Changing Tools and Methods in Engineering Education

Jerzy Moscinski
The Silesian University of Technology, Institute of Automation,
Akademicka 16, 44-100 Gliwice, Poland
jerzy.moscinski@polsl.pl

Abstract - **The paper concerns the changes in teaching tools and methods with respect to engineering education, especially with respect to laboratory exercises and project work and international students' exchange. The author analyses the phenomenon using experience from teaching computer controlled systems oriented courses at large European university of technology. Traditional teaching of courses concerning control systems programming, adaptive control, estimation and identification and similar included in many European universities extensive laboratory exercises during which students were supposed to develop pieces of software needed for system simulation and analysis. In many cases such system simulation was developed in C/C++ type of programming language and knowledge concerning e.g. numerical properties of algorithms and optimization methods was required. There were several factors that affected the above mentioned teaching methods and toolset – some of them are discussed in the paper using courses on Control Systems Programming and Adaptive Control for selected Electrical Engineering students as example.**

Index Terms – Engineering education, Modern tools in education, Methodology of engineering education, Internet technologies in education, E-learning.

INTRODUCTION

This paper discusses the changes that were observed by the author and his colleague teachers when working with electrical engineering students during the last few years. The experience reported in the paper concerns mainly the changing programming languages and platforms as well as students' attitude towards programming as part of engineering education. The examples given in the paper include lectures and laboratory and project work with respect to courses on Control Systems Programming and Adaptive Control, at the Faculty of Automation, Electronics and Computer Science, Silesian University of Technology (abbreviated to SUT in what follows).

It is argued in the paper that tools and methods used for teaching the courses mentioned above have changed during the last decade considerably. First, there appeared environments of MATLAB/Simulink kind which relieved students from lower level of system simulation programming. Second, there appeared languages like Java and concepts of building such simulation pieces of software in applets form which are ideal for Internet presentation and e-learning. Third, new environments and systems enable to build advanced graphically oriented and interactive interfaces which become more interesting for students than the simulation contents of the software. Fourth, large scale international students' exchange caused the meeting of students prepared to project kind of work with students preferring using the closed software for partial results. Fifth, because of labour market needs the students in many cases are more interested in mastering specific programmable logic controllers programming environments than the general ones.

The author's view is that the changes in tools and methods mentioned above and analysed further in the sections below should be carefully observed by engineering education professionals and thoroughly taken into account while deciding upon the choice of laboratory and project work topics and software tools used by students.

CONTROL SYSTEMS PROGRAMMING COURSE OVERVIEW

As it was mentioned above the ideas presented in this paper resulted from the author's experience in teaching – among other – courses on Control Systems Programming and Adaptive Control. Below, the contents of these courses – with respect to lectures as well as laboratory exercises and projects – will be presented in order to emphasize what kind of programming skills are useful while working with these topics and what programming languages and platforms seem to be the most appropriate in this context.

Lecture Contents

The course on Control Systems Programming concerns several important groups of topics from the control system design and implementation field [1], including at first introductory information on importance of control system programming phase, programming languages and packages used in control system synthesis, low level control systems programming, real time programming considerations and mechanisms of real time operating systems.

Further lectures cope with the problems of continuous time system simulation, discrete process simulation scheme, MATLAB with toolboxes as one possible control system design programming environment, automatic generation of control system design and simulation code, code generation for real time implementation of control systems. The topics concerning communications schemes in distributed control systems, software for computer control networks transmission and distributed systems processing are also covered.

Other topics include serial port programming, assembler type programming languages in control, analysis of applicability of C programming language for low-level control systems programming and its flexibility and portability features importance. Object oriented programming languages and their applicability in control system design and simulation are also covered, together with numerical efficiency of control systems programming issues, effective control systems algorithms, numerical stability of identification and control algorithms in micro-controller based control systems.

