A High-Accuracy, Real-Time Digital Computer for Use in Continuous Control Systems*

W. J. MILAN-KAMSKI†

IT HAS become evident during the last few years that the accuracy requirements of analog computers have become too difficult to be easily satisfied. The rising pressure to achieve better computational accuracy has led to significant improvements in the computational techniques used in analog computers. These new improvements have made it possible to achieve a high degree of precision so that a 0.1 per cent accuracy has gradually become a realistic figure in many analog machines.

However, present-day analog computer technology is completely helpless if accuracy requirements approach the magnitude of 1 part per million, or 0.0001 per cent. The only available computers which can achieve this degree of accuracy are obviously digitial computers.

Many attempts have been made to design digital computers so that they might be used as direct replacements for analog computers. However, a rather unexpected difficulty has arisen. Digital computers, which have received a great deal of publicity as being the fastest computational tools, are extremely slow when compared to analog computers. Since the comparison is made between digital and analog computers, the operation of the digital computer must be such as to satisfy the bandwidth requirements of the analog computer. By this equivalence, the bandwidth of a digital computer can be defined as the bandwidth of an equivalent analog computer.

There are three distinct approaches in solving the problem of designing high-accuracy, real-time digital computers. All three of these approaches are directed toward building high-accuracy digital computers which can replace analog computers in applications where accuracy requirements exceed present capabilities of these machines.

At least one approach has come from engineers whose experience and background have been chiefly in the field of analog computers. Their basic approach was to replace various analog computer elements by equivalent digital operational blocks. For example, an integrator which consists of a motor with appropriate velocity control can be replaced by a reversible counter; a potentiometric multiplier can be replaced by a digital element which is called a rate multiplier, and so on.

* Presently being developed under a subcontract from the Military Products Dept. of Detroit Controls, in Norwood, Mass., for the U. S. Navy.

† Epsco, Inc., Boston, Mass.

Since the operation of a computer of this type is incremental, its design approach led to the development of a family of computers called incremental digital computers.

The second approach was to translate the problem into a differential equation and then to solve the differential equations by integration. Since the solution of differential equations is done using finite increments, the family of digital differential analyzers is closely related to the family of incremental computers. The output function of incremental computers and of the digital differential analyzers is determined by the increment of the input function and by the internal state of the machine. These computers, therefore, can be regarded as deterministic transducers with infinite memory.

The third family of real-time digital computers is represented by machines which go through a complete computational cycle every time a new input sample is taken. These computers normally adopt computational techniques which have been developed in programming general-purpose digital computers.

These machines normally have short memories or, in many cases, no memory at all. Their output is always uniquely determined by the input.

The latter group of computers is particularly suited to applications in which a number of problems must be solved simultaneously and concurrently. It is achieved usually by interleaving several programs.

The computational speed of digital computers is usually defined as the number of additions or multiplications which the computer can perform within a certain period of time. This computational speed is extremely high when compared to the computational speed of a desk calculator. In real-time computation, however, the speed of operation is defined as the ability of the computer to generate output functions, which vary rapidly with time. Not only must the output function contain large values of higher order derivatives, but also must not be delayed by the finite computational time of the computer. The transfer function of real-time computers is often complicated and usually contains trigonometric functions. If a high degree of accuracy is desired, the word length required may be as large as 30 binary digits or more.

It is possible to show that a high-accuracy machine has a limited ability to generate output functions which contain large values of output function derivatives. The computational time increases very rapidly as the word length increases.

The design of a real-time digital computer is usually based on an input-output accuracy specification and on the bandwidth requirements. For a digital computer, the bandwidth requirement can usually be expressed in terms of the amplitudes of output function derivatives. Maximum possible values of the derivatives can normally be determined by analyzing the geometry and the dynamic character of the output function.

The first trial in the determination of the maximum permissible computational time can be accomplished by first calculating the greatest possible velocity of the output function, and then by selecting a computational time such that the change of the output function within the computational time will not be greater than a maximum permissible error.

Errors due to quantization, truncation, round-off, function approximation, etc., must be considered separately as additional system errors. In certain problems, the computational time calculated from the investigation of the maximum output velocity may be extremely short. Extremely short computational times can be realized with incremental computers. However, internal rates of several megacycles are necessary in order to construct incremental machines which have equivalent bandwidths equal to the bandwidth of analog computers and accuracy of 1 part in 10,000.

