ACES: An Interactive Software Platform for Self-Instruction and Self-Evaluation in Automatic Control Systems

Vassilios Petridis, Spyros Kazarlis, and Vassilis G. Kaburlasos [1]

Abstract — This work presents an interactive-, menu-driven prototype software platform, namely *Automatic Control Educational Software* (*ACES*), for self-instruction and self-evaluation in automatic control systems. ACES is used for enriching instruction in automatic control at Aristotle University of Thessaloniki, Greece in the Department of Electrical and Computer Engineering. The ACES platform includes theory with hyperlinks, a concept-graph, and a database with exercises. Students' answers to exercises are evaluated automatically "on-line". Furthermore exercises can be proposed automatically by ACES. An instructor /supervisor can support in person the learning-effort of a student, monitor the progress of a student and, also, tailor a course's contents on the modular ACES platform. Two statistical hypothesis tests on both attitude questionnaires and student marks in the final (written) exam confirmed that the employment of ACES in the educational process can improve the performance of students in an automatic control course, whereas the attitude of students towards the course does not change significantly with the use of ACES.

Index Terms — Control systems education, interactive educational software, HTML, self-instruction, self-evaluation.

[1] The authors are with the ECE Dept., Aristotle Univ. of Thessaloniki, 540 06 Thessaloniki, Greece.

I. INTRODUCTION

Different authors have recognized that engineering education is not keeping up with the rapid changes, which take place in the practice of engineering [1]. Moreover the IEEE Education Society has acknowledged that this is a period of rapid change in engineering education, especially at the undergraduate level [2].

Lately, the ever increasing power and availability of computers – both hardware and software – has encouraged efforts of using computers to enhance traditional ways of delivering education [3]. In this vein, the European Union has launched a number of research projects for the development of information technologies for learning and training [4].

This work describes a prototype software platform for delivering undergraduate education on automatic control systems. More specifically, the *Automatic Control Educational Software* platform or *ACES* for short, has been developed for self-instruction and self-evaluation as detailed here. ACES is used in the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece since 1999. Preliminary results have been presented in [5].

In electrical engineering various computer-assisted learning applications have been reported for various courses including circuit analysis [6], discrete-time systems [7], etc. Regarding automatic control, in particular, note that instruction at an undergraduate level is a challenging task due to the fairly high mathematical contents which should bear, however, an immediate practical utility [8].

With the advent of World Wide Web (WWW) new opportunities arose for delivery of education. For instance hypertext markup language (HTML) provides a mechanism for allowing media rich representations to be made on the WWW, and methods for developing hypertext learning courses have been reported [9]. Regarding automatic control in particular, note that an overview on Controls Education on the Web has been presented in [10]. Interactive-learning software tools for automatic control courses have been developed in HTML [11]. Likewise, the work in [12] presents an interactive software tool for iterative root locus and Bode plot control system classroom design. Another educational software package for teaching automatic control via the web is described in [13]. Moreover, the work in [14] describes a new technology that uses the web to teach feedback control; in particular a user can modify interactively a controller for a helicopter model located remotely. To advance the state-of-the-art in computer-assisted delivery of education in automatic control at undergraduate level the ACES software platform has been developed as detailed here.

The layout of this work is as follows. Section II presents general characteristics of software platform ACES. Section III describes technically in detail the various software components of ACES. Section IV discusses practical issues pertaining to employment of ACES, moreover an example of using ACES is demonstrated. Section V presents a statistical assessment of ACES. Section VI concludes by summarizing the contribution of this work including potential future work. Three short Appendices A, B, and C provide specific technical details.

II. ACES CHARACTERISTICS

A user-friendly software platform, that is ACES, was developed in modules so as to facilitate maintainability. Seeking for a longer-term employment of ACES via WWW most software modules of ACES have been developed in the HTML programming language and some in Visual C++. Moreover ACES runs on PC platforms including Windows 95 /98 /2000 and Windows NT.

In order to maximize its utility, ACES is able to run either on standalone PCs or on network environments. In the latter case client versions of the software are installed on network workstations, which can communicate interactively with the server.

ACES includes a theory module as well as a database with exercises which can be solved “on-line”. Also marks to a student’s answers can be delivered “on-line” automatically. That is, ACES can be used by the students for self-instruction and self-evaluation. More specifically, the student marks in an exercise are stored in a database together with a student’s *Learning State Vector*, or *LSV* for short. LSV is a three-dimensional vector of marks, which express a student’s competence in : 1) designing, 2) comprehension of selected concepts, and 3) comprehension of specific algorithms. Based on a student’s LSV, ACES can propose exercises to a student. In this way ACES is able to address an individual student’s learning style.

Several tools for interactive system design have been incorporated to visualize the effects of varying various control system parameters, e.g. system poles and zeros. Furthermore, a Concept Graph illustrates relations between useful concepts in automatic control.

It should be pointed out that ACES was not meant to dispense with traditional instruction. Moreover, ACES was developed for computer-based enrichment of the educational process rather than for providing with more homework problems.

III. TECHNICAL DESCRIPTION AND CONTENT

The software components of ACES are described technically in this section. Note that an internet browser Active-X control was used by ACES to view either theory- or exercise- software modules written in HTML. Furthermore ACES requires an Internet browser, e.g. Internet Explorer.

A. Pull-Down Menus

The Main Menu Window includes several pull-down menus as described in the following. The *File* pull-down menu includes switches for connecting-to /disconnecting-from a local server. The *View* pull-down menu specifies the appearance of the "toolbar" and the "status line"; from the latter menu it is also possible to launch the browser for the pages of theory. The *Students* pull-down menu is used to register a student. The *Theory* pull-down menu can activate both hypertext with theory and Concept Graph hypertext pages. The *Exercises* pull-down menu includes buttons for displaying /searching /selecting exercises as well as for displaying: first, the list of solved exercises, second, a student's detailed marks in an exercise and, third, a student's own *LSV* (*Learning State Vector*); also, from here a student can activate an interactive Pole-Zero Design Window. Finally, the *Help* pull-down menu includes all conventional buttons for help. Most of the aforementioned buttons are also available as "stand-alone" buttons on the application's floating toolbar for fast launching.

B. Theory Modules

The Theory Module includes six chapters on 1) System Modeling, 2) Time Response, 3) Frequency Response, 4) Stability, 5) Practical Design Issues & Specifications, and 6) Design Methodologies. Links to other chapters are included as well as an Appendix with useful mathematical formulas and procedures. The user can also access the Concept Graph (detailed below), click on to other Automatic Control web sites, and email to the developers of ACES. Note that the resizable Theory Module Window includes the conventional Back /Forward /Refresh /Home /Search /Close and /Help buttons for hypertext navigation. A button for launching the Exercise Window is also included.

C. Exercises Modules

A database including 168 exercises is available. In particular, there are 9, 13, 21, 49, 22 and 54 exercises for chapters 1, 2, 3, 4, 5 and 6, respectively. The difficulty level of an exercise is denoted by an integer 1, 2 or 3. Moreover an exercise is classified in one of categories: *design*, *concept comprehension*, or *algorithm comprehension*. Each category corresponds to an entry in LSV.

The Exercises Module Window includes sub-windows for displaying both exercise contents and various useful attributes including "student-defined filters" applicable so as to shorten selectively the list of exercises (Fig.1).