Specific control systems analysed during lectures include PID control algorithm, with emphasis put on digital implementation of PID controllers and safety nets necessary to use while implementing it. The rest of lecture topics include control system software development in industrial environment, using AD/DA I/O control system hardware, programming I/O parts of control system, signal processors usage in control systems, as well as design techniques, testing and implementation of control systems software using specialised hardware with signal processors.

Code Generation and Programming

It is clearly seen from the above course description that much information presented to the students during lectures concerns the code generation and other programming issues as they are especially important for efficient control systems programming. It is obvious that mainly these topics – including tools and methods used – should be trained while working with students during laboratory exercises and projects accompanying the CSP lectures.

Software development in control systems, or programming control systems, is a major task while implementing control system in reality. Programmable control systems are incomparably flexible and enable easy control software upgrade and maintenance. The CSP course students at the Faculty of Automation, Electronics and Computer Science are presented with detailed consideration on the programming language and platform choice and they should have a chance to check these suggestions during their self-laboratory-work. The information on these issues, which is transferred to students, is summarized below – in parts concerning different software tools groups.

It is obvious that using high level programming languages – especially highly acclaimed C language – should result in less effort in developing control system software – especially in the debugging phase. High level languages are well known and usually easy to use and coming with a lot of additional software tools enhancing the software development process. However, such languages are also not optimised with respect to the code efficiency, signal processing tasks, and application size. Time-critical signal processing applications may suffer from such code inefficiency.

Assembly language is the best tool from the efficiency of the resulting code point of view. It is common to code some part of the algorithm in assembly language even if one uses high level language as basic programming environment. Obviously assembly language programming is target hardware application specific and seeks thorough processor knowledge from the programmer. It is generally agreed that in digital signal processing applications one uses assembly language for some parts of the software project.

There exist specific languages for signal processors which share some features possible to be found in either high level languages or assembly language. Usually such languages offer the ease of development of considerably greater level when compared with assembly language programming environments, and the resulting code efficiency is much better that high level language programming code efficiency. Such languages are not numerous and they obviously seek some standards.

There are available some code generation packages which enable an automatic code generation for some specific processor hardware platforms. There exist some packages of this kind e.g. for digital signal processors from TMS320 family. Such packages enable the designer to concentrate on the control system design leaving the processor architecture specific issues to the built–in development package features.

There are many control system design and simulation packages available, including most popular MATLAB / SIMULINK environment. Such packages enable fast and easy design of complicated control algorithms as well as excellent possibilities of control system performance simulation and optimisation – thus greatly improving the control system design and implementation phase.

There are available device simulators for several digital signal processor devices that may be run on popular hardware/software platforms. Such tools provide full simulation of the specific signal processor instruction set and instruction timing. Full effects of control software implementation may be checked in such simulator environment without absorbing and building measurement environment of the plant hardware.

Laboratory Exercises and Project Work

Typical project exercises accompanying the Control Systems Programming lecture concern the control system design using the MATLAB environment, control software generation in MATLAB as well as C/C++ programming languages, implementation of control algorithms in the hardware environment including signal processor, programming I/O subsystem services using specialised AD/DA hardware as well as standard PC communication ports.

The students are also obliged to work on the set of laboratory exercises which are integral part of the Control Systems Programming course. Typically there are about six specific laboratory exercises topics of the following kind:

- peripheral devices communication algorithms programming in computer systems;
- testing numerical stability and efficiency of identification and control algorithms in C/C++ language implementation;
- control system design and simulation using object oriented techniques and C++ language;
- control system design using MATLAB environment;

- continuous and discrete time control systems simulation using MATLAB/Simulink environment;
- design and software implementation in microcomputer control system using TMS320C31 signal processor.

It is clear again that laboratory and project work needs quite substantial programming skills from students, including control systems programming in languages of C/C++ kind.

ADAPTIVE CONTROL COURSE OVERVIEW

The other course that was clearly affected by the changes in programming tools and methods concerned Adaptive Control. This course is offered to electrical engineering students specialising in computer control systems as well as to special group of Macrofaculty students with curriculum composed of chosen courses from automation and robotics, electronics and telecommunication and computer science fields.