In many applications, long computer memory is undesirable as, for example, in all real-time control and stabilization computers. Computer response to step inputs in target tracking applications must be excellent. Errors must be self-correcting, and the accuracy of the computer must be independent of the accuracy of previous computations.

These requirements cannot be readily satisfied by purely incremental computers. The selection of a certain type of real-time computer should be based on the specific requirements of each problem.

The best results can be achieved if the design of real-time computers is specially tailored to each problem. The specification for a real-time computer is usually determined by accuracy requirements and the characteristics of the time function to be controlled.

There are usually several other factors which are normally well specified; for example, the weight and size of the computer and the type of hardware to be used. These requirements, combined with the environmental specification, usually determine the maximum practical internal rate of the machine.

Several design parameters must be considered to determine the optimum combination of computer accuracy, internal speed of operation, approximations used, sampling rate, and the time of computation.

The maximum permissible computational time can be determined by analyzing the nature of the output function. The output function can always be expressed in terms of a Taylor series. The actual mathematical manipulation can be quite involved. It may also be difficult to determine the maximum possible values of all the derivatives of the output function. However, if the motion of a physical object is considered, it is usually sufficient to analyze only the first two or three derivatives in order to describe adequately the output function. Rapid changes in acceleration are very rare, and, therefore, higher order terms of the expansion can be disregarded.

The Taylor expansion can be regarded as a polynomial in t. It is possible then to substitute a polynomial for the output function. The period of time in which the polynomial substitution is valid can be determined by calculating the difference between the polynomial approximation and the output function. The difference must be less than the maximum permissible error. The higher the order of the polynomial used, the longer the period of time over which the substitution is valid. The computational time can then be determined by the time it takes the output function to diverge by a certain predetermined amount from the polynomial approximation.

The minimum sampling rate and the maximum computation time can then be determined for each order of the polynomial used as the output approximation. Computation times are progressively greater as the order of the polynomial increases.

The determination of the coefficients of the polynomial require the determination of the appropriate derivatives of the output function. The polynomial coefficients can be calculated on the basis of several samples computed at given time intervals. Using Newton's backward interpolation formula, it is possible to determine the coefficients of the polynomial by simply calculating the differences on the basis of several samples of the output function. (See Fig. 1.)

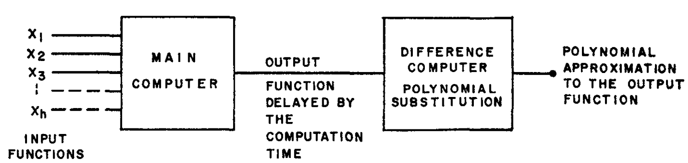

Fig. 1—Computer block diagram. Output function is compensated for the computational delay by means of a polynomial substitution.

The computation of function differences involves subtraction. Since random errors are not correlated, they are not subject to cancellation. In systems in which random and bias errors are of the same magnitude, a second order polynomial is probably the highest order which can be practically used. The computation of the terms of the polynomial makes it necessary to memorize the results of several computations. In other words, it is impossible to construct a computer which uses polynomial approximation and has no memory. However, the memory is relatively short. If a second order polynomial is used, the computer memory is equal to only three computation cycles.

The use of a polynomial approximation to the output function offers an added advantage which may be important in certain applications. The output function can be generated in steps which are smaller than the maximum permissible errors. The need for this form of output may arise if a high performance servo is controlled by the output of the computer. It can be seen from Fig. 2 that the actual output of the computer consists of a sequence of polynomial segments and that there is a discontinuous jump from a polynomial to the polynomial whose terms have just been calculated. This discontinuity can be made as small as desired. The reduction of the output function steps, however, can be achieved only at the expense of the computational time. It is possible, then, to trade computation speed for accuracy and vice versa.