(Fig.1 goes around here)

After solving an exercise the student turns in his /her answers in a specified vector format. Then, ACES grades the answers and produces a mark in the range 0…10 according to the "grade of correctness" of a student's answer; that is, instead of giving full credit for correct answers and zero credit for wrong answers, a fuzzy grading system was used as detailed in Appendix A. The marks of solved exercises define the *Learning State Vector* (*LSV*) which comprises three different marks for 1) designing, 2) concept comprehension, and 3) algorithm comprehension, respectively. The LSV is recalculated each time a new exercise is solved. The aforementioned exercise solving procedure is demonstrated by an example in section IV.

D. Pole-Zero Design

The user can place poles and zeros interactively on the scalable complex plane at locations of his /her choice. Furthermore the "gain" of a specific control system can be defined (Fig.2). Then, with a single mouse click, ACES builds a system model, it calculates and, finally, it displays the corresponding Step Response, Bode, and Nyquist Diagrams, and Root Locus on the System Response Window (Fig.3). The user can insert-, move-, or delete- poles /zeros, and refresh the screen. Note that a pole (zero) can be inserted either by clicking the left (right) mouse button on a specific location of the plane or by typing in specific x and y coordinates. A pole /zero can be moved to another location by clicking on it and dragging it.

(Fig.2 goes around here)

(Fig.3 goes around here)

E. Concept-Graph

The Concept Graph was meant to illustrate intrinsic relations between useful concepts in automatic control by a tree-like hyperlink structure. The Concept Graph contains a total of 20 pages, whereas the tree structure has a maximum depth of 4 nodes /pages. The five “root nodes” of the aforementioned tree structure are labeled as 1) System Categories, 2) System Characteristics, 3) System Response, 4) Stability, and 5) System Design. A student can click on a keyword to branch on to a new page with illustrations. A student can also view relations of a concept with other concepts.

F. User Interface

Running in a windows environment the User Interface is fully based on windows, dialogs, mouse input and events. All menu selections are available through either mouse clicking or keyboard commands. The basic buttons also appear in a floating toolbar for fast launching. The toolbar is initially docked under the main menu, but can be reformatted and be docked to alternative locations. A status line displays useful information about the state of the application as well as help hints. Most of the user input is provided by single clicking on buttons and controls. A complete help system has been developed for all application modules. Moreover, multiple module windows can be open at the same time, e.g. Theory Window and Exercise Window.

G. The ACES Supervisor Application

As a supplement to ACES a separate application, namely *ACES Supervisor*, has been developed for the instructor /supervisor. The aim of ACES Supervisor is to enable both "on-line" monitoring of student progress as well as "on-line" definition of a course's contents. For instance using *ACES Supervisor* the instructor can view both the "exercises solving record" of a student and a student's LSV vector. Note that the ACES Supervisor can be installed on any workstation that has network access to the lab server, e.g. in an instructor's office.

IV. USING ACES

Various technical issues are addressed in this paragraph. ACES was installed on PCs using a customized "setup program", which can be either downloaded via the internet or be copied from an installation CD-ROM. Note that ACES software is not transferable from a PC to another by simply copying all application files. This was achieved by inserting certain registry entries at "installation time" which (entries) are queried at "run time". Moreover, at "run time" the application runs a pre-launch diagnostic routine, which produces a *checksum* integer number from all application files. More specifically, *all bytes* in *all application files* are summed up resulting in integer number *checksum*, which is required to be identical to a reserved integer number in order for the application to start. Note that the aforementioned "checksum diagnostic routine" is very fast, more specifically it takes only around 5 seconds to sum up about 7.5 Mbytes of data.

By connecting to a server, the software runs in a “client” mode where all files are accessed from the server. There are at least three benefits from a single (server) copy of ACES: First, a student is not bound to sit at the same place during different laboratory sessions. Second, the instructor can monitor the performance of a student by accessing a single database file in the server PC. Third, the instructor can update the contents of any data or software module of either theory or exercises by replacing a single file in the server PC. By disconnecting from the server, ACES software enters the “stand-alone” mode where all files are accessed from local drives.

ACES was used in a “constrained educational environment” as explained in the following. In one year, students were instructed using traditional classroom techniques. In particular, an instructor taught an automatic control course to one group of students in the classroom two hours a week for one semester. In addition, there was a classroom tutorial two hours a week. Moreover, work was assigned to the students to be carried out either in the lab or at home. Next year, an instructor taught the same automatic control course to another group of students in the classroom two hours a week for one semester. Likewise, there was a classroom tutorial two hours a week. Finally, work was assigned to the students to be carried out using ACES either in the lab or at home. In both aforementioned years the grades of students in the final exam were determined the same way, that is the students using ACES were not treated differently. The following example illustrates in detail the employment of ACES for solving an exercise.

Example

Let a student’s Learning State Vector (LSV) be initially as shown in Fig.4(a). In particular, Fig.4(a) shows that a student has already achieved 8 /10 in ‘designing’, 9 /10 in ‘comprehension of concepts’ and, 6 /10 in ‘comprehension of algorithms’.

Fig.4(b) shows an exercise of type ‘algorithmic’ including hints to a student for inserting his /her answers in a specific vector format. Recall that an exercise is shown in the largest sub-window of the Exercises Module Window (Fig.1). A student may calculate the quantities requested using any tool of his /her choice. After solving an exercise the student needs to click on button ‘Solve Exercise’ (Fig.1) to pass the answers to ACES for grading them. Note that as soon as button ‘Solve Exercise’ is activated certain tasks are carried out automatically by ACES. More specifically, the corresponding exercise’s name is used as a key index to retrieve a specific file, which contains the solutions of the exercise in question. A copy of the latter file is renamed “evaluate.exe” and stored in a pre-specified location on the hard disc. Note that file “evaluate.exe” contains the correct answers of an exercise as well as instructions for grading a student’s answers.

After a student has calculated the answers in an exercise, he /she enters the answers in a *specific vector format* as specified in Fig.4(b), finally the student types in code word ‘evaluate’ as shown in Fig.4(c) then presses the ‘enter’ key. Executable file “evaluate.exe” starts running. Student answers are matched against the correct answers and, using “fuzzy grading principles” as explained in Appendix A, an overall grade is produced (Fig.4(d)). Finally, the Learning State Vector (LSV) is updated such that no more than one entry of LSV increases whereas the other two entries of LSV remain the same (Fig.4(e)).

□

(Fig.4 goes around here)

The entries of vector LSV cannot decrease, they can only increase. The software, which updates the entries /marks in vector LSV, was written such that a student needs to supply correct solutions to exercises in all categories and all levels of difficulty in order for his /her LSV vector to get high entries. In conclusion, execution of file “evaluate.exe” terminates and software control returns again to the Exercises Module Window (Fig.1).

Regarding vector LSV note that the entries of vector LSV reflect a student’s “exercise solving competence”. The entries of LSV can be updated only by ACES and only automatically. Nevertheless either a student or the supervisor can view the LSV entries. Hence, based on a student’s LSV, a course of study can be decided /proposed by either a student or the supervisor. Moreover, ACES software itself has the capacity to propose automatically exercises to a student based on a student’s LSV vector.

