THE IMPACT OF IEC (6)1131-3 ON THE TEACHING OF CONTROL ENGINEERING

Andy Verwer

1. Introduction

It is generally agreed that developments in engineering education should be driven by the needs of industry. For many years industry has been telling us that it requires broadly educated, multi-disciplinary engineers who can deal with the many different aspects of modern machines, products and processes in design, manufacture and maintenance roles. This is particularly true in the field of instrumentation and control where engineers must understand the overall process and trace cause and effect relationships through mechanical, electrical and software elements of complex systems[1].

Graduates also need to possess a balance of practical skills and theoretical knowledge. Within control engineering students must learn the basic principles of feedback system stability but also how to design, install and tune a practical control loop. The correct balance between principles, theory and technology is sometimes difficult to achieve. Education which is centred on particular equipment cannot give the breadth of exposure required. Further, when using a certain manufacturer's hardware or software, there is a danger that the exercise becomes simply "training" rather than broader based "education".

This paper looks at the teaching of traditional control engineering in further and higher education and examines the impact that IEC (6)1131-3 can have. An approach in which IEC (6)1131-3 compliant programming software is used with realistic simulations of plant is described. This approach has been found to have many benefits for teaching and also for practising instrumentation and control engineers.

2. Quantitative and logical control systems

For the purpose of this discussion, it is useful to divide control systems into two categories[2]:

Logical control systems - these predominantly manipulate on/off or binary signals using logical rules together with timing and counting functions to provide combinational and sequential control. Programmable Logic Controllers (PLCs) with mainly digital I/O are the traditional technology for implementing low speed logical control systems. For high speed logical control, industrial computers with hard real-time capability are often found.

Quantitative control systems - these are concerned with controlling the actual value of plant or machine quantities and thus predominantly manipulate analogue signals. Quantitative control is often referred to as ***feedback control***, however this name is confusing since logical control also normally involves feedback. Quantitative control is traditionally implemented using dedicated hardware and/or software, such as found in proportional, integral derivative (PID) controllers, drive systems for motor control, numerical control (NC) and robotic systems etc.

A A Verwer is a Principal Lecturer in the Department of Mechanical Engineering, Design and Manufacture at Manchester Metropolitan University. (Email : A.Verwer@MMU.AC.UK)

Both logical and quantitative control systems are heavily reliant on computer technology and software solutions. However, control engineers rarely need to be involved with the details of the computer code which is used to implement quantitative control. Often standard control laws are used and where more advanced control is implemented (for example: predictive control, adaptive control and self-tuning) the details of the algorithms are hidden from the user. On the other hand, logical control systems are often engineered at fairly low level with the major engineering effort being directed towards code generation. Within most industries, the number of logical control loops far exceeds the number of quantitative control loops[3], consequently, far greater engineering effort and cost is tied up with implementing logical control than with quantitative control.

3. Teaching control theory or control technology?

Engineering education has always had to strike a balance between mathematical theory and engineering practice/technology. Too much emphasis on theory and we produce graduates who find it difficult to tackle or understand practical problems. Too much technology and we produce graduates who cannot analyse problems or synthesise solutions. Nowhere is this difference more apparent than in the field of control engineering.

Further education has traditionally been closely associated with engineering practice. Logical control has usually been a core part of instrumentation and control courses in this sector, PLC programming and applications being widely taught. In higher education, on the other hand, the teaching of control engineering tends to be centred on quantitative control. Here the approach is normally very mathematical, involving mathematical modelling, dynamic system analysis and control law synthesis. Practising control engineers, however, rarely use a Laplace transform or Nyquist diagram in anger, they are much more likely to be concerned with technical problems associated with control system implementation.

So why aren't PLC applications more widely taught at University? The following are possible reasons:

- PLC programming tends to be thought of as a "technician level" activity rather than "engineer level".
- PLCs are seen as being limited to logical control applications.
- Traditional PLC programming is very manufacturer specific and so is seen more as a "training" exercise rather than broad based "education".
- Absence of a theoretical (mathematical) basis means that the topic is thought of by academics as relatively undemanding.
- The mathematical theory associated with quantitative control can be taught predominantly in the classroom. PLC programming requires much greater hands-on activities and use of laboratory equipment with all the problems that are involved. Numbers of students are thus limited by equipment and staff availability.

For the teacher, traditional PLC programming language differences, their limited analogue capability and the obscure nature of so-called "advanced" programming features can cause major problems. These difficulties tend to encourage pathological solutions to problems, fragmented and unreadable code and a limited approach to problem solving. Perhaps more importantly, the details of program implementation can tend to obscure basic principles.

