessing

Counter #2 is 306 hexadecimal. The base address can be set by flipping the DIP switches on the PC-412A board (refer to Section 2.2 of the [DATEL PC-411/412 Hardware and Software Manual, 1991]). Currently, the BASE address in the prototype is set at the factory standard of 300 hexadecimal.

The addresses can be write only, or can be both read and write, depending on the function that they are referred to. For example, the digital input and output ports are located at the same address of BASE + 3. This address can be written to (for digital output) and read in (for digital input). The digital-to-analog ports (BASE + 8 to BASE + 15) are, however, write only addresses.

On the PC-412A board, there is a First-in, First-out (FIFO) data memory, which temporarily stores analog-to-digital data until the data is read by the host computer. The data stored in the FIFO is transported to the host computer by reading the FIFO A/D data register (read BASE + 1). Reading the FIFO reset register (read BASE + 2) will clear all data stored in the register. The status of the FIFO data register can be checked by reading the status register (read BASE + 0), which indicates whether the FIFO data register is empty, half full, or full.

For detailed instructions on programming the PC-412A board, refer to the [DATEL PC-411/412 Hardware and Software Manual, 1991].

General Programming for US Digital PC7166

The general programming scheme for the US Digital PC7166 encoder board is relatively simple because of two reasons:

1. It only has the function of reading encoder count values.
2. The software driver source code developed by the manufacturer for this board is included, and only minor modifications were required.

The board uses 16 consecutive I/O addresses in the host computer. In the prototype set-up, the base address for this board is set to be 330 hexadecimal to avoid conflict with the PC-412A board and the SE100 board. A command bit is written to latch the counter value of a specified counter,

which is then read from the counter data register. See Table 3 for the memory map of this board.

I/O Address (hexadecimal)	Description
BASE + 0	1st Counter Data Register
BASE + 1	1st Counter CMD/Status Register
BASE + 2	2nd Counter Data Register
BASE + 3	2nd Counter CMD/Status Register
BASE + 4	3rd Counter Data Register
BASE + 5	3rd Counter CMD/Status Register
BASE + 6	4th Counter Data Register
BASE + 7	4th Counter CMD/Status Register
BASE + 8	Latch Pulse

Table 3. Memory Map for US Digital PC7166

General Programming for Video Blaster SE100

Due to the large number of applications of the Video Blaster SE100 board, general programming is fairly complex. However, when focused on one specific application, such as in this case, using the video board to capture a frame for target location, the task becomes more straight forward.

The manufacturer of the video board included a DOS application programming interface (API) in the programming kit to aid developers in low level programming of the board. A terminate-and-stay-resident (TSR) driver is used to access the various functions of the video board. Upon loading, the TSR driver is installed at a free interrupt vector from 80h to BFh and the API may be accessed through an interrupt call to this vector.

To activate a function of the video board, the appropriate API function number and a SET or GET flag should be written to the BL and BH registers of the board respectively.

Function	Description
Function 00h	Get Driver Version
Function 01h	Initialize / Warm-Start Driver
Function 02h	Terminate Driver
Function 03h	Base I/O Port Address
Function 04h	Video Base Address
Function 06h	Interrupt Number
Function 0Ah	Video Input Freeze Flag
Function 12h	Video Display Status
Function 1Ch	Read / Write Frame Buffer
Function 26h	Translate Coordinates

Table 4. Functions of SE100 API

Over 30 functions can be called upon in the API. The main functions that were are in the vision system for this project are listed in Table 4.

For more programming information about the SE100 video board, refer to [Video Blaster SE100 DOS Application Programming Interface, 1995].

RTC Building Blocks

The RTC building blocks can be categorized by the functions they perform. They are listed in Table 5.

This section will discuss more in depth each building block listed above, according to the following sequence: Data Acquisition, Controller, Command Signal Output, Timer, Reference Signal, Input, and finally, Output.

Function	Block Name
<i>Data Acquisition</i>	Analog-to-digital (A/D) Block Encoder Block
<i>Controller</i>	Open Loop Block PID-Controller Block
<i>Command Signal Output</i>	Digital-to-analog (D/A) Block
<i>Timer</i>	Timer Block
<i>Reference Signal</i>	Vision System Block
<i>Input</i>	Configuration Block Graphical User Interface Block
<i>Output</i>	Data Block

Table 5. RTC Building Blocks

Data Acquisition

Analog-to-Digital (A/D) Block

The function of this building block is to control the operation of the A/D channels on the DATEL PC412A multifunction I/O board, which read in analog signals generated from the sensors in the physical modules. There are, altogether, 16 A/D channels on the PC-412A board. The operations of the A/D function are controlled from the command register (write BASE + 0), status register (read BASE + 0), channel address register (write BASE + 1), FIFO (first-in-first-out) A/D data register (read BASE + 1), and FIFO reset register (read BASE + 2). A schematic of the A/D building block is shown in Figure 16.

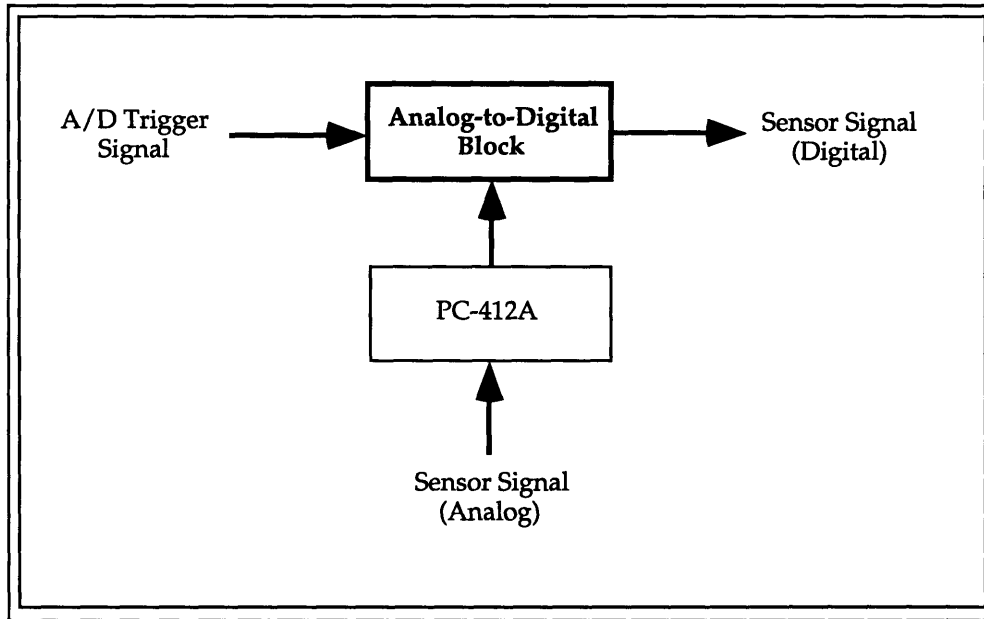


Figure 16. Schematic of A/D Building Block

The A/D block is programmed, prior to the running of the real-time control algorithm, with the information such as the trigger source, single/multi-channel, unipolar/bipolar, single-ended/differential, reference voltage, and input channel expansion. The detailed description of these functions can be found in Section 3.2 of [DATEL PC-412A Hardware and Software Manual, 1991]. During the running of the experiment, the A/D trigger command will be sent to this block to activate A/D conversion.

Inside the analog-to-digital building block, the algorithm can be described by Figure 17. When the A/D block receives an A/D trigger signal, it will quit any jobs the A/D converter may be doing and clear the buffer. This is done by first writing a bit value of zero to the first bit at the command register bit field, which disables the A/D converter, followed by reading the FIFO reset register, which clears the FIFO and sets the empty flag to true. Next the A/D conversion is enabled by writing a bit of value 1 to the first bit of the command register bit field.

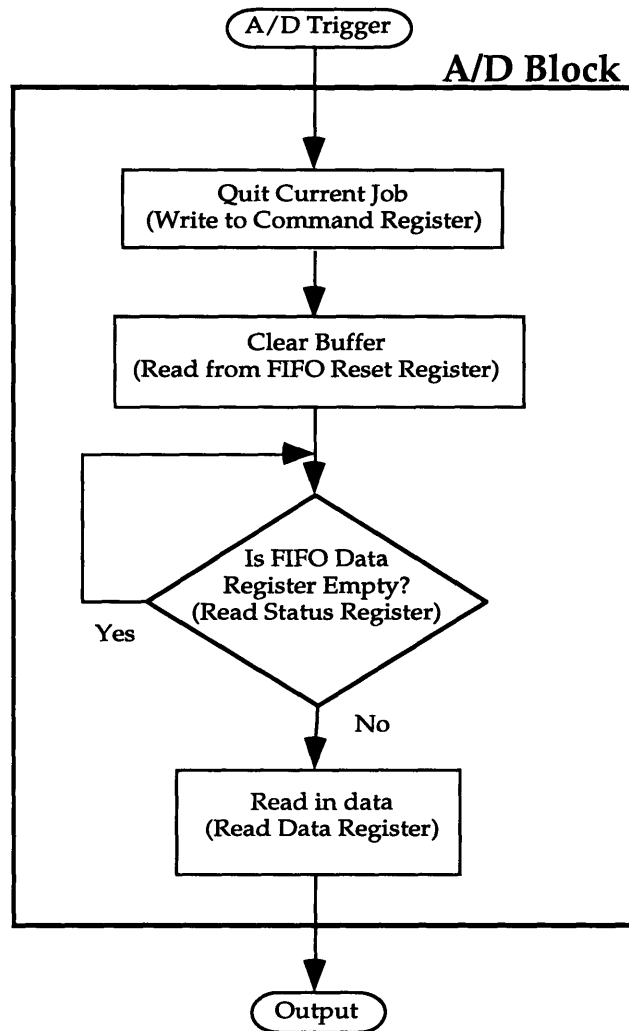


Figure 17. Internal Structure of A/D Building Block

After the A/D conversion is enabled, the empty flag for the FIFO buffer is monitored. As soon as the empty flag becomes false, indicating that a data has been read by the A/D, the buffer is read by the program from the FIFO A/D data register and is passed on to the next block in the RTC. This completes a cycle of the A/D building block.

Encoder Block

The US Digital PC7166 encoder board has four encoder channels. The encoder board comes with an accompanying software driver, from which the encoder building block is adopted and modified [Hintersteiner, 1995]. Figure 18 shows a schematic of this block.

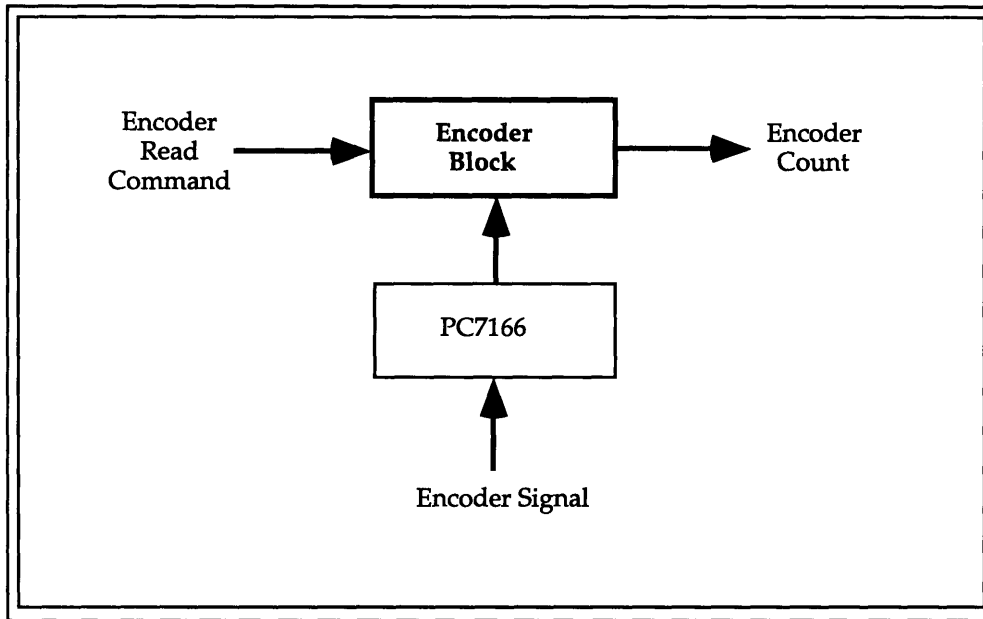


Figure 18. Schematic of Encoder Block

Controllers

Open Loop Block

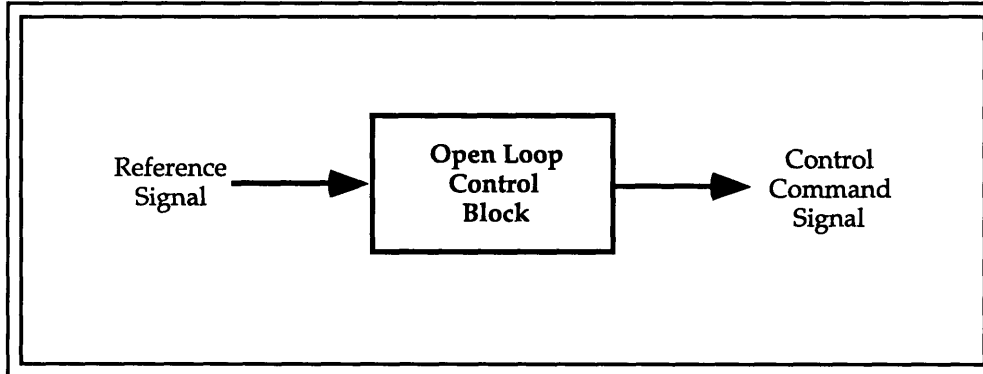


Figure 19. Schematic of the Open Loop Controller Block

The open loop block is used when an open loop control system is being developed. A reference input signal is received by this block. Inside this block, the reference input signal is converted to a command output signal by multiplying it to a unit-to-volt conversion factor. The offset is also accounted for inside this block. The command signal is then passed on to the next block.

Figure 19 shows a schematic of this block. Refer to [Hintersteiner, 1995] for a more comprehensive description of this block.

PID-Controller Block

The PID-controller building block is responsible for calculating the command signal from the error signal, as shown in Figure 20. Readers can refer to [Hintersteiner, 1995] for a detail explanation the digital PID-controller algorithm used in this block.

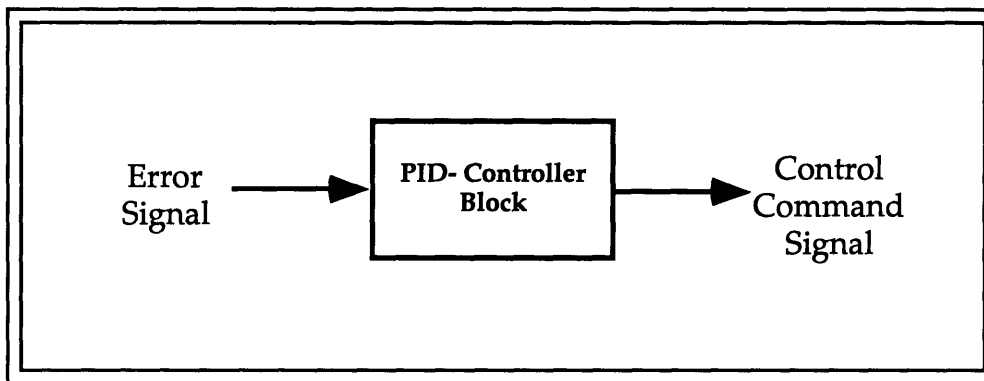


Figure 20. Schematic of PID-Controller Building Block

The proportional gain, derivative gain, integral gain, sampling frequency, offset, and zero position information are supplied to this block by the *.cfg* and *.gui* files (see section on "Input" later in this chapter).

Command Signal Output

Digital-to-Analog (D/A) Building Block

The algorithm for the Digital-to-Analog building block is relatively straightforward. Simply speaking, this function block takes in the command signal output from a controller block and writes it to the I/O address of the D/A port, as shown in Figure 21.

The pre-programmed information for this block is the D/A channel number. The D/A port on the PC-412A will hold steady the analog output value until the next command output signal is received by the D/A block.

Two issues related to the D/A building block deserve special attention: the digital nature of the desired output command voltage and the bit length of the data register.

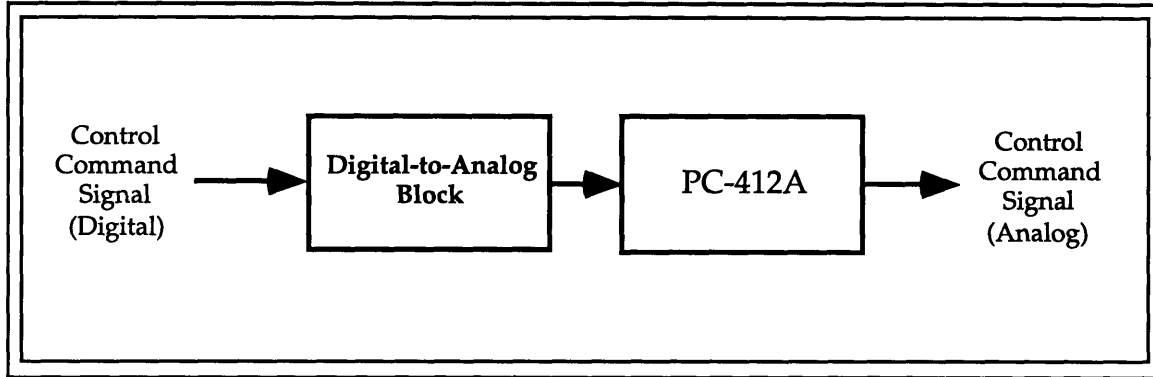


Figure 21. Schematic of D/A Building Block

The four D/A channels supported by PC-412A have resolutions of 12 bits and a range of -5 volts to +4.9976 volts. Because of the discrete nature of the D/A converter, a 12 bit resolution translates to 4096 intervals covering the whole output voltage range, with the value of 0 corresponding to -5 volts, the value of 2048 corresponding to zero volts, and the value of 4096 corresponding to +4.9976 volts. Therefore, the command output voltage received from the controller block has to settle to the nearest lower discrete output voltage supported by the D/A converter.

Since the PC-412A board is designed for connection with the 8-bit ISA bus of the host computer, the transport of the 12-bit word from the host computer to the board via the 8-bit ISA bus has to be done in two parts. The 7 least significant bits (LSB) are transported first, followed by the 5 most significant bits (MSB). For more information on writing to the digital-to-analog ports, refer to Section 3.11 of the [DATEL PC-411/412 Hardware and Software Manual, 1991].

