Discussion

M. W. Marcovitz (Burroughs Corp., Paoli, Pa.): Is the program for this machine stored in the core memory? If not, where?

Mr. Randall: Yes, the program is stored in the core memory.

T. A. Dowds (Burroughs Corp., Paoli, Pa.): Is this machine available? How are nonsignificant zeros stored when reading from punched cards and from ledger cards? Is this a single-address program?

Mr. Randall: No, this computer is not yet commercially available. The method of storing nonsignificant zeros is to first clear (write all zeros into) the memory cell to be loaded. The data are then loaded into the cell starting at the low-order end of the word until a signal is received that there are no more significant digits for that word. When reading punched cards this signal is an "end-of-field" signal from the card punch. When reading magnetic ledger cards the signal is an "end-of-word" symbol read from the card. The instruction format is not single address, but rather "three-plus-one" address.

R. A. Wallace (Burroughs Corp.): What is the method of feeding the ledgers? Is part of the information used to store line information? What provisions are there for checking?

Mr. Randall: The ledger cards are driven into the carriage by a mechanism added to the mechanical accounting machine. Line information is stored on the card so that it is stopped on the proper line for posting. There is practically no internal checking. However, checking can be accomplished by normal programming methods.

The Synthesis of Computer-Limited Sampled-Data Simulation and Filtering Systems*

ARTHUR S. ROBINSON†

THIS PAPER concerns the synthesis of systems in which a single digital computer is to be used in conjunction with an array of output "holds" or filters either to simulate the dynamic transfer characteristics of a number of linear continuous systems or to filter random messages from a number of continuous inputs, each of which consists of a mixture of random message plus random noise.

A block diagram of such a system is shown in Fig. 1. The system inputs are the continuous time functions

$$r_1(t),\ r_2(t),\ \cdots\ r_{N-1}(t),\ r_N(t).$$

The system outputs are the continuous time functions

$$c_1(t),\ c_2(t),\ \cdots\ c_{N-1}(t),\ c_N(t).$$

Each system output is obtained from an individual continuous output filter. This filter is in turn actuated by sampled signals periodically derived by the computer as it moves sequentially from channel to channel.

In systems of this type, in which a digital computer has available and is to supply continuous data, *the computer operates on sampled data only because a series of sequential operations are to be performed, each requiring a finite amount of time.* A typical simulator channel requires time for switching between channels and for analog-to-digital and digital-to-analog conversions, defined as s_j, and

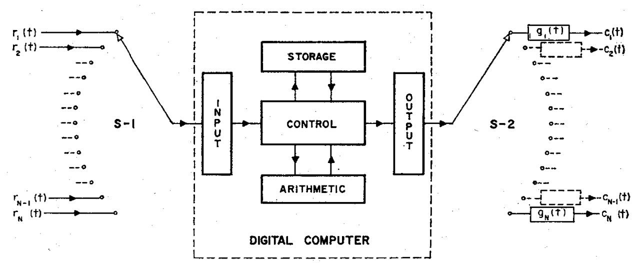

Fig. 1—Simulation system functional block diagram.

time for internal computer data transfers, multiplications, additions and subtractions, defined as k_j. The time required for switching and conversion is generally constant, whereas the time required for internal data processing and computation is governed by the complexity of the computer program. That is, in general, k_j will be a function of the number of terms in the computer program for that channel, so that T, the system sampling period, will be a function of the total number of program terms the computer is required to process for all channels.

The term Computer Limited has been coined to designate sampled data systems of this type, in which the system sampling period T is a function of limitations imposed by the computer implementation. Computer Limited sampled data systems can be contrasted to Data Limited systems, in which the system sampling rate is limited by some fixed external constraint in the data measuring equipment, such as the speed of rotation of a radar antenna. It is important to understand this basic difference between sys-

*Details pertinent to both this paper and footnote 1 are contained in footnote 2.

† Eclipse-Pioneer, Div. Bendix Aviation Corp., Teterboro, N.J.

tems, since synthesis techniques that have been derived heretofore for the optimization of Data Limited sampled data systems are not applicable to the Computer Limited problem.

It is interesting to note that Computer Limited sampled data systems are not restricted to the simulation systems described in this paper. *In general, any system that utilizes a digital computer in real time is a Computer Limited sampled data system, unless the available input data rate is slower than the speed with which computer solutions can be obtained.* Many analog computing systems that utilize multiplexed elements are also subject to Computer Limited constraints.

An application of Computer Limited theory to control system synthesis has already been presented.[1] It is suggested that the basic techniques to be presented in this and in supplementary papers[2] can be used as tools in the general synthesis of Computer Limited sampled data systems.

Returning to the stimulation and filtering application. When the system illustrated in Fig. 1 is to simulate the dynamic transfer characteristics of a number of linear continuous systems actuated by deterministic input signals, the ideal system outputs are as defined by the block diagram of Fig. 2, where

$$h_1(t),\ h_2(t),\ \cdots\ h_{N-1}(t),\ h_N(t)$$

are the impulsive responses corresponding to the ideal input-output relationships. When the system inputs are random functions of time of known statistical characteristics, or when the function of the system is to filter random messages from the continuous inputs, each of which consists of a mixture of random message plus random noise, the ideal system outputs are as defined in Fig. 3. As indicated, the ideal filtering response for each channel would result if a signal equal to the noise input could be effectively subtracted from the total input, so that only the random messages

$$r_{m_1}(t),\ r_{m_2}(t),\ \cdots\ r_{m_{N-1}}(t),\ r_{m_N}(t)$$

actuated the ideal filters. Since the block diagrams of the filtering and simulation problems are almost identical, it is convenient to visualize the filtering problem as the simulation of ideal filters, and so to combine the discussion of system synthesis techniques under the single heading of "simulation."