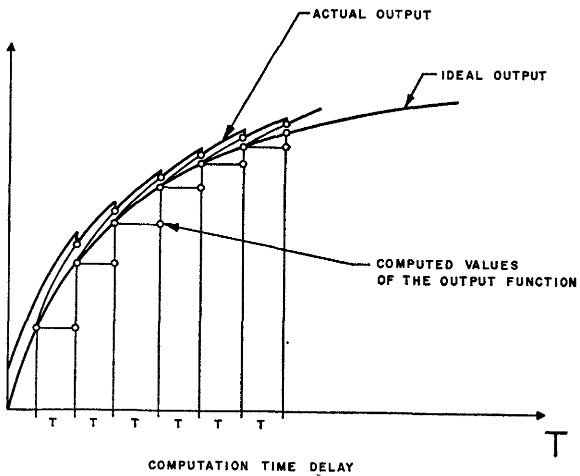

Fig. 2—Computer output function. Output function is approximated by polynomial substitution.

In between the computational times, the output function is not directly controlled by the input functions. However, the nature of the ouput function is such that it cannot possibly diverge from the approximated value by more than a certain predetermined value. This maximum deviation can be calculated by taking the terms of the Taylor expansion of the output function which do not appear in the polynomial approximation.

Once the sampling rate and the order of the polynomial approximation of the output function is determined, it is possible to determine the bandwidth of the computer. The bandwidth can be calculated by evaluating the accuracy of the computer as a function of the output function frequencies.

The frequency of the output function is postulated, and the rms value of the errors due to the polynomial approximation is calculated. For every frequency, a certain value of the rms error can be determined. The bandwith of the computer can then be defined as the maximum frequency at which the rms error is still within the permissible limits.

In all real-time control and stabilization computers, it is always necessary to compute some trigonometric functions. There are many ingenious schemes of computing these functions by using the incremental techniques. All these techniques, however, suffer from the limitation of having infinite or very long memories. In the Epsco STARDAC Computer, the trigonometric functions are calculated using the Tchebycheff polynomials. Sine and cosine functions are usually needed simultaneously. In the Epsco STARDAC Computer they are calculated concurrently by using the powers of the argument and the multiplying the result by appropriate Tchebycheff coefficients. A very high degree of accuracy can be realized if the Tchebycheff polynomial is used within an interval of 0° to 90°. Simple logic is used to accommodate arguments outside of this range.

In this high-accuracy, real-time system, error analysis is probably the most important phase of the system design. All possible sources of accuracy-limiting factors must be carefully analyzed.

In the applications in which the computational time cannot be disregarded, a polynomial substitution for the output function is used to offset errors due to the computation time. The polynomial substitution can be only approximate and consequently an error is introduced. Truncation and round-off errors can be determined by analyzing the number of significant digits lost in the computations. Errors introduced by the substitution of Tchebycheff polynomials for the trigonometric functions can be determined.

Output errors due to the errors present in the input functions must be carefully analyzed since these errors determine the maximum realizable accuracy of the system.

The accuracy of the input function has a profound effect on the decisions which must be made in the design of the computer. If the computer is designed correctly, the errors it introduces are normally smaller than the output errors caused by the errors in the input functions. However, the propagation of the input errors through the computer must be carefully analyzed since some of them can be amplified in the computer more than others. The input function errors can be divided into two categories, bias and random.

Bias errors can be defined as those whose magnitude is consistent. In other words, the magnitude of an error can be predicted with a certain accuracy on the basis of the errors present in several previous measurements. On the other hand, random errors can be defined as unpredictable. The random error in any sample has a probability which is independent of the errors present in the previous samples.

The propagation of these errors through the computer can be traced easily by using appropriate partial derivatives. This error analysis is well known to those who have designed fire control computers. However, the relative magnitude of bias and random errors in real-time digital computers is normally different from the

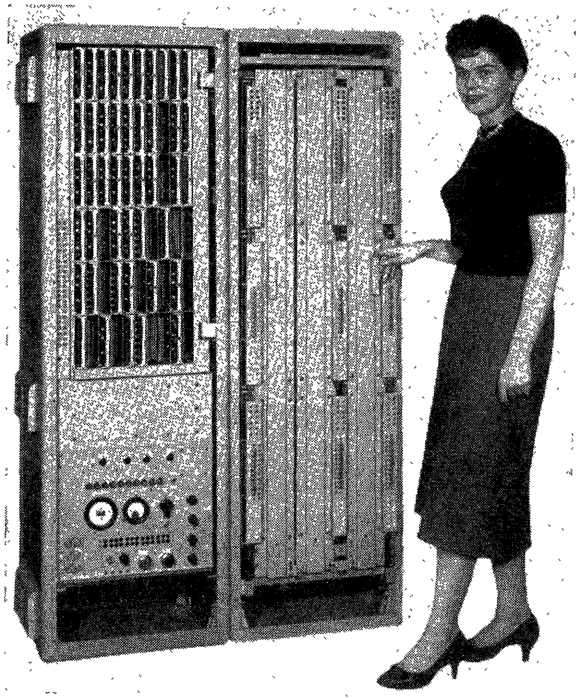

Fig. 3—Computer with covers in place.

