Software and Laboratory Experiments Using Computers in Control Education

Mohamed Mansour and Walter Schaufelberger

ABSTRACT: Personal computers and workstations are being introduced into university education at an increasing rate. Classical and modern methods as well as sequencing control form the basis of control education where students are used for classroom presentations, exercises, laboratory experiments, and in projects. Commercially available simulation and computer-aided control system design packages may be used in teaching together with real-time software and with special programs developed for specific purposes. These widespread possibilities are discussed and illustrated by using many examples of learning and teaching control systems with computers.

Introduction

Workstations and personal computers form a new medium for teaching. It differs from old media in that two-way communication is possible. The use of computers in the laboratory has a long tradition, whereas the use in exercises is fairly recent.

To investigate the potential of the possibilities offered by modern workstations, a pilot project for computer-based education in control was started in spring 1985 at the Swiss Federal Institute of Technology (ETH), Zurich. Most of the interactive software described in this paper has been developed as part of this project.

We have been using computer-controlled laboratory experiments in teaching since the early 70s. Program packages also have been in use for about 10 years. Interactive, graphically oriented screen experiments were introduced only about three years ago. In our view, this combination now provides many possibilities. With little investment of time, students may use our training programs. These are so easy to operate that little time is lost in handling, and students really study the control problem. A "feeling" can be obtained for nonlinear systems or systems with many parameters, and many "what if" questions can be answered easily. Program packages need much more time for introduction, but students soon realize that they can use the knowledge gained in their projects. The laboratory should convey mainly the idea that the theories developed in class can be applied to real problems.

Our experience shows that a reasonable balance of teaching methods allows many students to master a field often known for its theoretical difficulties and requirements on abstraction.

An attempt is made to show different possibilities that computers offer in control education mainly for short assignments (exercises, laboratory experiments, etc.). Environments and basic ideas for the development of teachware also are described briefly.

Control Education

This brief overview should provide some background information on the teaching of control in the Electrical Engineering Department at ETH so that the software and laboratory experiments to be discussed later can be put into a context.

Teaching control to the electrical engineering student at ETH begins in the fifth semester (third year) with a course on linear control systems, where classical and state-space methods of representation, analysis, and synthesis are discussed. This course is followed by a course in the sixth semester on digital control systems, where the theory of computer control together with implementation problems are dealt with. During the fifth and sixth semesters, students can choose some laboratory experiments that deal mainly with computer-controlled physical models, some of which are described in this paper. In the seventh and eighth semesters, four electives are offered dealing with the theory of nonlinear control systems, optimal control, and practical control engineering. Computer-based exercises are done in all these courses.

A student in electrical engineering majoring in control can do two research projects in control. The first project is carried out in the seventh or eighth seminar, half time; the second is a diploma project in the ninth or tenth semester, full time. Some of the computer control experiments were developed by the students during these semester and diploma projects. A student working toward a Ph.D. degree in control should take a certain number of credit hours in the postgraduate control courses to reach an adequate level before being admitted to the doctorate. These courses deal with advanced control and system-theoretic problems, such as stochastic control, adaptive control, robust control, and advanced system theory. A postgraduate project is also required at this level.

Special Training Programs

Special training programs are used to gain insight and experience into very specific situations. A set of such programs has been in use at ETH for more than two years [1]-[8]. As an example, a screen used for two-term controller tuning is shown in Fig. 1. The students are asked to obtain controller settings from heuristic rules such as Ziegler-Nichols or Chien-Hrones-Reswick by carrying out the corresponding experiments. The simulation environment is used mainly to verify these designs and to gain insight into the correlation between the pole locations displayed in the lower right-hand corner and the transient behavior. A total of fifty Macintosh computers is available in three classrooms for the exercises where the students are carrying out the experiments individually. The experiments are used to replace the paper-and-pencil work. Extended versions of the programs are also used for classroom demonstrations.

Our teaching language is German, and the text in the programs is therefore German, too. We have changed the text in the outputs for better readability of the paper.

Similar programs are available to study the antiwindup problem for two-term controllers with control value saturation and for state-feedback control with an observer. In the latter case, the design may be done with the program package MATLAB and the investigation by simulation.

The idea of simulation also may be used for more advanced topics, such as adaptive...
control and identification. In another experiment (Fig. 2), a recursive mean-square identification algorithm with a forgetting factor is used to identify and then control a first-order plant. Input and output signals as well as the real and identified parameters are displayed. The behavior of the system may be observed during transient response, demonstrating the sensitivity with respect to changing input signals, etc. Signals stored during operation may be replayed later.

