ABSTRACT: This paper describes two Stanford University courses of one quarter (10 weeks) each, covering the material of "Digital Control of Dynamic Systems" by the authors. The courses have neither homework nor exams but are based on laboratories using IBM PC computers with locally designed and constructed analog plants. Rather formal reports are also required covering the seven labs in the two courses. The first course (4-unit credit) has labs on digital filters, control design by transform methods, control design by state-space methods, and quantization effects. The second course (3 units) has labs on identification, Kalman filters, and optimal design. The courses are taught several times a year by faculty from the Electrical Engineering, Mechanical Engineering, and Aeronautics and Astronautics Departments, and satisfy degree requirements in each of these departments; the students are mainly first-year graduate students from these departments with a few undergraduates.

Introduction

Sampled-data control and the related control by digital computer have been taught at Stanford University since 1957. For at least the past 20 years, the courses for digital control have been laboratory-based; our first computer was the HP2114 with paper-tape program and data storage. This system was followed by an HP211MX, and now we use the IBM PC. The plants to be controlled were simulated first on EAI analog computers, including the TR-20, TR-48, and Miniac. Now we use a locally designed and constructed analog computer whose "program board" is a printed circuit board hand-wired for specific experiments and plugged into the main box via an edge connector. The models used are second and fourth order but could be almost any order desired.

For such courses, one needs both real-time software to control the clock, and analog-to-digital (A/D) and digital-to-analog (D/A) operations and off-line software to aid the control system design. Our philosophy from the start has been to concentrate on the control aspects of the problem and avoid a significant requirement for computer programming. We selected the language BASIC as being so simple that the essential ideas could be learned by the students after no more than a half hour of instruction followed by practice with prepared programs. We have provided subroutines accessible via the CALL instruction to set the clock and to do input and output operations. In recent years, because many students have learned PASCAL in computer science courses, input/output (I/O) routines have been provided in that language and many students do their real-time programming in PASCAL. For control system design, many programs written by the several instructors, the teaching assistants, and some students have been collected and made available to the students. Over the years, programs have been in APL, FORTRAN, PASCAL, BASIC, and C. One of the first of these was a program for plotting the root locus written largely by O. H. Schuck while he was a visitor at Stanford. Today, we provide MATLAB (The MathWorks, Inc., 21 Eliot St., South Natick, MA 01760) on the IBM PCs in the lab and also a cluster of these machines provided by the School of Engineering for all engineering students. The package includes a TRANSLATE program to convert data files generated by BASIC or PASCAL to MATLAB format. With this package, we have an integrated design tool covering both transform and state-space methods that includes excellent plotting facilities for both design and analysis of experimental data.

At the start, we decided that well-written reports would constitute the students' record of work done. The reports are required to be bound with a table of contents and sections on supporting analysis, experimental results, discussion, and conclusions. The students work in groups of two and may sign up for experimental times in 2-hr blocks any time. The laboratory access is by combination lock and is used 24 hr a day, seven days a week. (The three days before the state-space design lab is due, there are very few unused hours in the lab!) Teaching assistants are available for substantial time in the laboratory, and the instructor holds office hours each week. Each person writes independently the discussion and conclusion sections. To assist the report writing, the program PC-WRITE is on all the lab machines, but many students have access to other computers and editors, both IBM and Macintosh, with a few using LaTeX and laser printers.

The philosophy of the course is to keep the intellectual content as high as possible while keeping the drudgery as low as possible. Key factors that have contributed to this end are the use of BASIC and MATLAB software and the simplified analog electronics that have been constructed at Stanford.

The students are predominantly first-year graduate students and have completed a first course in control system design. Many have had some matrix theory, but that is not required. Furthermore, no prior experience with PCs or programming is required, although most have it.

In the remainder of the paper, the laboratory equipment is described in more detail, the topics and typical results from the labs are described, and a brief summary is presented.