The ideal system responses shown in Figs. 2 and 3 define reference outputs against which the performance of the actual system shown in Fig. 1 must be compared to evaluate the effectiveness of the simulation. When system inputs are deterministic, the synthesis techniques to be described permit the system designer to determine the linear digital computer program that will minimize the integrated error squared between ideal and actual continuous system outputs. When system inputs are random, the synthesis techniques lead to minimization of the mean squared error. When only the form of the channel output filters are specified, the synthesis procedure permits the filter parameters to also be optimized.

[1] A. S. Robinson, "The Synthesis of Computer Limited Sampled Data Control Systems," presented at AIEE Computer Conf., Atlantic City, N.J., October, 1957.

[2] A. S. Robinson, "The Optimum Synthesis of Computer Limited Sampled Data Systems," D. Eng. Sc. Dissertation, Columbia University, New York, N.Y.; May, 1957.

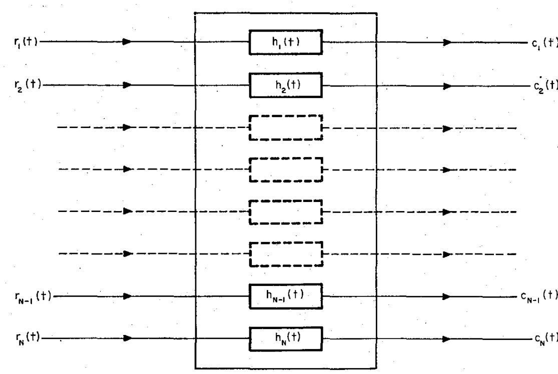

Fig. 2—Ideal simulator input-output relationships —deterministic inputs.

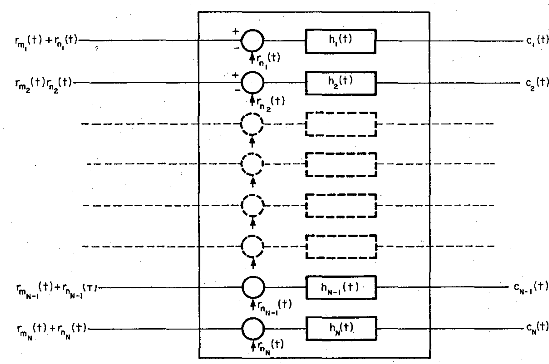

Fig. 3—Ideal simulator input-output relationships—random inputs.

Returning to Fig. 1, a mathematical model for a given channel can be obtained by tracing the channel from its input to its output. At the input the continuous signal $r_j(t)$ is periodically connected to the digital computer input, where the sampled analog voltage is converted to a digital number. Each such digital number is effectively an instantaneous sample of the value of the continuous function. If the period between sampling instants is denoted by T, the conversion of the continuous deterministic input signal $r_j(t)$ to the series of digital numbers

$$r_j(0),\ r_j(\mathrm{T}),\ r_j(2\mathrm{T})\ \ldots$$

can be visualized as a process of impulse modulation, with the area of each impulse equal to the value of the corresponding digital number.

Each time the computer receives new input data it proceeds through a new series of computations and delivers a new solution at its output after the time delay required to perform these computations. The computed problem solution is then used to actuate the output filter, characterized by its impulsive response, $g_j(t)$, so that as the computer moves on to other simulator channels, the output filter provides a continuous output response for the jth channel.

The mathematical model for a single channel with a deterministic input is shown in Fig. 4 in terms of Laplace and Z transforms. It is convenient to describe the digital computer by an instantaneous response, characterized by $C^*(Z)$, followed by a time delay, described by $Z^{-k/T}$. The transfer function of the continuous output filter is defined by $G(s)$. The meaning of each of these symbols is shown more precisely in Table I. $r^*(t)$ describes the train of impulses at the computer input. Z^{-1} is, of course, simply the delay operator e^{-sT}. The Z transform of the computer input is $R^*(Z)$. $G(s)$ is the Laplace transform of the output filter impulsive response.

Dr. Salzer[3] has shown that $C^*(Z)$, the linear program of a digital computer operating in real time, is physically realizable only when it can be described by a ratio of polynomials in Z^{-1}, as indicated in Table I. Referring to Table II, this constraint states that the impulse sequence corresponding to the effective computer instantaneous output, characterized by $P_i^*(Z)$, can only be formed as the sum of appropriately weighted past and present values of the computer input, and past values of the computer output. That is, $R^*(Z)a_0$ corresponds in the time domain to the input impulse sequence multiplied by a_0. $R^*(Z)Z^{-1}a_1$ corresponds to the input impulse sequence delayed by T and multiplied by a_1. $R^*(Z)Z^{-2}a_2$ corresponds to a delay of $2T$ and multiplication by a_2, etc. $P_i^*(Z)Z^{-1}b_1$ corresponds to the computed output sequence delayed by T and multiplied by b_1. $P_i^*(Z)Z^{-2}b_2$ corresponds to a delay of $2T$ and multiplication by b_2, etc. Only past and present terms can be utilized in a real time digital computer because future terms (positive powers of Z) will not, of course, be available. Table II also indicates the existence of the constraint that exists in an actual system implementation between the number of computer program terms, the channel computing time k_j, and the system sampling period T. This relationship between computer program complexity and required computing time must always form part of the basic problem statement. The relationship need not be linear. Any nondecreasing function can be accepted, defined in analytical, graphical, or tabular form.

Fig. 5 summarizes the relationships between true and desired system outputs for a single simulation channel with a deterministic input. The object of the synthesis procedure is to determine the linear digital computer program $C^*(Z)$ that will result in minimization of the integral of the continuous error squared. That is, the quantity

$$\int_0^\infty [\epsilon(t)]^2 dt = \int_0^\infty [c_t(t) - c_d(t)]^2 dt$$

is to be minimized. Note particularly that it is the continuous error, squared and integrated, that is being used to evaluate system performance, not the error at sampling instants only. Thus inter-sample ripple is automatically accounted for. When only the form of $G(s)$ is specified, it is also required that the optimum parameters for $G(s)$ be determined.