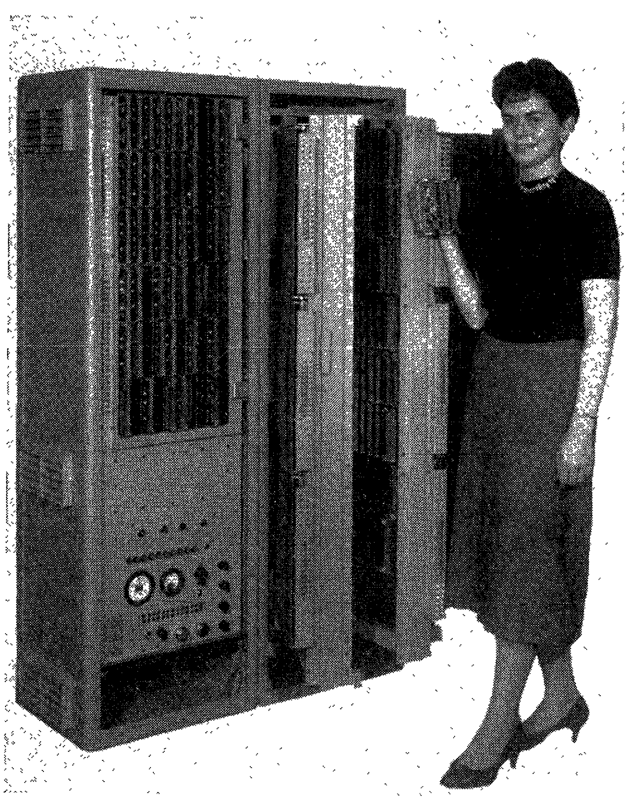

Fig. 4—Computer with covers removed, showing access for servicing.

relative magnitude of bias and random errors in, for example, radar returns.

In real-time control computers, input random errors are usually small and they are very often introduced only by the input quantization. The quantization random error has a rectangular probability distribution

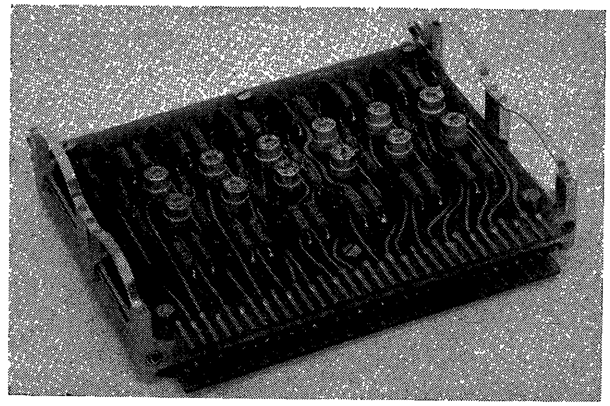

Fig. 5—Digital computer module, assembled.

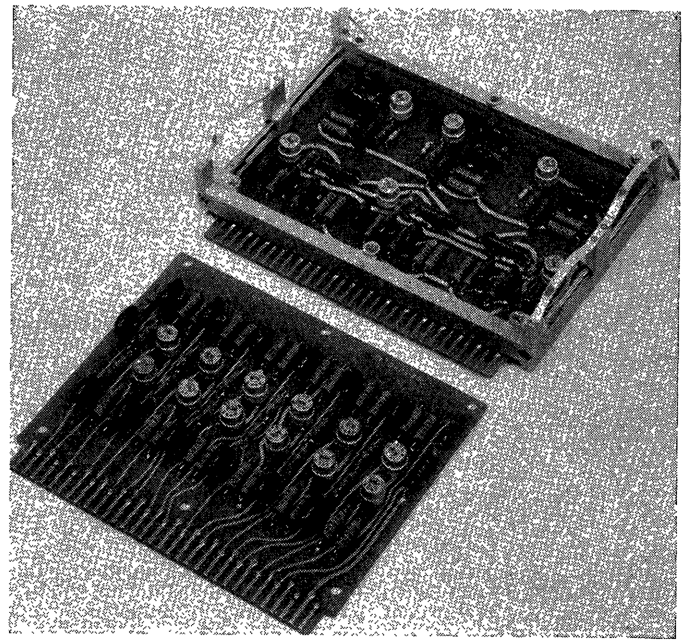

Fig. 6—Digital computer module, disassembled.

with a maximum possible error equal to one half of the least significant digit.