Stanford Control Laboratory

A photograph of one of the workstations is shown in Fig. 1. The basic computer is the IBM PC, which is shown in the center of Fig. 1. It has 512K bytes of memory, a math coprocessor, the Enhanced Graphics Adapter and color monitor, and a hard disk of 10 or 20MB running DOS-3.1. In one slot is a Data Translation 2801A I/O board with a 12-bit A/D converter, a 16-channel multiplexer, and two 12-bit, ±10-V D/A converters. The four input and two output ports are brought to a termination box fitted with BNC connectors, shown just to the right of the CPU in Fig. 1 and in more detail in Fig. 2. A dot matrix printer is on the instrument shelf above the monitor. Analog plants are simulated on a locally designed and built analog computer, the Stanford Universal Plant (SUP). The SUP is a 10×12×4-in. box containing power supplies, four precision pots, relays to switch from operate to reset, six output sensor lines, two input actuator lines, and a voltmeter with a selector switch that permits meter reading of any pot or output line. These circuits are wired to a 44-pin edge connector in the box. Figure 1 shows the SUP to the right of the computer and...
termination box, whereas Fig. 2 shows more details. Op-amp circuits for each plant (there are five plants for the seven labs) are wired on printed circuit boards labeled "LAB" etc., as appropriate. The boards have on-board trim pots that permit parameter changes from quarter to quarter as desired. The teaching assistant has a collection of boards and changes them as needed as the course progresses. Our most complex board, a "LAB 0" is assigned, which contains an interconnection box and the Stanford Universal Plant.

![Fig. 1. Overview of the digital control lab workstation.](image)

![Fig. 2. Close-up of the interconnection box and the Stanford Universal Plant.](image)

were) for use in observing the response of a controller without wasting paper, a low-frequency signal generator for generating sine waves for Lab I, and a digital multimeter for debugging connections, etc. Six other PCs, including two model ATs, are in the lab for design, program debug, and report editing.

We found that the sampling frequency we obtained for feedback control (sample, output, sample, output, . . . ) with the software provided with the DT-2801A was limited to be on the order of 30 Hz. Therefore, the students, Marc Ullman, wrote drivers that interfaced to BASIC and TURBO-PASCAL that provided sample rates up to about 300 Hz (on a 5-MHz XT). This has been more than adequate for the lab exercises. There are three BASIC routines: SETCLOCK initializes the real-time routines and sets the sample rate, READA2D obtains a sample from the A/D converter, and WRITEDA2 outputs a control value to the D/A converter.

The sampling rate that is possible depends on the complexity of the computation of the control algorithm. Typical values used in the lab are 5-100 Hz. On the first day of class, the students are given a tour of the lab, after which they are asked to design by pole placement a fourth-order plant wired on it. Except for the experiment on identification, the students are given the wiring diagrams. In addition to the PC and the SUP, each of the four real-time stations has an oscilloscope (only one is digital; we wish they all were) for use in observing the response of a controller without wasting paper, a low-frequency signal generator for generating sine waves for Lab I, and a digital multimeter for debugging connections, etc. Six other PCs, including two model ATs, are in the lab for design, program debug, and report editing.

We found that the sampling frequency we obtained for feedback control (sample, output, sample, output, . . . ) with the software provided with the DT-2801A was limited to be on the order of 30 Hz. Therefore, the students, Marc Ullman, wrote drivers that interfaced to BASIC and TURBO-PASCAL that provided sample rates up to about 300 Hz (on a 5-MHz XT). This has been more than adequate for the lab exercises. There are three BASIC routines: SETCLOCK initializes the real-time routines and sets the sample rate, READA2D obtains a sample from the A/D converter, and WRITEDA2 outputs a control value to the D/A converter.

The sampling rate that is possible depends on the complexity of the computation of the control algorithm. Typical values used in the lab are 5-100 Hz. On the first day of class, the students are given a tour of the lab and a "LAB 0" is assigned, which contains an example program in real-time data collection from an oscillator set up on the SUP and a tutorial exercise in MATLAB. The students are expected to explore programming (including writing "M" files in MATLAB) and collecting and plotting data before the end of the first week of the class. Since most students have had a course in computer programming before, the burden is not heavy at this point.