It is known from the literature that “effective learning”, in an hypertext-based learning environment, calls for an active participation of students by browsing, selecting, searching, scanning, and tracing [15]. Our experience with ACES has confirmed that students-users of ACES should themselves be self-motivated in order to maximize the effectiveness of ACES. Moreover our experience has shown that the plethora of software tools confused some students who were unwilling to make *active choices*. In addition, a few students were distracted (internet surfing) during a lab session. Note, however, that ACES was not designed for increasing the student interest in a course but rather it was designed for increasing a student’s understanding of course material, all other factors being equal. A statistical assessment has shown that ACES can affect the educational process in a positive way as detailed in the following section.

V. STATISTICAL ASSESSMENT

The effectiveness of ACES was assessed statistically as explained in this section.

A. Statistical Tests

The following two null hypotheses H0 and G0 were tested.

H0: Given the same resources, a group of students who used ACES receive the same marks in the final (written) exam as a group of students instructed with traditional classroom techniques.

G0: The student interest for the course is the same whether they use ACES or not.

Null hypothesis H0 was tested using the marks received by students in the final (written) exam, whereas null hypothesis G0 was tested based on student responses to customized *multiple choice* “attitude questionnaires” handed out to the students the day of the final exam - the latter questionnaires are described in Appendix B.

Statistical testing of either hypothesis H0 or G0 was effected based on two populations of random samples. The first population $\{x_1\}= \{x_{11},x_{12},\ldots, x_{1N_1}\}$ corresponded to N_1 students taught by traditional classroom techniques, whereas the second population $\{x_2\}= \{x_{21},x_{22},\ldots, x_{2N_2}\}$ corresponded to N_2 students who used ACES.

The non-parametric, rank-based Kruskal-Wallis statistical test was used which assumes both independence and identical distribution within each sample. An advantage of the Kruskal-Wallis test is that it can be used for any distribution. Nevertheless, the Kruskal-Wallis test assumes sample populations, which differ only in location but not in dispersion or distributional shape. Therefore an alternative statistical test was employed, that is “test A” which is described in Appendix C. Note that, in addition to the aforementioned advantages of Kruskal-Wallis test, an advantage of statistical “test A” is that it assumes arbitrary sample populations.

For course “Automatic Control Systems” the null hypothesis H0 was tested using populations of samples with sizes N_1=67 and N_2=53. The computed test-statistic value of Kruskal-Wallis was significant at a “less than 0.1%” level. Moreover, using statistical “test A” the null hypothesis H0 was rejected with confidence 99.9% as explained in Appendix C. Therefore hypothesis H0 is not statistically supported. Fig.5 shows two (percentage) histograms of student marks in this course for two groups of students. In particular the histogram in Fig.5(a) was produced from marks of a group of students instructed in “Automatic Control Systems” using traditional classroom techniques, whereas Fig.5(b) was produced from the marks of another group of students instructed in “Automatic Control Systems” using ACES. In view of the aforementioned statistical rejection of null hypothesis H0, it is reasonable to conclude that the use of ACES has improved student marks.

Further, the null hypothesis G0 regarding the student interest in course “Automatic Control Systems” was tested using populations of samples with sizes N_1=50 and N_2=79. For Kruskal-Wallis the test-statistic value was significant at levels in the range 25%-50% for the various questions in the attitude questionnaire. Moreover using statistical “test A” (Appendix C) the null hypothesis G0 was accepted for the various questions in the attitude questionnaire, in particular the difference between μ_1 and μ_2 was not statistically significant (see in Appendix C & Table T1). Therefore it is reasonable to conclude that the use of ACES in the educational process did not change student interest in course “Automatic Control Systems”.

(Fig.5 goes around here)

For course “Classic Automatic Control” the null hypothesis H0 was tested using populations of samples with sizes N_1=149 and N_2=73. For Kruskal-Wallis the test-statistic was significant at a level near 50%, furthermore using statistical “test A”

(Appendix C) the null hypothesis H0 was accepted as explained in Appendix C & Table T1. Hence it is reasonable to accept the null hypothesis H0. Fig.6 shows two (percentage) histograms of student marks in this course for two groups of students. More specifically the histogram in Fig.6(a) was produced from marks of students instructed in “Classic Automatic Control” using traditional classroom techniques, whereas the histogram in Fig.6(b) was produced from the marks of students instructed in “Classic Automatic Control” using ACES. In view of the aforementioned statistical acceptance of null hypothesis H0, Fig.6(a) and Fig.6(b) jointly confirm that the use of ACES in the educational process did not affect student marks in course “Classic Automatic Control”.

Further, the null hypothesis G0 regarding the student interest in this course was tested statistically using populations with sizes N_1=142 and N_2=184. For Kruskal-Wallis, the test-statistic value was significant at a level around 25% for the various questions in the attitude questionnaire. Furthermore using statistical “test A” (Appendix C) the null hypothesis G0 was accepted for the various questions in the attitude questionnaire, more specifically the difference between μ_1 and μ_2 was not statistically significant (see in Appendix C & Table T1). Hence it is reasonable to conclude that the use of ACES in the educational process did not change the student interest in course “Classic Automatic Control”.

(Fig.6 goes around here)

A remark is useful here regarding the applicability of “test A”. More specifically, the reader is cautioned not to be misled by the shapes of histograms (of populations) in either figure 5 or figure 6 which (histograms) apparently do not correspond to a normal probability distribution. Nevertheless, the average value of any of the aforementioned populations is (for all practical purposes) *normally distributed* as explained in Appendix C, therefore statistical “test A” is applicable.

B. Explaining the Results

Regarding the null hypothesis H0, since the employment of ACES resulted in different final exam performances in two different automatic control courses, an explanation was sought. The following explanation is postulated from an instructor’s viewpoint.

On the one hand, an improvement of student marks in course “Automatic Control Systems” has been attributed to the fact that the aforementioned course is basically concerned with novel concepts and algorithms. It appears that the use of ACES in the educational process improved understanding of both novel concepts and algorithms resulting in a clear improvement in the marks. Nevertheless, on the other hand, course “Classic Automatic Control”, apart from novel concepts and algorithms, is also concerned with hand-drawing diagrams such as approximate Bode plots. We hypothesize that the use of ACES by itself cannot improve students’ skill for hand-drawing diagrams. Therefore it can be concluded that the latter is the reason why the use of ACES did not result in any statistically significant improvement in student marks in course “Classic Automatic Control”.

Regarding the null hypothesis G0 the above mentioned statistical results confirm similar results reported in the literature [13], [16].

VI. CONCLUSION

Learning “on-line” is considered to be one of the fastest-moving trends in higher education [17]. In this work an interactive software platform, namely *Automatic Control*

Educational Software or *ACES* for short, was presented for self-instruction and self-evaluation in automatic control systems.

The effectiveness of ACES on the educational process was assessed statistically as detailed in this work based on both student marks in the final (written) exam and on student answers in multiple-choice attitude questionnaires in two different automatic control courses. We remark that no special treatment or (grade) bonuses were given to the group of students who used ACES. Results of two different statistical tests have confirmed that ACES can improve performance of students in the final (written) exam in an automatic control course, whereas the student attitude towards a course did not change with the use of ACES.

From our experience it appears that ACES has motivated students for further study. That is, it seems that ACES has encouraged students to study at home. Nevertheless, no relevant statistical study has been documented in the context of this work.