[3] J. M. Salzer, "Treatment of Digital Control Systems and Numerical Processes in the Frequency Domain," D. Eng. Sc. Dissertation, Dept. of Elec. Eng., M.I.T., Cambridge, Mass.; 1951.

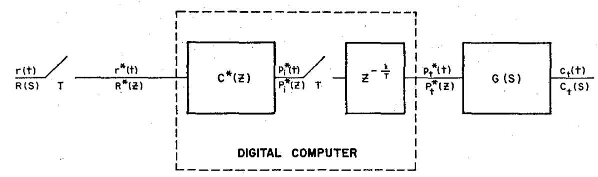

Fig. 4—Simulation system single channel—block diagram.

TABLE I
BLOCK DIAGRAM DEFINITIONS

$$r^*(t) = \sum_{n=0}^{\infty} r(t)\delta(t - nT) \quad (1.1)$$

$$Z^{-1} = e^{-ST} \quad (1.2)$$

$$R^*(Z) = \sum_{l=0}^{\infty} r(lT)Z^{-} \quad (1.3)$$

$$C^*(Z) = \frac{\sum_{p=0}^{p_m} a_p z^{-p}}{1 + \sum_{q=1}^{q_m} b_q z^{-q}} \triangleq \frac{a^*(z)}{b^*(z)} = \frac{P_i^*(Z)}{R^*(Z)} \quad (1.4)$$

$$G(S) = \int_0^\infty g(t)e^{-st}dt \quad (1.5)$$

TABLE II
COMPUTER PROGRAM RELATIONSHIPS

$$C^*(Z) = \frac{\sum_{p=0}^{p_m} a_p z^{-p}}{1 + \sum_{q=1}^{q_m} b_q z^{-q}} = \frac{P_i^*(Z)}{R^*(Z)} \quad (2.1)$$

$$P_i^*(Z) = R^*(Z)[a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots a_{p_m} z^{-p_m}] - P_i^*(Z)[b_1 z^{-1} + b_2 z^{-2} + \cdots b_{q_m} z^{-q_m}] \quad (2.2)$$

$$Z^{-\frac{k}{i}}, \quad k = f(p_m + q_m) \quad (2.3)$$

$$T_j = k_j + s_j \quad (2.4)$$

$$T = \sum_{j=0}^{N} T_j \quad (2.5)$$

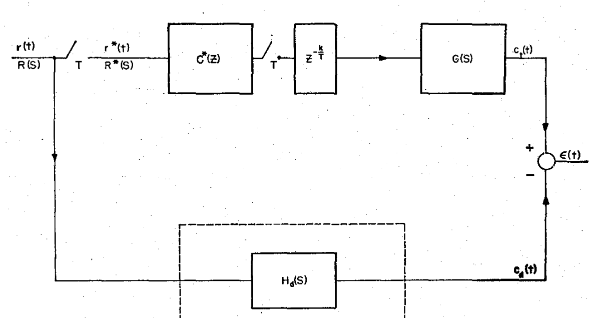

Fig. 5—Simulation channel error—deterministic input.

TABLE III

OPTIMIZATION RELATIONSHIPS FOR COMPUTER LIMITED SIMULATION SYSTEMS—DETERMINISTIC INPUTS

(a)

$$\int^{a_p} E^{2*}(Z) = 2\gamma^2 R^*(Z) R^*\left(\frac{1}{Z}\right) \Phi_{g_k g_k}{}^*(Z) \frac{a^*(z) z^{+p}}{b^*(z) b^*\left(\frac{1}{z}\right)} - 2\gamma \frac{R^*\left(\frac{1}{Z}\right) \mathcal{F}^*(Z) Z^{+p}}{b^*\left(\frac{1}{z}\right)} \quad (3.1)$$

$$\int^{b_q} E^{2*}(Z) = -2\gamma^2 R^*(Z) R^*\left(\frac{1}{Z}\right) \Phi_{g_k g_k}{}^*(Z) \frac{a^*(z) a^*\left(\frac{1}{z}\right) z^{+q}}{b^*(z) b^*\left(\frac{1}{z}\right)^2} + 2\gamma R^*\left(\frac{1}{Z}\right) \frac{\mathcal{F}^*(Z) a^*\left(\frac{1}{z}\right) z^{+q}}{b^*\left(\frac{1}{z}\right)^2} \quad (3.2)$$

$$\int E^{2*}(Z) = R^*(Z) R^*\left(\frac{1}{Z}\right) C^*(Z) C^*\left(\frac{1}{Z}\right) \gamma^2 \Phi_{g_k g_k}{}^*(Z) - 2\gamma R^*\left(\frac{1}{Z}\right) C^*\left(\frac{1}{Z}\right) \mathcal{F}^*(Z) + \Phi_{c_d c_d}{}^*(Z) \quad (3.3)$$

(b)