**Laboratory Experiments**

The first course, Engineering 207, has four experiments. Lab I is concerned with digital filters and is given two weeks. A specific continuous filter such as the bandpass transfer function

\[ H(s) = \frac{50s}{(s + 5)(s + 50)} \]  

is given, and the students are asked to design a digital equivalent using either rectangular or trapezoid (bilinear) integration or pole-zero mapping or the zero-order-hold equivalent. The supporting material is in the first four chapters of the course text [1], which describes the sample-and-hold operation and z-transforms as well as the principle of the discrete equivalent. Significant points in this lab include the delay of $T/2$ introduced by the sample and hold and the difference between sample-to-sample transfer functions represented by $H(z)$ and a continuous transfer function, $H(s)$. The approximation of a discrete transfer function plus sample and hold, a time-varying operation, by a constant continuous transfer function is to be explored both in theory and in experiment. The students are asked to evaluate the frequency-response magnitude and phase using several approximation techniques and to investigate the effects of different sample rates.

Lab II, also two weeks, covers design by transform methods as treated in Chap. 5 of the text. A plant, such as the Professorial Servomechanism, $1/(s^3 + 1)$, is given with suitable performance specifications, and design is to be done in two ways. First, a continuous design for a controller that meets the specs is to be done. Then a discrete equivalent using one of the methods of Lab I is to be computed, implemented, and tested for performance. The delay of $T/2$ due to the sample and hold may be inserted in the continuous design, and the resulting design used to correct for this effect. Second, the discrete model of the plant is to be computed and a discrete compensator designed using either root locus in the $z$ plane or frequency (Bode) plots from the unit circle. Designs are done for two sample rates and compared. The data are usually selected so that method I at the slow rate results in an unstable digital design! A typical root locus in the $z$ plane is shown in Fig. 4. In this lab, the ideas of digital equivalents studied in Lab I are transferred to control, and the need for reasonably fast sampling is emphasized in order to achieve good design accuracy.

Lab III has three weeks rather than two and calls for the design of a positioning system using state-space methods described in Chap. 6 of the text. The system to be controlled is the two-mass structure shown in Fig. 5. First, a design by pole placement for the $1/s^2$ plant is investigated using a model
with a rigid "spring." State feedback, state estimators, command inputs, and disturbance forces are included. Next, the design for \( \frac{1}{s^2} \) is tried (as if an unmodeled vibration were present) on the fourth-order plant:

\[
G(s) = \frac{k(s + 100)}{s^2(s^2 + 0.3s + 30)}
\]  

The result is almost unstable (by design of the instructors!), as shown by the response in Fig. 6. Finally, a full fourth-order controller is designed and implemented for this plant, and is quite satisfactory, as shown in Fig. 7.

Lab IV studies the effects of quantization by looking at the sensitivity of a discrete frequency response to the coefficients of the corresponding difference equation. Also studied are the effects of signal quantization by taking a signal with fewer and fewer digits in the filter implementation. The experimental mean-squared error is compared to the result obtained by modeling quantization error as uniformly distributed white noise as described by Widrow [2] and others. The match appears to be surprisingly good. These matters are discussed in Chap. 7 of [1].

In the second course, Engineering 208, three laboratory experiments are done. The first is on system identification, and students are given data from a simulation of a two-mass, one-dimensional system equivalent to that of Fig. 5, but with some proportional feedback added for stabilization. The data are arranged to contain only one sinusoid at first, and the problem of inadequate excitation or else (alternative view) overparameterized models is explored. Second, a full excitation is done, and a good model estimate results. The students are asked to explore the effects of sample rate, model order, and data length using the least-squares identification method described in Chap. 8 of [1]. The residual errors are determined by comparing the true system with the estimated model on repeated trials so that error means and covariances can be compared between theory and experiment.

In Lab II, E208, the Kalman filter is explored. Sensor data corresponding to the slant range of a vehicle moving on a line of constant altitude and constant horizontal velocity as shown in Fig. 8 are sampled, and an estimate of the horizontal range is to be computed. The issues of linearization and the relation between the solution to the Riccati equation and the accuracy of the estimate are explored. The noise on the slant range measurement is augmented with added noise from MATLAB's random noise generator so that the filter performance can be investigated with sensor errors ranging from the background A/D quantization up to a factor of 100 or so higher. Also considered is the effect on the state estimates of different as-
s umptions for the covariances of the sensor and process noises and the initial covariance and state values. The students are asked to compare the square root of the covariance diagonal elements with the actual errors time histories and to comment qualitatively on their consistency.

