Schools in control? : questions about the development of the teaching of cybernetics

Martin Owen
School of Education, University of Wales

Abstract
The paper considers the role of he teaching of cybernetics and control technology in the National Curriculum (Technology) in England and Wales, and the limitations of its horizons. The scope of software currently available to schools is surveyed and the range of possibilities of this software is considered. Alternatives to current practice drawn from modern technologies which embody their own design methodologies are considered. These methods include object oriented methods, machine learning and neural computing. The fact that these systems are based on clear conceptual design methodologies should make them attractive to educators. The paper concludes that there is a need to develop new curriculum methods to accommodate newer methods.

Introduction

The study of Control Technology is a compulsory part of children's education in the United Kingdom. Following the 1988 Education Reform Act there has been an element of compulsion in what is taught in schools. In National Curriculum (Technology)[1]. There has been a degree of flexibility in content because the curriculum was specified by processes rather than knowledge. A Programme of Study (PoS) and an assessment system has defined what is needed to be known, and if we note page 40 of the PoS we see control is exemplified. Also one of the elements which was offered for practical assessment of technology at the end of Key Stage 3 (pupils aged between 13-14) was labelled "Control"[2].

In Attainment Target 5 of the National Curriculum, which deals with the aspects of Information Technology, *Measurement and Control* is delineated as one of five strands which is to be studied as part of the NC.

The draft revised standing orders for the National Curriculum (Technology)[3] have an even greater emphasis on control, with a major theme within the Programme of Study "Control & Energy". This programme mandates the study of control from Primary School through compulsory secondary education, and includes electronics and pneumatic/ fluidic systems. The authors accept that they did not complete work on the match between Attainment Target 5 (Information Technology) and the Design and Technology components. This may be particularly critical in considering control technology.

Control technology is thus considered important enough to be a compulsory subject for all pupils during their compulsory education.

My problem

It is a personal belief that this study of control in the education system in England and Wales, although laudable and innovatory in itself, is flawed. There is a risk that it will be badly taught because insufficient curriculum thinking has been invested into it. There has been no development in the subject from the 1970's neither in the real ways in which "control" is being developed as a technology nor into the conceptual basis of the processes of control which give the subject a meaningful intellectual identity rather than a collection of heuristics and examples. Neither is there any thought given to the progression and development of complexity of thinking about the subject.

I believe the science and technology of Cybernetics: the study of automatic communication and control, is culturally important for pupils, has an intellectual and conceptual underpinning and as a subject can be approached in a developmental and progressive way. I also feel that if we are teaching a technology, we should have good reasons.

Ausubel suggests that for a subject to be *learnable* it should be non-arbitrary and have logical meaningfulness. We should look to the subject to see whether clear conceptual patterns with implementable developmental paths exist. Most developments in cybernetics are based on the conceptualisation of the problems of machine communication and control, and therefore new methodologies inherently start from a fundamental conceptual basis.

I also believe that what we teach as technology should have some cultural validity. I once taught in a school where pupils made fire-side combination sets in their CDT (then Metalwork) classes. However all the pupils lived in a smokeless zone, and most lived on a housing complex with under floor electric heating. The technology we teach should either be seminal, historically important or have genuine contemporary value. It should not be arbitrary.

Current practice with hardware

There are three aspects to control in school: electronics, pneumatics/fluidics and computer based systems. I intend to say little about pneumatics. It is reliable, expensive, and has a direct correspondence to the ways that the systems are applied in the industrial context. There is a simple consistency in the process it presents and it has a clear conceptual structure. It is eminently learnable.

Electronics teaching is variable. It can, at one level, be a process of simple (or complex) materials handling in assembling components of formula circuits. This is typically exemplified by the "Control" task for KS3 in 1993 where students assemble a flashing device. At other levels there are well conceived modular, systems approaches using prefabricated system elements which fit reliably together to teach basic systems understanding in a problem solving context[4] . At an intermediate stage there is electronics teaching which relies on increasingly ingenious use of the 555 Timer circuit.

All the above concentrate on manufacture and design. Fault finding and repair, which are more common activities for individuals with electronics skills, are not elements of most teaching.