where

$$C^*(Z) = \frac{a^*(z)}{b^*(z)} \quad (3.4)$$

$$\Phi_{g_k g_k}(Z) = Z[G(S)G(-S)] \quad (3.5)$$

$$\mathcal{F}^*(Z) = Z[G(-S)e^{-sk}R(S)H_d(S)] \quad (3.6)$$

$$\Phi_{c_d c_d}(Z) = Z[R(S)R(-S)H_d(S)H_d(-S)] \quad (3.7)$$

γ = impulse duration

TABLE IV

SUMMARY OF BASIC SYNTHESIS PROCEDURE—SIMULATION SYSTEMS—DETERMINISTIC INPUTS

Step Number	Evaluate	From
	(a)	
1	$\Phi_{g_k g_k}{}^*(Z)$	$\Phi_{g_k g_k}(S) = G(S)e^{-sk}G(-S)e^{-sk} = G(S)G(-S)$
2	$\mathcal{F}^*(Z)$	$\mathcal{F}(S) = G(-S)e^{-sk}R(S)H_d(S)$
3	$R^*(Z)$	$R(S)$
4	$\frac{\partial}{\partial a_p}\int \epsilon^2 = 0;\ p = 0, 1 \cdots p_{ms}$	The values of the $(p_{ms}+1)$ time functions corresponding to $\int^{a_p} E^*(Z)$ (3.1) evaluated at $t=0$.
	(b)	
5	$\frac{\partial}{\partial b_q}\int \epsilon^2 = 0;\ q = 1, \cdots q_{ms}$	The values of the q_{ms} time functions corresponding to $\int^{b_q} E^{2*}(Z)$ (3.2) evaluated at $t=0$.
6	$C^*(Z)$	Simultaneous solution of the $(p_{ms}+q_{ms}+1)$ equations defined by steps 4 and 5.
7	$\int \epsilon^2$	The value at $t=0$ of the time function corresponding to $\int E^{2*}(Z)$ (3.3), with T determined from the stated relationship between sampling period and computer program complexity.

Detailed derivations of the synthesis procedures to be described are available.[2] It is possible, without proceeding through these derivations, to acquire an understanding of the capabilities and limitations of the technique, and a physical insight into the fundamental processes that are automatically brought into play by the synthesis procedures.

The three key synthesis equations for systems with deterministic inputs are listed in Table III. In essence, these equations summarize in transform form all of the time domain convolutions, integrations and differentiations required to establish the conditions for optimum operation of a given channel, given the total number of terms in the computer program numerator (p_m+1) and in the program denominator (q_m). These equations contain both negative and positive powers of Z, so that their corresponding time functions have values for both positive and negative time. The synthesis procedure requires that the value at $t=0$ of the time functions corresponding to (3.1) $(p=0, 1 \cdots p_m)$, (3.2) $(q=1, \cdots q_m)$, and (3.3) be evaluated. The results of each of the (p_m+1) evaluations of (3.1) and of the q_m evaluations of (3.2) are then set equal to zero. This effectively constrains the derivatives of the integrated error squared with respect to each of the computer program parameters to be zero. The resultant (p_m+q_m+1) simultaneous equations are then solved for the (p_m+q_m+1) unknowns $(a_0, a_1 \cdots a_{p_m}, b_1, b_{q_m})$. This establishes the optimum values for the computer program parameters. When these values are substituted in the result of the evaluation of (3.3), the resultant number, (still a function of k and T), will correspond to the system integrated error squared. Finally, k, which is a function of single channel computing time, and T, which is a function of total system computing time, can be evaluated, and the integrated error squared will then correspond to a known number. Table IV summarizes the steps involved in the determination of the optimum program parameters and integrated error squared for a computer program of stated complexity (p_m, q_m). In the event that only the output filter form is specified, the filter parameters can also be optimized by differentiating the integrated error squared with respect to each parameter, setting each equation equal to zero, and solving for the optimum parameters.

Note that the factor γ that appears in the basic synthesis equations of Table III corresponds to the duration of the computer output pulse that actuates the channel output filter. Normally, the duration of this pulse is important, since the output filter is specified in terms of its impulsive response, and both the amplitude and duration of the computer output affect the filter output. When the output filter consists of a hold circuit that responds only to the amplitude of the computed output, and is unaffected by the time it takes to actuate the hold, a factor $1/\gamma$ must appear in the hold transform to compensate for this effect. That is, the transform of a first-order hold is simply

$$\frac{(1 - e^{-sT})}{\gamma s}.$$

Data Limited theory provides an upper limit to the number of potential computer programs that must be considered before the optimum Computer Limited program can be established. Dr. Franklin[4] has shown that the optimum Data Limited $C^*(Z)$ is given by the equations listed in Table V. The derivation of the optimum Computer Limited program requires consideration of both this Data Limited solution and all less complex programs, and the evaluation in each case of the corresponding integrated error squared, subject to the computer complexity-sampling period constraint. Based on these evaluations, the optimum Computer Limited program will be evident as the program resulting in least integrated error squared.

For example, if a given Data Limited solution results in the program form shown in Table VI, (6.1) Computer Limited theory would require, in effect, that each of the potential programs shown in (6.2) through (6.4) be considered. Programs of greater length would not have to be considered, since they would always lead to greater integrated error squared. This is the case because Data Limited theory imposes no penalty for computer program complexity, so that the Data Limited program cannot be improved by increasing the number of program terms. However, when the sampling period—computer complexity constraint is taken into account, a longer program would result in a longer sampling period and therefore in greater integrated error squared. In general, the advantage of the Computer Limited approach lies in its ability to indicate shorter programs, and therefore shorter sampling periods, than those nominally dictated by Data Limited theory.

In implementing the procedure described above the system designer can, if he wishes, use only the five term program shown in (6.5), solve the required five simultaneous equations to obtain a general solution to the problem, and simply reduce the unused parameters to zero when considering each potential program in turn. This approach is to be compared with the solution of three sets of three simultaneous equations

$$(p_m = 2,\ q_m = 0), \quad (p_m = 1,\ q_m = 1), \quad (p_m = 0,\ q_m = 2).$$

The selection of a particular procedure can be made to suit the convenience of the system designer, since the results are independent of the procedure.

[4] G. Franklin, "The Optimum Synthesis of Sampled-Data Systems," D. Eng. Sc. Dissertation, Columbia University, New York, N.Y.; May, 1955.