Various methods can be used in order to minimize the effect of random errors on the output function. Input random errors are particularly harmful if differences are employed in the computation of the polynomial which is used as the approximation to the output function. For example, if a second order polynomial is used, the third difference is calculated and is used to smooth out the output function. This compensation is valid only if the noise level is such that the third difference of the output function is much smaller than the measured third difference due to random input errors. This method, however, leads to relatively complicated equations. It is often possible to obtain a significant improvement by simply reducing the quantization errors. This is obvious since bias errors are not amplified as much in the computation of differences as are random errors.

Accuracy analysis would not be complete without a description of the selection of the control equations. In real-time, digital, control computers, accuracy can be greatly limited if a large number of mathematical operations must be carried out in order to compute the output function. Long computations are undesirable for two reasons. Large numbers of computations are time-consuming; and also, in each arithmetic addition as much as one half of the least significant digit may be lost. It is then necessary to know exactly what is the largest possible number of operations which might be necessary under the worst possible combination of input variables. The number of computations, sometimes, is very difficult to predict. This is particularly true if the computer function involves division and if the denominator, under certain conditions, approaches zero.

Unfortunately, this condition arises often in all problems in which spherical geometry is involved; this happens, for example, if it is necessary to compute an angle whose tangent is determined by a ratio of two expressions which, in turn, are determined by some other trigonometric functions. The angle itself is uniquely determined for the whole interval from 0° to 360°; however, the tangent is discontinuous at 90° and 270°.

In the STARDAC Computer, this problem was solved by the use of an iterative routine, which made it possible to compute the argument even if the tangent of the angle approached infinity.

As mentioned before, the STARDAC Computer has a built-in sine-cosine function generator. First a number is substituted for the value of the argument and the computer calculates the sine and the cosine. Then the sine of the argument is multiplied by the denominator and the cosine of the argument is multiplied by the numerator. In the second step of the computation, a comparison is made between the two products. The difference is then added directly to the number which was substituted for the argument. Then the cycle is repeated.

Mathematical justification for this operation is almost self-evident if the numerator of the fraction is represented as $\sin A$ and the denominator as $\cos A$. The term which is added to the argument can be expressed as

$$\Delta = \sin \theta K \cos A - \cos \theta K \sin A,$$

but θ was selected at random and was not equal to A. So the equation can be rewritten as:

$$\Delta = \sin (A + \Delta\theta) K \cos A - \cos(A + \Delta\theta) K \sin A$$

or

$$\Delta = K \sin \Delta\theta.$$

For small $\Delta\theta$, the value of Δ is equal to $K\Delta\theta$. The function converges rapidly if the value of the coefficient K is close to unity, and in a few iterations the error becomes negligible even for systems which require extremely high accuracy. The program is simple. No ambiguities arise and the arithmetic operations contain only multiplications, additions, and complementing. All these operations are particularly easy if performed in straight binary code.

The packaging techniques used in the construction of the STARDAC Computer can best be presented by referring to Figs. 3–6. Fig. 3 illustrates the computer complete with power supplies and input-output equipment. Fig. 4 shows the computer with covers removed and the frames pulled out for servicing. Figs. 5 and 6 show typical modules used in the computer.

It is felt at Epsco that a family of real-time computers such as described in this paper will find broad application in the field of high-accuracy real-time control systems such as stabilization computers, fire control computers, navigation computers, autopilots, etc.

A computer whose design is based on the approach outlined in this paper can offer an ideal solution to the problem of maintaining extremely high internal accuracy. It is believed that the need for these computers will grow together with the need for miniaturized, general-purpose computers. It is felt that this new type of computer will soon establish itself as a member of the family of computers together with the stored program, general-purpose machines and analog computers.