Fig. 5. Spectrum of a 1 kHz square wave.

tional tool in a senior year communications laboratory, introducing the students to various signal processing concepts.

REFERENCES

- [1] *TMS32010 User's Guide*, Digital Signal Processor Products, Texas Instruments, Inc., 1983.
- [2] *TMS32010 Evaluation Module*, Texas Instruments, Inc., 1983.
- [3] *TMS32010 Analog Interface Board*, Texas Instruments, Inc., 1983.
- [4] *Precision Digital Sine-Wave Generation*, Digital Signal Processing Appl. Rep., Texas Instruments, Inc., 1984.
- [5] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, pp. 383-390.
- [6] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, pp. 239-242.
- [7] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms*. New York: Wiley, 1985.

TUTOR301—A Successful Application of CAI for Electrical Engineering Problem Solving

J. M. PULLEN AND R. T. MERCER

Abstract—A computer-assisted instruction (CAI) system has been developed at West Point. The system is used to teach electrical engineering problem solving in an introductory course for nonengineers. The system, known as TUTOR301, takes the place of an individual tutor or coach in problem solving methods. The simple form of solution text employed minimizes faculty time required to prepare solution scripts for the computer. It has been accepted well by students, and appears to improve the academic performance of those students who use it.

Manuscript received January 30, 1987; revised July 20, 1987.

The authors are with the United States Military Academy, West Point, NY 10996.

IEEE Log Number 8717787.

I. INTRODUCTION

In the twenty-odd years since educators first began experimenting with computer-assisted instruction (CAI), its potential benefits have been well established by a variety of efforts [1]–[8]. However, among the major factors limiting the application of CAI has been the great investment of faculty time necessary to develop instructional programs or “courseware” for the computer. This has been particularly true of CAI for engineering education. Where CAI has been used, the focus has been more on teaching facts than on developing complex skills such as engineering problem solving. We report here on a CAI application developed at the U.S. Military Academy, West Point, NY, which is an exception to both of these generalities. Our CAI system requires only a modest effort by the faculty, and supports teaching of electrical engineering problem solving methods.

Our CAI system, known as TUTOR301, grew out of a desire to automate the process of “additional instruction” which is traditional at the Military Academy. The West Point cadet’s workload (both academic and otherwise) is highly demanding and time driven. A cadet experiencing difficulty in any subject is expected to contact the course instructor for what amounts to a private tutoring session. This is a natural extension of the small-group method of instruction at the Academy. Over the years, it has served well in helping students to master difficult subject material in the shortest possible time.

Every cadet at West Point takes at least one course in electrical engineering. The course taken by those concentrating in the humanities and public affairs disciplines (about 250 per semester) is known as EE301, Basic Electrical Systems. It is understandable that students in this course often experience difficulty with engineering material. Although problem solutions were posted in classrooms, often the students needed further live demonstrations of problem solving methods. As a result, EE301 instructors were spending many long hours providing needed additional instruction.

We noticed that the process of tutoring soon became repetitive; the instructor would lead the student through a problem solution, encouraging as much unassisted effort as possible, and correcting whenever a false step was taken in the solution. It occurred to us that, in many cases, the tutoring could be performed by an interactive computer just as effectively. This would free the instructor to concentrate on students having unusual difficulties, while making the assistance available at all hours right in our dormitories (barracks). It might also overcome the natural reluctance of students to approach an instructor for assistance.

We implemented TUTOR301 in the Spring of 1984 with about five hours of programming and 100 hours of solution-writing, the latter being distributed over the semester. (Up to half of the solution-writing time would have been required anyhow, to prepare and post solution books.) The result was highly gratifying. Whereas the course is not generally a popular one, the cadets were enthusiastic about the value of the “computer tutor” as a learning aid. Time spent in “live” additional instruction diminished by a factor of at least five. Test scores improved noticeably, about five percentage points on the average. While other factors may have contributed to this improvement, the “computer tutor” was the only major change in teaching methodology over the period in question.