Lecture Contents

The course on Adaptive Control [2] concerns in its introductory part such topics from the adaptive control field as process control system tuning and adaptive control system classification. Next, the students learn some basic types of adaptive control including model reference adaptive control systems, adaptive control systems with programmed changes of the controller parameters and adaptive control systems with system model identification.

Stability and minimum-phase properties of the adaptive control system plant are covered next as well as stability, convergence and robustness related requirements in the adaptive control system. Direct and indirect adaptive control systems are defined and analysed as well as transfer function and predictive model of the plant in adaptive system. Identification and parameter estimation in adaptive control systems are taught as next step and transfer function and predictive model identification options are analysed.

Important part of the lectures concerns stochastic models of disturbances in adaptive control systems as well as deterministic disturbances - including representation of such disturbances, disturbance model, deterministic disturbance types, disturbance rejection possibilities and other.

The lecture includes also extensive coverage of dynamic system simulation topics, including simulation experiment as a tool for adaptive control system analysis and synthesis. Adaptive control system performance indices evaluation is also covered.

There are two main parts of the Adaptive Control lecture and the firs of them concerns the control algorithms used in adaptive control systems. The lectures from this part introduce to the students pole placement adaptive control systems, pole-zero placement adaptive control systems, reconsider model reference adaptive systems and describe various versions of famous – and difficult – adaptive minimum variance control systems. The other adaptive control algorithms discussed during AC lecture include long range predictive adaptive systems, MAC and GPC examples of predictive control algorithms, adaptive control systems based on step and pulse responses of the plant and auto-tuning and adaptive versions of PID controllers

Second main part of Adaptive Control lecture concerns identification and estimation schemes and algorithms. The topics covered during lectures include recursive estimation algorithms in adaptive control systems, forgetting factor choice in recursive estimation schemes, numerical properties of recursive estimation algorithms, stability of adaptive control systems with various estimation schemes, as well as convergence of recursive estimation algorithms in adaptive systems.

There are also additional and non-obligatory parts of Adaptive Control course which include lectures and exercises on continuous time model based adaptive control systems, fuzzy logic based design and synthesis of control systems, fuzzy adaptive control design schemes and applications., evolutionary optimisation algorithms as tools for identification and choice of structure in adaptive control systems and multivariable adaptive control systems – multivariable models of control system plant and multivariable plant model identification.

Course Objective and Essence

The main objective of the Adaptive Control course is to provide the students with basic and advanced knowledge concerning theory, analysis and synthesis of adaptive control systems. During the course the students should develop the skills concerning the methods of theoretical analysis and synthesis of adaptive control systems as well as the skills of building and using computer simulation packages for analysing the behaviour of such complicated control systems.

It is well known that analysis and synthesis of adaptive control systems is a difficult task [3]. Usually engineers think about implementing adaptive control when the plant to be controlled or its environment is nonstationary, which also means that it is very difficult to analyse from the control theory point of view. Other situation that leads to the adaptive control system is the nonlinearity of the controlled plant and set-point changes – which is rather typical but once again complicates considerably analysis and synthesis of such systems because of their nonlinearity.

In addition to the bad news above one should understand that in adaptive control system there are two groups of changing elements – signal values and controller parameters, and because these elements appear in products in almost all models of such systems, this makes such systems inherently nonlinear, i.e. difficult for analysis from the control theory point of view. Unfortunately this is not everything: one should also remember that there is an important duality in adaptive control systems with respect to control task and corresponding performance index and the identification task with its own performance index. The two tasks and indices are not separate and it is not easy to solve them both at the same time.

The above remarks were put in order to justify the importance and usefulness of simulation methods for the synthesis of adaptive control systems and analysis of its properties. It seems especially reasonable to use control system simulation tools for adaptive control systems as we lack clear and sure procedures for adjusting adaptive control schemes parameters and criteria that would guarantee stability of such systems and convergence of estimated parameters. This means that engineers implementing adaptive control systems and students attending the Adaptive Control course should be accustomed to the dynamic systems simulation problem and software used for this task.