Tools
A tool differs from a special training program in that it may be used for professional work later. MATLAB and its descendants are the favorite computer-aided control system design systems in use in our education [4], [9]. By including pole placement, Riccati-solving routines, etc., a control system design environment easily can be made available to every student, especially if it is based on a public-domain version of MATLAB. We also use professional versions with the students, but only in small numbers. The use of tools requires considerably more time for an introduction than the use of special-purpose programs.

ModelWorks [10], [11] is a window-based general software simulation environment that was developed in our pilot project. Plant and observer outputs of a state-variable feedback system are displayed in the graph window in Fig. 3. The differential equations are defined in a Modula-2 module and compiled and linked to the system (a matter of 30-50 sec with the new one-pass compiler of N. Wirth). The initial conditions, model parameters, and the display organization may be changed interactively. A modified version allows entering differential equations of up to sixth order directly on the screen. The package Stella is also used for simulations.

A simple expert system shell has been developed in cooperation with the Asea Brown Boveri Research Center, based on their Modula-2 PROLOG. The shell may be operated together with simulation programs, as shown in Fig. 4. The result of the consultation displayed is to use a three-term controller and to tune it according to Chien-Hrones-Reswick. A mouse click will switch the user to the experiment displayed in the background, where simulations can be performed. Switching back and forth between an expert system shell and a simulation is seen as a useful vehicle for introducing rule-based controller design.

Sequential control education often is neglected if compared to feedback control. A course on sequential control has been taught at ETH for almost 10 years [8]. Decision tables and Petri nets have emerged as two techniques that may be used in the design of sequencing control systems. The Petri net models the dynamic behavior of the system, whereas the decisions that have to be made in certain states are investigated by decision-table analysis. This provides one of the few possibilities to verify a design by a theoretical investigation (completeness, contradictions, etc.). A PROLOG program for decision-table analysis is used for exercises.

Petri nets are a useful tool for the simulation of relatively simple event-driven systems. A simulation program and a program to compute the reachability tree are used in the exercises. Both are programmed in PROLOG. The software described so far is freely available to every student. Commercial products such as ACSL, SIMNON, MATLAB with control and identification toolbox, MACSYMA, SLAM, etc., are often used in the students’ projects. As a result of licensing conditions, these packages are available only on university machines.

Levels of Software
The availability of good and reasonably priced software will be crucial for the future of computer-assisted instruction in control. From our teaching, a somewhat more detailed distinction than the one used earlier is as follows.
The exercises that have been shown so far are not used to replace laboratory sessions, but paper-and-pencil work. We use many processes in laboratory assignments—from simple drives and elevators to advanced electromechanical structures with several degrees of freedom.

First, a brief overview of the many laboratory assignments is given [12]-[14]. Three experiments are then described in more detail. We will discuss only undergraduate assignments here. Laboratory experiments are also important for our graduate education.

**Control Laboratory**

**Traffic Lights** A model of a street intersection with five lanes in the city of Zurich is controlled by a computer. In automated operation, the system has six different states. Students write a program for a Siemens PC in Modula-2 for fully automated operation of the corner and carry out experiments on the installation.

**High-Bay Warehouse** Wooden blocks can be stored and retrieved from a fully automated model warehouse with 360 storage places. Experiments with the crane-position control subsystem (preprogrammed in PASCAL) are carried out by students, using different control algorithms.

**Coin-Exchanging Machine** A fully automated coin-exchanging machine using small balls as coins is available. Students write real-time PASCAL programs for simple changing strategies and operate the installation. Several balls may reside concurrently at different stages of the plant.

**Heating and Ventilation Control Systems** A model of a room is available in this experiment. An industrial controller is used for temperature and airflow control.

**Three-Basin Level Control System** On-off and proportional-integral controllers are used to control two water levels in this system. Students verify the operation of all parts and the model obtained theoretically. They then operate the system with a preprogrammed solution.

**"Helicopter" Model** A two-degree-of-freedom model of a helicopter with two drives is controlled by analog or digital control.

*Level 1: Special Training Programs*

These are programs used in a specific situation during the education. They serve no other purpose than to aid the student in mastering a certain subject, e.g., tuning of a controller.

*Level 2: Mini Tools*

Mini tools can be used in different situations and in several courses. They allow handling simple tasks such as drawing the graph of a function, inverting matrices of restricted order, solving differential equations with limited order, etc.

*Level 3: Tools*

Tools may be used for professional work, without necessarily providing a fully developed environment as the professional tools do.

*Level 4: Professional Tools*

These are the fully developed systems usually marketed by software houses. It is hoped that agreement may be reached that software up to level 3 must be made available for all students in control engineering at minimal cost. All level 1-3 programs and packages discussed in this paper are available to all students of ETH at no cost. Our programs and documentations are available to other universities, unfortunately, most are in German, only a few examples have been translated into English.
Three-Mass-Spring System Output feedback and adaptive controllers can be designed and implemented on this sixth-order system.