There can be no doubt that understanding of electronics at a systems level can be the only intellectually honest way forward. Fabrication and construction techniques are tractable activities in the school workshop, however in most cases the only relevance of these skills is as a hobby interest and bear no relationship to what happens in SONY or Thorn-EMI. The application of these skills that do arise in the context of repair (an important technological process) does not form part of the curriculum.

That electronics is a developing technology can not be denied. Whether National Curriculum (Technology) electronics is sufficiently flexible enough, and is treated with sufficient conceptual abstraction to reflect those changes is yet to be established.

Computer control

My main thesis concerns the impoverished provision in the curriculum of computer based control. This in turn reflects directly on the teaching of electronics. Recent practice in contemporary control circuitry includes the use of embedded programmable control integrated circuits. These are inexpensive devices that operate entirely on the control system designed in computer software. Programs designed on computer are loaded into the silicon chip. Thus single devices, carrying messages written on the computer constitute sophisticated control systems. Thus the notion of teaching electronics control systems, crafted out of sub-components is a vanishing methodology. Designing and even building your circuit in software is a current vision of the future of control systems. The components are less expensive to fabricate into devices and the development time is quicker.

Current school approaches to computer control.

There are three approaches in the UK for schools based on the materials commercially available for typical school microcomputers. These are broadly a subset of an imperative style procedural language like Pascal, a variant on the BASIC or Fortran language represented as a flow diagram, and a simple time-line approach. These are essentially sequencing activities.

The procedural language approach is the most common. It is exemplified by a number of systems from a variety of suppliers such as SMART form Economatics. CoCo from Commotion, Control LOGO from Longman-Logotron, Contact 2000 from NCET/Educational Electronics.

All feature the ability to write procedures in an interpreted environment of the type:

```
To React-to-light-for-5-secs
IF INPUT1 = TRUE TURNON Output1
WAIT 5
TURNOFF Output1
END
```

a procedure to sense an input and trigger an output for 5 secs if the input is true.

In the more sophisticated options there is sensitivity to analogue inputs and pulsed digital outputs so we might have:

```
To-react-differently-to-light
MAKE "Scale-factor 7
MAKE "Lightval INPUT1
PULSE OUTPUT1 :Lightval * Scale-factor
END
```

A programme which assigns values to two variables, a scale factor, and a value for light intensity. These values are then used to calculate the magnitude of pulses driving a *device at the output.*

These allow the construction of typical feedforward and feedback control systems. There are many good examples of this kind of work being carried out in primary schools, however there is little progression beyond this type of procedure at the end of compulsory schooling. Where should the progression and development of ideas be?

There is also a belief that some of these systems are LOGO-like (for Control LOGO it is obviously true). This is an important claim, for if the control environments were truly LOGO like, their combination with other LOGO programmes/ microworlds would allow for many of the potentially interesting activities suggested below. The systems borrow some keywords from LOGO like the use of MAKE for the variable assignment statement, however in truth the systems are closer to BASIC for they do not support the sorts of data structures and their manipulation primitives which occur in LOGO which allow for Artificial Intelligence style of programming.

"Control by Design"™, an Economatics product allows for similar programming to the systems described above, however the system differs in that programs are constructed by drawing the conventional Fortran style flow charts on the screen. This is an interesting innovation in that the step by step sequencing is seen in an iconic form. However the use of this flow charting methodology in modern computer program design has long since passed out of use in favour of more structured methods.

LEGOLines™, from LEGO allows switching on and off, referral to digital input signals, and time delays along a matrix of commands in which each successive row is processed in turn. It accomplishes the same type of tasks as the simpler implementations of the above software, however less typing is needed as keywords appear on pressing specific function keys.

All the above are variants on a single vision of the problem of the control of automatic systems. In each case it is a sequencing activity, often a feedforward system, with the behaviour of the system being entirely predictable. Some conditional branching is included to vary the sequence, however the behaviour of the system can only be described as a sequence. This is an important way of thinking about control problems. It is still widely used, it is learnable by primary school pupils. However, it is not the only way of thinking about cybernetic problems. Technology has moved on and has opened up new ways of thinking about and designing control systems. Should the National Curriculum, and school technology reflect these other systems as well?