Timer

Timer Block

The timer building block carries the important responsibility of controlling the timing of each digital control loop. The algorithm for the timer building block is arguably the most conceptually complicated of all building blocks described in this chapter. This complexity has two causes:

1. Although it is included on the DATEL PC-412A board, the timer chip was developed by Intel, and has a different programming specification from the rest of the PC-412A board.

2. The timer block is heavily coupled with the real-time control algorithm, while the other building blocks are more independent.

The 82C54 Counter/Timer Chip

First, a brief description of the 82C54 chip is necessary for understanding the timer building block. The 82C54 timer features 3 counters with six different counter modes, 3 of which are used on the PC-411/412 board (refer to Section 4, [DATEL PC-411/412 Hardware and Software Manual, 1991]) The individual functions and modes set for the three counters in the chip are as follows:

Counter #0

This counter specifies the number of A/D conversions per trigger (maximum number of 65,536 conversions per trigger). For real-time control purposes, only one A/D conversion per control cycle is necessary, and the trigger is controlled by the software. Therefore, this counter is unnecessary and is disabled by setting it to Mode 4.

Counter #1

This counter is responsible for controlling the A/D conversion rate, and it is clocked by a 1 MHz internal clock. If an external clock input is used, the maximum conversion rate can go as high as 10 MHz. However, for the purpose of this laboratory, the 1 MHz clock is more than sufficient.

Since the key idea of the timer building block rests in this counter, it deserves a more detailed discussion. Initially, a count number is pre-programmed to this counter by the user through the host computer's local bus and the PC-412A board as described earlier in this chapter.

At the beginning of the timing function, the counter gate signal is high, and this high level validates counting of the 1 MHz clock tick for counter #1. When the counter reaches the pre-programmed count number, the counter gate signal goes low for one clock tick. This low signal triggers an A/D conversion in the analog-to-digital converter. The output signal at the gate returns high again after 1 clock tick, the pre-programmed count number is reloaded, and the cycle repeats.

Obtaining the desired rate of A/D conversion is done by controlling the time duration of the high output period, which can be controlled by writing the count number to the timer/counter chip. For example, if an A/D sampling rate of 1 kHz, or a period of 1 millisecond, is desired. Since counter #1 is clocked by a 1 MHz clock, i.e. 1 microsecond clock tick period, a desired period of 1 millisecond means the counter should count up to 1000 when triggering an A/D conversion. Therefore, to command the counter/timer to generate a rate of 1kHz, one should set this counter to Mode 2 (Rate Generator) and write a value of 1000 to counter #1's register I/O address (BASE + 5).

Counter #2

This counter, clocked by an internal 31.25kHz clock, is responsible for generating the internal trigger signal to start A/D scans. In the real-time control program, a single trigger for A/D conversion is performed by the algorithm during every control cycle. Therefore, this counter is disabled by setting it to Mode 4, as in the case of counter #0.

Table 6 summarizes the function of the counters and the mode they are set to:

Counter Number	Function	Mode Number
0	Count number of A/D conversions per trigger.	4
1	Count number of clock tick of the 1MHz internal clock; control A/D conversion rate.	2
2	Count number of clock tick of the 31.25kHz clock; generate internal trigger signal for starting A/D scans.	4

Table 6. Summary of the Counter Functions and Mode Settings

The 82C54 chip cannot be directly programmed by the real-time control software. Instead, it has to be done through writing command bits to the PC-412A board. An example for programming the timer/counter chip through the PC-412A board is available in PC-412A users manual (refer to Section 4.5, [DATEL PC-411/412 Hardware and Software Manual, 1991]).

The Logic behind the Timer Building Block.

For the real-time control algorithm, the timer needs to answer the two following questions:

Question 1: Is it time to begin another control loop?

Question 2: Did the previous control loop complete within the time of one sampling period?

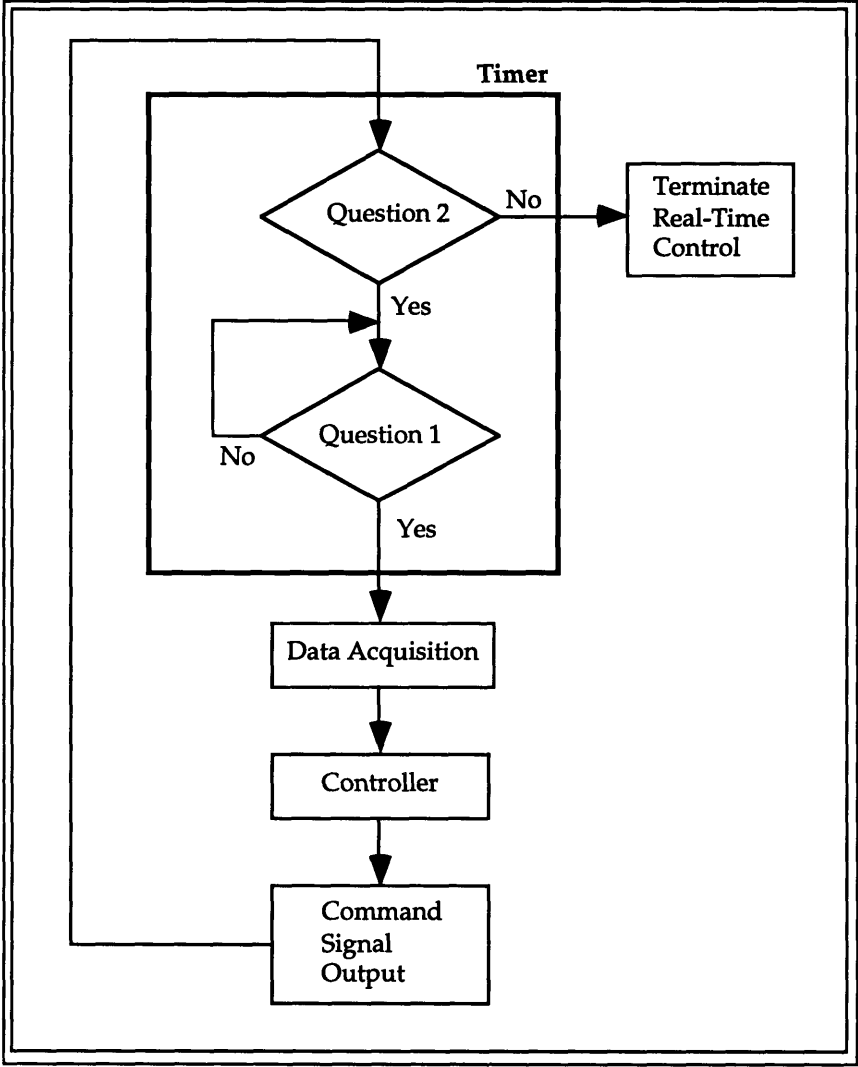


Figure 22. Schematic of the Timer Building Block

The method used by the timer block to answer these questions is described below with the help of Figure 22.

Before the beginning of the real-time control loop, the timer chip's counters are programmed to the set mode as described in Table 6 and the desired control cycle period.

Inside a RTC program is a "while loop" which contains, among others, a block for data acquisition, a controller block, and a command signal output block. The true/false argument for the "while loop" is determined by the timer building block. When the argument for the while loop is true, the blocks inside the loop are executed.

Having set up this structure, the task of the timer building block can be specifically defined as giving out a true value if the answer to both questions listed above are true; a false value when the answer to the first question is false; or termination of the program if the answer to the second question is false.

To answer the two questions, the timer block makes use of the analog-to-digital conversion function of the DATEL PC-412A board, which reads in Counter #1's gate signal from the 82C54 chip for triggering A/D conversions.

The first procedure the timer block performs is to check for an empty flag in the FIFO buffer (read BASE + 0, bit 4). The reason for this will be discussed later in this section. Next, the timer block sends a command to the PC-412A board to enable A/D conversion (write BASE + 0, bit 1). Immediately afterwards, the timer block checks whether the FIFO buffer is empty from the status register (read BASE + 0, bit 4). If the FIFO buffer is empty, it means that the timer chip has not yet triggered an A/D conversion, and the timer block continues to check for the false empty flag again.

As soon as the empty flag for the FIFO buffer is false, indicating that the timer chip has triggered an A/D conversion, the timer block writes to the FIFO reset register (write BASE + 2), which clears all data inside the FIFO data register, and returns a true value to the "while loop" argument, which enables the execution of the real-time control algorithm. This takes care of the first question of knowing when it is time to begin another control loop.

The next question is: how does the timer building block check whether the previous control loop was completed within the time frame of one sample period? The answer to this question is important to determining

whether the sampling rate being used is too fast for the system to handle. The answer to this question is to check the FIFO data register's empty flag at the beginning of the timer building block.

Since the FIFO data register is cleared at the end of the timer block every time it runs, the empty flag will remain empty as long as the timer chip has not triggered another A/D conversion. If, by the next loop, the timer block is called from the "while loop" in the main program and the empty flag is still true, the time it took for the host computer to perform the data acquisition, the control algorithm, the command signal output, and other functions inside the "while loop", was shorter than the time it took for the timer chip's next trigger.

If the completion of the algorithm took longer than one trigger period of the timer chip, the timer building block will find the empty flag for the FIFO data register false as soon as it enters the timer block, thus indicating the sampling rate specified by the user was too fast for the system to handle. If this is the case, the timer block will give an error message to inform the user of this problem and exit the real-time control program.

Reference Signal

Vision System Building Block

The function of this block is to read in data from the video board, which is connected to a CCD camera. Calculations are performed inside this block to find out the exact location of a target object, which is then passed to the main portion of the real-time control algorithm as the reference input signal. Figure 23 shows a schematic of this building block.

Algorithm for Locating Target Objects

To understand the algorithm for locating target objects, it is important to know the fundamentals of how data is collected by the CCD camera and how this data is interpreted and stored in the video board's internal frame buffer.

The black and white CCD camera used in the prototype set-up captures live images and puts them in NTSC format consisting of a total of 324 horizontal pixels and 246 vertical pixels. The video board is connected to the

camera via an RCA cable and connector. The NTSC signals are converted to RGB (Red-Green-Blue) format in the video board. Inside the frame buffer, the values of R, G, and B for each pixels are stored in 3 consecutive 8 bytes, which means that the values of R, G, and B can range from 0 to 256.

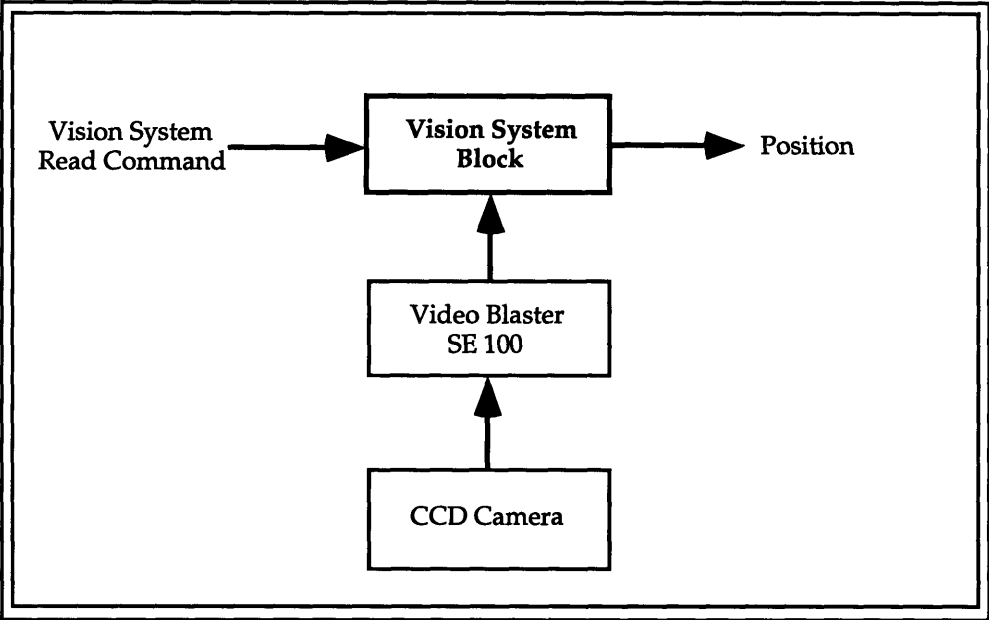


Figure 23. Schematic of the Vision System Building Block

Since the image from the CCD camera is black and white, the magnitudes of the R, G, and B components of each pixel varies proportionally with the brightness. For example, if the image is dark, then the R, G, B values will be low; if the image is bright, then the R, G, B values will be high.

The algorithm downloads the captured image from the frame buffer into the host computer’s memory, and examines the brightness of each pixel. If the target is dark colored and is located in a bright background, to locate this object, the algorithm will search for pixels with “low” R, G, and B values. The criterion for “low” is dependent on the object color, the background color, and the lighting conditions. Usually, it is determined by experimentally placing the target object in front of the background and comparing the R, G, B values for the object’s pixels and the surrounding pixels.

Once the target object is found, the location of the target object must be converted from pixel number to inches. The conversion of pixels to inches is done by placing the target object in a known location and finding the

corresponding pixel number. By this means, a pixel-to-inch conversion factor is obtained.

The location of the target, in inches, is first written to a text file, then loaded into the MATLAB workspace for generating the *.gui* file (see later sections in this chapter, Chapter 6, and [Hintersteiner, 1995] for more detailed discussions on *.gui* files), which is ultimately used in the real-time control program as the reference input signal.

Input

Several parameters are vital to the operations of the RTC software, and they have to be set by the user. These parameters include the controller gains, input type, input magnitude, input frequency (if the input is a sinusoidal one), sampling period, run time, data collection factor, A/D channel, D/A channel. These parameters are stored in two separate text files, one with a *.gui* extension, and the other with a *.cfg* extension.

Configuration Block

The parameter files with *.cfg* as the extension are created by the developer. They contain the values for sampling frequency, run time, data collection factor, A/D channel number, and D/A channel number. For more information regarding the *.cfg* files, refer to [Hintersteiner, 1995].

The function of the configuration input block in the RTC is to retrieve the data contained in the *.cfg* parameter file. The schematic of a configuration block is shown in Figure 24.

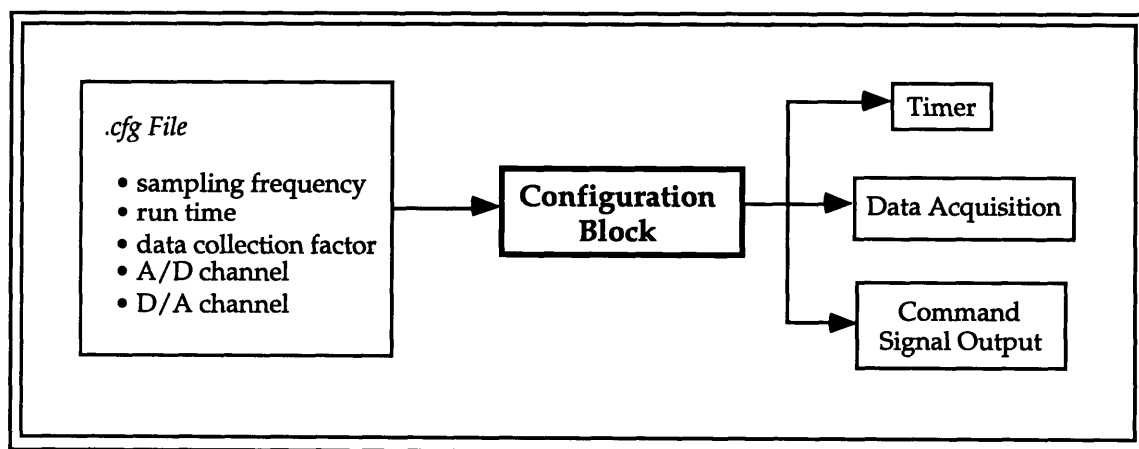


Figure 24. Schematic of the Configuration Block

Graphical User Interface Block

The parameter files with *.gui* as the extension are created by MATLAB *.m*-files, and they contain the values for controller gains (K_p , K_i , K_d), input type (type 1 = step, type 2 = sinusoidal, type 3 = ramp), input amplitude, and input frequency. For more information regarding the *.gui* files, refer to [Hintersteiner, 1995].

The function of the configuration input block in the RTC is to retrieve the data contained in the *.gui* parameter file. The schematic of a graphical user interface block is shown in Figure 25.

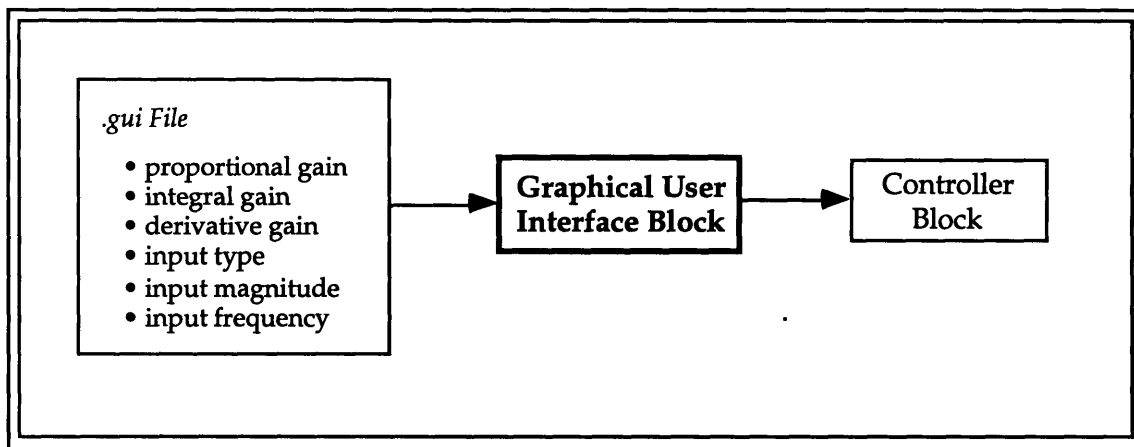


Figure 25. Schematic of the Graphical User Interface Block

Output

Output Block

The output of the RTC experiment needs to be stored for plotting and analysis purposes. The purpose of the output block is to collect the experimental data and write them to a text file with a *.dat* extension. This data file will later be retrieved by MATLAB for plotting and analysis.