II. A SIMPLE CONCEPT FOR CAI

In seeking to automate the process of coaching for problem solving, we considered two CAI packages available to us, PILOT [9] and KATY [10]. While these seemed promising, both appeared more complex than we required. What we sought was an approach mirroring the human method of coaching, where the instructor asks a question leading toward the problem solution. If the student an-

swers the question successfully, another question is asked. The process continues until a wrong answer is given; at that point, the instructor explains the concept involved and asks the question again, if necessary demonstrating the method of solution. This continues to the end of the solution, with the result that the student is reinforced in the concepts already understood, retaught those not understood, and exposed to the general method of solution for the problem at hand. This stepwise solution paradigm of tutoring was, we felt a good approach to teaching and also one which would lend itself to automation.

It was necessary to distribute solution authorship among the four faculty members teaching the course. All four were, of course, very familiar with the tutoring process, but none knew any CAI language. Rather than familiarize all solution authors with one of the available CAI languages, we prepared a simple Fortran program to select a disk file containing the problem solution and to display it piecemeal. The solution author uses a standard text editor program to prepare a file, consisting of a sequence of questions, along with their correct answers and explanations as to how the answers are obtained. As in human tutoring, the questions are planned to lead the student through the solution to one particular problem. Also, as in human coaching, the program reinforces correct concepts with immediate positive feedback. The text showing the method of arriving at the correct answer to the question is printed only when the student's response is incorrect. This method is highly motivational, and allows the student to proceed rapidly to the point of difficulty, while it repeatedly reinforces the logical progression of problem solving.

For many students, the simple fact that the "computer tutor" was available at their own convenience was sufficient incentive to use it. Others began to use TUTOR301 when they realized that it produced a detailed description of each solution method, which could be captured on a printing terminal for later study. To encourage more cadets to try the system, it was modified to produce a usage log for those who chose to enter their names, although anonymity was permitted. We informed the students that their successful efforts were being recorded along with their names, documenting their efforts. (Of course, only persistence is required for "success," since the "tutor" will eventually give the entire solution.) Those availing themselves of this option typically began with a low course average and were hoping to avoid failure (some of the better students also used the system, because they found it was a more efficient way to study). It was very gratifying to observe that the low averages mostly went away after CAI was used, because it relieved the source of difficulty. At the end of the term, no student making regular use of TUTOR301 had a grade below C-. On the other hand, those few who failed the course had never used the program at all!

III. THE TUTOR301 PROGRAM

The success of our CAI system is undoubtedly due in part to the computer support provided at the Military Academy. The academic area and barracks at West Point are served by a local-area network which supports terminals in all academic buildings, faculty offices, and barracks. (Eventually each cadet room will have access to the network.) The instructional support computers on this network are PRIME 850 superminis with shared disk storage. System reliability is very high due to the ability to shift users to another processor if any unit fails. Response time is typically less than a second, although it occasionally degrades under peak usage.

The only limitation posed by our computer support is that graphic output is not available on most terminals. This was easily overcome by incorporating any required diagrams in the problem statements. While this limited interaction with the student somewhat, it also simplified the process of preparing solutions. We were lucky in that our notation required only minimal use of Greek letters and subscripts. In these cases we substituted a similar notation such as "BETA" for β or "hfe" for h_{fe} .

We feel that another key aspect of TUTOR301's success is its

very simple logic structure. Rather than the elaborate paths through the terminal session commonly available with CAI packages, our program allows only the simple structure shown in Fig. 1: Any amount of text is followed by a single question, which may require either a character or numeric response. Character responses must match exactly (most are single letters, often Y or N). Numeric responses match if within 1 percent of the nominal answer. Whenever the correct answer is not given, an explanation of the correct method (but not the actual answer) is provided. After three reasonable tries are made, the program gives the answer and proceeds to the next text section. (The definition of a "reasonable" numeric answer is one which is within an order of magnitude of the correct value. Unreasonable answers result in messages indicating that the answer is far too large or small, a sense which we feel is important for the student to develop.)

Problem solutions for TUTOR301 are generated using a standard text editor program. The text of the solution is printed out for the student exactly as entered, except for the first character of each line. The first character, which is not displayed to the student, indicates the purpose of the line:

```
T  solution text
Q  question having numeric solution
N  numeric solution value
C  question having character solution
A  character solution value
P  page eject/clear screen
*  author comment (not shown to student)
E  explanation of correct method (follows each answer but is
    only printed when response does not match solution value).
```

The TUTOR301 program itself is quite simple. Although made more elaborate and "user friendly" since its inception, it still consists of less than ten pages of Fortran code, more than half of which is documentation. The general structure of the program is shown in Fig. 2. A sample session and the solution file from which it was generated are given in the Appendix.