TABLE V

OPTIMIZATION RELATIONSHIPS FOR DATA LIMITED SIMULATION SYSTEMS DETERMINISTIC INPUTS

$$C^*(Z) = \frac{W_1^*(Z)}{\gamma \left\{ R^*(Z) R^*\left(\frac{1}{Z}\right) \right\}^+ \left\{ \Phi_{g_k g_k}^*(Z) \right\}^+} \qquad (5.1)$$

$$W^*(Z) = \frac{R^*\left(\frac{1}{Z}\right) \mathit{IF}^*(Z)}{\left\{ R^*(Z) R^*\left(\frac{1}{Z}\right) \right\}^- \left\{ \Phi_{g_k g_k}^*(Z) \right\}^-} \qquad (5.2)$$

$$= \underbrace{\underset{\text{Poles Inside}}{W_1^*(Z)} + \underset{\text{Poles Outside}}{W_2^*(Z)}}_{\text{The Unit Circle}}$$

$$\Phi_{g_k g_k}^*(Z) \overset{\Delta}{=} \underbrace{\underset{\text{Poles Outside}}{\left\{ \Phi_{g_k g_k}^*(Z) \right\}^-} \underset{\text{Poles Inside}}{\left\{ \Phi_{g_k g_k}^*(Z) \right\}^+}}_{\text{The Unit Circle}} \qquad (5.3)$$

TABLE VI

EXAMPLE OF POTENTIAL COMPUTER LIMITED PROGRAMS

$$\text{Data Limited Program } C^*(Z) = \frac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}} \qquad (6.1)$$

Alternate Programs to be Considered

Program	Period	
$C^*(Z)_{a-1} = a_0$	T_a	(6.2)
$C^*(Z)_{b-1} = \dfrac{a_0}{1 + b_1 z^{-1}}$; $C^*(Z)_{b-2} = a_0 + a_1 z^{-1}$	T_b	(6.3)
$C^*(Z)_{c-1} = \dfrac{a_0}{1 + b_1 z^{-1} + b_2 z^{-2}}$; $C^*(Z)_{c-2} = \dfrac{a_0 + a_1 z^{-1}}{1 + b_1 z^{-1}}$; $C^*(Z)_{c-3} = a_0 + a_1 z^{-1} + a_2 z^{-2}$	T_c	(6.4)

General Program Containing All Applicable Terms

$$C^*(Z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \qquad (6.5)$$

When N similar problems are being simulated, the system sampling period T is N times the channel sampling period, so that a channel sampling period T_j will result in a system sampling period $T = NT_j$. The synthesis of the entire system can then be accomplished by optimizing the program for a single channel. When the N problems to be simulated are not similar it is necessary to consider each combination of potential single channel programs that could produce a given system sampling period, and each potential system sampling period, evaluating in each case the corresponding integrated error squared for each channel. The sum of all channel integrated error squared terms can then be observed and the least value selected as the optimum system operating point. Further details on such a procedure are available.[2]

When the system inputs are mixtures of random message plus random noise, a slightly modified mathematical

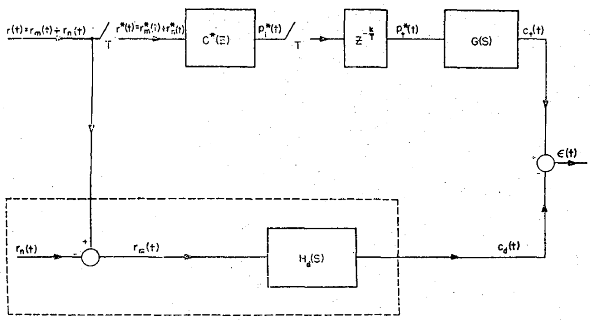

Fig. 6—Simulation channel error—random inputs.

TABLE VII

OPTIMIZATION RELATIONSHIPS FOR COMPUTER-LIMITED SIMULATION SYSTEMS RANDOM INPUTS

(a)

$$\overline{\overline{E^{2*}}}^{a_p}(Z) = \frac{2\gamma^2}{T}\Phi_{rr}^*(Z)\mathbf{\Phi}_{g_kg_k}^*(Z)\frac{a^*(z)z^{+p}}{b^*(z)b^*\left(\frac{1}{z}\right)} - \frac{2\gamma}{T}\frac{F^*(Z)z^{+p}}{b^*\left(\frac{1}{z}\right)} \qquad (7.1)$$

$p = 0, 1, \cdots p_m$

$$\overline{\overline{E^{2*}}}^{b_q}(Z) = -\frac{2\gamma^2}{T}\Phi_{rr}^*(Z)\mathbf{\Phi}_{g_kg_k}^*(Z)\frac{a^*\left(\frac{1}{z}\right)a^*(z)z^{+q}}{b^*\left(\frac{1}{z}\right)^2 b^*(z)} + \frac{2\gamma}{T}\frac{a^*\left(\frac{1}{z}\right)}{b^*\left(\frac{1}{z}\right)^2}F^*(Z)z^{+q} \qquad q = 1, \cdots, q_m \qquad (7.2)$$

$$\overline{E^{2*}}(Z) = \frac{\gamma^2}{T}\frac{a^*(z)a^*\left(\frac{1}{z}\right)}{b^*(z)b^*\left(\frac{1}{z}\right)}\Phi_{rr}^*(Z)\mathbf{\Phi}_{g_kg_k}^*Z - \frac{2\gamma}{T}\frac{a^*\left(\frac{1}{z}\right)}{b^*\left(\frac{1}{z}\right)}F^*(Z) + \Phi_{c_dc_d}^*(Z) \qquad (7.3)$$

(b)

$$\mathbf{\Phi}_{g_kg_k}^*(Z) = Z[G(S)G(-S)] \qquad (7.4)$$

$$\Phi_{rr}^*(Z) = Z[\Phi_{rr}(S)] \qquad (7.5)$$

$$F^*(Z) = Z[G(-S)e^{sk}H_d(S)\Phi_{rr_m}(S)] \qquad (7.6)$$

$$\Phi_{c_dc_d}(Z) = Z[\Phi_{r_mr_m}(S)H_d(S)H_d(-S)] \qquad (7.7)$$

model must be used to define the error between true and ideal system outputs for a given channel. As illustrated in Fig. 6, the main difference lies in the fact that the ideal output $c_d(t)$ is assumed to be derived from an ideal transfer characteristic $h_d(t)$ that is actuated by the random message alone. Since the stationary random signals are assumed present over all time, the continuous mean squared system error, rather than the integrated error squared, is to be minimized. The techniques required to synthesize a system of this type are essentially the same as those required to synthesize a system with deterministic inputs, although a different set of optimization equations now apply. The optimization equations for the synthesis of simulation systems with random inputs are presented in Table VII, the corresponding step by step synthesis procedure is tabulated in Table VIII, and Prof. Franklin's Data Limited solution to this problem is shown in Table IX.