In the *.dat* files, the time, output variable signal, the reference signal, the command signal, and an extra signal type are stored in 5 columns. The extra signal can be any signal which the developer finds necessary. For example, in the velocity control RTC program, the extra signal was chosen to be the position of the linear table. In the case where no extra signal needs to be stored, the elements in the extra signal column will be all assigned zero

value. Refer to [Hintersteiner, 1995] for more description of the output block. Figure 26 shows a schematic of the output block.

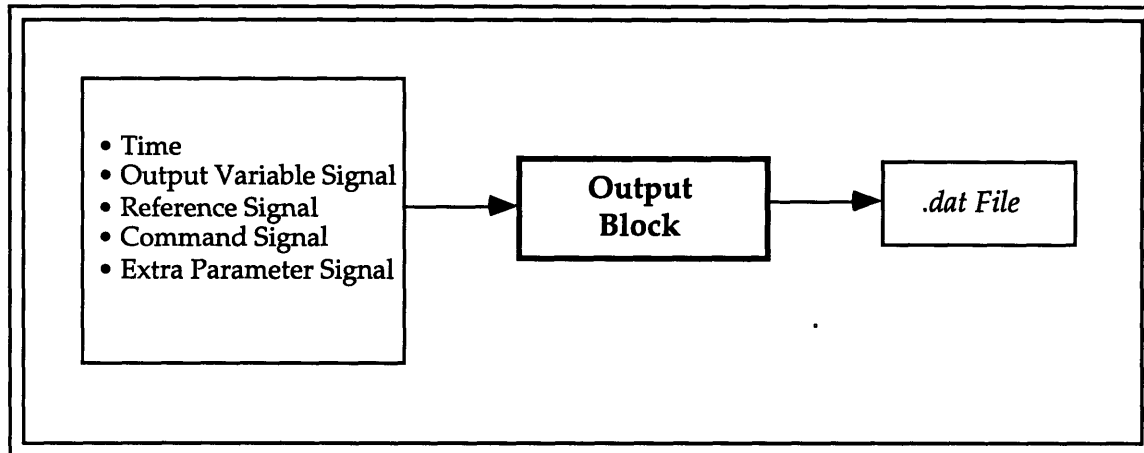


Figure 26. Schematic of the Output Block

Summary

This chapter documented the design and development of the real-time control software for the new System Dynamics and Controls Laboratory. Description of the building block design approach and a general presentation on programming the I/O boards were included. Lastly, in-depth discussions of the RTC blocks were given.

The next chapter is dedicated to documenting the development of the graphical user interface (GUI) software of the new System Dynamics and Controls Laboratory, including descriptions on MATLAB GUI, MATLAB Notebook Suite, Microsoft Word, and Mosaic.

Chapter 6 Development of Graphical User Interface

Introduction

In Chapter 5, the software for linking the Computer Module to the physical hardware modules was discussed in detail. This chapter focuses on the discussion of the software to provide an interface between the users and the Computer Module. Figure 27 shows the overall software architecture which was originally presented in Chapter 5.

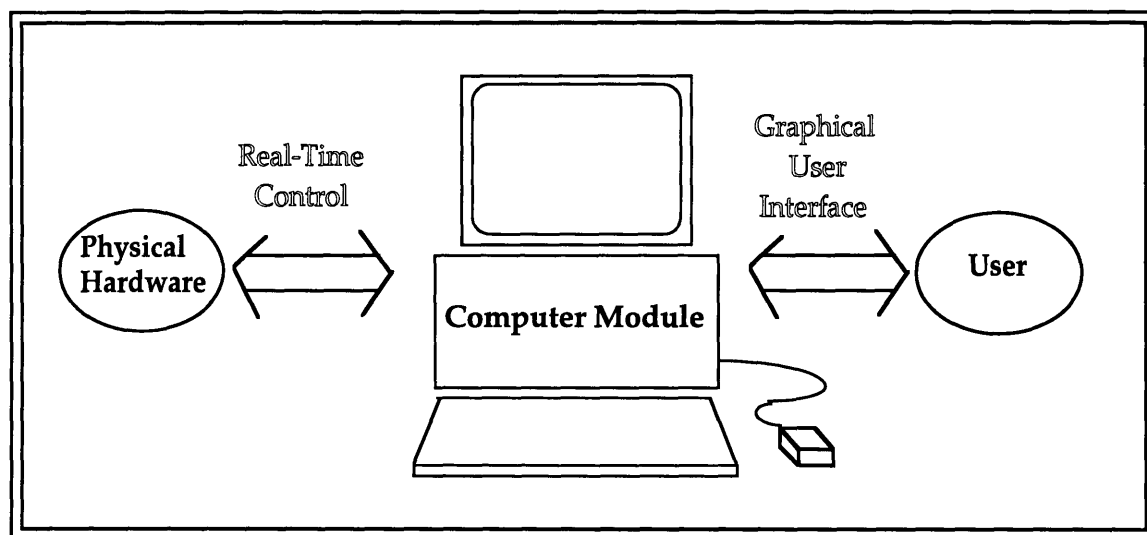


Figure 27. Schematic of Software Communication Architecture

The software enabling the users and the Computer Module to interact is generalized as the user's interface. The approach taken in the development of the new System Dynamics and Controls Laboratory is to present the user's interface with extensive graphics features, such as menus, buttons, lists, and editable text fields, as compared to the command-line oriented interface. In today's terminology, the type of user interface developed in the new System Dynamics and Controls Laboratory is called the Graphical User Interface (GUI).

Four graphical user interface tools are used in the laboratory prototype: Mosaic, MATLAB Notebook Suite, Microsoft Word, and MATLAB Graphical User Interface. This chapter will present the use of these interfaces to accomplish the objectives of the laboratory. It will begin with a brief overview of the GUI functions and the architecture, followed by in depth discussions of each GUI. Particularly, emphasis will be placed on the development of the MATLAB Graphical User Interface.

Overview of GUI Functions

Figure 28 shows the four graphical user interface software packages and their functions. Mosaic is a web browser for searching and obtaining the information available in the World Wide Web. It is mainly used in the laboratory for retrieving laboratory materials and as a means for instructor-student communication.

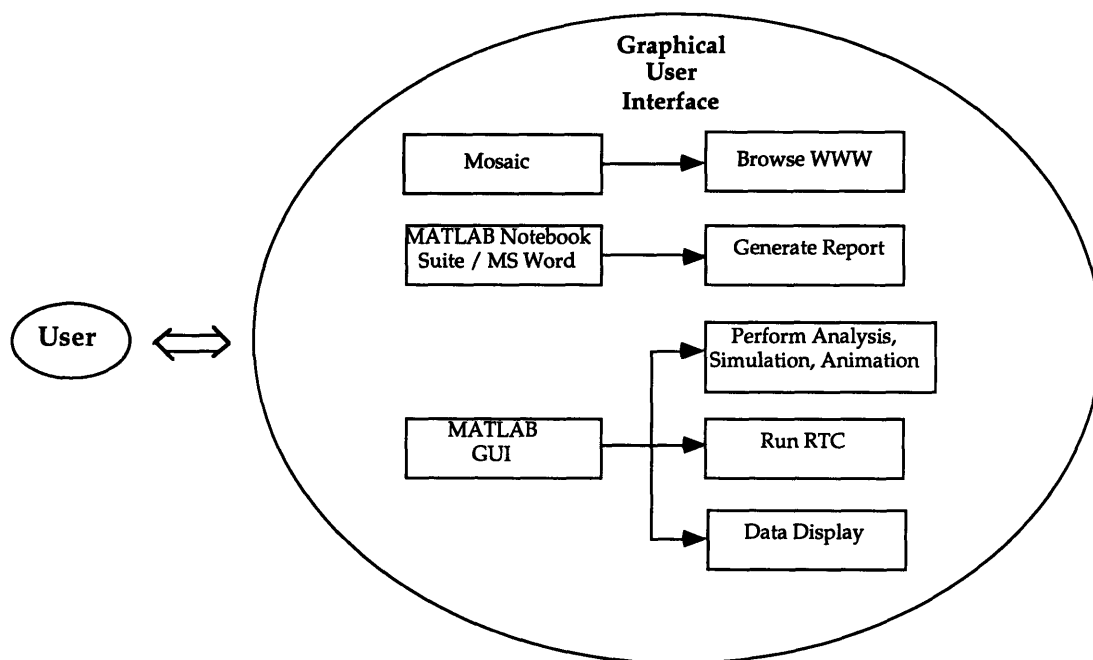


Figure 28. Functions of the Graphical User Interface Software

The student laboratory reports are generated using the MATLAB Notebook Suite and Microsoft Word. The combination of these two software packages enables rapid production of laboratory reports, with plots from simulated and experimental results, as well as other features, in a neat and easy manner.

The MATLAB GUI is responsible for the execution of the laboratory exercises. Through this interface, students can carry out control analyses and simulations, see an animation of the plant response, execute real-time control experiments on the physical system, and retrieve and plot the experimental data in a very user-friendly environment. Icons, pop-up menus, push buttons, and many other easy-to use features are incorporated in the GUI pages to assist the students in navigating among the various functions. More description of the GUI's and their functions are presented in later sections of the chapter.

The GUI Architecture

This section presents how the GUI's are organized and used by the students to connect themselves to all of the components both inside and outside the laboratory. Figure 29 is a schematic of the GUI architecture.

First, the students obtain the laboratory materials from the World Wide Web through the ethernet connection and Mosaic. They then carry out the laboratory procedures through the MATLAB GUI, which connects to the RTC software to run physical experiments. The results from the laboratory exercises are then passed on to a Microsoft Word document via the MATLAB Notebook Suite. The experimental results can also be transported to a student's account on Athena via the ethernet connection.

This architecture design has the of advantage of protecting the software from being tampered by unauthorized personnel, yet allowing the students sufficient access to information for getting a good understanding of the course material. The PC's software can be altered only inside the laboratory, while the World Wide Web software is well protected by the Athena network security. Therefore, it would be very difficult, if not impossible, for any computer "heckers" to break into the laboratory computer system from remote sites and alter the software codes.

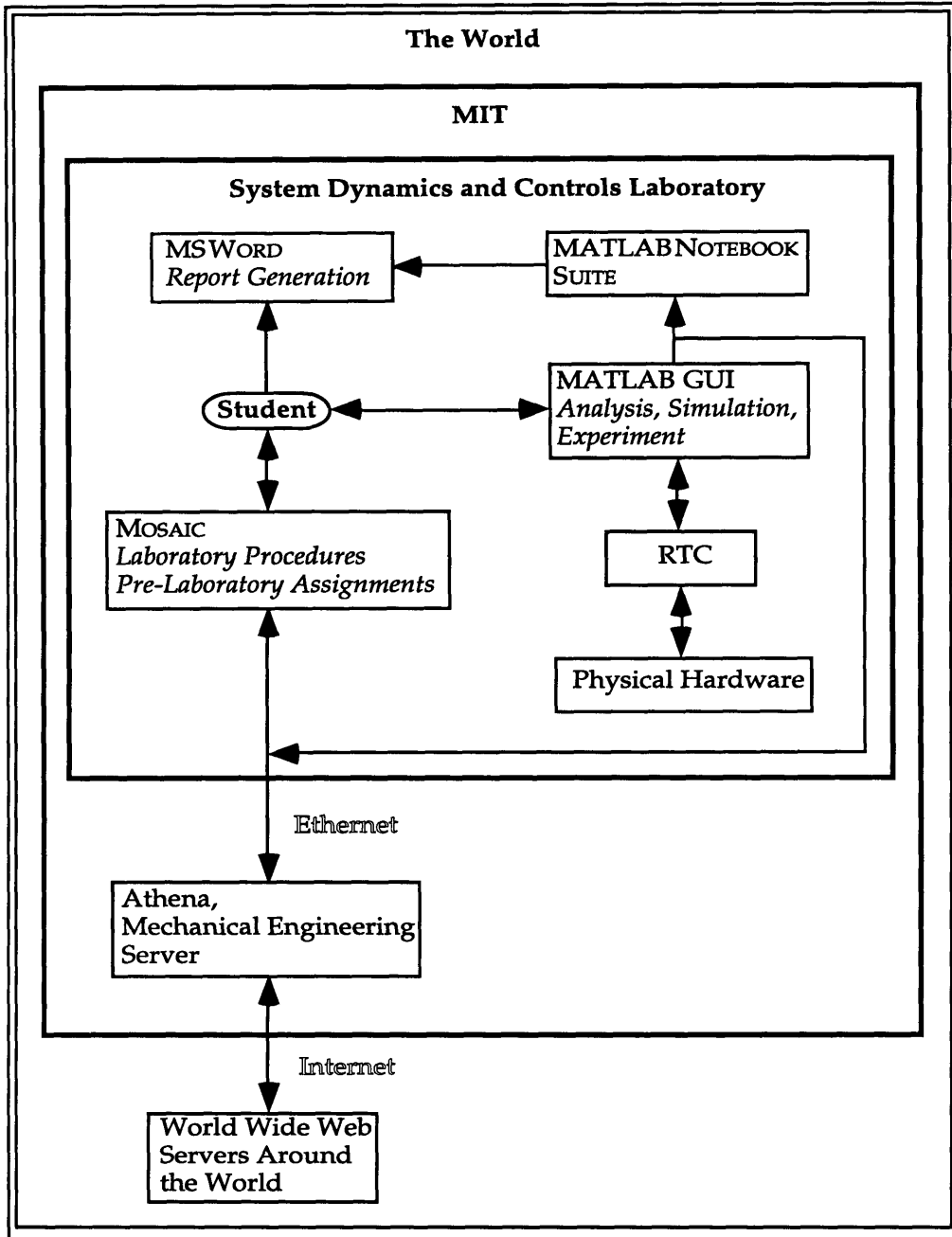


Figure 29. GUI Architecture of the Laboratory

Mosaic and the World Wide Web

It was described in Chapter 3 that the laboratory will tap into the resources of the World Wide Web by via ethernet connections between the laboratory computers and the MIT school-wide network, Athena. In order to

view the materials from the World Wide Web, a “Web Browser” is necessary. Mosaic has been selected as the browser for it is supported by Athena.

Mosaic provides the users the convenient features of hypertext and scroll bars. The operation of this web browser is very intuitive, and students can easily obtain the documentation describing the operations of it, as well as the World Wide Web, from many computer support services inside the Institute.

The World Wide Web URL for the new System Dynamics and Controls Laboratory is <http://me.mit.edu/2.14/>. The web pages, as well as the network connection, are developed by J. Hintersteiner and are fully documented in [Hintersteiner, 1995].

MATLAB Notebook Suite / Microsoft Word

At the end of each laboratory, students are required to hand in a laboratory report. The report should include simulation and experimental data and plots, as well as observations and comments from the students.

MATLAB is well known for its for analysis, simulation, and graphics capabilities; Microsoft Word has become a standard word processing software package for technical reports in the science and technology field today. With the newly developed MATLAB Notebook Suite by Mathworks, Inc., the bests of both worlds are brought together.

The Notebook Suite enables the user to input MATLAB commands directly from a Microsoft Word document. The results generated by MATLAB will then be transported back for display in the Word document using the Window Dynamic Data Exchange (DDE) capability. With this feature, the graphs generated by MATLAB can be plotted in the Word document where the students are writing their laboratory reports.

Laboratory Report Template

In the laboratory, the students are given a laboratory report template in Microsoft Word. The template includes questions needed to be answered by the students along with the spaces for the answers. When students complete a section of the laboratory experiment with the MATLAB GUI, they will plot

the results in the laboratory report in Microsoft Word and type in their observations and comments in the pre-defined spaces.

How to Create an Laboratory Report Template

Creating a laboratory report template is relatively straightforward and requires only general knowledge of Microsoft Word, MATLAB, and the [MATLAB Notebook Suite User's Guide, 1994]. To create a laboratory report template, the developer first needs to know the questions for the students and the graphs that need to be plotted. Next, the developer opens a new Microsoft Word document, selecting the *mbook* template from the Word pop-up menu. When the new document window opens, the developer types in the laboratory questions as a regular Word document. Once all the questions are finished, the developer types in the appropriate MATLAB commands, and defined them as MATLAB commands using the window's menu.

For more details regarding the specifics about how to create an mbook document, refer to [MATLAB Notebook Suite User's Guide, 1994] and [Tesciuba, 1995]. A sample laboratory report is included in Appendix G for reference.

MATLAB GUI

The real-time control software (discussed in Chapter 5) and MATLAB graphical user interface make up the backbone for the laboratory's operation. Through the MATLAB GUI, students can conduct control analysis, simulation, animation, run real-time control, plotting simulation and experimental results, and more.

The primary way to operate the MATLAB GUI is via the mouse. In certain cases, students are prompted to input values by the keyboard. However, most function input simply requires clicking with the mouse at the buttons or menu pages displayed on the computer monitor.

MATLAB GUI Architecture

MATLAB GUI's are organized in groups according to their functions. A schematic of this organization is shown in Figure 30.