In the final lab, the design of a linear multivariable plant by using the minimization of a quadratic cost function is studied. The plant corresponds to a direct-drive reel-to-reel tape drive, with actuator saturation included. Time-domain specification for step responses and the final tape position and tension are given. The basic task is for the students to find weighting matrices for the control design that meet these specifications, including control saturation effects. Here, MATLAB is especially important to carry out the design iterations necessary. Additional tasks include the design of an integral control scheme, use of alternate state definitions, and a classical design by decoupling the plant into two single-input/single-output systems.

Conclusions
The laboratory courses in digital control at Stanford University have been very successful and are seen as very valuable by the students. The students do, however, warn their successors that the courses are extremely time-consuming. The laboratory equipment now seems to be inexpensive and acceptably reliable. Plans for the future are to introduce servomechanisms and, perhaps, a robot arm to provide more realistic and physical plants to be controlled. We are always searching for ways to reduce the time-consuming aspects typical of laboratory exercises, while keeping the intellectual content as high as possible. The key ideas on this point have been the use of BASIC, the use of MATLAB, and the Stanford Universal Plant. Student excitement and enthusiasm have always been high, and this makes the teaching of these courses most satisfying.

References

Workshop on Computer-Aided Control System Design

The IEEE Workshop on Computer-Aided Control System Design (CACSD) will be held Saturday, December 16, 1989, in Tampa, Florida. The workshop is organized by the Control Systems Society Technical Committee on CACSD and is sponsored by the Society. The workshop is a one-day event to be held at the Hyatt Regency Tampa Hotel in Tampa, Florida, immediately after the 1989 IEEE Conference on Decision and Control. The General Chairman of the workshop is Douglas Birdwell of the University of Tennessee at Knoxville. The workshop provides a forum for the presentation and discussion of new directions in research, which involve significant linkages between systems and controls and relevant areas of computer science and computer engineering.

The workshop will contain presentations of both contributed and invited papers. Both applied and theoretical papers are solicited in the following and related areas: computer-aided modeling, analysis, and design environments; advanced modeling concepts; artificial intelligence, expert systems, and machine learning; and computer-assisted optimization.

All papers accepted for presentation will be published as part of the conference proceedings, which will be available at the workshop.

Instructions to Authors
Prospective authors should submit seven (7) copies of the full paper, headed with the paper title, names, affiliations, complete mailing addresses of all authors, and the statement "CACSD '89." The first-named author will be used for all correspondence unless otherwise stated. Submissions must be made by May 1, 1989, to:

CACSD '89
Prof. J. D. Birdwell

Gene F. Franklin received the B.S. degree in electrical engineering from Georgia Tech in 1950, the M.S. degree from the Massachusetts Institute of Technology in 1952, and the D.E.Sc. degree from Columbia University in 1955. He taught at Columbia University from 1955 to 1957 and has been at Stanford since 1957, where he is now a Professor in the Information Systems Laboratory in the Electrical Engineering Department. His research encompasses all aspects of control incorporating digital logic, including adaptive control of both nonlinear systems and systems with multiple-data sampling.

J. David Powell received the B.S. degree in mechanical engineering from the Massachusetts Institute of Technology in 1960 and the M.S. and Ph.D. degrees in aeronautics and astronautics from Stanford University in 1966 and 1970. He worked (1960-1961) at Outboard Marine Corporation in engine design and testing prior to entering the aerospace guidance and control field at Lockheed (7 years), Analytical Mechanics Associates (1 year), and Systems Control (3 years). He has been on the faculty at Stanford University since 1971 and is currently a Professor with a joint appointment in aeronautics and astronautics and mechanical engineering. His research is concerned with issues in digital sampling policies and the control of tethered spacecraft, fluid flows, and internal combustion engines.

Dept. of Electrical and Computer Engineering
University of Tennessee
Knoxville, TN 37996-2100 USA
Phone: (615) 974-5468/3461
E-mail: birdwell@cascade.engr.utk.edu or birdwell@utk.bitnet

Authors will be notified regarding acceptance of their papers for presentation at CACSD '89 by July 15, 1989. Authors of accepted papers will be provided publication kits and instructions for preparation of their manuscripts for the proceedings. Manuscripts are restricted to eight proceedings pages or less. Authors of accepted papers are expected to attend the workshop to present their work. The registration fee for one author is required when the manuscript is returned on conference mats, as a condition of publication.