Enhancements to the program since the original version have had two purposes: to improve data collection about usage, and to encourage cadets to use the system. For the former purpose, features were added to capture the user's account number, time and data of usage, and individual problems attempted, for statistical analysis (the log of "successful completions" is still available for classroom instructors). For the latter purpose, the system now includes a "bulletin board" for course memos, allows the user to copy the session to a printer if a nonprinting terminal is used, makes "escape" from frustrating problems easy, and ends the terminal session with a "fortune" quip for the day.

IV. RESULTS

As of this writing, TUTOR301 has been in use for five semesters, and as far as the faculty and students are concerned, it is an unqualified success. Several problem solutions have been prepared for each lecture, with more added each semester, so that there are now several hundred problems available. After preparing two or three problems, a faculty member is sufficiently experienced that each new problem takes about an hour to code, enter, and test. It is thus possible to add new problems each semester with a modest effort, divided among the instructors teaching EE301. With a student population between two and three hundred, we have recorded over 6000 completed problems per semester. Clearly, the program is successful beyond our expectations.

As described above, the overall improvement in test scores after TUTOR301 became available was clear. We thus are certain that adding the "computer tutor" to the general course environment enhanced learning of the entire group. We are also certain that the system, to a large extent, takes the place of a faculty tutor for students having difficulties with the course material.

An open question is whether all students who make frequent use of TUTOR301 perform better, on the average, than those who do

Fig. 1. Basic solution structure.

Fig. 2. TUTOR301 program structure.

not. It is our impression that this is the case, so we have used the program's data collection facilities to analyze grades for 120 students taken from the middle of the class (cumulative averages between "C" and "B"). We found that those cadets (about half of the group) who used the program more than 20 times during the term did in fact earn better grades than would be predicted by their grade-point averages, although the difference (a little over a tenth of a grade point) was not statistically significant for the sample size. Our results here may be skewed by the fact that when students work in groups on the system, only one account is recorded. Also the printed solutions are at times shared among students, which amounts to noninteractive use of the system without any record of its use.

V. CONCLUSIONS

We enthusiastically advocate the "keep it simple" approach embodied in TUTOR301. Although there are undoubtedly cases where a more elaborate control structure could be advantageous, the overall benefits of simplicity in preparing solutions are great. Clearly our method is not applicable to all learning environments, but within the scope of basic electrical engineering problem solving we

have yet to find a case where the strategy embodied in Fig. 1 is less than adequate. We believe this is because that strategy reflects an excellent approach to coaching problem solving, with or without CAI. The strategy is one which could and should be expanded to a wide range of scientific and engineering problems. (To do this, a graphics capability would be useful but not necessary. It is, however, important that the computer terminals used support a character set adequate for the symbology of the disciplines involved.) We are willing to make copies of our program available upon request.

We also believe that the success of TUTOR301 lies at least as much in its compatibility with human nature as in its approach to problem solving. The combination of immediate feedback with incentives to practice problem solving is undoubtedly a powerful aid to learning (and, thus, to teaching). We are convinced that our method is applicable to a broad class of introductory engineering topics. While there is surely a limit to the level of abstraction which can be dealt with successfully in this way, we feel that the approach we described in this paper will be found to have merit in any course where concrete principles and structured problems with numeric solutions are included.

APPENDIX
SAMPLE TUTOR301 SESSION AND SOLUTION FILE

TERMINAL SESSION (student input underlined)

WELCOME TO THE EE301 TUTOR PROGRAM!
THIS PROGRAM PROVIDES THE USER WITH SOLUTIONS TO ALL
STUDY AND DRILL PROBLEMS IN EE301, BASIC ELECTRICAL SYSTEMS
FOR AN OVERVIEW OF THE USE OF THIS PROGRAM, REQUEST THE
FOLLOWING SOLUTION WHEN PROMPTED:

LESSON # --> 1
PROBLEM # --> 1

YOUR LAST NAME? Mercer
YOUR FIRST INITIAL? R
YOUR MIDDLE INITIAL? T
LESSON NUMBER? 3
PROBLEM NUMBER ? 1

** NOTE **

TO LEAVE PROBLEM PRIOR TO COMPLETION ENTER "QUIT"
AS THE RESPONSE TO ANY QUESTION ASKED!