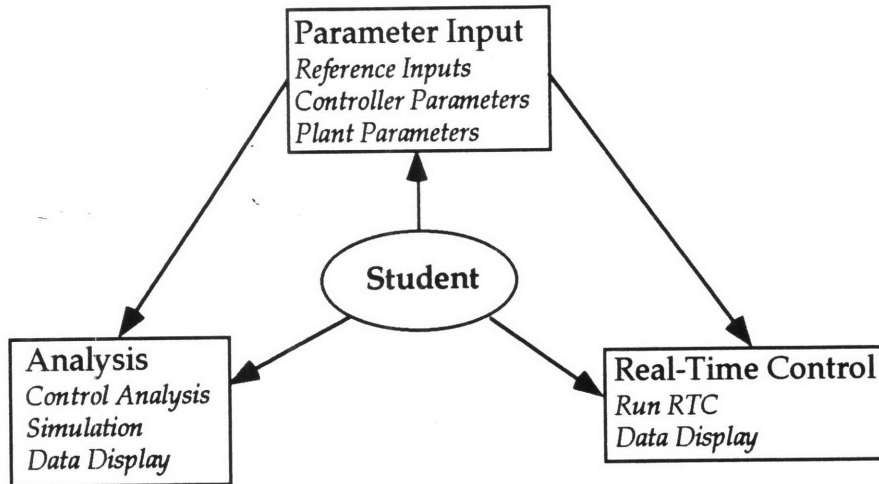


Figure 30. MATLAB GUI Architecture

The functions of the GUI pages can be generalized into three categories:

- Parameter Input
- Analysis
- Real-time Control

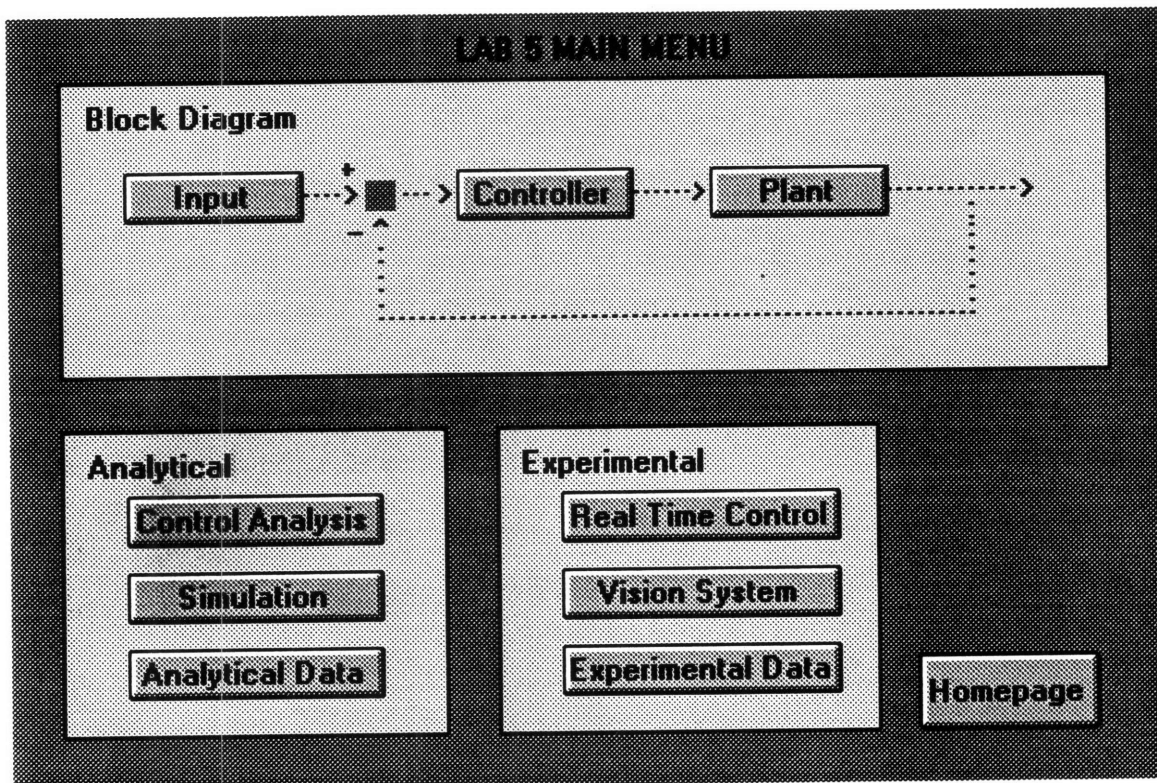


Figure 31. Typical MATLAB GUI Menu Page

The layout of the GUI pages are organized by their function types. For instance, on the menu GUI page, the parameter input buttons are grouped together at the top portion of the window page, the analysis buttons are grouped together at the lower left portion, and the real-time control portions are grouped together in the lower right portion (see Figure 31). This layout is designed so that the students can learn to “get around” the GUI pages with minimal effort.

How is a MATLAB GUI Page Written?

Before proceeding to in depth discussions of each GUI function groups, it is important to have a brief understanding of how a MATLAB GUI page is written.

First, a GUI map should be designed. This GUI map describes the organization and sequence of the GUI pages. An outline of the function of each GUI page should also be included in the GUI map.

Prior to actual programming each GUI page, the layout of a GUI page should be designed on paper. What type of control buttons to be used? Where to put these buttons? What color should it use? What information should be conveyed in this page? Are these buttons and information neat and easy to understand? These are all questions to be answered by the developer.

<i>Script</i>	<i>Description</i>
<code>uicontrol(gcf,...</code>	Initiate User Interface Control
<code>'Style', 'push',...</code>	Define style of button
<code>'Position',[10 10 100 25],...</code>	Define the location of the button
<code>'String','Start Plot',...</code>	Define the text on the button
<code>'Callback','plot(time, value)');</code>	Define the action by MATLAB

Figure 32. A Typical MATLAB GUI Script

Finally, the developer carries out the actual programming of the GUI page. Each GUI page is generated by a set of MATLAB commands. These command scripts are stored under one or more MATLAB *.m-files* (refer to

[MATLAB User's Guide, 1994] for more information on *.m-files*). The command scripts contain the information about a specific button, such as what type of button it is, where the button is located, what string is displayed on the button, and a callback command which tells MATLAB to perform an action when the button is activated. Figure 32 shows a typical MATLAB command script for creating a GUI "push" button.

A typical MATLAB GUI page contains a few of these scripts, depending on how many buttons are included. More examples can be found in the MATLAB's manual [Building a Graphical User Interface, 1994]. The following sections will be in depth discussions on each GUI function groups.

Parameter Input Group

The MATLAB GUI pages in the parameter input group carry the function of retrieving parameter values from the users. There are three primary types of parameters: reference input parameters, controller parameters, and plant parameters. The buttons for this group of function are located at the top of the menu page (see Figure 31) in the form shown in Figure 33.

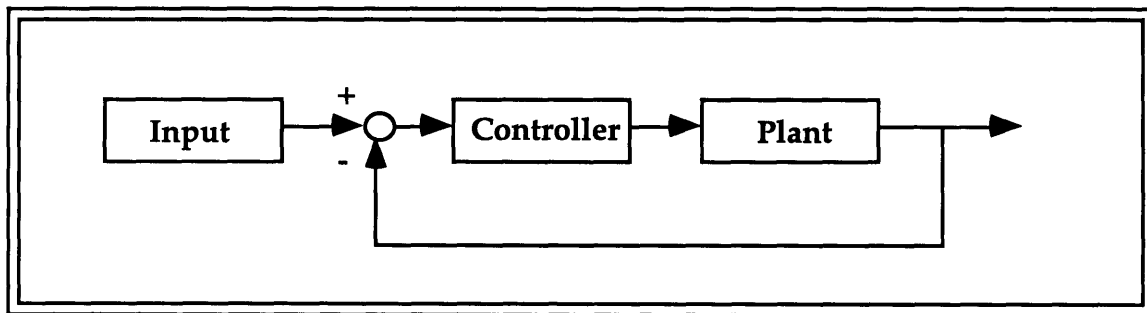


Figure 33. Parameter Input Group

Reference Input Parameters

This set of GUI pages is specifically for prompting the users the reference input for the control system. To enter this set of GUI pages, click the "Input" push button shown in Figure 31.

Inside this domain, users can select three types of reference inputs: step, sinusoidal, or ramp. The selections and the necessary input parameters are summarized in Table 7.

Input Selections	Input Parameters
Step	Magnitude
Sinusoidal	Magnitude, Frequency
Ramp	Slope

Table 7. Reference Input Parameters

After choosing the input selection and entering the corresponding parameter values, the user can return to the main menu by clicking the “Main Menu” button, or go to real-time control group by clicking the “Run RTC” button. The input parameters are saved in the MATLAB workspace automatically.

Controller Parameters

This set of GUI pages is specifically for prompting the users for the controller parameters of the controller. To enter this set of GUI pages, click the “Controller” push button shown in Figure 31.

Inside this domain, users can adjust the proportional gain, the derivative gain, and the integral gain of the controller by a slide bar or by editable text. The user can return to the main menu or go to real-time control mode by clicking the “Main Menu” or “Run RTC” buttons, respectively. The controller parameters are saved in the MATLAB workspace automatically.

Plant Parameters

This set of GUI pages is specifically for prompting the users for the plant parameters. To enter this set of GUI pages, click the “Plant” push button shown in Figure 31.

Inside this domain, users can input the plant parameter by two ways: directly input the plants transfer function, or directly input the plants physical parameters. If the user selects the first way, the plant’s transfer function will be input to the GUI page as two entries: the numerator of the transfer function, and the denominator of the transfer function, both in descending order of power. For example, if the transfer function of a plant is:

$$\text{Transfer Function} = \frac{s}{s^4 + 2s^3 + 4s^2 + 2s + 1}$$

Then input to the GUI should be in the following format:

numerator: [1 0]
denominator: [1 2 4 2 1]

The second way to input the plant parameters is to enter a list of plant physical parameter values. For example, the physical parameters necessary for determining the linear slide system’s transfer function are time constant (τ) and open loop gain (K). The GUI will prompt the user for these parameter values.

The transfer function can be generated from a set of MATLAB command scripts stored in a separate *.m-file*, which performs the calculations based on the physical values input by the user. This *.m-file* is activated by the “CallBack” string in the “Transfer Function” click button’s script.

After inputting the plant parameters, the user can return to the main menu or go to real-time control group by clicking the “Main Menu” or “Run RTC” buttons, respectively. The plant parameters are saved in the MATLAB workspace automatically.

Analysis Group

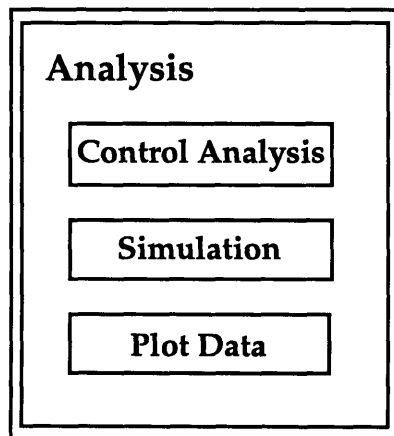


Figure 34. Analysis Group

The MATLAB GUI pages in the analysis group carry the functions for enabling users to perform control analysis, simulation, animation, and

simulation result displays. The buttons for this group of functions are located at the lower left portion of the menu page (see Figure 31) in the form shown in Figure 34.

Control Analysis

This set of GUI pages activates a number of MATLAB commands such as *rlocus* (Root Locus analysis) and *bode* (Bode analysis). The user can select the desired control analysis tool by a pop-up menu. The parameters needed to perform these analysis are automatically loaded up from the MATLAB workspace. The analysis result will be displayed on a separate page. A sample output of Bode plot is shown in Figure 35.

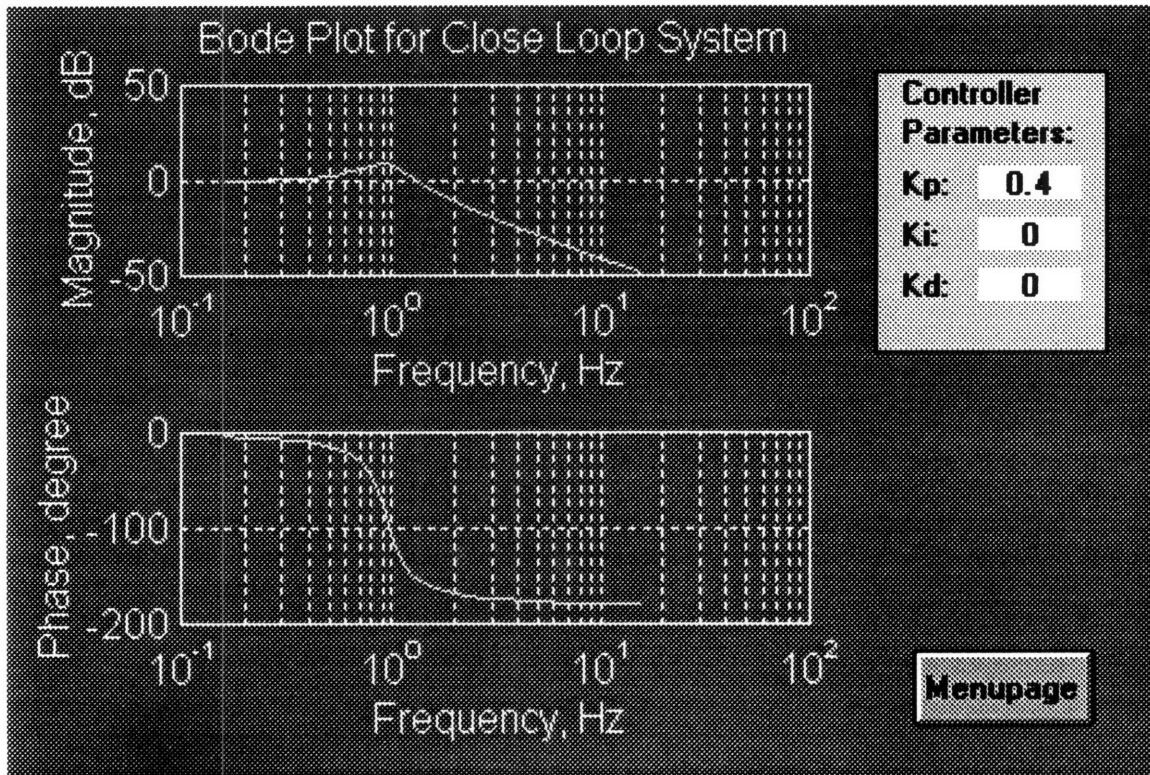


Figure 35. Sample Output from Control Analysis Function

Simulation

This set of GUI pages activates the Simulink software for simulation. The users can select the model they wish to simulate from a pop-up menu

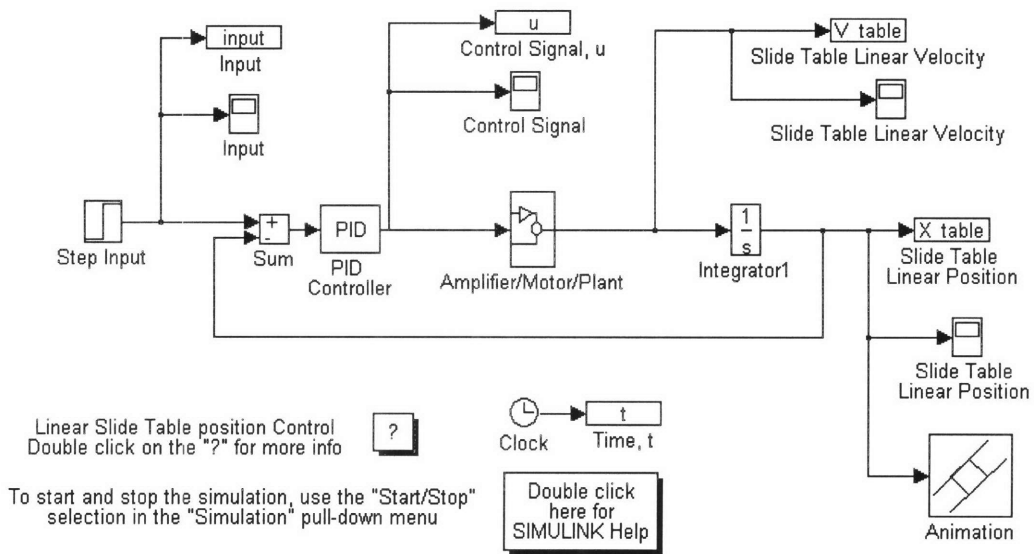


Figure 36. Typical Simulink Block Diagram

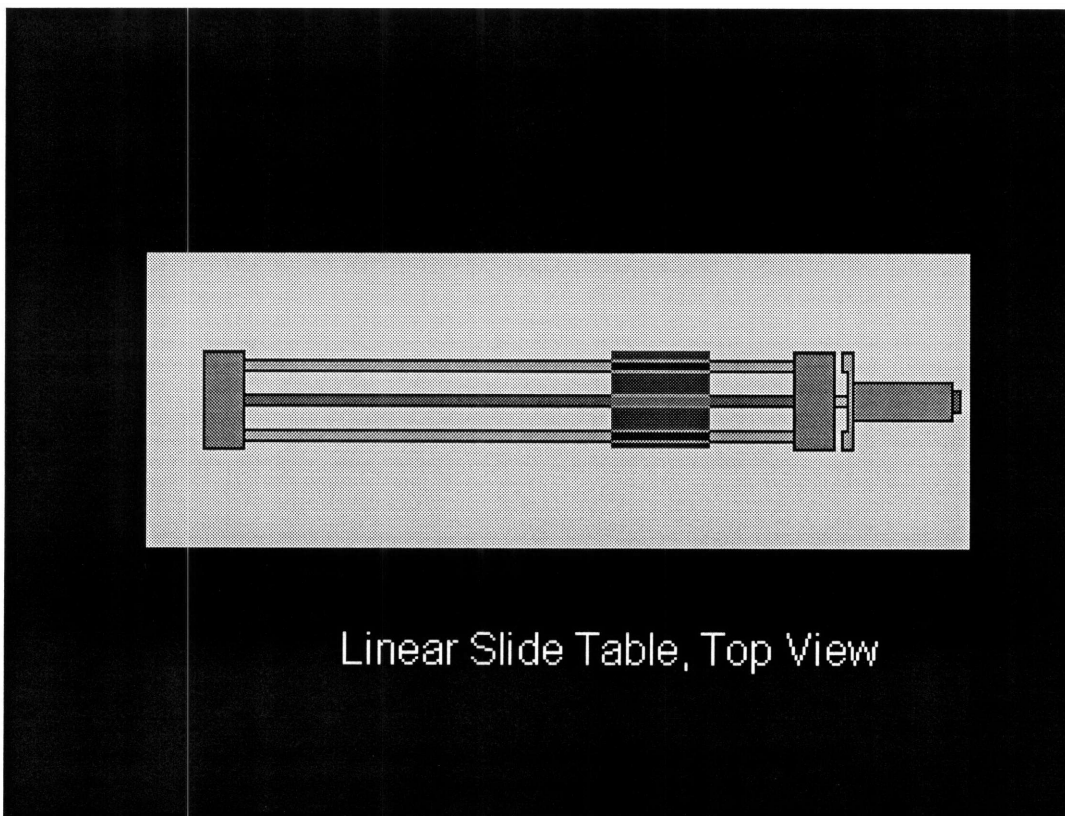


Figure 37. Typical Frame of Animation

and the corresponding file is loaded up by Simulink. The parameters needed to perform the simulation are automatically loaded up from the MATLAB workspace. See Figure 36 for a typical Simulink block diagram.

A MATLAB animation window [Tesciuba, 1995] displaying the simulation results will automatically pop up once simulation begins. It is aimed that the users can gain a better feel and understanding from watching an animation of the system response than by just looking a data plot. A typical frame of an animation is shown in Figure 37.

Simulation Data Display

This set of GUI pages displays the data generated from simulations. Users can select the parameter they wish to display from a pop-up menu. A typical display of simulation data is shown in Figure 38.