DC Servo State-variable feedback control can be implemented on a fourth-order servo by using an observer.

Inverted Pendulum An inverted pendulum is controlled by various control algorithms. Swing-up strategies have been developed for a single-stick pendulum, and stable controllers have been realized for all equilibrium states of a two-stick pendulum.

Model Railway A model railway is used for projects in real-time programming. Program packages written in Modula-2 as well as in the real-time language Portal or a programmable logic controller may be used to control and supervise several trains simultaneously [15].

Speed and Tension Control in a Tape Transporting Machine Design and implementation of a controller consisting of two proportional-integral controllers and a static decoupling network is another example of a laboratory assignment at this stage.

Electric Power Network Model This realtime process is used to monitor effects in power systems control and to investigate specific problems. The central part of the model consists of three machine groups, each composed of an SM (synchronous machine) generator (5 kVA) and a DC motor as a turbine, and the infinite bus, a three-phase transformer connected to the local power supply. The special construction of the synchronous machines makes them behave like 400-kVA generators; the electric time constants are all in the same range as the ones of common generators of this size. This makes it possible to use a 1:1 time scale in the experiments. For an experiment, these machines can be interconnected by 50-km transmission lines, hardware simulated with modules. External three-phase connections make it possible to use any loads suitable for the experiment. For switching operations, eight switchyards displayed on the front panel are used; the front panel is also used to show the actual system configuration. Control and instrumentation are centered on a PDP 11/60, which has access to any of the following quantities and system states: actual connections in the system, access via control of the circuit breakers, and recognition of manual switching; and frequency of the machines via Gray code disk interfaces.

Manufacturing Cell This is a test bed for assembly research based on an IBM 7575 manufacturing system. It consists of a SCARA manipulator, a servo power module, an industrial version of an IBM AT with dedicated axis control cards for control of the manipulator, and an IBM AT for program development. Programming is done in AML/2, a general-purpose programming language with robot-specific functions. Features of AML/2 are an interactive interpreter with debugging aids, functional transparency, object-oriented programming, exception handling, and treatment of program statements as data. Peripheral devices of the manipulator include an ADEC servo gripper, two fiber-optic infrared reflectance sensors in the gripper fingertips, and a vision system based on binary images. The vision system determines object identity and location, classic global features such as area, form factor, extensions in the direction of main axes, etc., as well as the presence and location of holes within objects.

Three of these experiments will now be described in more detail.

Instructional Robot System In addition to an industrial robot and a mobile robot, which are used in student projects for automatic assembly and factory automation, an instructional robot system [16] (Fig. 5) is used as a normal one-afternoon experiment that can be chosen by any third-year student in electrical engineering. This experiment, like many others, was developed during a student project. Its main goal is to introduce the students to main areas of robot technology. A commercially available instructional Rhino XR-3 robot is used for manipulating aluminum or brass blocks. It is mounted on a platform with eight switches and light-emitting diodes (LEDs). They can be used to influence the control program or to show specific program states, respectively. Four position switches are used to detect the presence of a block at a specific location; one switch is mounted in a spring balance, which allows discrimination between aluminum and brass blocks. The other switches are used for manual input. The robot can be programmed either by a teach pendant or by textual programming on a per-
sonal computer. Textual programming is done using an AML/X (IBM) interpreter. Cartesian coordinates of a specific gripper position are input from which the robot joint angles are calculated. In order to simplify the calculation of the inverse kinematics, the robot's degree of freedom is restricted to four, i.e., the gripper is always kept in a vertical position.

The following experiments are done by the student:

1. Study homogeneous transforms and their application to robot kinematics (Denavit-Hartenberg), apply them to the Rhino robot, and verify the theoretical results, using the real robot;
2. Program the robot, using the teach pendant, to grasp blocks that are detected on one of the position switches, and dump them somewhere else;
3. Program the robot, using textual programming, to weigh the blocks and to sort them accordingly.

**Helicopter Model**

The control of a two-degree-of-freedom "helicopter" model [3], [12] (Fig. 6) is another typical one-afternoon experiment for the students. As with many other experiments, the design and implementation was first done in undergraduate and graduate projects. The electromechanical structure can be modeled reasonably well by a linear, sixth-order, unstable, nonminimum-phase system with two inputs (drives) and two outputs (angles). This experiment was first implemented on an HP-1000 computer in PASCAL and is now transported to a PC in Modula-2. In short assignments, students are given a model and asked to carry out the following steps: compute the discrete-time description; solve the Riccati equation for state-variable feedback; simulate state control; introduce an observer; design and simulate with observer; replace the simulated process with the real one; and carry out experiments. Typical results are shown in Fig. 6, where the process is controlled by an HP computer.