4. IEC (6)1131 Standardisation

Traditional PLC programming has been closely tied in to the vendors. Each PLC manufacturer has their own particular language variation. This situation tends to tie PLC users to a particular manufacturer because of past training, and experience. IEC (6)1131-3, "Programming Languages for Programmable Controllers", was issued in 1993 providing an open standard for PLC software. The IEC working group aimed to produce a comprehensive open standard which overcame many of the difficulties associated with traditional PLC programming. The working group did an excellent job and the resulting standard is applicable to a wide range of PLCs and industrial computer systems.
The relative advantages of using IEC (6)11131-3 are well documented[4,5] and include:

- Vendor independent programming - Users can more easily move between vendors, education and training can be non-manufacturer specific.
- Extended language capability - Better support for quantitative control, numerical manipulation and sequencing. Rich set of data types and attributes and strong type checking leading to more robust code.
- Structured programming facilities - making team working and software maintenance easier and leading to software reuse.
- Facilities for execution control - provides real-time capability, regular sampling rates, interrupt handling etc.
- Configuration facilities - provides support for a wide range of hardware including multi-processor and distributed systems, adaptable as system requirements change.

Initially, the industry was slow to adopt the standard, probably because of the large number of existing PLC systems and installations and because of the significant effort required to produce IEC (6)1131-3 compliant systems. Most PLC manufacturers are however adopting the standard and many end users are now insisting on compliant PLC systems. An increasing number of compatible programming packages are becoming available. Many of these packages are true compilers which produce native microprocessor code which will run on a range of hardware. Some packages are still linked to traditional PLC architectures and produce PLC manufacturer specific code. Full compliance with IEC (6)1131-3 is still scarce, however, programming software and hardware is rapidly moving towards this goal.

5. The impact on Education and Training

The adoption of IEC (6)1131-3 means that the teaching of PLC programming and applications no longer needs to be manufacturer specific. The techniques which are taught are widely applicable to a variety of hardware from traditional PLC systems to embedded microprocessor systems. Further, the standard provides support for all types of control system and control methodologies in current use.

The rich set of languages and hierarchical facilities incorporated in the standard provide support for the different types of control system and various design and implementation aspects from overall top level system organisation[6] and hardware configuration[4] to low level code generation. Combinational and sequential logical control systems are straightforward to implement as is the coding of quantitative feedback control.

Many well documented software engineering aspects are incorporated into the standard, modular structured programming, encapsulation, data hiding and data type checking. These allow a systematic approach to control system design and software development to be taught.

6. An approach to PLC software development

One of the problems in teaching programming for industrial control systems is how to give students exposure to a wide range of realistic applications with limited laboratory hardware. Connecting PLCs to real equipment in the laboratory is clearly the best way to provide such exposure. However the cost of providing multiple sets of PLCs and target rigs can be prohibitive and a large variety of equipment is required to show the range of combinational, sequential and quantitative control systems.

A technique developed at Manchester Metropolitan University is to use simulated processes on which students develop and test their control programs. The simulations are written by the tutor in one or more of the IEC 1131-3 languages to run on a compliant package on a PC. Animated graphical displays are also provided to add realism and interest to the simulation. The control programs are written by the student as separate IEC 1131-3 programs which initially interact with the simulation. Later the simulation code can be removed from the project and the control program applied to the real plant by connecting to physical I/O or downloading to real PLC hardware. In this way students can be exposed to a wider range of control problems than would otherwise be the case. A further advantage is that faults can be easily and safely introduced into the simulation to test the robustness of the control scheme.

The technique of writing a plant simulation also has major benefits for practising engineers working with real processes and plant. Typically the simulation will involve some modelling of the process dynamics which initially can be quite crude. The accuracy of the modelling can be improved as and when data becomes available from the real plant. The approach allows control programs to be developed and tested off-line. Reaction to faults and control system robustness can also be easily checked. Preliminary system testing can be done off-line and the simulation is invaluable for use in training engineers for maintenance and trouble shooting. The technique has been adopted by several companies whose engineers have attended courses at Manchester Metropolitan University.

References

1 Verwer, A. A.: “Teaching Open Standards for PLC Programming”, IEE Colloquium - Mechatronics in Education, Manchester, March 1996, Digest No: 1996/071.

2 Golten, J. W. and Verwer A. A.: “Control System Design and Simulation”, McGraw-Hill, London, 1991.

3 Frost and Sullivan: “World programmable logic controller markets: increasing functionality promises continued growth”, Code 5191-10, January 1995.

4 Lewis, R. W.: “Programming industrial control systems using IEC 1131-3”, IEE publications, 1995.

5 Bonfatti, F, et. al.: “IEC 1131-3 Programming Methodology”, CJ International, France, 1997

6. Golten, J. W. and Verwer A. A : “GEMMA - Graphic guide for top-level automation system design”, Manchester Metropolitan University internal document, January, 1997, a translation of “Guide d’Étude des Modes de Marches et d’Arrêts”, ADEPA, France

Printed and published by the IEE, Savoy Place, London WC2R 0BL, UK.