With some additional work ACES platform can be developed in the Java programming language and thus be launched in the WWW. In the long-run ACES platform could accommodate educational software for other courses as well.

APPENDIX A

This Appendix explains, by-example, how “fuzzy grading principles” have been applied, furthermore the corresponding rationale is explained.

Let the correct answer to a problem question be x=7. Should the student turn in x=7 he /she receives full credit (100%) in this question. Any other answer than x=7 is wrong and it should not receive full credit. Nevertheless it makes common sense to argue that a wrong answer in the neighborhood of the correct answer such as “x=6.9” might “deserve” some (partial) credit, whereas another wrong answer far from the correct answer such as “x=14” does not “deserve” any credit. In this context fuzzy grading principles [18] have been applied as illustrated in the following.

Consider a fuzzy set with isosceles triangular membership function as shown in Fig.7. Hence for student answer x=7 full credit (100%) is given, furthermore for x=6.5 partial credit (50%) is given (Fig.7), etc. Moreover, for turning in answers either smaller than 6 or larger than 8 a student receives no credit.

The exact shape of the fuzzy membership function employed (e.g. triangular, bell-shaped, etc.) as well as the corresponding span of the fuzzy membership function are user-defined. In the context of this work triangular fuzzy membership functions have been used with variable span.

In sum, note that “fuzzy grading principles” have been employed as explained in this Appendix for giving students progressively larger credit for turning in answers ever closer to the correct one.

APPENDIX B

This Appendix describes the attitude questionnaires used in courses “Automatic Control Systems” and “Classic Automatic Control”. Note that an attitude questionnaire included 16 and 22 multiple-choice questions for courses “Automatic Control Systems” and “Classic Automatic Control”, respectively.

On the one hand, typical questions in the attitude questionnaire for course “Automatic Control Systems” included: “Do you find this course interesting?”, “How well are you familiar with the concepts Controllability /Observability?”, “How easily can you design a PID controller for a second order system such that the closed loop system satisfies certain steady state error, and overshoot specifications?”, etc. On the other hand, for course “Classic Automatic Control” typical questions in the attitude questionnaire included: “Do you find this course interesting?”, “How well are you familiar with drawing a Bode Plot?”, “How difficult is it for you to sketch a Root Locus?”, etc.

For each multiple choice question a student was requested to tick off in one of four boxes standing out for: *little* (1), *enough* (2), *much* (3), *very much* (4), where a number within parentheses was meant to quantify the corresponding linguistic term.

(Fig.7 goes around here)

APPENDIX C

A statistical test, namely “test A”, is described in this Appendix for testing either null hypothesis H0 or G0. Test A is proposed here for overcoming a restrictive assumption of Kruskal-Wallis test, that is in particular the assumption that “sample populations should differ only in location but not in dispersion or distributional shape”.

Recall that for testing either hypothesis H0 or G0 two populations $\{x_1\}$ and $\{x_2\}$ of samples were available having sizes N_1 and N_2, respectively, where population $\{x_1\}$ corresponded to students taught by traditional classroom techniques and population $\{x_2\}$ corresponded to students who used ACES.

Based on the Central Limit Theorem it follows that the average of a population of independent, identically distributed (i.i.d.) samples is, for all practical purposes, normally distributed, for population sizes greater than 30 [19, p214]. Note that all the populations of samples used in this work had considerably larger sizes than 30.

For each sample $\{x_1\}$ or $\{x_2\}$, unbiased estimates for both the *mean* and the *standard deviation* were calculated, respectively, [19, p188] by

$\bar{x}_j = \frac{1}{N_j}\sum_{i=1}^{N_j} x_{ji}$ $\quad s_j^2 = \frac{1}{N_j - 1}\sum_{i=1}^{N_j}(x_{ji} - \bar{x}_j)^2$, j=1 and 2, respectively, for samples $\{x_1\}$ and $\{x_2\}$.

Statistical "test A" is described in the following. The *null hypothesis* that "samples $\{x_1\}$ and $\{x_2\}$ derive from the same probability distribution" was tested versus the *alternative hypothesis* that "samples $\{x_1\}$ and $\{x_2\}$ derive from the different probability distributions with means μ_1 and μ_2, respectively".

It holds *probability*$(\bar{x}_1 \le \bar{x}_2) \ge$ *probability*$(\bar{x}_1 \le a)$*probability*$(\bar{x}_2 \ge a)$ for real number *a*. Hence, denoting by z_u the u percentile of the standard normal density [19, p247] it follows, with very good approximation for the population sizes considered in this work, that $\left|\bar{x}_1 - \bar{x}_2\right| \ge z_u\left(\frac{s_1}{\sqrt{N_1}} + \frac{s_2}{\sqrt{N_2}}\right)$ implies $\mu_1 \neq \mu_2$ with *confidence* $c=uu=u^2$.

Let for two populations of samples $\{x_1\}$ and $\{x_2\}$ be $\left|\bar{x}_1 - \bar{x}_2\right| = a\left(\frac{s_1}{\sqrt{N_1}} + \frac{s_2}{\sqrt{N_2}}\right)$. Then if $a \ge z_u$, it follows that μ_2 is different than μ_1 with *confidence* larger than c.

Confidence numbers c can be calculated from standard Gaussian distribution tables. Selected values for percentiles z_u as well as the corresponding *confidence* numbers $c=uu=u^2$ are shown in Table T1.

(Table T1 goes around here)

REFERENCES

[1] W.A. Wulf, "How Shall We Satisfy the Long-Term Educational Needs of Engineers?", *Proceedings of the IEEE*, vol. 88, no. 4, pp. 593-596, 2000.

[2] Strategic Plan, *IEEE Education Society*, http://www.ewh.ieee.org/soc/es/plan.html, 2001.

[3] M.A. Vouk, D.L. Bitzer, and R.L. Klevans, "Workflow and End-User Quality of Service Issues in Web-Based Education", *IEEE Trans. on Knowledge and Data Engineering*, vol. 11, no. 4, pp. 673-687, 1999.

[4] Information Technologies ESPRIT Theme: IT for Learning and Training in Industry, European Commission: Directorate General III (Industry), October 1998, http://www.cordis.lu/esprit/home.html.

[5] V. Petridis, V.G. Kaburlasos, S. Kazarlis, L. Petrou, and G. Hassapis, "Simulation & Hypertext: Software for Instruction and Practice in Real-time Systems" (in Greek), *Proc. Panhellenic Conf. on the Greek Educational System* sponsored by the Greek Ministry for Education, Athens, Greece, 21-23 Sept. 2000, pp. 200-206.

[6] B. Oakley, "A Virtual Classroom Approach to Teaching Circuit Analysis", *IEEE Trans. on Education*, vol. 39, no. 3, pp. 287-296, 1996.

[7] T.G. Engel, and M. Jackson, "Discrete-Time Analysis of Linear and Nonlinear Systems Using Analog Circuit Simulators", *IEEE Trans. on Education*, vol. 42, no. 3, pp. 205-211, 1999.

[8] I. Levin, and D. Mioduser, "A Multiple-Constructs Framework for Teaching Control Concepts", *IEEE Trans. on Education*, vol. 39, no. 4, pp. 488-496, 1996.

[9] C. Chou, "Developing Hypertext-Based Learning Courseware for Computer Network: The Macro and Micro Stages", *IEEE Trans. on Education*, vol. 42, no. 1, pp. 39-44, 1999.