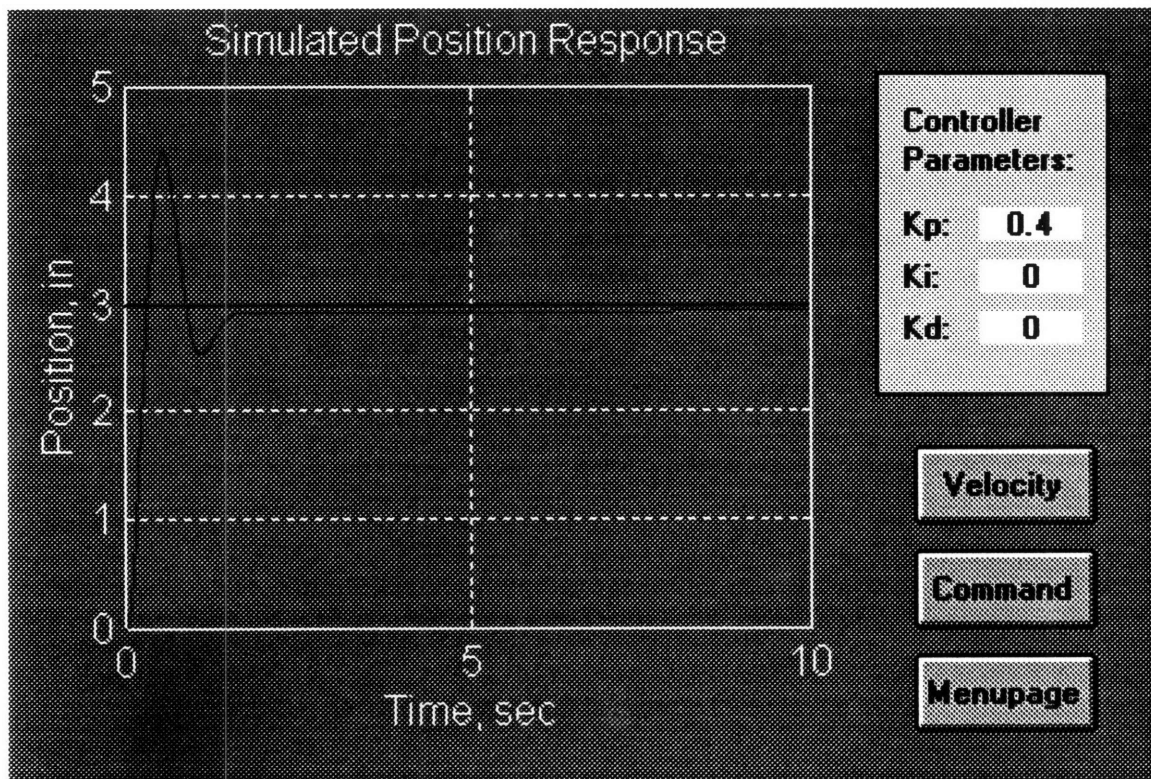


Figure 38. Typical Simulation Data Display

Real-Time Control Group

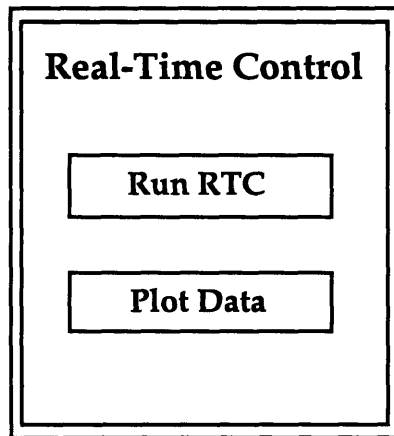


Figure 39. Real-Time Control Group

The MATLAB GUI pages in the real-time control group carry the function of enabling users to perform real-time control to the experiment physical set-ups. The buttons for this group of function are located at the lower right portion of the menu page (see Figure 31) in the form shown in Figure 39.

Run RTC

When the “Run RTC” button is clicked, a MATLAB GUI page summarizing the controller values and reference inputs is displayed. The purpose of this is to allow the user to check whether these values are correct prior to running the real-time control experiment. If any of these values need to be modified, two push buttons are available for direct access to the parameter input group for inputting the correct reference input and controller values. If the controller and reference input values are correct, the user can click a “Start” button, which creates a *.gui* file and calls up the RTC program.

Creating a .gui File

The nature of *.gui* files were discussed in Chapter 5 and [Hintersteiner, 1995]. They store the necessary information, such as controller gains, input types, input magnitude and/or frequency, for running a real-time control experiment. To create a *.gui* file, a MATLAB script using the commands *fopen*, *fprintf*, and *fclose*, was written and is executed when the “Start” button is clicked.

Calling RTC Programs from MATLAB

The execution of RTC programs from MATLAB can be done by a command which consists of an exclamation point (!) before the RTC executable file name. The exclamation point indicates to MATLAB that the rest of the input line is issued as a command to the operating system, in this case, DOS. For example, to run the RTC program for closed loop position control for the linear slide table plant, write the following script in a real-time control *.m-file*:

```
!ls.exe
```

where *ls.exe* is the executable file for running the real-time control experiment.

Experimental Data Display

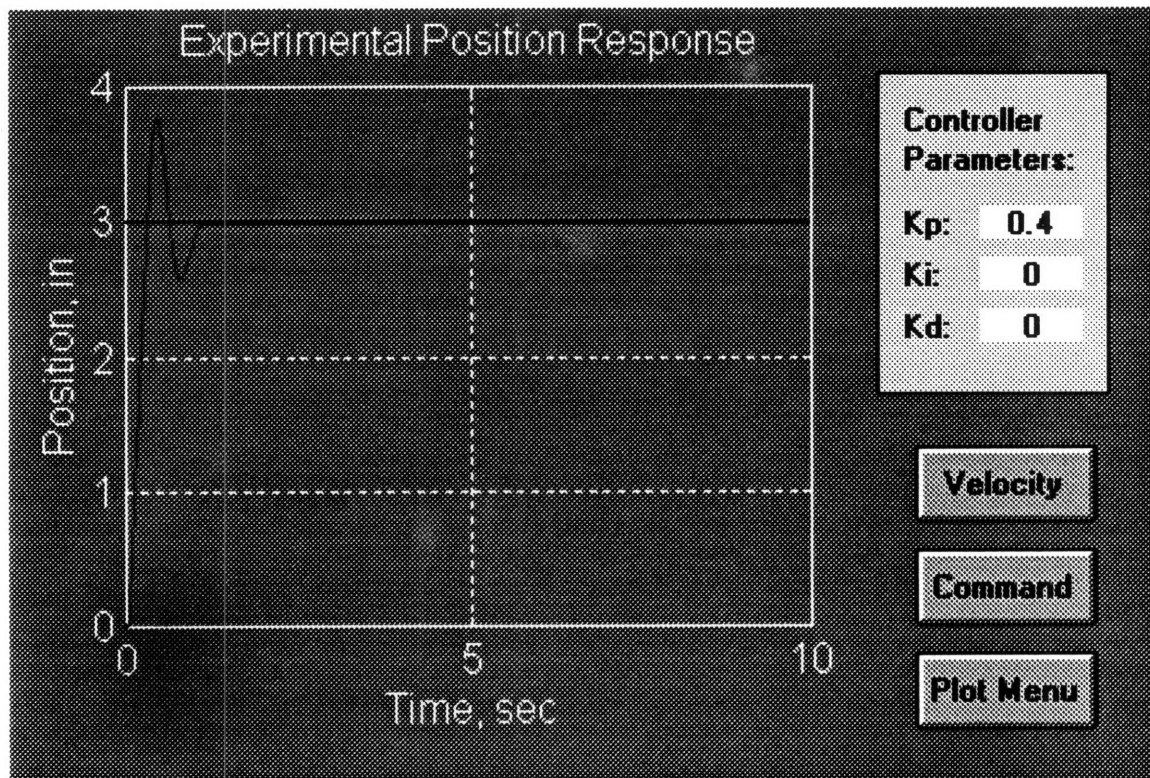


Figure 40. Typical Experimental Data Display

This set of GUI pages displays the data generated from the real-time control experiments. Users can select the parameter they wish to display from a pop-up menu. A typical display of experiment data is shown in Figure 40.

Developing New MATLAB GUI Sets

The discussion in this section focuses on the overall development of the MATLAB GUI pages rather than on developing each single page. Developers can refer to MATLAB's [Building a Graphical User Interface Manual, 1994] for details in creating MATLAB GUI pages.

The development of the MATLAB graphical user interface runs into a problem of the pages being too laboratory specific. Since the function of GUI pages are heavily dependent on each laboratory set up and experiment procedures, it is difficult to create generic blocks for each purposes similar to the design approach taken for the real-time control program discussed in Chapter 5. Moreover, although MATLAB programming language is more user friendly than C language, it is far less modular. Therefore, rather than writing building blocks for each specific tasks, a complete set of MATLAB GUI pages is written and stored in a separate directory, called *DEVELOP*.

To develop a new set of MATLAB GUI pages for a new laboratory, just simply load up the files in the *DEVELOP* directory and start modifying the programs to tailor the specific needs of the new laboratory. This way, the developer does not have to start from scratch every time he/she develops a new set of GUI's needs to be developed.

This design approach has the advantage of permitting the software to mature and expand in capabilities as more developers contribute new function libraries to the *DEVELOP* directory. This is in coherent with the open architecture design objective stated in Chapter 2.

In order to save valuable hard disk space, the *DEVELOP* directory need not be stored in every workstation's computer module, since it should never be used other than by the developer.

Summary

This chapter documented the design and development of the graphical user interface software for the new System Dynamics and Controls Laboratory. It began with an overview of the user interface software architecture, followed by descriptions for the Mosaic and the World Wide Web, MATLAB Notebook Suite and Microsoft Word. These were followed by

an in depth discussion on the MATLAB GUI and some examples of actual display output. Lastly, a guideline for developing a new MATLAB GUI set was presented.

The next chapter is on the student testing of the prototype set-up. Description of the student testing procedures, results of this test, and suggestions on future improvements are presented.

Chapter 7 Student Testing

Introduction

Student testing of the new System Dynamics and Controls Laboratory prototype set-up was conducted. These tests served two purposes: to obtain overall feedback from the end users, and to explore the areas that may require debugging or improvement. This chapter documents the student testing procedures and results. At the end of this chapter, suggested improvements in future set-ups are presented.

Student Testing

Demographics of students

Altogether, 11 students volunteered to do the student testing. Among them, 10 are Mechanical Engineering majors and 1 is a double major in Mathematics and Mechanical Engineering.

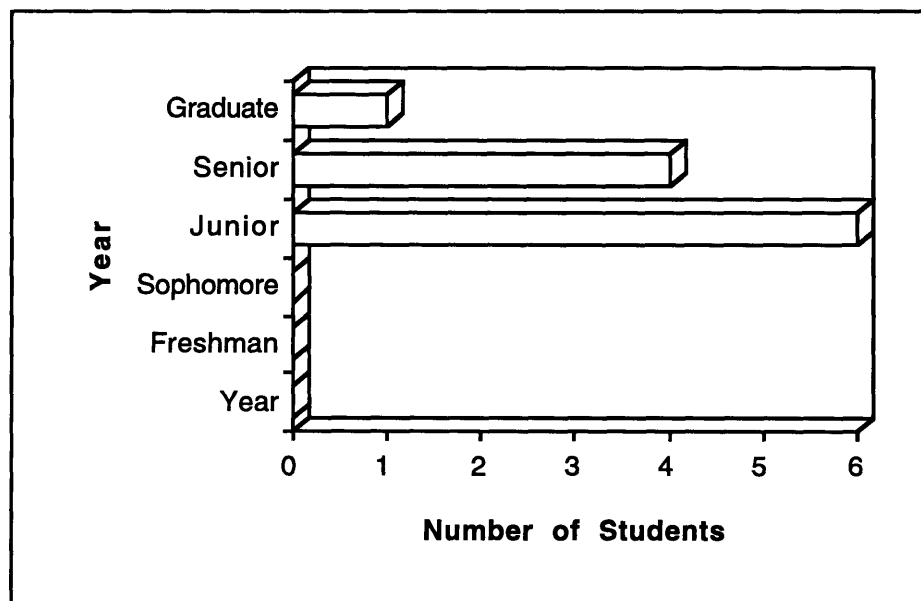


Figure 41. Volunteer Student Demographics by Year

All of them are currently enrolled in the Mechanical Engineering Department's Controls Systems Principles course, and have worked in the existing Controls Systems Laboratory. The years of the students are shown in Figure 41.

Test Method

The students were separated into two main groups, one worked on the electromechanical module and the other worked on the fluids module. Most of the student volunteers worked in groups of two, in keeping with the proposed course logistics described in Chapter 3.

A total of six sample laboratory exercises were designed, three for the electromechanical module (see Appendix G for one of the sample exercises) and three for the fluids module (refer to [Tesciuba,1995]). The three sample laboratories for the electromechanical module cover the topics of modeling, velocity control, and position control of the linear slide plant.

Six students performed the sample laboratories on the electromechanical module and five students performed the sample laboratories on the fluids module. All the student volunteers performed at least one sample laboratory exercise, after which they were asked to complete a survey form.

Survey

The survey form was primarily written by M. Tesciuba with additional input from the author (Appendix H). In addition to asking for student backgrounds (such as year and major), the survey focused on three topics: (1) How does the new laboratory prototype compare with the existing laboratory set-ups? (2) Is the new hardware and software capable of conveying the controls concepts effectively? (3) Are there any suggestions for improvements?

The survey form uses both numerical ratings as well as expository style questions.

Results

Comparisons of the existing laboratory set-up and the new laboratory prototype in three aspects were made on a 10 point scale: 1 being the new laboratory is much worse than the existing laboratory; 5 being about the same; 10 being much better. The survey results are shown in Figures 42 - 44.

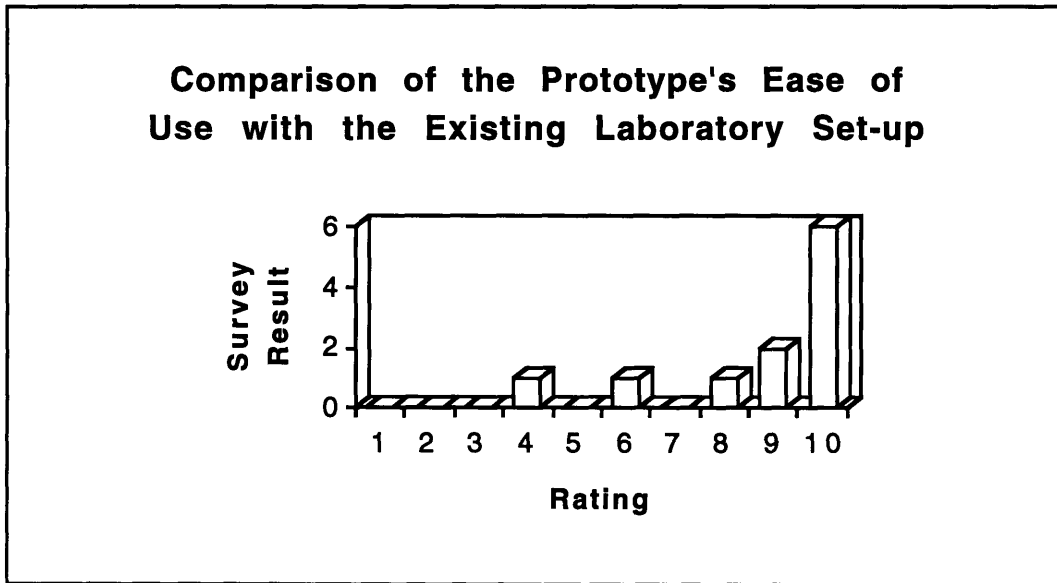


Figure 42. Survey Result

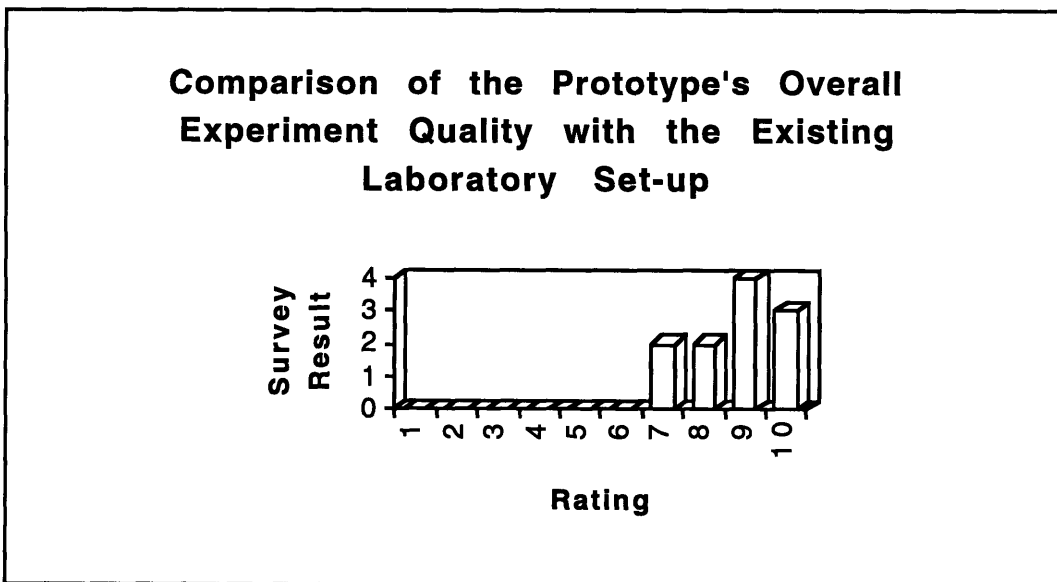


Figure 43. Survey Result

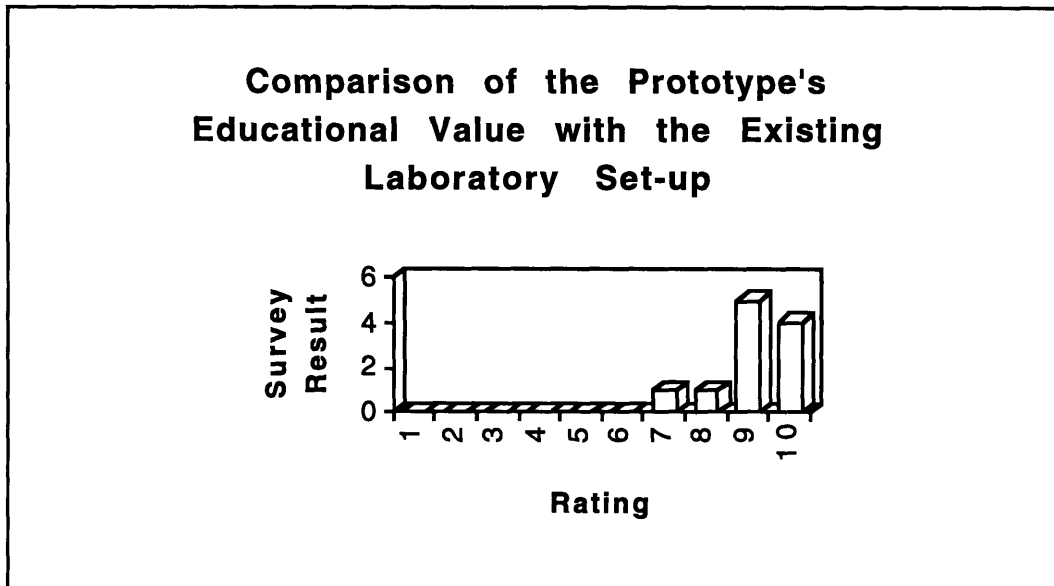


Figure 44. Survey Result

Comments on the GUI's

Most students responded very favorably to the evaluation of the educational effectiveness of the new prototype set-up. When asked to comment on the graphical user interface of the new set-up, most students stated that the GUI pages have a positive impact on enhancing the understanding of the laboratory materials. Some of the comments were: “very easy to get used to (the GUI’s)”; “The GUI allows me to forget about how to implement the experiment, and I can focus on the concepts being taught”; “much more intuitive than the old lab”; “pretty user friendly”; “very comfortable”.

