WORKING NOTE AC86

Alan Creak
14 January 1994

GOALS FOR ROBOTICS AND REAL-TIME CONTROL

This paper is a summary of the areas which I would like to cover in the 473 course, and for which I would like supporting equipment in the laboratory. It isn't very systematically composed, but I think it's a fair representation of my intentions. The equipment is listed to show that it's there (or, if in italic, would be nice) – most of the equipment isn't tied to any specific experiment, and gets used for other purposes too.

Topic	Equipment
Continuous control	Analogue computers; shaft encoders.
Sequential control	Train, *Programmable logic controller*.
Distributed control	Macintosh, BBCs, Acorns, *Single board computers*.
Communications	*Standard bus (STEbus ?)*
Manufacturing workcells	Conveyor belt, robot, etc.
Sensors	Infrared and ultrasonic transmitters and receivers.
Vision	*Television camera (CCD, probably).*
Control interface	Joystick.
Construction	Tools, scope, meters, power supplies, ... , *more*.
Stationary robots	Mentor robots.
Mobile robots	Stiquitos, turtle, *new turtle*.
Real-time operating systems	*A real-time operating system*.

NOTES ON NEW THINGS.

(Costs, where given, are approximate.)

Programmable logic controller (PLC).

These are widely used for industrial control systems where sequential control is important. They were originally just what they say – programmable boxes which replaced hard-wired relay logic. Now they commonly also include ways of handling analogue signals too, but the principle is the same. A real one would be nice; I'd use it both with the train and with the robot workcell.

Cost : ?

Single board computers (SBCs).

SBCs are commonly used for local control of continuous or sequential systems in a hierarchic system. Programming and development is typically carried out on a conventional microcomputer, and the resulting code copied to RAM on the SBC, or converted into ROM. We've used the BBCs in the past (usually between a Macintosh and some machine such as the turtle); they do the right thing, but they're clumsy and have far more bits and pieces than we need. They're also too expensive to get in quantity.

I'd been thinking of some things called Miniboards which are not very expensive and fairly widely used, but if we choose to work with STEbus (see below) it would be better to get something more compatible therewith.

Cost : ?

Standard bus.

I'd always hoped that 473 would accumulate its own collection of interesting things which future generations of students would build on to make even more interesting things, but it's never happened. One reason has been that every design has been individual, and connecting things together has therefore been a formidable task.

We might be able to get round this if we based all our construction (and the communications used by the existing machinery) on some standard communications bus. Paul Q has been investigating such things, and it's clear that STEbus would be very appropriate for our environment. This would also give me a real example of a bus to talk about when I discuss communications, distributed systems, and so on.

Cost : $2000

Television camera.

Vision is an important component of many control systems. For a little while in the early days of 473 we were able to borrow a small camera from the mechanical engineers, but we accidentally burnt out the vision chip and I haven't been able to get a replacement. While it lasted, though, it was very good, and a replacement of some sort – preferably more robust – would restore a very useful opportunity for experiments.

Cost : $2000

More.

For constructing bits of hardware, we've been buying components as required and sticking them on boards. This is good for students who are experts, but we don't get many such. A different approach is available if we move to using STEbus a lot : you can get development cards for the bus which already have power supplies etc. in place, and that would be a lot easier. It would also help to solve the compatibility problem, which I mentioned earlier.

Cost :

New turtle.

Well, it would be nice, but it isn't a priority. The old one still works, despite many generations of alterations and rebuilding, but it's expensive on power, so isn't really a good base for an autonomous machine. (We've tried with various sorts of battery.) So if I see something turtlish but lighter and self-contained I might be interested. I've considered radio-controlled cars, which are the right sort of thing, but short of sensors and not too good at fine control. (No, I don't really need the pen, the eyes, or the horn !)

Cost : dunno yet.

A real-time operating system.

So far, we haven't had anything of the sort, and students have written bits of code to do their own scheduling where it's been needed. That's been easy enough (and educative) because the assignments have been tiny. It's also been not unreasonable, because we haven't had anything sufficiently ambitious to demonstrate proper operting systems.

It would be good to be able to do a bit more, and real real-time systems are developing now. We could write one, but it would probably take for ever, and I have an ulterior motive (Paul Q) in wanting a fairly ambitious one now.

Cost : Sourcetask + Fastcycle-88 $2000 (using Peter F's 386 PC, which could be a bit of a nuisance); $4000 (using a 486 4Mbyte processor board for the A5000 - preferred).