WORKING NOTE AC52

Alan Creak
21 May 1986

ROBOT DYNAMICS AND THE DIGITAL DIFFERENTIAL ANALYSER.

Here are some well known facts, which may or may not be true :

The calculations needed to handle the dynamics of robots are hard [4].
It is important to complete these calculations quickly [4].
Iterative methods are generally too slow to be of use [1].

The complexity of the calculations and the need for speed of execution suggest that systems using several processors ought to be useful. In fact, such systems have been used for exactly this purpose [4, 5].

The focus of attention seems to have been on rather conventional computer hardware and software for these jobs. The object of this note is to suggest an alternative which may be worthy of attention.

The equations to be solved are ordinary differential equations in terms of the various joint coordinates of the robot, with time as the independent variable. They are complicated non-linear equations (whence the difficulty of solution), and they are customarily expressed in some form of matrix notation, which perhaps makes them look even harder than they really are.

Analogue computers are well adapted to solving large sets of simultaneous linear equations, though conventional analogue machines do not perform startlingly well on very large sets of equations, nor with equations where non-linearity is marked. They do, on the other hand, provide the multiprocessing which seemed to be a good idea.

The difficulties of using analogue computers with large numbers of equations and with non-linearity are nowhere near so significant if the same problem is solved with a digital differential analyser [3]. These machines combine the topology, and the multiprocessing, of the analogue computer with the precision and flexibility of digital data representation and operations. The digital signals used are also well adapted to ready interfacing with other digital equipment, so configurations and values can be calculated on conventional computers and loaded into the digital differential analyser as required.

It seems to me, therefore, that an interesting thing to investigate would be something equivalent to a digital differential analyser set up to solve the appropriate equations of motion of a robot and to calculate the control signals required to constrain the robot to follow a predefined path.

The digital differential analyser solution – if it *is* a solution – is based on iterative techniques. (So is a "real" analogue computer method, for that matter – in fact, so is the robot itself.) The proposed method must still be shown to be fast enough to be useful. Of course, "fast enough" here is partly a matter of implementation. A digital differential analyser simulated on a conventional sequential machine might be fast enough to control a robot snail, but there are other possibilities :

- Implement the machine on a network of coupled microprocessors. In its simplest form, this model would use one microprocessor to simulate each unit of the digital differential analyser; but, as a microprocessor is much more elaborate than is needed for this purpose it might be possible to model several units of the analyser on each microprocessor. Again, timing considerations will probably loom large in determining whether or not this is feasible.

  I would want to look closely at "Transputers" [6] for this function. They are designed from the outset with an eye on interconnection into massively parallel systems, and the associated language occam [7] is explicitly designed to describe sets of simple intercommunicating processes – which exactly matches the problem in hand.

- Implement the machine as an integrated circuit. The digital differential analyser is constructed of very simple units; they seem very well suited for fabrication in quantity by techniques such as large scale integration. To make a general-purpose programmable digital differential analyser by such means is not necessarily as easy as it sounds, though, if only because of the problem of interconnecting the units according to changing requirements from time to time; but a "fairly good" topology incorporating "enough" switched links could perhaps be devised. To make special integrated circuits implementing a digital differential analyser for a specific system seems to me – speaking from a position of profound ignorance – unlikely to be extraordinarily difficult.

Whether it works or not, it's a different approach, and is perhaps worth pursuing to some extent simply to achieve a different perspective on the problems. It largely eliminates one unwelcome characteristic of conventional methods : the scheduling problem [(4)]. It provides an algorithm which translates the differential equations into circuit diagrams which, in effect, schedule themselves. In addition, the method is based on a fairly direct mathematical representation of the robot structure; as such, it is tempting to speculate that it might be more able to cope with small differences between robots which at present make robot programmes essentially untransportable [(8)].

I don't pretend that this is some sort of magic wand, which will eliminate all known problems of controlling robots. There are still plenty of problems to be solved – here are some examples :

- Will it work ? Can it be made fast enough to solve the actual equations sufficiently precisely in the time available ?
- What sort of communications do you need ? How do you set up the initial values ? How do you incorporate the planned motions ? How do you cater for errors ?
- What are the merits of different strategies for implementing the processor ? How flexible must it be ? How adaptable can it be made ?

Finally, it's interesting to observe that the idea of using DDAs to control the motion of a machine has a very long history : Appendix B of Sizer's book [(3)] considers using a digital differential analyser to construct a path for a numerically controlled machine tool by interpolating between a few given points.

REFERENCES.

(1) J.J.Craig : *Introduction to robotics : mechanics and control* (Addison-Wesley, 1986) p127 (though others [(2)] disagree.)

(2) A.A. Goldenberg, B. Benhabib, R.G. Fenton : *A complete generalized solution to the inverse kinematics of robots*, IEEE J Robotics and Automation **1**, 14 (1985).

(3) T.R.H. Sizer (Ed) : *The Digital Differential Analyser* (Chapman and Hall, 1968).

(4) H. Kasahara, S. Narita : *Parallel processing of robot-arm control computations on a multimicroprocessor system*, IEEE J Robotics and Automation **1**, 104 (1985).

(5) G. Gini, M. Gini, R. Gini, D. Giuse : *A multi-task system for robot programming*, Sigplan Notices **14 #5**, 11 (May 1979).

(6) M. Stevens : *Transputers : Lego for fifth generation machines*, Computer Bulletin Series 2, **#42**, 21 (December 1984).

(7) D. May : *Occam*, Sigplan Notices **18 #4**, 69 (April 1983); D. May : *Occam*, Computer Bulletin Series 3 **1 #1**, 14 (March 1985).

(8) B.I. Soroka : *What can't robot languages do ?* Proceedings of the 13th international symposium on industrial robots, 1983, p12-2.