Comments on the Electromechanical Module

The students also liked the electromechanical module. They believed it was safe to operate, especially with the Plexiglas cover, the limit switches at the ends of the table, and the speed limit of the table. Some comments which reflect what they liked best about the electromechanical module were: “The

ease of repositioning the module, adding additional inertia (and) friction elements”; “(The electromechanical module) is much more self explanatory than using Comdyna (and the rotational inertial plant) because it (is) translational instead of rotational”; “Neatly designed with housing (and) relatively quiet while in operation”; “Encased in clear container (safety cover) allows you to see the plant in action”

Comments on the fluids and electrical modules can be found in [Tesciuba, 1995]. In addition to the general comments on the new prototype set-up, the volunteers were also asked to give suggestions on improvements. These suggestions are summarized in the next section.

Suggested Improvements

Graphical User Interface

Although the frame work for the graphical user interface software has been established, was well received by students, and is fully implementable, additional improvements can be made to make the GUI even better. Some improvements suggested by the students are listed in the following:

- allow the experimental and simulated results to be plotted in the same window simultaneously;
- use bigger windows;
- provide a menu bar on top of the window;
- use another GUI builder which allows for more features and faster speed.

Electromechanical Module

The following summarizes the students suggestions on the electromechanical module:

- include a reset button for returning the linear slide to the zero position;
- reduce noise and friction;

- install a linear optical encoder to compare actual linear position with a reference position;
- include a design project/game/competition such as putting an egg on the handle and designing a motion controller such that the egg will not fall off.

Summary

This chapter described the student testing of the new laboratory set-up. The results of the survey were summarized and suggestions on future improvements were presented.

The next chapter is the conclusion to this document. It will state the contribution of this work towards the whole project, as well as the impact of this project on the development of future educational laboratories.

Chapter 8 Conclusion

Contribution of This Work

This thesis describes the design and development of an educational laboratory for teaching system dynamics and controls in the Mechanical Engineering Department at MIT. A prototype laboratory set-up is fabricated and software is created to demonstrate the feasibility of the design concept.

A summary of the intellectual requirements and objectives was presented. An overview of the whole system design was also given.

The design and development of an electromechanical system was presented in this thesis. Details on components selection and fabrication were included.

The fundamental software programs for operating the physical hardware are documented. The software architecture was presented and the individual components of the software were discussed in detail.

Design and development of user interface software was described. An overview of the user interface structure was presented. The approach of using extensive graphics in the user interface, and the use of commercial and public software for development, were discussed. The design and implementation of MATLAB graphical user interface was presented in great detail.

Student testing of the prototype set-up was documented. The results of the tests were summarized and suggestions on future system improvements were given.

Significant amount of details from this work are documented in the appendices, including hardware specifications, a total cost breakdown of the Computer Module, and logistics on the course operation. This will hopefully provide some benchmarks for future developers of this laboratory.

Impact of This Project on Future Laboratory Course Developments

The results of this project have impact not only on the development of the new System Dynamics and Controls Laboratory, but possibly also on other future laboratory developments as well. Design ideas and approach taken in this project can be applied to the development of educational laboratory in many fields of engineering.

For example, the framework of the user interface software demonstrated in the prototype set-up can be adopted for use in other laboratory environments. Another example will be the modular design approach of the physical hardware which permits new experiment set-ups to be incorporated easily.

The documents which describe the design and development of this laboratory prototype are available, so are the source codes for the software, as references for developers who are interested in creating educational laboratories.

Bibliography

Advance Motion Controls, *Advance Motion Controls PWM Servo Amplifiers Catalog*, Camarillo, CA, 1992.

K. J. Astrom, A. Ostberg, *A Teaching Laboratory for Process Control*, IEEE Controls System Magazine, October 1986.

K. J. Astrom, *Education in Automatic Control at Lund Institute of Technology*, American Control Conference Proceedings, Boston, MA, June 1991.

D. M. Auslander, R. Horowitz, *Mechanical Engineering Control System Curriculum*, American Control Conference Proceedings, Atlanta, GA, June 1988.

J. W. Baish, E. J. Mastascusa, *Inexpensive Experiments for Undergraduate Control System Laboratories*, American Control Conference Proceedings, Atlanta, GA, June 1988.

J.E. Beaini, L. Althaus, J. C. Olabe, E. Welch, *Integrated Electromechanical Controls Laboratory using Programmable Logic Controllers*, American Control Conference Proceedings, San Francisco, CA, June 1993.

M. Bodruzzaman, S. Zein-Sabattoo, S.S. Devgan, R. Shanidain, *PC-Based Data Acquisition and Analysis in Circuits Lab*, ASEE 1992.

K. Bonner, R. Fisch, P. Kalata, *A generic, Hands-on Control System Laboratory*, American Control Conference Proceedings, Atlanta, GA, June 1988.

E. K. Boukas, *Automatic Control Laboratory and Project Course*, American Control Conference Proceedings, Boston, MA, June 1991.

K. C. Cheok, N. A. Kheir, *A Computer Visualization Teachware for Evaluating the Performance of Control System*, American Control Conference Proceedings, San Francisco, CA, June 1993.

Creative Technology Ltd., *Video Blaster SE100 DOS Application Programming Interface*, Milpitas, CA, 1995.

M. K. Das, K. C. Cheok, *Electrohydraulic Servomechanism - A Laboratory Project for Undergraduate/Graduate Control Courses*, American Control Conference Proceedings, San Francisco, California, June 1993.

DATEL, Inc., *DATEL PC-411/412 Hardware and Software Manual*, Mansfield, MA, December 1991.

J. K. Eaton, *LabTutor Course for Instruction in Laboratory Computer Use and LabView Programming*, Stanford, CA, 1994.

K. T. Erickson, *Experiments for An Undergraduate Automation Laboratory*, American Control Conference Proceedings, San Francisco, CA, June 1993.

A. Feliachi, *Sontrol Systems Curriculum National Survey*, American Control Conference Proceedings, Boston, MA, June 1991.

G. F. Franklin, J. D. Powell, *Digital Control at Stanford*, American Control Conference Proceedings, Atlanta, GA, June 1988.

J. Hintersteiner, *Development of Network and Real-Time Control Software for the System Dynamics and Controls Laboratory*, MIT, June 1995.

J. Hintersteiner, G. Soto, M. Tesciuba, R. Wang, *Laboratory Manual*, MIT, June 1995.

G. M. Huang, L. Fu, J. Fleming, *The Development of An Undergraduate Laboratory for Control System Design*, American Control Conference Proceedings, San Francisco, CA, June 1993.

E. Ibrahim, R. Cockrum, G. Herder, F. Smith, *Integrating Computers and Industrial Hardware in Instrumentation and Control Education*, ASEE Annual Conference Proceedings, 1993.

A. Jana, S. S. Cheh, *Developing Instrumentation Laboratory with Real Time Control Component*, American Control Conference Proceedings, San Francisco, CA, June 1993.

R. E. Klein, *Novel Systems and Dynamics Teaching Techniques using Bicycles*, American Control Conference Proceedings, Atlanta, GA, June 1988.

P. Kopacek, N. Girsule, R. Probst, *Education in Robotics by Means of PC's*, American Control Conference Proceedings, Atlanta, GA, June 1988.

B. T. Kulakowski, *Automatic Control Laboratory: New Approach to Classical Problems*, American Control Conference Proceedings, Atlanta, GA, June 1988.

M. K. Masten, *Challenges for Control System Educators*, American Control Conference Proceedings, Boston, MA, June 1991.

Mathworks, Inc., *MATLAB Building a Graphical User Interface*, Natick, MA, June 1994.

Mathworks, Inc., *MATLAB Notebook Suite User's Guide*, Natick, MA, June 1994.

Mathworks, Inc., *MATLAB User's Guide*, Natick, MA, June 1994.

W. S. Levine, N. Enrich Leonard, W. P. Dayawansa, *An Integrated Undergraduate Controls Experience*, American Control Conference Proceedings, San Francisco, CA, June 1993.

M. McDermott, O. Masory, B. Mooring, *Laboratory for a First Course in Modeling and Automatic Control*, American Control Conference Proceedings, Atlanta, GA, June 1988.

P. D. Olivier, L. Boyce, *Networked Digital Control and Data Acquisition for an Undergraduate Controls and Robotics Laboratory: A Progress Report*, American Control Conference Proceedings, San Francisco, CA, June 1993.

U. Ozguner, *A Tree Course Control Laboratory Sequence*, American Control Conference Proceedings, Atlanta, GA, June 1988.

W. Schaufelberger, *Teachware for Control*, American Control Conference Proceedings, Atlanta, GA, June 1988.

W. Schaufelberger, *Educating Future Control Engineers*, IFAC, 11th Triennial World Congress, Tallinn, Estonia, USSR, 1990.

G. Soto, *Design and Fabrication of the Fluids Module for a Dynamics and Controls Laboratory*, MIT, June 1995.

M. Tesciuba, *A Modular Design for an Undergraduate System Dynamics and Controls Laboratory*, MIT, June 1995.

Thomson Industries, Inc., *Thompson Industries Inc. Catalogue*, Port Washington, NY, February 1992.

J. Vagners, *Control Systems Laboratory Development*, American Control Conference Proceedings, San Francisco, CA, June 1993.

S. Yurkovich, *The Instructional Control Laboratories in Electrical Engineering at The Ohio State University*, American Control Conference Proceedings, San Francisco, CA, June 1993.

Appendix A Interview Contacts

Mechanical Engineering Department at MIT

Name	Number of Meetings
Professor A. Annaswamy	2
Professor H. Asada	1
Professor S. Crandall	1
Professor D. Gossard	1
Professor D. Hardt	1
Professor N. Hogan	2
Professor J. Lienhard	1
Professor B. Mikic	1
Professor I. Paul	1
Professor D. Rowell	1
Professor W. Seering	2
Professor T. Sheridan	1
Professor A. Slocum	1
Professor J-J. Slotine	2
Professor D. Trumper	2
Professor D. Wilson	1
Professor K. Youcef-Toumi	1

Other Departments at MIT

Professor C. Chryssostomidis (Ocean Engineering)

Outside MIT

Professor W. Durfee (University of Minnesota)

Professor J. Eaton (Stanford University)

Professor J. Henry (University of Tennessee, Chattanooga)

Professor G. King (Purdue University)

Professor Shafai (Northeastern University)

Professor Z. Shiller (University of California, Los Angeles)

Professor J. Vagners (University of Washington)

Professor B. Wilson (Northeastern University)

Appendix B Class Enrollment History

Table 8 lists the enrollment history for the past five years for the controls system course offered in the Mechanical Engineering Department at MIT (Course # 2.14).

Year	Semester	Number of Enrollments
1995	Spring	60 (as of 4/12/95)
1994	Fall	54
	Spring	64
1993	Fall	67
	Spring	62
1992	Fall	78
	Spring	71
1991	Fall	94
	Spring	109
1990	Fall	75
	Spring	110
1989	Fall	81
	Spring	103

Table 8. Class Enrollment History for Course 2.14 (Control System Principles)

Appendix C Proposed Course Schedule

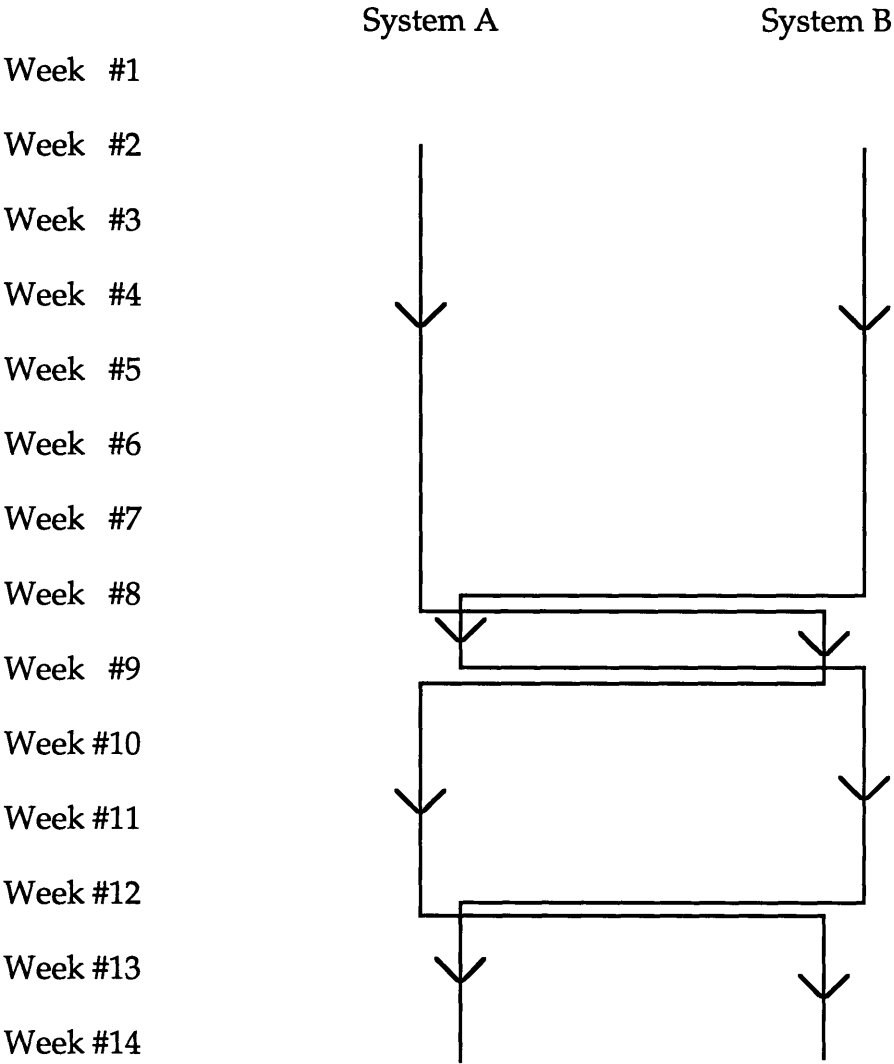
The proposed course schedule includes two parts: (1) the week-by-week control concepts coverage sequence, and (2) the schedule on which physical modules used.

I. Control Concept Coverage Schedule

	<i>Laboratory</i>	<i>Laboratory for other controls courses</i>
Week # 1	X	X
Week # 2	Modeling	X
Week # 3	Modeling	X
Week # 4	1st Order System	X
Week # 5	X	Modeling
Week # 6	2nd Order / Feedback Control	X
Week # 7	2nd Order / Feedback Control	X
Week # 8	X	X
Week # 9	2nd Order / Stability	X
Week #10	Root Locus	X
Week #11	X	Transient
Response		
Week #12	Frequency Response	X
Week #13	Design	X
Week #14	Design	Frequency
Response		

X = No laboratory session this week.

II. Student Interaction with Laboratory Physical Modules



Appendix D Proposed Teaching Assistant Schedule

Under this proposal, the laboratory teaching assistant will save 45 hours per semester and can utilize this amount of time for instructing laboratory sessions for other system dynamics and controls courses.

Current TA Schedule

Total # of hrs. available per term	$20\text{hr/wk} * 13\text{wks}$	260 hrs
Hrs. spent in lab sessions	$9\text{labs} * 2.75\text{hr/lab} * 6\text{sect.}$	(148.5 hrs)
Hrs. spent grading	$9\text{labs} * 5\text{hr/lab}$	(45 hrs)
Hrs. spent preparing the labs	$9\text{labs} * 3\text{hr/lab}$	(27 hrs)
Hrs. spent in lectures	$3\text{hr/wk} * 13\text{wks}$	(39 hrs)
Total # of hrs. spent per term		(259.5 hrs)

Proposed TA Schedule

Total # of hrs. available per term	$20\text{hr/wk} * 13\text{wks}$	260 hrs
------------------------------------	---------------------------------	---------

For Controls System Course

Hrs. spent in lab sessions	$9\text{labs} * 2.15\text{hr/lab} * 6\text{sect.}$	(121.5 hrs)
Hrs. spent grading	$9\text{labs} * 4\text{hr/lab}$	(36 hrs)
Hrs. spent preparing the labs	$9\text{labs} * 2\text{hr/lab}$	(18 hrs)
Hrs. spent in lectures	$3\text{hr/wk} * 13\text{wks}$	(39 hrs)
Total # of hrs. spent per term		(214.5 hrs)

- The TA has $260 - 214.5 = 45.5$ hrs available to instruct laboratory sessions for other system dynamics and controls courses.

For Other System Dynamics and Controls Courses

Hrs. spent in lab sessions	3labs * 2.25hr/lab * 4sect.	(27 hrs)
Hrs. spent grading	3labs * 4hr/lab	(12 hrs)
Hrs. spent preparing the labs	3labs * 2hr/lab	(6 hrs)

Total # of hrs. spent in 2.02 per term (45 hrs)

Total # of hrs. spent per term 214.5 hrs + 45 hrs = (259.5 hrs)

Appendix E Cost Breakdown of Computer Module

The Computer Module's hardware and software costs are summarized in Table 9.