It should be also realized that control system simulation – both in the continuous time and discrete time domains – is a difficult task and should include very careful choice of numerical methods, programming languages and software development platforms. Also, the identification and estimation algorithms used in the adaptive control systems are quite complicated pieces of software that have to provide the rest of the system with the best possible estimates of model parameters very fast and with the best possible numerical properties.

It should be also noted that many adaptive control systems are implemented and simulated in the form of discrete time systems – because of the choice of the control algorithm. In many cases it may be advantageous to use in such case as small sampling interval as possible. It is well known that too small sampling interval may cause problems in real adaptive control system implementation but also during simulation, therefore special transforms and models are sometimes used and such aspects should be also included in the possibilities of programming language and platform choice.

Laboratory Exercises Topics

There are typically twelve laboratory exercises within Adaptive Control course that demand from students programming work on the following group of topics:

- choice of structure of adaptive control systems, sampling period, measurement devices and actuators representation in simulation experiments;
- comparison of adaptive predictive control algorithms, minimum variance control algorithms and pole/zero placement controllers;
- choice of structure and parameters in adaptive control systems with open loop unstable and/or nonminimumphase plants;
- numerical properties of recursive estimation algorithms in adaptive control systems;
- synthesis of adaptive control system for nonstationary plant with typical static nonlinearities – comparison of control results for various estimation algorithms and parameters;
- advanced MATLAB/Simulink based design and simulation of adaptive control systems.

Again – as with CSP course – it is clear that laboratory exercises accompanying Adaptive Control course lectures need a lot of nontrivial programming from students.

CHANGING TOOLS AND METHODS

The success of Java language and Java platform, together with wide usage of Internet technologies and e-learning concepts in education, changed the practice of teaching programming languages and system simulation in many universities. At present many university teachers are eager to use multimedia tools and modern education concepts with their courses. The Java platform is especially useful while considering the modernization of engineering education lectures and laboratory exercises because it enables using applets for easy Internet based teaching with system simulation examples and because it is capable of performing serious calculation in reasonable time, which is in many cases necessary while talking about engineering education topics examples, especially with respect to control systems simulation and system identification.

Java language offers full-featured, general purpose and object orient programming environment which makes it good candidate for replacing the basic computer programming language in engineering education – replacing e.g. C/C++ languages. Many students like to learn Java because of the language popularity, connections with web programming and usually better prospects of finding job after studying in the programmers' sector. Unfortunately, the numerical efficiency of Java language is not among the highest ones, the code is also in a way superfluous because of the strong object oriented character of the language, and sometimes even the basic operations need looking for classes outside of the main Java package. These and similar reasons cause that while talking about control system simulation, implementation of specialized estimation schemes and testing features of advanced adaptive control algorithms during laboratory exercises and projects the Java platform does not seem to be the most suitable one.

When the Computer Control Systems students at SUT are supposed to develop program within courses on Control Systems Programming and Adaptive Control they usually want to use environments and languages of Rapid Application Development type. The reason is simple: they can "rapidly" get an application that is running, has a professional look and is typically capable of producing complex and eye-catching plots. In many cases building the computer program interface, dialogue layer and presentation oriented elements is easier than implementing complex algorithm concerning estimation and control algorithm, as well as dynamic system simulation. On the other hand it is perfectly possible to spend really much time on such work – extending and enhancing the simulation program interface, even using excellent Builder-, Delphi-, Visual-like programming environments. Unfortunately this leads to the result that the students feel that they worked a lot on their program but from the point of view of control or identification system simulation the program is either poor in features, or full of bugs or both.