What might be alternatives?

There are certain key ideas which out to be part of any study of control technology. These include: data abstraction; virtual systems; object oriented programming; machine learning; and neural networks.

Data Abstraction is an important idea in computer science. If we are trying to develop technological literacy, I think this concept would be on my list. Essentially data abstraction refers to the idea that you can assume a lot of your or other computer scientists' previous thinking. After all, if the computer is so smart why can't it figure out what to do for itself. If I order meal in a restaurant I do not have to issue explicit instructions to the waiter to get to the kitchen to instruct the chef... to instruct the food wholesaler... to instruct the delivery man... to instruct the farmer... to instruct the seed merchant... I assume all that is going on in the background.

Data Abstraction is a methodology that enables us to isolate how a compound data object is used from the details of how it is constructed from more primitive data objects[5] .

A further break with the forms of computer programming which currently influence how we approach control in schools is that the methods described, and particularly where data abstraction plays a part, is there is an increasing blurring between what was "data" in a computer program and what is a " procedure".

Increasingly the process of engaging in control is a process of producing a model of what is to be controlled as a computer model. We produce "virtual systems" which have all the characteristics of the real system but are behind the glass screen of the computer. The real major change in thinking is that the virtual system can be in one to one correspondence with the real system. The embedded controller circuits described above can have the same program which controlled the virtual environment. The process between simulation, and manufacture of a hardware control system is almost seamless. This should raise serious questions about why are we teaching some topics.

A currently available, but costly, virtual development system is LabView from National Instruments. The design process in this system is interesting. Design and learning are processes of defining system function in these systems. One starts by designing the functionality of the system by devising the front panel of a virtual instrument: where the knobs, switches, dials and meters will fit, and labelling their function and giving them style. The front panel of this instrument having been designed is then opened up and the components are then “wired” together using a store of functional elements such as comparators or amplifiers or differentiators. The control functions of conventional languages like iteration, loops and branching are also incorporated iconicly. When connected to the appropriate outside world, the device ceases to be a virtual simulation, but becomes the genuine working application.

Object Oriented Methods are of increasing importance in computer science. Here the model of design is that of communicating objects inside a computer. Design is a process of modifying existing objects, juxtaposing objects or describing the behaviour of objects when they communicate with each other. So that if you already have a computer object which represents a ball with size, roundness and bounciness and so on, to have a red ball one need not program the “ball” from computer primitives but merely clone the existing ball and make it red.

Many users of modern Paint programmes on graphical computers will already be familiar with the concepts and take them for granted. In such a program the following object oriented activities may take place:

The *pallet* object may send a message to the *paint pot* object (the flood fill tool) to take the property redness. The *paint-pot* tool may then send a message to the instance of the *rectangle* object to become filled.

It takes little imagination to see how such message passing amongst communicating objects can simulate real situations and consequently act as control systems for them.

The notion that such a programming system would be useful for children was perceived fairly early in the history of computing, and Goldberg(1980)[6] and Kay developed the computing environment SmallTalk for that purpose. Their system has perhaps proved too powerful, however the software features we take for granted in our most powerful desk top computers were developed as part of the project. Reenskaug(1980)[7] demonstrates such systems have application in control, as they describe systems as users perceive them.

The ideas of objects or actors following scripts to describe their varying behaviours is a powerful way of describing system behaviour, and can allow for much more complexity than sequential programming. Whalley(1992)[8] has developed environments for children in which motorised LEGO constructions are controlled by representations of the system on screen. The programmer is presented with questions like:

> “What do you want the train to do to do when the signal is down?”

Responses a train object may make to signal objects might be speed up, slow down, continue as at present, start, stop or toot. In time honoured ways down signals presumably mean stop. The previous work of other programmers will have abstracted stop to sending a “0” to the appropriate motor control using some deep rhunes of computer science. That feature, having been programmed once and fully tested, can be taken for granted. What is important for so many reasons involving clarity and thought about the real activity being controlled is that the control activity is being defined at a level close to reality and thus better understood.

Whalley has achieved some success with these actor based environments. The systems are both virtual and object oriented. It opens a lead for interesting ways forward.