SOLUTION FOLLOWS: PRESS RETURN FOR NEXT PAGE

SOLUTION TO PROBLEM 3-1 (2.3-1 IN TEXT)

A Light bulb is rated at 120 V, 100 W. Find the current and
resistance. How much energy in kilowatthours is required to keep
the bulb lighted for 24 hours?

We will use 3 key equations in this section:

$$\begin{aligned} V &= I \times R \\ P &= V \times I \\ W &= P \times t \end{aligned}$$

We know the voltage across and power dissipated by the bulb.

What is the value of the CURRENT (to 3 sig digits)? 1.2

I AM SORRY BUT YOUR ANSWER WAS INCORRECT.

If $P = V \times I$, then $I = P/V$.
TRY AGAIN.

What is the value of the CURRENT (to 3 sig digits)? .833

CORRECT

We have a current of .833 A through the bulb.

What is the RESISTANCE of the light bulb? 144

We now have a resistance of 144 ohms for the bulb.

Our last requirement is to determine the energy used during a
24 hour period.

How many KILOWATT-HOURS are used? 2.4

CORRECT

GOOD JOB, CADET MERCER
SOLUTION COMPLETE.

WORK ANOTHER SOLUTION ? (Y OR N) n

*** STOP

SOLUTION FILE

SOLUTION TO PROBLEM 3-1 (2.3-1 IN TEXT)

A light bulb is rated at 120 V, 100 W. Find the current and
resistance. How much energy in kilowatthours is required to keep
the bulb lighted for 24 hours?

We will use 3 key equations in this section:

$$\begin{aligned} V &= I \times R \\ P &= V \times I \\ W &= P \times t \end{aligned}$$

We know the voltage across and power dissipated by the bulb.

What is the value of the CURRENT (to 3 sig digits)?

N.833

E

Elf $P = V \times I$, then $I = P/V$.

We have a current of .833 A through the bulb.

What is the RESISTANCE of the light bulb?

N144

E

Ohm's law states that $V = I \times R$, so $R = V/I$.

T

We now have a resistance of 144 ohms for the bulb.

T

Our last requirement is to determine the energy used during a
24 hour period.

T

How many KILOWATT-HOURS are used?

N2.4

E

Energy is the dissipation of power over a period of time (t):

Therefore, $W = P \times t$

EA Kilowatt = 1000 watts

T

REFERENCES

- [1] N. Balabanian, "The educational engineering called programmed instruction," *IEEE Trans. Educ.*, vol. E-9, pp. 49-57, June 1966.
- [2] R. D. Strum and J. R. Ward, "Some comments on computer-assisted instruction in engineering education," *IEEE Trans. Educ.*, vol. E-10, pp. 1-3, Mar. 1967.
- [3] H. Bestougeff, J.-P. Fargette, and R. Jocoud, "Computer-aided control of learning," *IEEE Trans. Educ.*, vol. 12, pp. 4-7, Mar. 1969.
- [4] I. Pressman, "Computer-assisted instruction: A survey," *IEEE Trans. Educ.*, vol. E-13, pp. 134-141, Sept. 1970.
- [4] H. L. Plants and W. S. Venable, "Teacher (A revised edition)," in *Proceedings 1971 Frontiers Educ. Conf.*, IEEE, 1971.
- [5] E. B. Koffman, "Design techniques for generative computer-assisted instructional systems," *IEEE Trans. Educ.*, vol. E-16, pp. 182-189, Sept. 1973.
- [6] D. M. Levy, "Computer-managed instruction in electromagnetic field theory," *IEEE Trans. Educ.*, vol. E-20, no. 1, pp. 60-63, February 1977.
- [7] D. A. Carlson, "Integrating computer-aided instruction into the engineering program at the Univ. of Massachusetts," in *Proc. 1984 IEEE Conf. Comput. Educ.*, IEEE Computer Society, 1984.
- [8] J. R. Anderson and E. Skwarecki, "The automated tutoring of introductory computer programming," *Commun. ACM*, vol. 29, Sept. 1986.
- [9] *A Guide to the PRIME Implementation of the PILOT Programming Language*, Univ. Surrey, Guilford, England, Mar. 1981.
- [10] J. Benton, *The KATY System*. Schenectady, NY: Logic World, 1984.