Item	Description	Price
PC 486	Dell 466MX, 210MB Hard Drive, 20MB RAM, 17" monitor.	\$3,078
Multifunction I/O Board	DATEL PC412A	\$695
Encoder Board	US Digital PC7166	\$150
Video Board	Video Blaster SE100	\$310
VGA Board	SpeedStar Pro	\$100
Ethernet Board	3COM 3C509	\$150
MATLAB	MATLAB 4.2c Classroom Kit, Simulink, Control Toolbox, Notebook Suite	\$1,200
Multifunction I/O Board Software Driver	DATEL PC411SRC	\$395
MS Word	MS Word 6.0	\$90
	Total:	\$6,168

Table 9. Cost Breakdown of the Computer Module

Appendix F Dynamics of Linear Slide Table System

This appendix describes the tests performed to obtain the values of time constant, open loop DC gain, and static friction for the linear slide module. Figure 45 shows a schematic of the linear slide table.

Theoretical Background

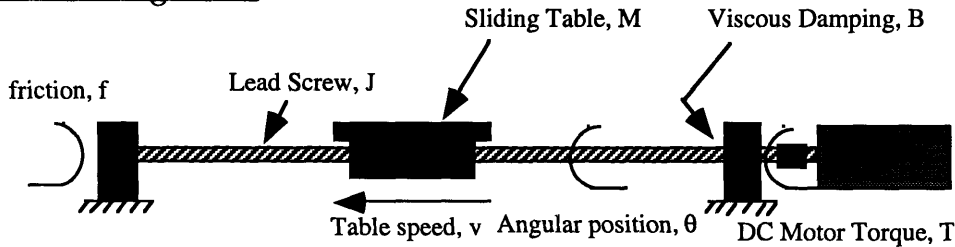


Figure 45. Schematic of the Linear Slide Table

M	Slide table mass	T	Motor torque
J	Ball screw rotational inertia	f	Friction
p	Pitch of ball screw	x	Linear slide table position
B	Viscous damping coefficient	V	Voltage to the Amplifier

The system's transfer function can be written in the form:

$$\frac{X(s)}{V(s)} = \frac{K_{system}}{s(\tau_{plant}s + 1)} \quad \text{eq. (F1)}$$

Rewrite Equation E1 by substituting $x = v/s$, where v is the velocity of the linear table:

$$\frac{X(s)}{V(s)} = \frac{K_{system}}{(\tau_{plant}s + 1)} \quad \text{eq. (F2)}$$

For a step voltage input of magnitude V_{in} at time = 0, i.e. $V(s) = V_{in}/s$, then the velocity response can be described in the S-domain as:

$$\frac{X(s)}{V(s)} = \frac{K_{system}}{s(\tau_{plant}s + 1)} \quad \text{eq. (F3)}$$

or, in time domain as:

$$v(t) = V_{in} * K_{system} (1 - e^{-t/\tau_{plant}}) \quad \text{for } t \geq 0 \quad \text{eq. (F4)}$$

Figure 46 shows a typical velocity response of the linear slide table for a step input. From the response curve, τ_{plant} can be determined. Also, the value of K_{system} from the value of v_{ss} can be determined:

$$K_{system} = \frac{v_{ss}}{V_{in}} \quad \text{eq. (F5)}$$

From the experimental values of K_{system} , and the specs for K_A and K_M , the value of K_{plant} can be determined:

$$K_{plant} = \frac{K_{system}}{K_A * K_M} \quad \text{eq. (F6)}$$

To determine the value of friction, f , for the system, the “stiction test” is used. Static (coulumbic) friction is assumed. This static friction is overcome at the instant when the plant begin motion. So to determine f , a ramp input voltage is applied to the system. From the specification value of K_a and K_M and controlling the input voltage, the amount of torque applied to the plant can be calculated. The plant begins motion as soon as the motor torque exceeds f . Therefore, the friction value, f , can be determined by measuring the voltage at the instant when the plant begins motion.

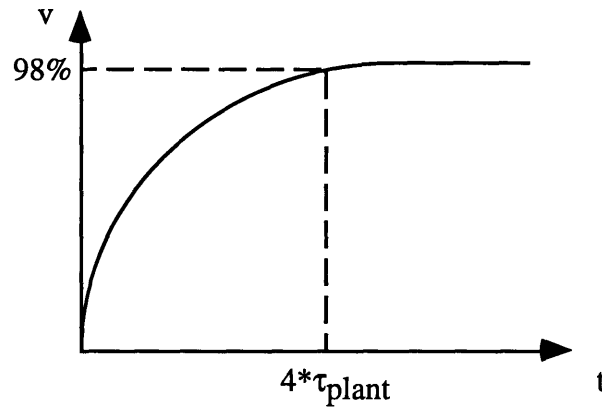


Figure 46. Typical Velocity Response to Step Input

Manufacturer’s Specifications

$$K_M = 14.8 \frac{\text{oz} - \text{in}}{\text{amp}}$$

$$K_A = 4 \frac{\text{amp}}{\text{volt}}$$

Experimental Results

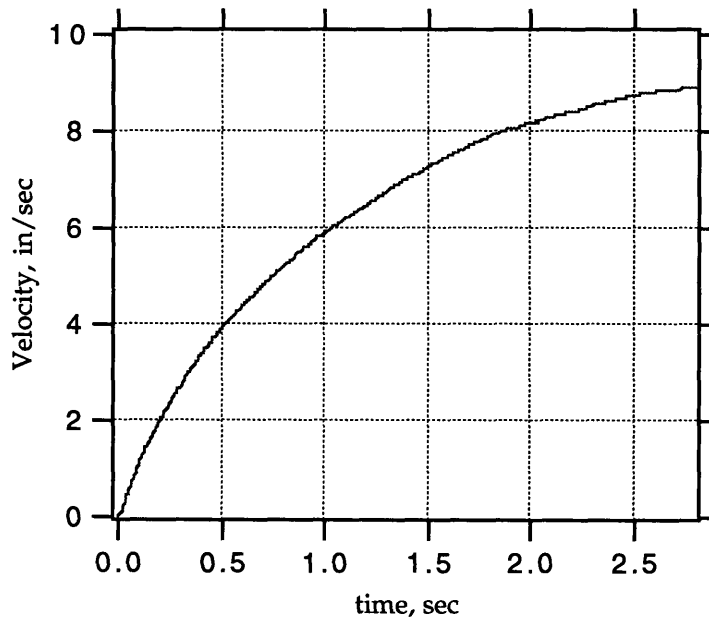


Figure 47. Typical Experimental Open Loop Step Velocity Response

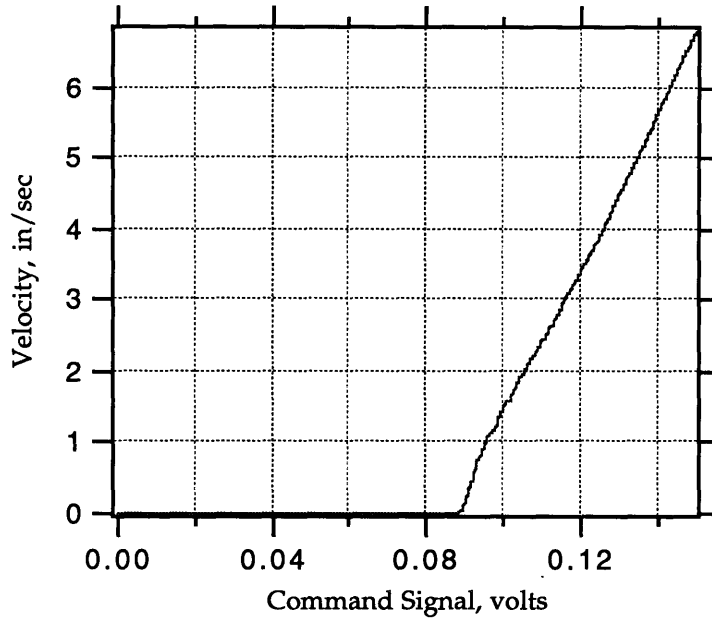


Figure 48. Typical Friction Value Determination Plot (without friction device)

From Figures 47 and 48, the following plant parameters are obtained:

- Time constant (τ_{plant}): 0.50 seconds
- DC gain of plant (K_{plant}): $0.76 \frac{\text{in/s}}{\text{oz-in}}$
- Static Friction: 5.4 oz-in

Other Experimental Results

This section presents the experimental results when the friction device and additional inertial load is applied to the linear slide module.

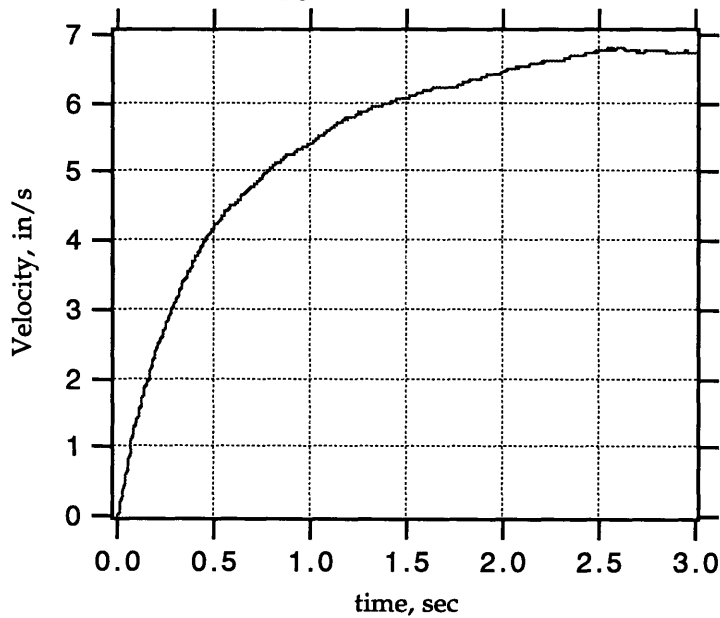


Figure 49. Velocity Response to Step Input (with additional inertial load)

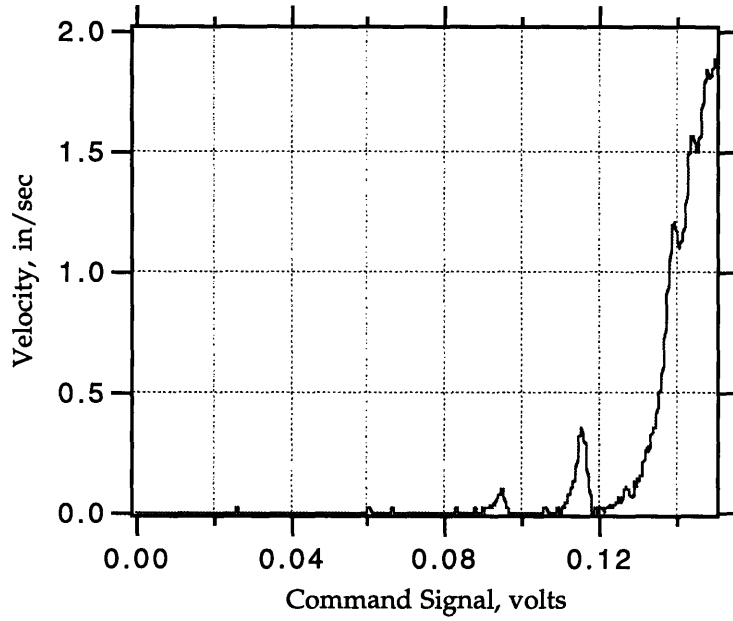


Figure 50. Typical Frictional Value Determination Plot (with friction device)

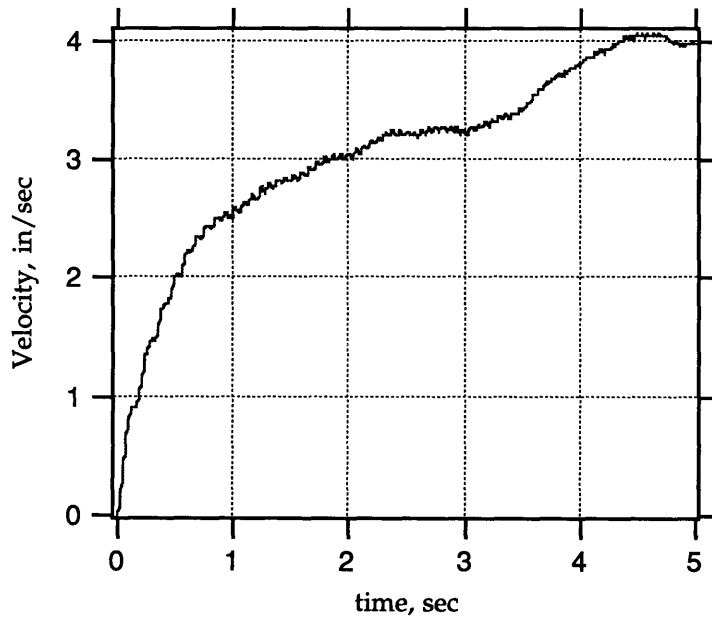


Figure 51. Typical Velocity Response to Step Input (with friction device)

Appendix G A Sample of Laboratory Handout and Report

This appendix includes a sample of laboratory procedures, handout and a completed report for a PID-position-control laboratory exercise.

Linear Slide System

Lab 5 PID Position Control

Introduction

During the last lab exercise, you have implemented a P and PD controllers and found that PD control allows you to vary the damping ratio and natural frequency of the system independently. However, you also observed that P and PD controllers are poor at eliminating steady state errors due to external disturbances. In this laboratory exercise, you will add an integral controller to the system and examine its ability to reduce steady state errors. You will also explore the toll on other performance aspects caused by the integral control, namely, instability.

There is a design project at the end of this lab, where you will design a PID-controller for the linear slide table in conjunction with the vision system to trace an object.

Objective

In this lab, you will:

- (i) implement a PID controller to the linear slide system.
- (ii) observe steady state error reduction.
- (iii) get a first look at instability.

Pre-lab Assignment

Closed loop PID Control

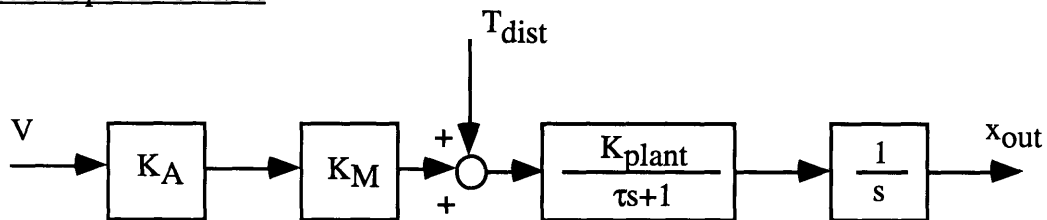


Figure 1: Amplifier/Motor/Plant

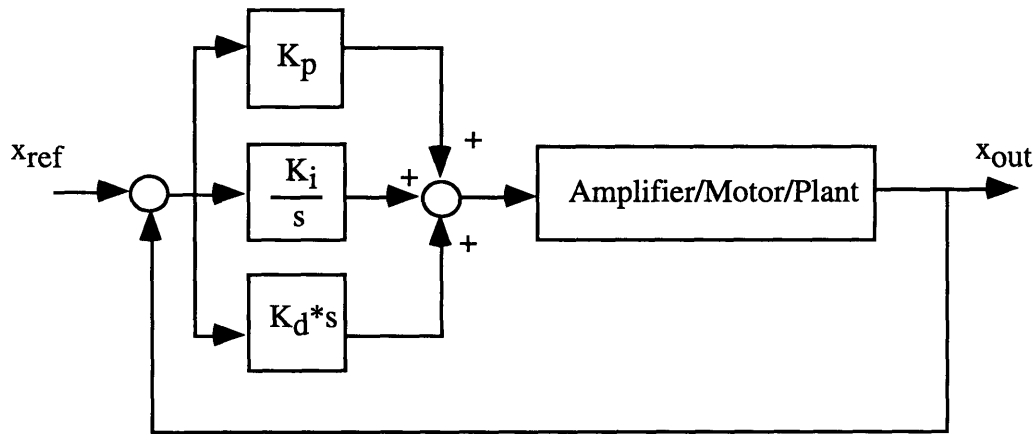


Figure 2: Linear Slide System with PID controller

For this laboratory, K_p and K_d are set to 0.25 and 0.02 respectively.

Question 1

Derive the transfer function for the system with PID controller. What is the order of this system?

Question 2

From Figures 1 and 2, determine the transfer function for x_{out}/T_{dist} . Using Final Value Theorem, determine the steady state value of x_{out} for a unit step T_{dist} .

Question 3

Using the Routh-Hurwitz method, and the numerical value of the system's parameter, find the critical value of K_i that will result in an unstable system.

Simulation 1

For $K_p = 0.25$, $K_d = 0.02$, and $K_i = 0$, simulate the response of the system for $x_{ref} = 3$ in. The controller gains can be set by clicking the **Controller** button; the input can be set by clicking the **Input** button, and the simulation can be started by clicking the **Simulation** button. Plot the response by clicking the **Plot** button and place it in your lab report document.

Simulation 2

For the same K_p , K_d , and x_{ref} , simulate the response of the system with K_i set to 0.1. Plot and save the response to your lab report document.

Repeat the simulation for $K_i = 0.75$, and the theoretical value of critical K_i obtained from Routh-Hurwitz method.

Question 4

Compare the results from Simulations 1 and 2 on the system's performance. Particularly, focus on the system's steady state error, percentage overshoot, settling time, and number of oscillations. Also comment on the effect of increasing value of K_i on the system's response.

Question 5

At what value of K_i did the system become unstable? Did the theoretical value match the simulated value?

Lab Assignment

We will begin lab by implementing a PD controller similar to the previous lab. Then an integral controller will be added to study the effects of integral control. Afterwards, the experimental results will be compared with the theoretically predicted and simulated results.

PD Control

Experiment 1

Implement a PD controlled system by clicking **Run RTC** and entering $K_p = 0.25$, $K_d = 0.02$ and step input size of 3 in. Observe the initial starting position and the final position of the linear table.

Did the table travel 3 inches? Did you observe any overshoot and steady state error? Plot the response by clicking **Plot** and save the results to your lab report document.

PID Control

Experiment 2

Keep the values of K_p , K_d , and input step size unchanged, enter $K_i = 0.1$. Click **Run RTC** to start experiment. Observe the initial starting position and the final position of the linear table.

Did the table travel 3 inches? Did you observe any overshoot and steady state error. Plot and save the results to your lab report document.

Repeat Experiment 2 with $K_i = 0.75$, the theoretical critical gain, and the simulated critical gain value. Use smaller input step size to avoid system saturation.