[10] S.E. Poindexter, and B.S. Heck, "Using the WEB in your courses: what can you do? What should you do?", *IEEE Control Systems*, vol. 19, no. 1, pp. 83-92, 1999.

[11] B. Wittenmark, H. Haglund, and M. Johansson, "Dynamic Pictures and Interactive Learning", *IEEE Control Systems*, vol. 18, no. 3, pp. 26-32, 1998.

[12] R.C. Garcia, and B.S. Heck, "Enhancing Classical Controls Education via Interactive GUI Design", *IEEE Control Systems*, vol. 19, no. 3, pp. 77-82, 1999.

[13] G.J.C. Copinga, M.H.G. Verhaegen, and M.J.J.M. van de Ven, “Toward a Web-Based Study Support Environment for Teaching Automatic Control”, *IEEE Control Systems*, vol. 20, no. 4, pp. 8-19, 2000.

[14] J. Apkarian, and A. Dawes, “Interactive Control Education with Virtual Presence on the Web”, *Proc. of the American Control Conference*, June 2000, pp. 3985-3990.

[15] Y.B. Lee, and J.D. Lehman, “Instructional cuing in hypermedia: A study with active and passive learners”, *J. Educational Multimedia and Hypermedia*, vol. 2, no. 1, pp. 25-37, 1993.

[16] T.R. Rhoads, and N.F. Hubele, “Student Attitudes Toward Statistics Before and After a Computer-Integrated Introductory Statistics Course”, *IEEE Trans. on Education*, vol. 43, no. 2, pp. 182-187, 2000.

[17] J.M. Tien, “Individual-Centered Education: An Any One, Any Time, Any Where Approach to Engineering Education”, *IEEE Trans. on Systems, Man, and Cybernetics - C*, vol. 30, no. 2, pp. 213-218, 2000.

[18] J.R. Echauz, and G.J. Vachtsevanos, “Fuzzy Grading System”, *IEEE Trans. on Education*, vol. 38, no. 2, pp. 158-165, 1995.

[19] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, New York, NY: McGraw-Hill International Editions, Third Edition, 1991.

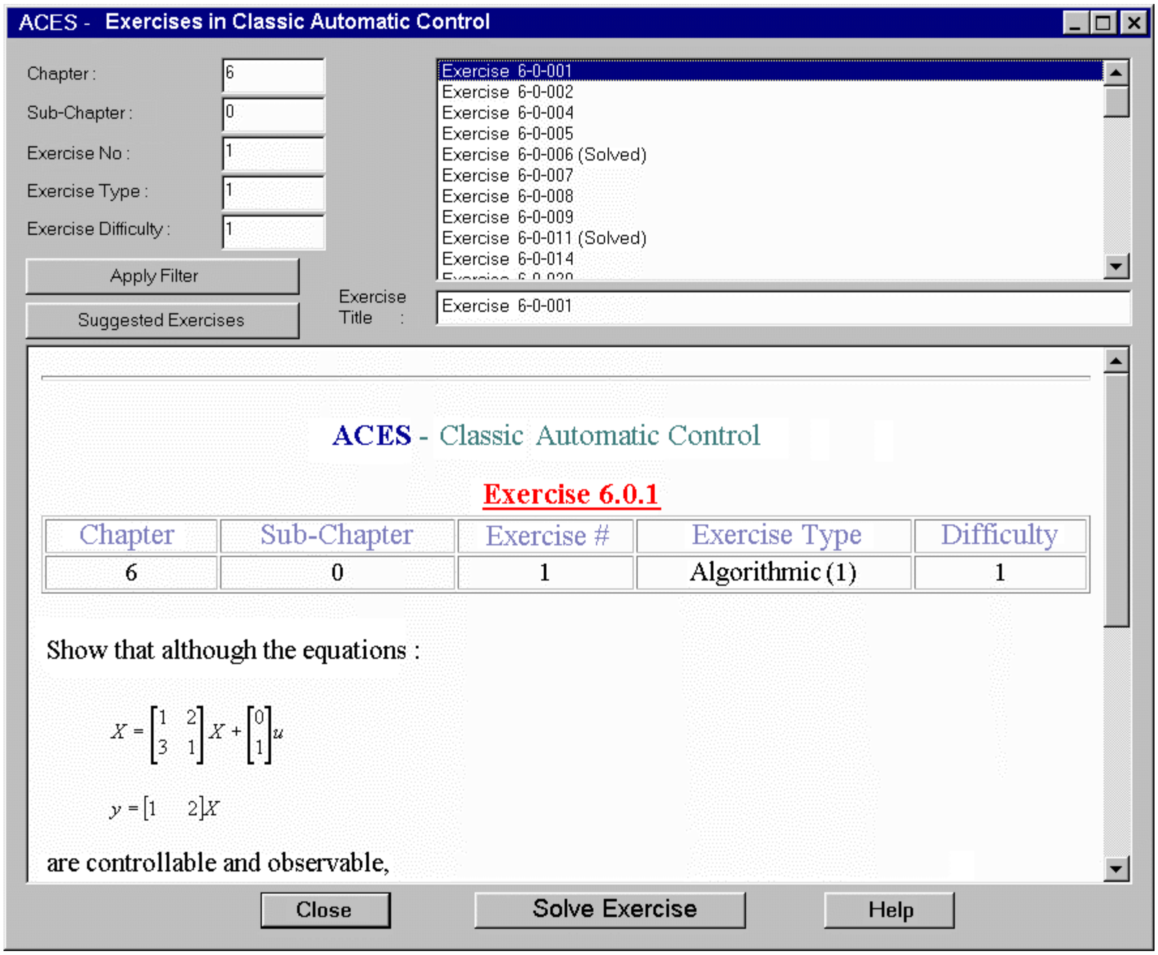


Fig.1 The Exercises Module Window. A selected exercise is displayed in the large sub-window. The "id numbers" of various exercises are shown in the top-right sub-window. Under the latter sub-window is shown the id number of a selected exercise. The five boxes in the top-left corner are either for displaying the attributes of a selected exercise or for defining a matrix of attributes used for filtering the list of exercises shown in the top-right sub-window.