Note that the channel inputs are now defined by the

TABLE VIII

SUMMARY OF BASIC SYNTHESIS PROCEDURE—SIMULATION SYSTEMS—RANDOM INPUTS

Step Number	Evaluate	From
	(a)	
1	$\mathbf{\Phi}_{g_kg_k}^*(Z)$	$\mathbf{\Phi}_{g_kg_k}(S) = G(S)e^{-sk}G(-S)e^{sk} = G(S)G(-S)$
2	$\Phi_{rr}^*(Z)$	$\Phi_{rr}(S)$
3	$F^*(Z)$	$F(S) = G(-S)e^{sk}H_d(S)\Phi_{rr_m}(S)$
4	$\frac{\partial\overline{e^2}}{\partial a_p} = 0;\ p = 1, \cdots p_{ms}$	The values of the $(p_{ms}+1)$ time functions corresponding to $\overline{\overline{E^2}}^{a_p}*(Z)$ (7.1) evaluated at $t=0$.
	(b)	
5	$\frac{\partial\overline{e^2}}{\partial b_q} = 0;\ q = 1, \cdots q_{ms}$	The values of the q_{ms} time functions corresponding to $\overline{\overline{E^2}}^{b_q}*(Z)$ (7.2) evaluated at $t=0$.
6	$C^*(Z)$	Simultaneous solution of the $(p_{ms}+q_{ms}+1)$ equations defined by steps 4 and 5.
7	$\overline{e^2}$	The value at $t=0$ of the time function corresponding to $\overline{E^{2*}}(Z)$ (7.3) with T determined from the stated relationship between sampling period and computer program complexity.

TABLE IX

OPTIMIZATION RELATIONSHIPS FOR DATA—LIMITED SIMULATION SYSTEMS—RANDOM INPUTS

$$C^*(Z) = \frac{1}{\gamma}\frac{W_1^*(Z)}{\{\mathbf{\Phi}_{g_kg_k}^*(Z)\}^+\{\Phi_{rr}^*(Z)\}^+} \qquad (9.1)$$

$$W^*(Z) = \frac{F^*(Z)}{\{\mathbf{\Phi}_{g_kg_k}^*(Z)\}^-\{\Phi_{rr}^*(Z)\}^-} \qquad (9.2)$$

$$= W_1^*(Z) + W_2^*(Z) \qquad (9.3)$$

($W_1^*(Z)$: Poles Inside; $W_2^*(Z)$: Poles Outside — The Unit Circle)

$$\Phi_{rr}^*(Z) = \{\Phi_{rr}^*(Z)\}^-\{\Phi_{rr}^*(Z)\}^+$$

($\{\Phi_{rr}^*(Z)\}^-$: Poles Outside; $\{\Phi_{rr}^*(Z)\}^+$: Poles Inside — The Unit Circle)

TABLE X

DETERMINING THE VALUE AT $t=0$ OF THE TIME FUNCTION CORRESPONDING TO $\Phi_{rc}^*(Z)$—METHOD 1

$$\Phi_{rc}^*(Z) = \frac{-1.2Z^{+1} + 4.24 - 0.8Z^{-1}}{(1 - 0.6Z^{+1} + 0.08Z^{+2})(1 - 0.4Z^{-1} + 0.03Z^{-2})} \quad (10.1)$$

$$\phi_{rc}(0) = \frac{1}{2\pi j}\oint_{\Gamma} \Phi_{rc}^*(Z)Z^{-1}dz \quad (10.2)$$

$$\Phi_{rc}^*(Z) = \frac{-1.2Z^{+1} + 4.24 - 0.8Z^{-1}}{(1 - 0.2Z^{+1})(1 - 0.4Z^{+1})(1 - 0.1Z^{-1})(1 - 0.3Z^{-1})} \quad (10.3)$$

$$\phi_{rc}(0) = \left.\frac{-1.2Z^{+2} + 4.24Z^{+1} - 0.8}{(1 - 0.2Z^{+1})(1 - 0.4Z^{+1})(Z^{+1} - 0.3)}\right|_{z=0.1} + \left.\frac{-1.2Z^{+2} + 4.24Z^{+1} - 0.8}{(1 - 0.2Z^{+1})(1 - 0.4Z^{+1})(Z^{+1} - 0.1)}\right|_{z=0.3} \quad (10.4)$$

$$\phi_{rc}(0) = 2.062 + 2.200 = 4.262 \quad (10.5)$$

to the form shown in (10.3), so that the integral can be evaluated using its residues inside the unit circle as shown in (10.4) and (10.5). The disadvantage of this approach lies in the requirement for factoring the denominator of $\Phi_{rc}^*(Z)$ in order to determine the pole locations.

A second approach to the problem is to factor $\Phi_{rc}^*(Z)$ into the sum of two parts, one inside and the other outside the unit circle. This approach is presented in Table XI. When the numerator of the assumed and actual polynomials are equated, as shown in (11.1), the set of linear equations listed in (11.2) result. These equations can be solved for α_{20}, the value at $t = 0$ of the time function corresponding to the original two-sided Z transform, as shown in (11.3).