**Control of a Double Pendulum**

This is a laboratory setup that serves as a demonstration object and as a controlled process in students' projects in the field of modern control theory [17]. Once again, the control algorithms were developed by undergraduate students in their semester and diploma projects.

The process consists of a cart with a double pendulum. The cart is guided by a heavy rail that supports two pulleys at its ends. Over these, a sheet-metal belt is drawn and fixed under some tension to the cart. One of the pulleys is driven by a DC motor, the other carries an angular transducer in order to measure the cart's position. The first pendulum is hinged to the cart with a bearing carrying another angular encoder. The second pendulum is hinged to the first one without any measuring device, so that in contrast to other laboratory double-pendulum systems, the angle between the two pendulums is not measured. A subordinate controller forces the cart to behave just like two chained integrators. Main control tasks are implemented on an HP-1000/F computer under the operating system RTE-6. Figure 7 shows the physical arrangement, and the block diagram of the linearized system is depicted in Fig. 8.

The problems dealt with in the control of this process are as follows:

1. Stabilization of the system in its four equilibrium positions by observer-based state-feedback control using the linearized model;
2. Testing of robust control algorithms;
3. Study of open-loop control strategies to transfer a nonlinear system from one equilibrium point to another.

**Authoring Systems**

To develop software as shown in this paper rapidly and efficiently is a problem of software engineering. Authoring systems that allow rapid development by nonspecialists are certainly important for education. One of the outcomes of our project is an authoring system called the DialogMachine [11], which has considerably shortened our development times. (It took 3 man-days to develop the program shown in Fig. 1.) It has proven very useful to base our development environment on the programming language Modula-2 (developed by N. Wirth), with its excellent facilities for data abstraction. Therefore, it is easily possible to write Macintosh applications without ever consulting the Inside Macintosh manuals. Our environment consists of about a dozen modules, containing procedures for menu and window handling, all types of input and output, graphic and numeric routines, etc. In addition, tools are available for the design of programs consisting of many screens in a rapid and efficient way (see [18] for details). We have experience with several other authoring systems and are convinced that environments such as Apple's HyperCard will play an important role in the future for computer-assisted education. (All our programs can be launched easily from within HyperCard.) The effort for actually programming applications is now comparable to the effort required for writing the documents that must accompany such a program. We are also transporting our development system to the IBM world.

**Man-Machine Interfaces**

We are using a consistent users' model of the system to design our screen experiments.
Experience shows that, in this way, we are able to develop programs that are extremely easy to handle. When a student does the first exercise, it takes about 10 min to become acquainted with the programs, which are all organized in exactly the same way. See [19]–[21] for general information. Many of our programs are based on the so-called textbook model. This means that the programs can be used in a way that is very similar to the use of a book. General commands are available to leaf through the book page by page, to close it, and to jump to the beginning of the chapters. The individual presentation unit comparable to a page in a book is one screen. The commands available on the active screen are displayed either in the menu bar or on the screen. We believe that it is important that programs used in teaching are well presented and easy to handle if we want this new medium to be accepted by noncomputer-enthusiasts.

**Conclusion**

From our experience, we are convinced that the appropriate use of computers greatly enhances control education.

**References**


Mohamed Mansour was born in Damietta, Egypt, in 1928. He received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Alexandria, Egypt, in 1951 and 1953, respectively, and the Dr. Sc. Techn. degree in electrical engineering from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 1965, when he was awarded the silver medal of ETH. He has been Professor and Head of the Department of Automatic Control at ETH Zurich since 1968; he was Dean of Electrical Engineering from 1976 to 1978. He has been a Visiting Professor at IBM Research Laboratories, San Jose, California, 1974; at the University of Florida, Gainesville, 1975; at the University of Illinois, Urbana, 1981; and at the University of California, Berkeley, 1983. He was President of the Swiss Federation of Automatic Control from 1979 to 1985. He is a member of the IFAC Council and Treasurer of IFAC. He has honorary degrees from Gansu University and from Guangxi University, China. His fields of interest are control systems, especially stability theory and digital control, stability of power systems, and digital filters.

Walter Schaufelberger is Professor of Control Engineering in the Electrical Engineering Department of the Swiss Federal Institute of Technology (ETH). He received the Diploma degree in 1964 and the Ph.D. degree in 1969, both from ETH, where he became Assistant Professor in 1972. His research interests are in nonlinear and intelligent control systems. Since 1986, he has been responsible for a schoolwide project for the integration of workstations into engineering and scientific education.