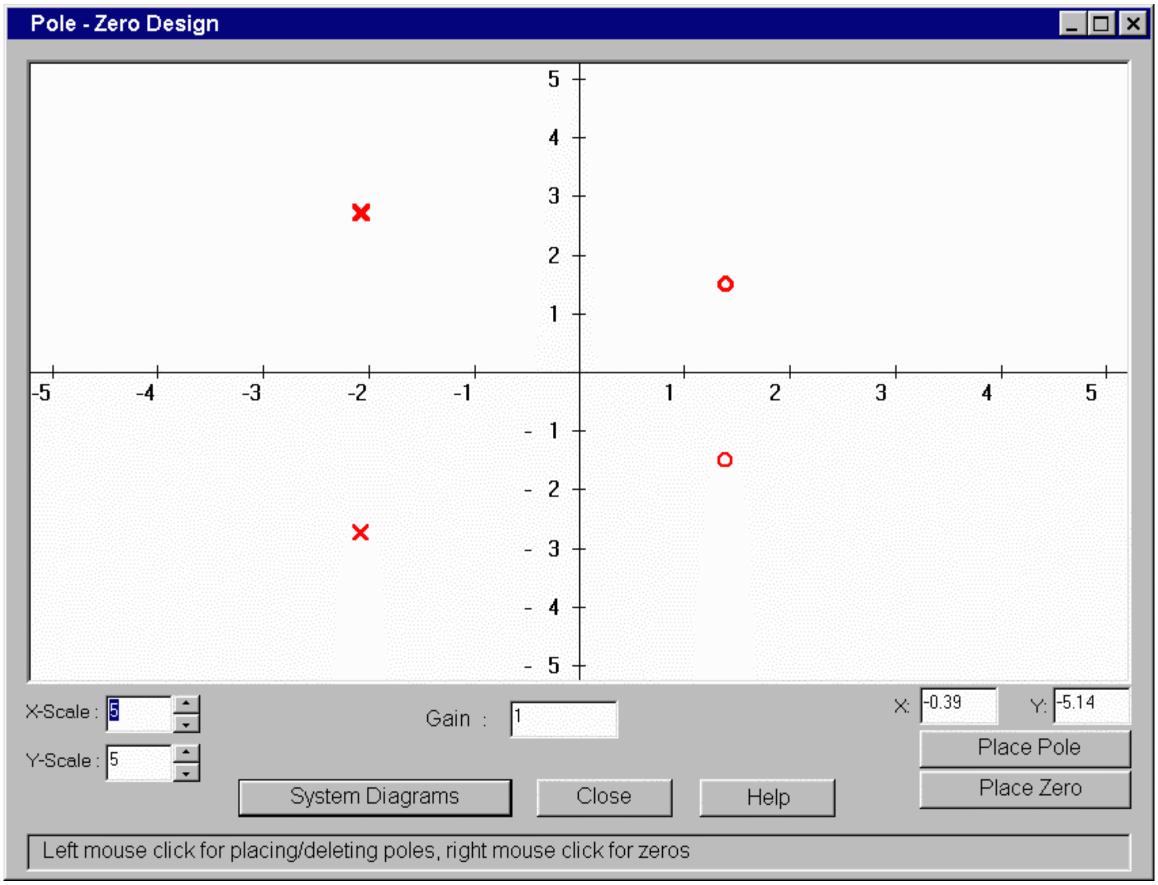


Fig.2 The Pole-Zero Design Window of ACES for inserting-, moving-, or deleting system poles (shown by an “X”) and zeros (shown by an “O”).

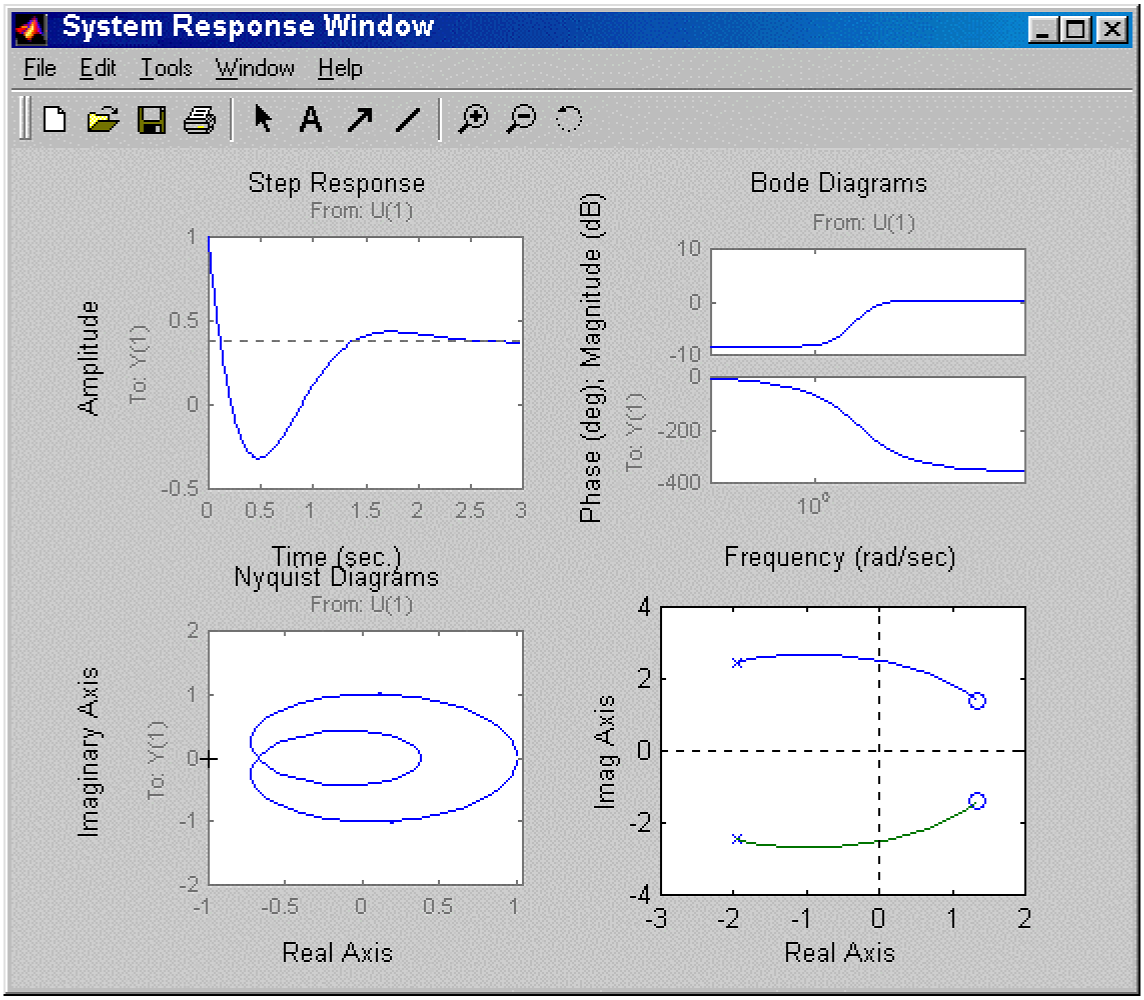


Fig.3 The System Response Window displays the Step Response (top-left), Bode Diagram (top-right), Nyquist Diagram (bottom-left), and Root Locus (bottom-right), of the system whose pole(s) and zero(s) have been defined by the user in the Pole-Zero Design Window (Fig.2).