TABLE XI

DETERMINING THE VALUE AT $t=0$ OF THE TIME FUNCTION CORRESPONDING TO $\Phi_{rc}^*(Z)$—METHOD 2

$$\Phi_{rc}^*(Z) = \frac{-1.2Z^{+1} + 4.24 - 0.8Z^{-1}}{(1 - 0.6Z^{+1} + 0.08Z^{+2})(1 - 0.4Z^{-1} + 0.03Z^{-2})} = \frac{\alpha_{20} + \alpha_{21}Z^{+1}}{(1 - 0.6Z^{+1} + 0.08Z^{+2})} + \frac{\alpha_{11}Z^{-1} + \alpha_{12}Z^{-2}}{(1 - 0.4Z^{-1} + 0.03Z^{-2})}$$
$$= \frac{(\alpha_{20} + \alpha_{21}Z^{+1})(1 - 0.4Z^{-1} + 0.03Z^{-2}) + (\alpha_{11}Z^{-1} + \alpha_{12}Z^{-2})(1 - 0.6Z^{+1} + 0.08Z^{+2})}{(1 - 0.6Z^{+1} + 0.08Z^{+2})(1 - 0.4Z^{-1} + 0.03Z^{-2})} \quad (11.1)$$

$$\begin{aligned} 0\alpha_{11} + \alpha_{12} + 0.03\alpha_{20} + 0\alpha_{21} &= 0 \\ \alpha_{11} - 0.6\alpha_{12} - 0.4\alpha_{20} + 0.03\alpha_{21} &= -0.8 \\ -0.6\alpha_{11} + 0.08\alpha_{12} + \alpha_{20} - 0.4\alpha_{21} &= +4.24 \\ 0.08\alpha_{11} + 0\alpha_{12} + 0\alpha_{20} + \alpha_{21} &= -1.2 \end{aligned} \quad (11.2)$$

$$\alpha_{20} = \frac{\begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & -0.6 & -0.8 & 0.03 \\ -0.6 & 0.08 & 4.24 & -0.4 \\ 0.08 & 0 & -1.2 & 1 \end{vmatrix}}{\begin{vmatrix} 0 & 1 & 0.03 & 0 \\ 1 & -0.6 & -0.4 & 0.03 \\ -0.6 & 0.08 & 1 & -0.4 \\ 0.08 & 0 & 0 & 1 \end{vmatrix}} = \frac{-3.317}{-0.778} = 4.262 \quad (11.3)$$

power density spectra of the input signals (random message plus random noise). Also note that $F(s)$, one of the transforms required in the synthesis procedure, is derived from $\Phi_{rr_m}(s)$, the power density spectrum corresponding to the cross-correlation between message plus noise and message alone.

In order to implement the synthesis procedures that have been described, it is necessary to determine the value at $t = 0$ of various two-sided Z transforms. Before proceeding to a problem example, three techniques for evaluating two-sided Z transforms will be described. The simple two-sided Z transform listed in Table X will be used as an example.

One direct and straightforward method for determining the value at $t = 0$ of the time function corresponding to this transform is to use the relationship presented in (10.2). This approach requires that $\Phi_{rc}^*(Z)$ be factored

The third approach to the problem is illustrated in Fig. 7 and tabulated in Table XII. Referring to the figure, it can be noted that the original two-sided Z transform is the product of a numerator and of two terms in the denominator, one with poles inside and the other with poles outside the unit circle. Both denominator expressions can be expanded to a finite number of terms rational in positive and negative powers of Z respectively. These expansions can be cross-multiplied to obtain a new two-sided Z transform. This transform can then be multiplied with the numerator, considering only terms that contribute to the value at $t = 0$ of the corresponding time function. The example in Fig. 7 shows only three numerator terms and therefore only three terms of the new two-sided transform need be considered. Table XII further demonstrates the process.

As a simple example of the theory that has been outlined, consider the problem of simulating a pure prediction

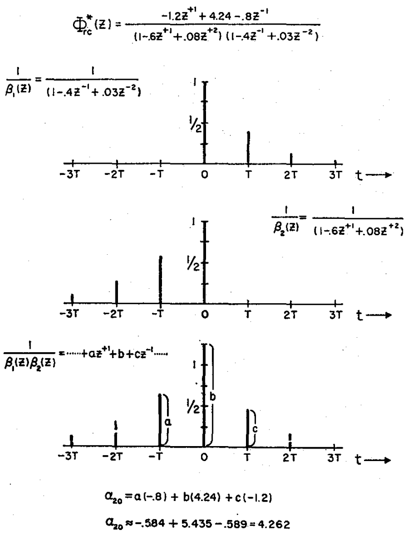

Fig. 7—Approximating the value at $t=0$ of the time function corresponding to $\Phi_{rc}^*(Z)$.

TABLE XII

DETERMINING THE (APPROXIMATE) VALUE AT $t=0$ OF THE TIME FUNCTION CORRESPONDING TO $\Phi_{rc}^*(Z)$—METHOD 3

$$\frac{1}{\beta_1^*(Z)} \approx 1 + 0.4Z^{-1} + 0.13Z^{-2} + 0.04Z^{-3} + 0.121Z^{-4} + \cdots \quad (12.1)$$

$$\frac{1}{\beta_2^*(Z)} \approx 1 + 0.6Z^{+1} + 0.28Z^{+2} + 0.12Z^{+3} + 0.0496Z^{+4} + \cdots \quad (12.2)$$

$$\Phi_{rc}^*(Z) = \frac{(-1.2Z^{+1} + 4.24 - 0.8Z^{-1})}{\beta_1^*(Z)\beta_2^*(Z)} \approx [0.0496Z^{+4} + 0.13984Z^{+3} + 0.33445Z^{+2} + 0.72958Z^{+1} + 1.28180 + 0.49065Z^{-1} + 0.15739Z^{-2} + 0.04726Z^{-3} + 0.0121Z^{-4}] [-1.2Z^{+1} + 4.24 - 0.8Z^{-1}] \quad (12.3)$$

$$\phi_{rc}(0) \approx 4.2624 \quad (12.4)$$

in each of N simulation channels when the system inputs are random messages with power density spectra

$$\Phi_{rr}(s) = \frac{2b}{b^2 - s^2},$$

each output filter is to be a simple exponential hold of the form $G(s) = 1/(s+a)$, and the constraint between computer program complexity and system sampling period can be approximated by a delay of T_1 seconds for each program term. Since N similar problems are to be solved, the system sampling period is given by

$$T = (p_m + q_m + 1)NT_1,$$

and the delay in a given channel is

$$k = (p_m + q_m + 1)T_1.$$

If the desired prediction for a given channel is α, then

$$H_d(s) = e^{\alpha s}.$$

A Data Limited solution to this problem specifies the program and mean squared error listed in Table XIII, (13.1) and (13.2).[4] When the computer complexity—sampling period constraint is applied to these results, $T = 2NT_1$ and the solutions are as shown in (13.3) and (13.4).[2]