The other possibility with respect to the choice of software for courses on Control Systems Programming or Adaptive Control is to use the widely known MATLAB software with many excellent toolboxes, including the famous Simulink for dynamic systems simulation. It is also possible to look for software with possibly less features but free for academic use, like Scilab. Such systems are like libraries of high-level routines that make programming considerably easier. In fact students can very quickly build running programs and dynamic systems models that are really advanced from the point of view of the system complexity and amount of numeric calculation involved. The problem is that when using such systems the students do not work on the implementation of modern control and estimation algorithm and dynamic system simulation but rather on the organization of calculation on the basis of existing software blocks. This is of course useful if we want the students to build software for testing concepts concerning control and estimation algorithms, but it is less useful if we want them to better understand these algorithms. Also, when students implement software with such integrated high level systems it is usually more difficult for them to find and understand the source of any odd behaviour of it.

The teacher has also to decide if the laboratory exercises accompanying the course on Control Systems Programming or Adaptive Control should be more like project work with some software development or should it be only the usage of closed software for performing experiments and drawing conclusions. In the latter case the students are not supposed to master their programming skills in order to accomplish their work on control and estimation concepts. On the other hand, also due to the growing international exchange of students between SUT and other European universities, the engineering education model at the Faculty of Automation, Electronics and Computer Science, SUT, includes more and more project work, also in the form of sequences of laboratory exercises, in order to make the education closer to the one typical for other European countries. Therefore the question of choosing the programming language and platform is becoming more and more important for the competencies achieved by students attending the Control Systems Programming and Adaptive Control courses.

CONCLUSIONS

The author concerned the problem related to the current trends in teaching programming languages in technical universities and the models of running laboratory exercises and projects for courses from the control and estimation algorithms and systems topical area. The experience presented in the paper results mainly from teaching courses on Control Systems Programming and Adaptive Control, for Computer Control Systems specialization students.

The author presented the typical contents of lectures and laboratory exercises concerning the mentioned courses, and argued that the choice of programming platform and skills in low-level programming of control and estimation algorithms were crucial for the good understanding of the course contents by students. Unfortunately, recently the students are in many cases worse prepared for developing software concerning control and estimation systems simulation, numerical methods and optimization methods – they are more interested and better educated in web-programming, building interfaces and Internet database systems. If the students are supposed to build control and estimation systems they do this using high-level libraries and environments, which does not imply that they know the theory behind the implemented algorithm sufficiently well.

The net result of the facts stated in the paper is the following question: are we currently able to teach the engineering students how to transform concepts and algorithms into reliable and useful software? The author’s opinion is that we have to be aware of threats to this education objective resulting also – paradoxically – from the development of programming languages and platforms – like Java, RAD systems and MATLAB/Simulink software.

The paper surely does not concern the threats that have to appear in all technical universities. The objective of the paper is rather to point out the other face of modern and otherwise extremely useful software tools and systems used in engineering education. It is maybe the question of the level of understanding of theory and algorithms implemented in such tools – by the technical universities graduates. If a teacher asks the student how to calculate the root of the first order polynomial and the answer is that the method is to use MATLAB, then something is wrong almost for sure. Similarly, it seems obvious that the university graduate should be able to calculate the scalar product of vectors without software tools of such kind. For more complex tasks the answer is more difficult but it seems obvious that in a vast majority of cases the student should understand how the new tools work – in order to be able to interpret the experiments results, to diagnose possible sources of mistakes and to build similar tools. The possible answer is that we – the teachers – have to teach our students not only the algorithms and software tools for accompanying advanced calculation, but also the proper attitude towards such tools. The students have to know not only how to use such modern tools technically but also to understand that such tools’ role is not to replace the part of the student brain – and this is our, the teachers, task and responsibility.

.

REFERENCES

[1] Astrom, K. J., Wittenmark, B., "*Computer Controlled Systems*", Prentice-Hall, 1997.

[2] Astrom, K. J., Wittenmark, B., *"Adaptive Control*", Prentice-Hall, 1989.

[3] Middleton, R. H., Goodwin, G. C., "*Digital Control and Estimation. A Unified Approach*", Prentice-Hall, 1990.