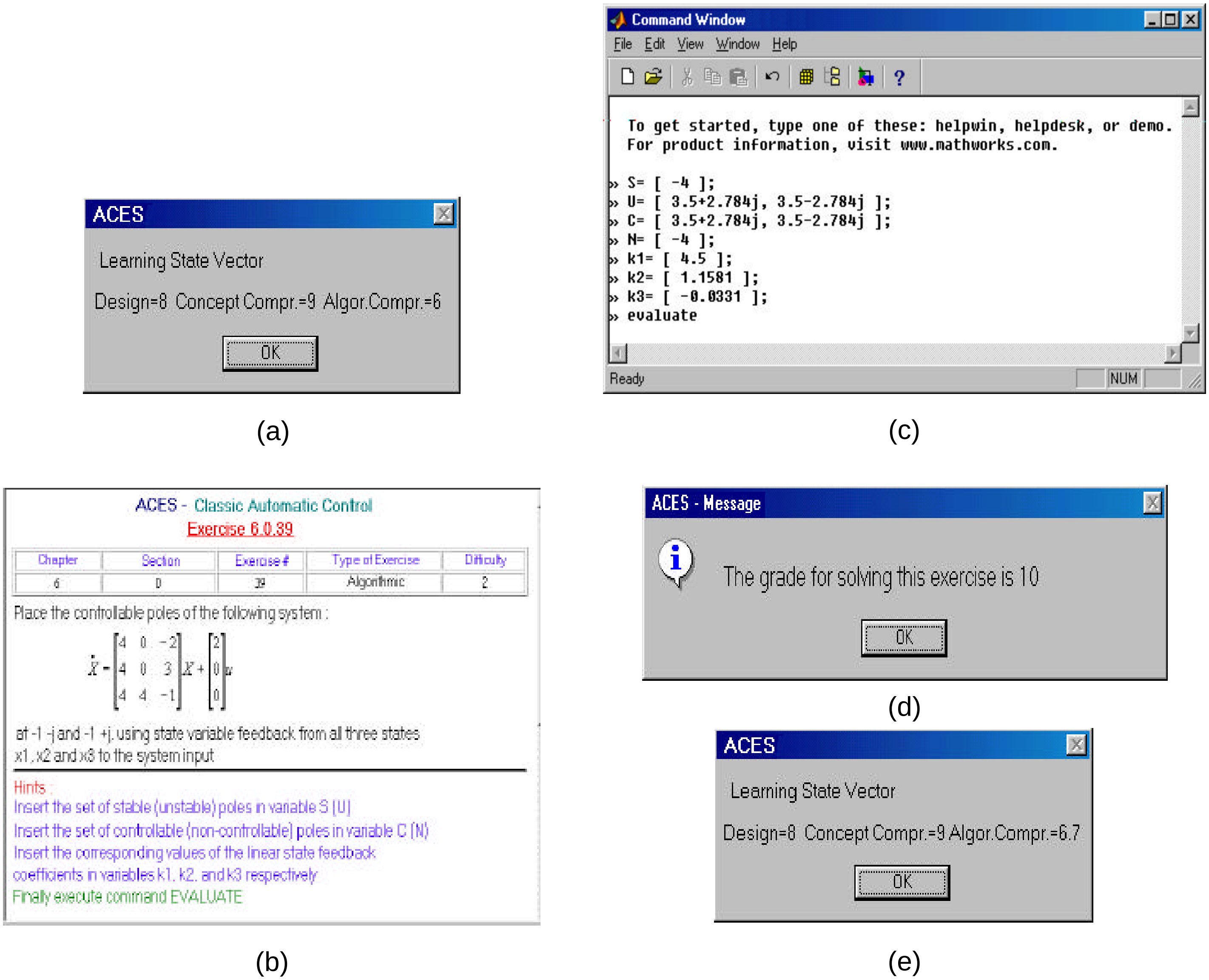


Fig.4 Demonstrating various steps in the solution of exercise item 6.0.39 using ACES.

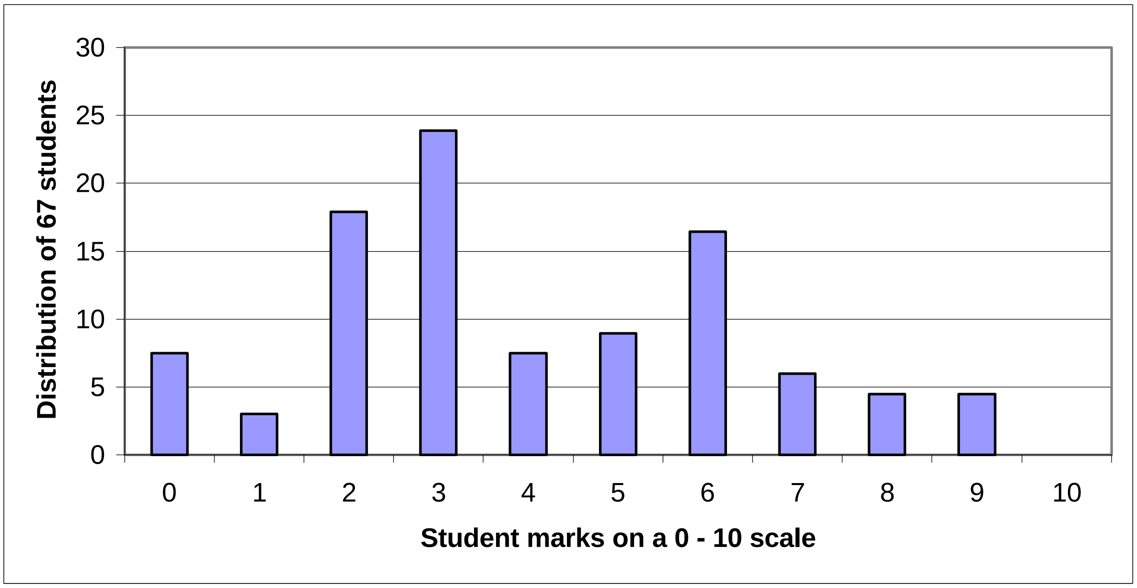
(a) Entries of LSV (Learning State Vector) before solving an exercise.

(b) A sample exercise in course ‘Classic Automatic Control’. Exercise answers are required in a specific (vector) format.

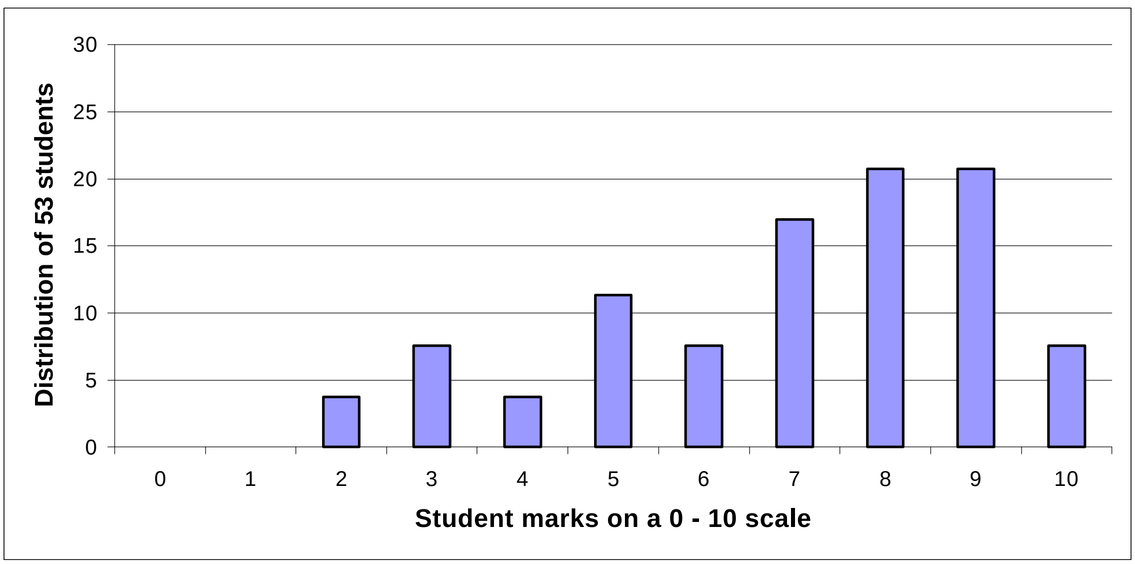
(c) A student turns in his /her answers as specified by ACES. Then a student activates executable program ‘evaluate’ for automatic grading.