Question 6

Compare the results from Experiments 1 and 2 on the system's performance. Particularly, focus on the system's steady state error, percentage overshoot, settling time, and number of oscillations. Comment on the effect of increasing value of K_i on the system's response. Also, how well did the simulated results agree with the experimental results?

Experiment 3

Determine experimentally the critical value of K_i by running RTC with step input size of 0.25 in. When the experimental value of K_i is found, save the response of the system.

Question 7

How well does this value of K_i compare with the theoretical and simulated values of critical K_i ?

Observations/ Comments/ Analysis

Question 8

Explain, using the materials you learned in the lecture, the effect of integral control on the system's percentage overshoot, settling time, and number of oscillations.

Question 9

Explain the discrepancies, if any, between the theoretically calculated critical value of K_i , the simulated one, and the experimentally determined one.

Project

The goal of this project is to design a PID-controller for the linear slide table to trace an object, with the camera serving as the sensor for locating the target. Two poles are placed at both ends of the camera vision range, which the linear slide table must not knock over. The performance of the controller must meet the following specifications.

For the target located in the vision range, the linear slide table must:

- have rise time less than 1.5 seconds;

- have steady-state error must be less than 0.25 inches;
- remain in the vision range at all times.

To activate the vision system, click the **Vision System** button in the main menu GUI page.

When you have successfully designed a controller meeting all of the above specifications, demonstrate it to your TA and he/she will check you off. Enjoy!

Linear Slide System

Lab #5 Report PID-Position Control

Section: _____ Date: _____

Name(s): _____

Question #1:

The transfer function of the linear slide table with PID-controller is:

$$\frac{x_{out}}{x_{ref}} = \frac{K_A K_M K_{plant} (K_d s^2 + K_p s + K_i)}{\tau s^3 + (1 + K_A K_M K_{plant} K_d) s^2 + K_A K_M K_{plant} K_p s + K_A K_M K_{plant} K_i}$$

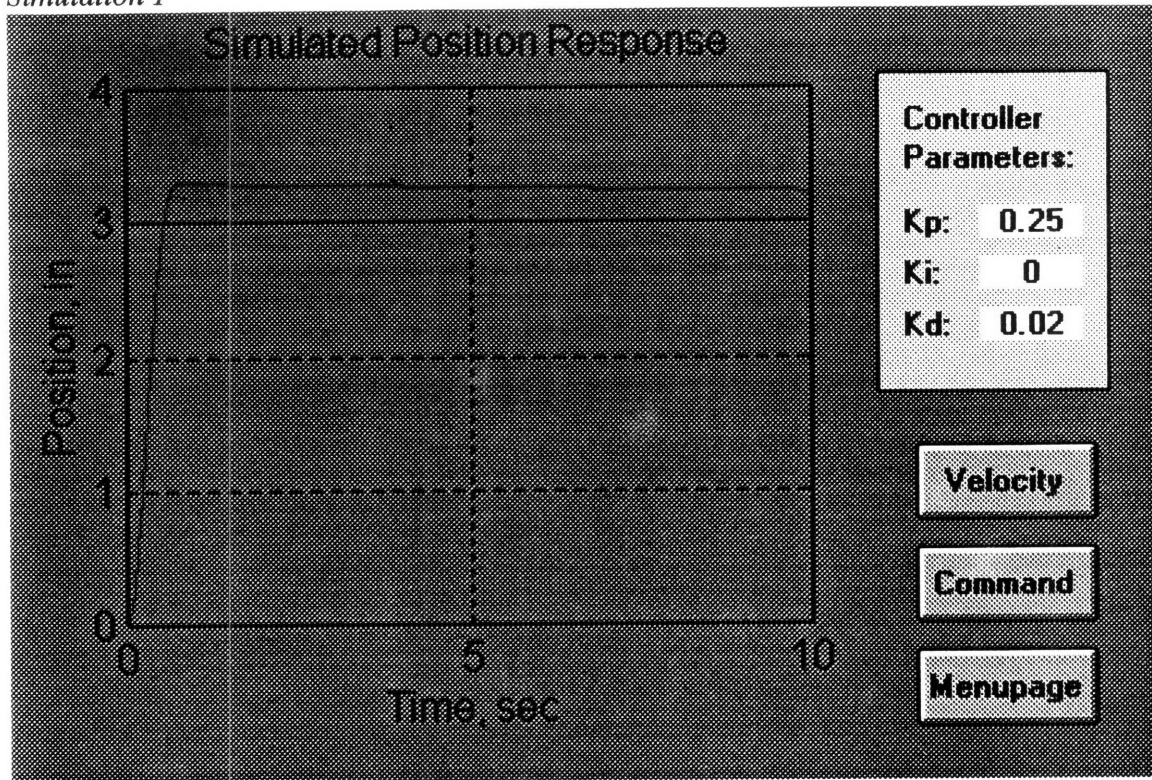
Question 2

$$\frac{x_{out}}{T_{dist}} = \frac{K_{plant}}{(K_A K_M K_d \tau) s^3 + (K_A K_M K_p \tau + K_A K_M K_d + \tau) s^2 + (K_A K_M K_i \tau + K_A K_M K_p + 1) s + K_i}$$

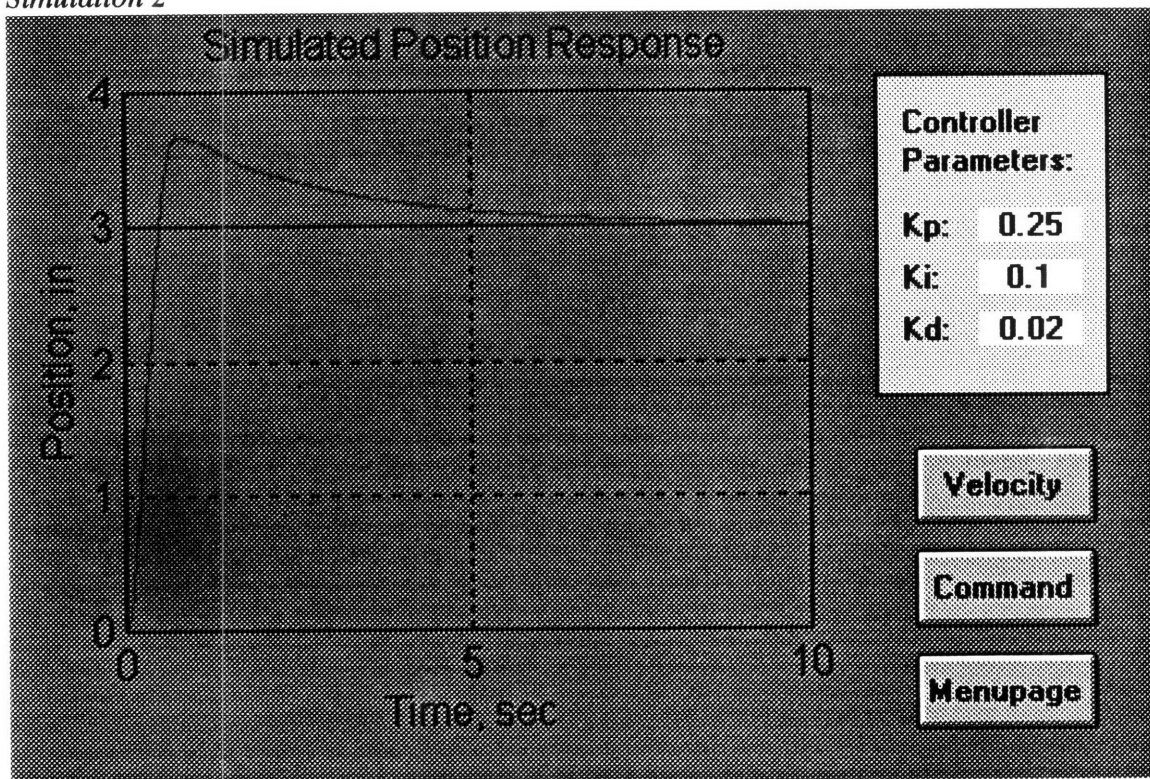
Question 3

$$K_{critical} = 0.948$$

Simulation 1



Simulation 2



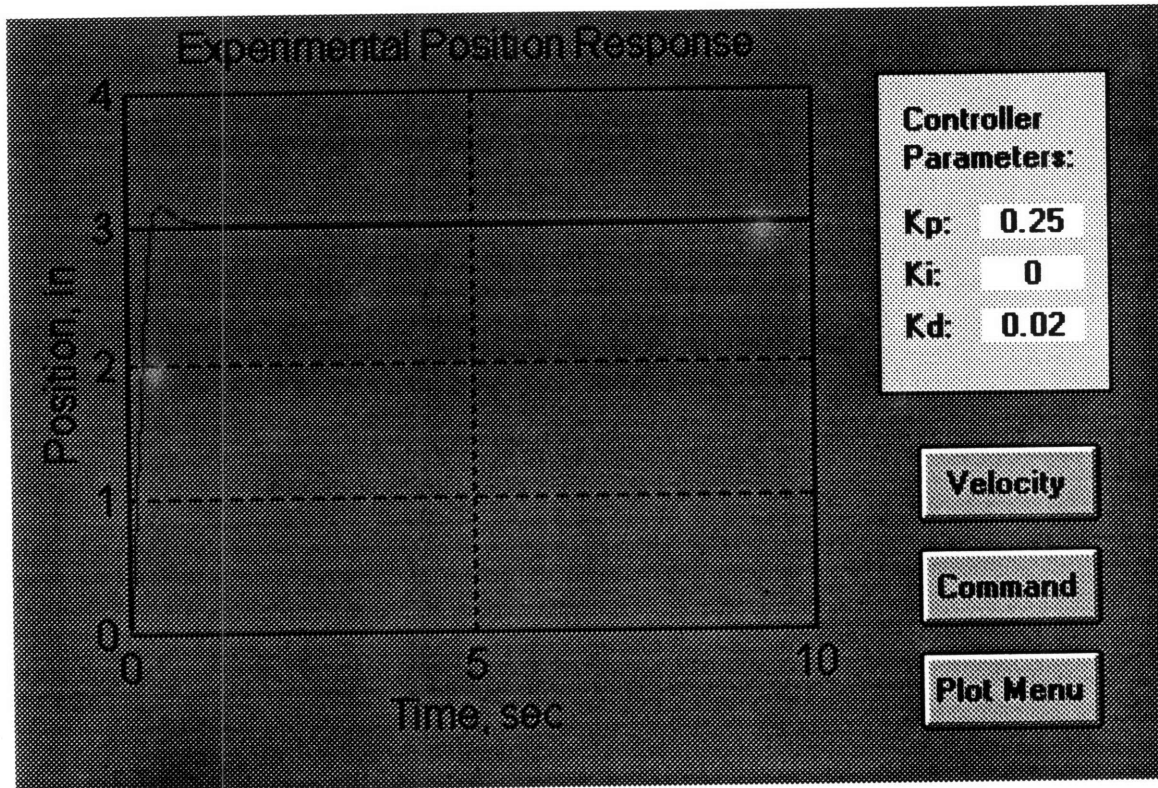
Question 4

When K_i is increased, the percentage overshoot is higher, settling time gets longer, number of oscillations increased, the effect of steady state error reduction is more dramatic (some times it over did it).

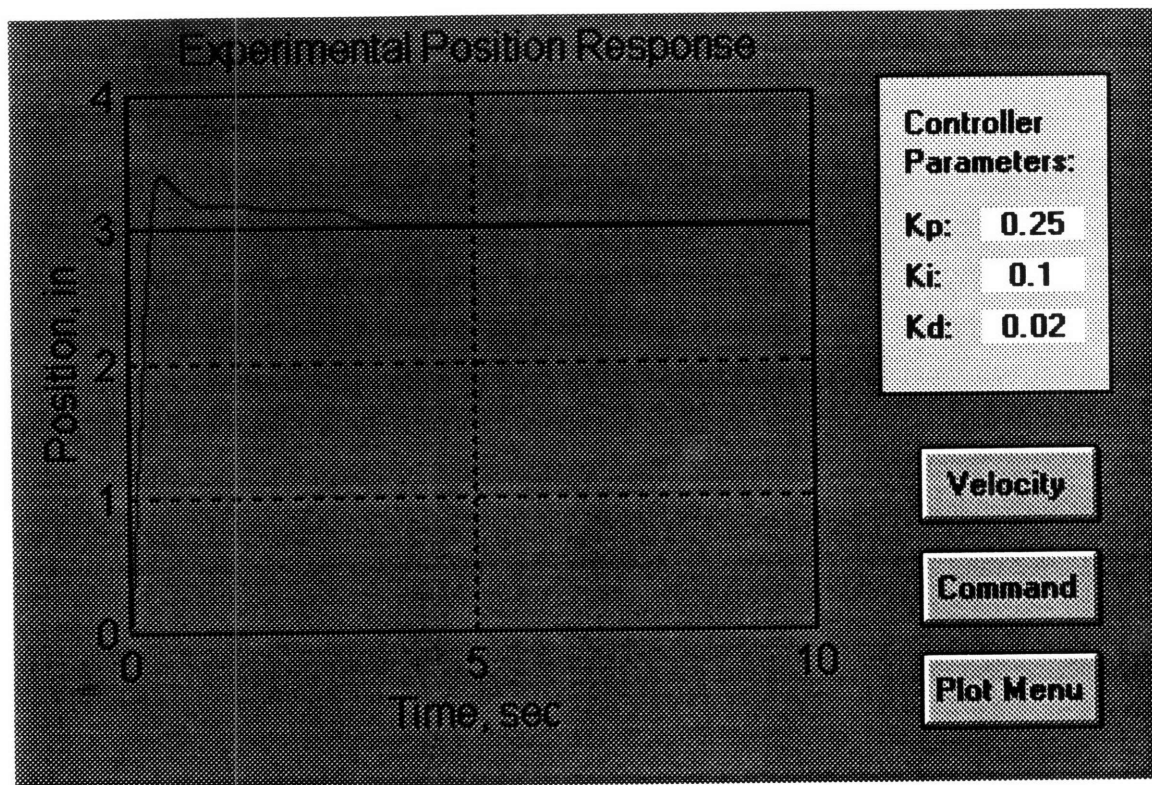
Question 5

The simulated $K_{i\text{critical}}$ is 1.15. This value is higher than predicted by Routh-Hurwitz method. It may be due to the friction effect is taken into account in the simulation.

Experiment 1



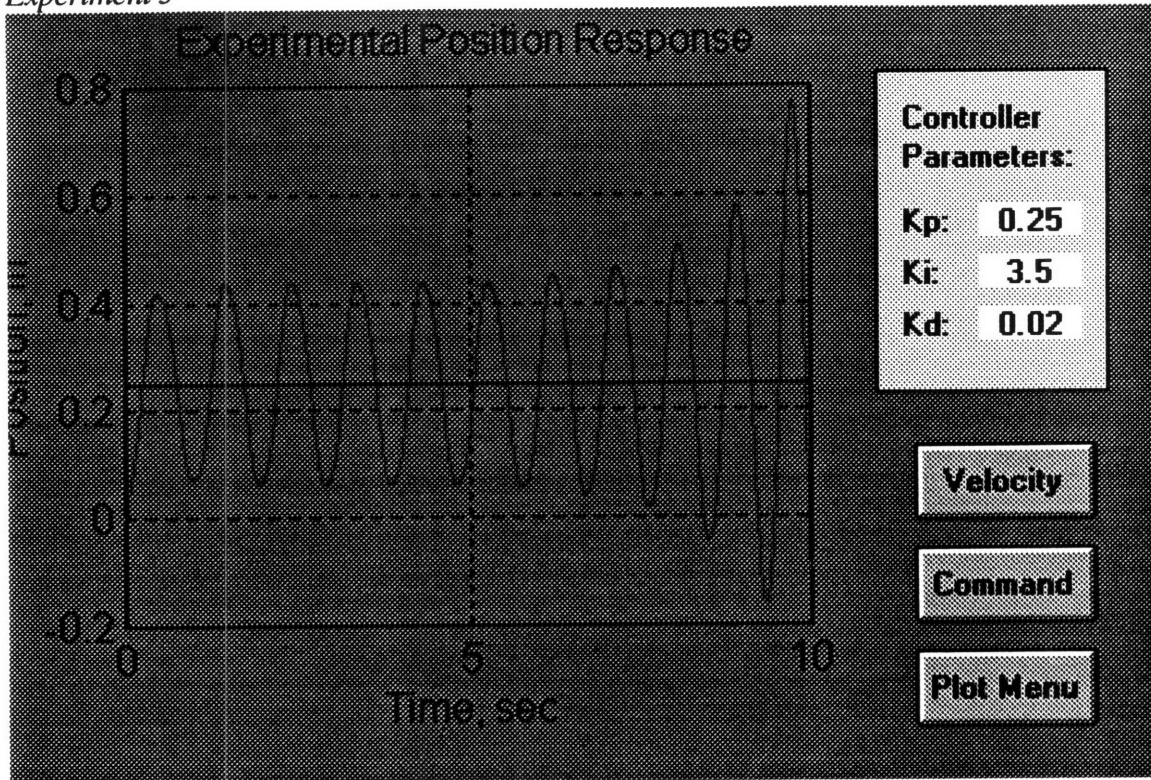
Experiment 2



Question 6

When K_i is increased, the percentage overshoot is higher, settling time gets longer, number of oscillations increased, the effect of steady state error reduction is more dramatic (some times it over did it). The simulation results and the experimental results agree to a large degree, however, the physical system did not go unstable for neither the theoretical critical K_i nor the simulation critical K_i .

Experiment 3



Question 7

The experimental value of $K_{icritical}$ is 3.5, which is 3.8 times bigger than the theoretically predicted value and 3.1 times bigger than the simulated value.

Question 8

From the root locus plot for a third order system, as the gain increases, two of the roots moves away from the real axis (thus more oscillation and more overshoot) and towards right (thus longer settling time).

Question 9

Mainly due to the friction present in the system, which is difficult to model correctly.

- 5) Did you find the hardware "comfortable" in terms of safety and robustness?
Electrical Module:

Fluid Module / Electromechanical Module:

- 6) What did you like best about:
the Electrical Module?:

the Fluid Module / Electromechanical Module?:

- 7) What needs to be improved in the prototype?
on the Electrical Module?:

on the Fluid Module / Electromechanical Module?:

- 8) Would you recommend this new system dynamics and control laboratory to your best friend? Why?

Any additional comments, thoughts or recommendations are welcome ...

Thanks for your input!