There are approaches which have developed from artificial intelligence and robotics. Declarative programming, as found in the language PROLOG and in Expert System software tools, provides a means of designing a system by declaring the constraints and limits of a system and leaving the rest to data abstraction. The fundamental idea behind such expert system approaches is that the you discover and then employ rules the best human operator would apply to governing a situation. A program may have the following kinds of statements:

Maintain energy efficiency if safety is positive.
A system is energy efficient if the rate of fuel consumption is optimal.
A system is energy efficient if the temperature gradients are optimal.
[(coolanttemp, safe):- (Less (coolanttemp, 40°))]

The real style may not be quite as natural language[9], however the thinking processes which are called upon are expressions of real understandings of the functioning of systems.

Other current thinking in robotic systems revolve around machine learning. Here there are two systems which ought to be of interest. The notion of genetic algorithms[10], computer algorithms that find solutions to problems, and then try to optimise those solutions by mutating themselves and comparing performance with the parent... with the survival of the fittest, has potential across a whole range of design activities. The ideal control system for a snooker playing robot should start with the rules of snooker, the ability to move around the table, and vary the cue aiming and force application. The robot should then *learn* about spin and side and the behaviour of the cushion to become an expert snooker player by observing and modifying its behaviour by experience!

Finally, a major focus of thinking about computer systems in this milieu is the idea of neural computing. This is particularly important when one is concerned with control through ideas like computer vision. Ritter *et. al* (1991)[11] describe how a robot arm can be taught to juggle using these strategies. In neural computing the essential element is one of confirming/rejecting a pattern representation of the phenomena with which the computational system is interacting. This corresponds to the hypothesis that in the brain the way we learn, and thus the way we are able to control arises form connections between neurons being made or deactivated.

Programming a computer in neural terms is exactly analogous to a connectionist view of human learning, of concepts and rules being built up by experience of good and bad examples of concepts and rules with provision made for increasing fuzziness in thinking when needed. This technology is the major thrust in developing machine senses and learning. Vision, and auditory powers have obviously massive implications for how machines will automatically control processes in future.

It should be possible to develop educational software which allows experience of these methods.

Implications for Education

There is a desire on the part of the drafters of the National Curriculum to teach all pupils about cybernetics. I have outlined my unease with current practice based on a survey of what is available as cybernetics in schools and what is happening in the real world of control. I believe that there is a need for an important rethink about cybernetics as it is taught in school. I have known for some time that there are intellectually honest ways of teaching about some of the ideas expressed here[12]. Educational software for teaching ideas about neural programming are already available. This is directed at undergraduates, however so was BASIC when it was introduced. Whalley has demonstrated that actor based systems can be effective. I have a number of guesses why there has been little development of the ideas of control in the curriculum, with inertia being the principal hypothesis. However there is a case which needs to be answered, and a need for further development.

References

1. DFE/Welsh Office, *Technology in the National Curriculum*, March 1990, HMSO

2. DFE/Welsh Office, *KS3 School Assessment Folder : Design & Technology*, 1993, HMSO

3. DFE/Welsh Office, *Technology for ages 5 to 16 (December 1992)*, 1993, HMSO

4. See for instance *Alphaboard* or *MFA* both produced by Unilab Ltd., Blackburn, England

5. Abelston H., Sussman GJ. and Sussman J., *Structure and Interpretation of Computer Programs*, MIT Press, Cambridge Mass, 1985 p74

6. Goldberg, A. and Ross, J., *Is SmallTalk80 System for Children?*, Byte, August 1980, pp34-38

7. Reenskaug, T., *User Oriented Descriptions of SmallTalk Systems*, Byte, August 1980, pp148-166

8. Whalley, P., Making Control Technology Work in the Classroom, *British Journal of Education Technology*, September 1992, Vol. 23, no. 3.

9. The last statement is in typical PROLOG syntax

10. See Goldberg, D., *Genetic Algorithms*, Addison Wesley, Cambridge, Mass, 1989

11. Ritter H., Martinez T., and Schulten K., *Neural Networks*, Addison Wesley, Cambridge, Mass, 1991

12. Owen and Sheath, *Teaching Robotics in the Middle School*, paper to AI-ED '81, Exeter, 1981