(d) An overall grade is produced as explained in Appendix A.

(e) After solving an exercise the LSV vector is updated accordingly.



(a)



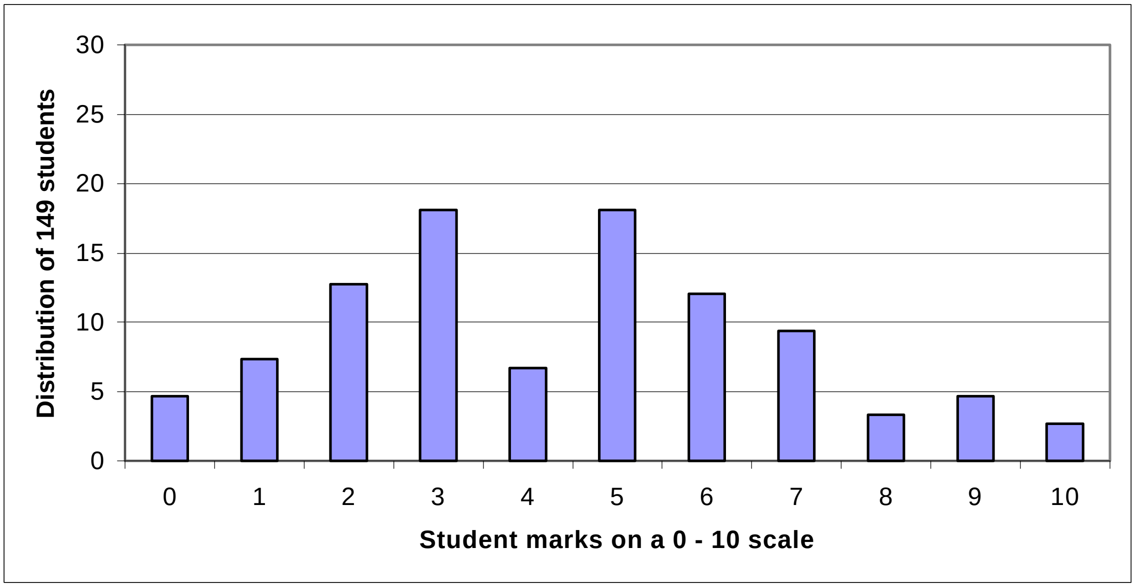
(b)

Fig.5 Histograms of student marks (percentage-wise) in course "Automatic Control Systems" for two groups of students:

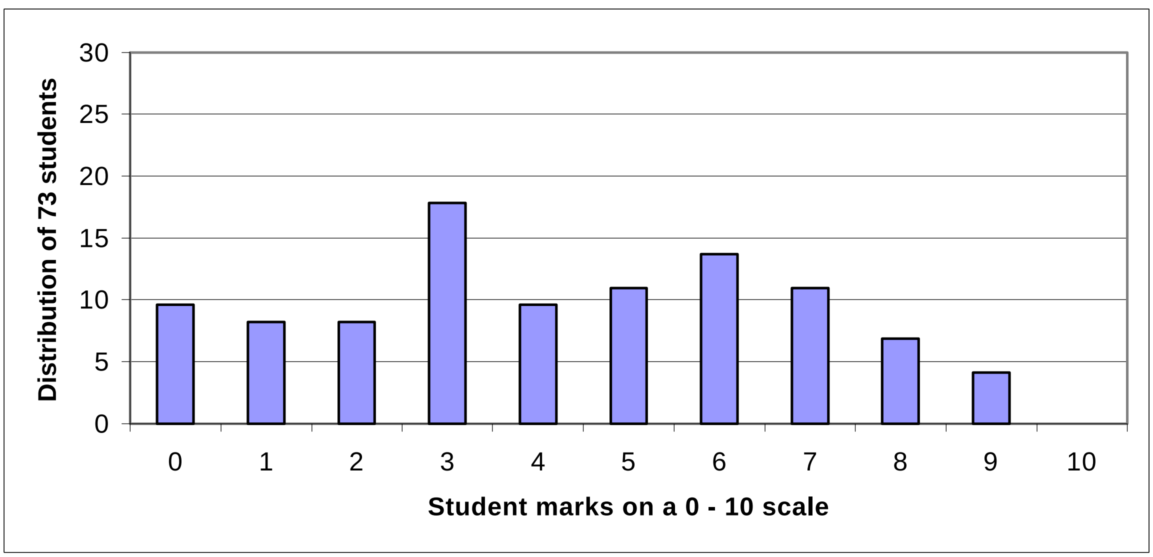
(a) a group of students instructed using traditional (classroom) instruction, and

(b) a group of students instructed using the ACES software.

An improvement of student performance in the final (written) exam has been inferred statistically when ACES was used in the educational process as explained in the text.



(a)



(b)

Fig.6 Histograms of student marks (percentage-wise) in course "Classic Automatic Control" for two groups of students:

(a) a group of students instructed using traditional (classroom) instruction, and

(b) a group of students instructed using the ACES software.

No change of student performance in the final (written) exam could be inferred statistically when ACES was used in the educational process.

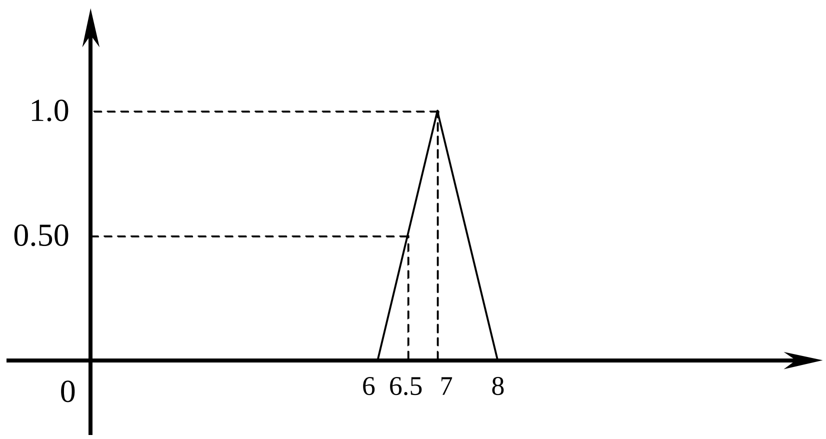


Fig.7 Fuzzy sets with triangular membership functions are used by ACES to give potentially partial credit to a wrong answer. For the fuzzy membership function shown above a student receives full (100%) credit for turning in the correct answer x=7, whereas a student receives partial (50%) credit for turning in x=6.5. For turning in either x≤6 or x≥8 a student receives no credit.

Table T1

Values for percentiles z_u and the corresponding confidence $c= u^2$

percentile z_u	confidence $c= u^2$
3.29	99.9 %
3.09	99.8 %
2.58	99.0 %
2.33	98.0 %
1.97	95.0 %
1.64	90.0 %
1.44	85.0 %
1.28	81.0 %

For $z_u < 1.28$ it follows that the difference between μ_1 and μ_2 is not statistically significant.