THE IMPACT OF MICROPROCESSORS ON CONTINUOUS SIMULATION

Everett L. Johnson

Wichita State University

ABSTRACT

With the advent of low cost microcomputer hardware the possibility now exists for the development of systems of parallel microprocessors. For simulation purposes the system of processors may take a form similar to that of an analog or hybrid computer with one processor performing an integration, another a multiple variable summation, etc. This paper presents a possible configuration for such a system and discusses briefly some problems to be resolved as well as some suggestions for multiprocessor integrations.

MICROPROCESSOR SIMULATION SYSTEM

The system proposed here is an all digital system consisting of a master or host computer, probably a mini-computer, which controls and communicates with a set of microprocessors. Each microprocessor has the ability to accept data from the host computer or from any other microprocessor. Integration routines are loaded into processors assigned as integrators, function generators, etc. Control words are also input to each processor to inform it as to what inputs to expect and where its outputs are to be routed.

The realization of such a system would allow structuring an analog-like computing system while eliminating some of the undesirable aspects of analog computers such as patching, scaling, pot setting and static checks.

Various analog functions have been shifted from the analog computer to the digital computer in many hybrid computer simulations. Digital computer speed has been the limiting factor. The availability of inexpensive multiprocessor systems will require a new look at each of the analog functions and its implementation digitally.

A. INTEGRATION

The trapezoidal integration algorithm, Eq. (1), is a simple algorithm which may be used to demonstrate a principle applicable to any algorithm. A computational tree for Eq. (1):

$$y_{n+1} = y_n + \frac{3T}{2} f_n - \frac{T}{2} f_{n-1} \qquad (1)$$

indicates that parallel implementation requires two processors. The parallel tasks of each processor may be controlled by a common stored program with each instruction to be executed being supplied to the instruction register of both processors.

Using the Intel 8008 8 bit processor and double length words the integration time T_i for the trapezoidal algorithm should be less than 10 milliseconds and occupy less than 256 8 bit words or 1 1024 bit random access memory. An input rate of 100 samples/sec could be accommodated allowing frequency components of around 10 Hz. Faster processors will soon be available which, along with new parallel processor integration techniques, will increase the integrator bandwidth.

For a given processor speed and algorithm an integration requires T_i seconds. Faster integrating speeds may be obtained by trading lag and additional processors for increased integrator speed. Consider a system of m multiprocessor digital integrators each requiring T_i seconds to perform an integration. The inputs are from an A/D converter with sampling interval T_i/m. On a cyclic basis the first sample goes to integrator 1, the second to integrator 2, etc. The output of all integrators are summed to form the result. For example consider a two integrator system. Each multiprocessor integrator can perform the integrating algorithm in T_i seconds. The A/D converter has a sampling interval, T, of $T_i/2$ seconds. New values are output from the combined integrator at every T seconds. For the sake of simplicity consider the trapezoidal integration algorithm. The outputs, y_i, of each integrator are:

$$y_{o,n+1} = y_{i,n} + \frac{3T}{2} f_{n-1} - \frac{T}{2} f_{n-2} \qquad (2)$$

n odd

$$y_{1,n+1} = y_{o,n} + \frac{3T}{2} f_{n-1} - \frac{T}{2} f_{n-2} \qquad (3)$$

n even

$$f(t) = 0 \text{ for } t < 0$$

The total output for all n is given by:

$$y_{n+1} = y_n + \frac{3T}{2} f_{n-1} - \frac{T}{2} f_{n-2} \qquad (4)$$

The z transform of the single integrator trapezoidal algorithm, Eq. (1), is:

$$\frac{Y(z)}{F(z)} = \frac{T}{2z} \frac{(3z-1)}{(z-1)} \qquad (5)$$

The z transform for the multiple integrator output, Eq. (4), is:

$$\frac{Y(z)}{F(z)} = \frac{T}{2z^2} \frac{(3z-1)}{(z-1)} \qquad (6)$$

or the same as Eq. (5), but delayed by T seconds. Recall that T in Eq. (6) is ½ the single integrator integration time, T_i.

In general, any number of integrators could be connected in parallel with a proportionate increase in integrator bandwidth. If the number of integrators is m then the A/D sampling interval is T_i/m and the transport delay introduced is:

$$\frac{(m-1)T_i}{m} .$$

The task for the kth processor is:

$$y_{k,n+1} = y_{k-1,n} + \frac{3T}{2} f_{n-m+1} - \frac{T}{2} f_{n-m}$$

$$n=k,\ k+m,\ k+2m,\ k+3m\ \ldots . \qquad (7)$$

Increased bandwidth has been attained but at the cost of possibly requiring compensation for the transport delay if the integrator is used in a closed loop. For open loop integrations, high bandwidths could be achieved by adding additional processors. All processors could be controlled with common memory. The trapezoidal integration algorithm used as an example here has poor amplitude characteristics and only fair phase characteristics. Higher order integration algorithms such as a fourth order Runge-Kutta should be considered. Parallel processors could significantly cut down on the Runge-Kutta algorithm execution time.

B. SUMMATION

Software is easily developed for digital summation and coefficient multiplication and though simple it has been an effective tool in the hybrid lab for reducing the number of amplifiers required for force and moment summations in aircraft simulations.

C. RESOLVERS

In aerospace simulations coordinate transformations are used extensively. The simple trigonometric functions can easily be programmed on a microprocessor. For rapid computations, table look-up, and interpolation formulas would be used.

D. FUNCTION GENERATORS

The digital portion of a hybrid computing system has probably contributed as much in the area of multi-variable function generation as in any other. Multi-variable function generation lends itself to speed improvements by use of multiprocessor systems since several tasks must be performed prior to arriving at the proper output given the input variable list. Many of these tasks could proceed in parallel.

Each analog-like device described above could be controlled or partially controlled from common memory, i.e. all integrators, all summers, etc.

All digital multiprocessing continuous system simulators exist today but in the form of large expensive digital systems such as those at NASA-Ames and McDonnell Douglas. For many small simulation labs the possibility of obtaining such systems is nonexistent. However with the advent of microprocessors the possibility of developing small special-purpose multiprocessing digital simulators is very real!