TABLE XIII

DATA LIMITED SOLUTION TO PROBLEM EXAMPLE RANDOM INPUT

$$C_{DL}^*(Z) = \frac{2a}{\gamma} \frac{e^{-\alpha b}}{(a+b)} \frac{(1 - e^{-aT}e^{-bT})}{(1 - e^{-2aT})} (1 - e^{-aT}z^{-1}) \quad (13.1)$$

$$\overline{\epsilon^2} = 1 - \frac{1}{T} \frac{2a}{(a+b)^2} e^{-2\alpha b} \frac{(1 - e^{-aT}e^{-bT})^2}{(1 - e^{-2aT})} \quad (13.2)$$

$$C_{DL}^*(Z) = \frac{2a}{\gamma} \frac{e^{-\alpha b}}{(a+b)} \frac{(1 - e^{-2aNT_1}e^{-2bNT_1})}{(1 - e^{-4aNT_1})} (1 - e^{-2aNT_1}z^{-1}) \quad (13.3)$$

$$\overline{\epsilon^2} = 1 - \frac{1}{NT_1} \frac{a}{(a+b)^2} e^{-2\alpha,} \frac{(1 - e^{-2aNT_1}e^{-2bNT_1})^2}{(1 - e^{-4aNT_1})} \quad (13.4)$$

TABLE XIV

COMPUTER LIMITED SOLUTION TO PROBLEM EXAMPLE RANDOM INPUT

$$C^*(Z) = \frac{2a}{\gamma} \frac{e^{-b(k+\alpha)}(1 - e^{-aT}e^{-bT})}{(a+b)(1 + e^{-aT}e^{-bT})} \quad (14.1)$$

$$\overline{\epsilon^2} = 1 - \frac{2a}{T} \frac{e^{-2b(\alpha+k)}(1 - e^{-aT}e^{-bT})}{(a+b)^2(1 + e^{-aT}e^{-bT})} \quad (14.2)$$

$$C^*(Z) = \frac{2a}{\gamma} \frac{e^{-b(k+\alpha)}(1 - e^{-aNT_1}e^{-bNT_1})}{(a+b)(1 + e^{-aNT_1}e^{-bNT_1})} \quad (14.3)$$

$$\overline{\epsilon^2} = 1 - \frac{2a}{NT_1} \frac{e^{-2b(\alpha+k)}(1 - e^{-aNT_1}e^{-bNT_1})}{(a+b)^2(1 + e^{-aNT_1}e^{-bNT_1})} \quad (14.4)$$

If the Computer Limited synthesis procedure tabulated in Table VIII were carried out for the program $C^*(Z) = a_0 + a_1 Z^{-1}$, a solution identical to the Data Limited result would be obtained. When the program $C^*(Z) = a_0$ is considered, however, the optimum program and mean squared error expressions are found to be as shown in Table XIV, (14.1) and (14.2), respectively. When the computer complexity—sampling period constraint is applied to these results, $T = NT_1$, and the solutions are as shown in (14.3) and (14.4).

A comparison (13.4) and (14.4) indicates that when $a = 1/NT_1$, the one term program always results in less mean squared error. When $b = 1/NT_1$, the one term program is superior whenever $a > 0.03/T_1$. The one term program is therefore superior under the majority of potential operating conditions. The reason for the success of the Computer Limited program is illustrated in Fig. 8, where the mean squared error corresponding to (13.2) and (14.2), $a = b = 1/NT_1$, $N = 20$, are plotted as functions of T. Note that if T were not related to computer program complexity the two term program would always be superior to the one term program. This is essentially the assumption of Data Limited theory. The one term Computer Limited program superiority arises because the two term program actually imposes the requirement for a system sampling period of $2NT_1$, while the one term program requires a system sampling period of only NT_1.

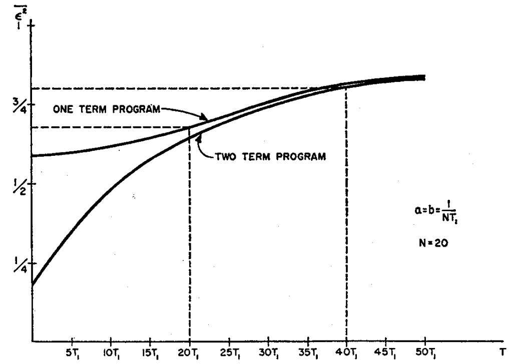

Fig. 8—Mean squared errors—one and two term programs —random inputs.

Further physical insight into the reason for Computer Limited program superiority can be gained by considering the same problem when the system input is an exponential $r(t) = e^{-bt}$ and $\alpha = 0$, $N = 1$. The Data Limited and Computer Limited solutions to this problem are tabulated in Tables XV and XVI respectively.

When $b = 0$, the Computer Limited solution is always superior to the Data Limited solution. The reason for this superiority can be seen from the time responses from the two systems plotted in Fig. 9. Note that after a delay of $2T_1$ the two term program immediately achieves the best response of which it is capable, and that while the one term program response is delayed by only T_1, it requires a transient period to reach its optimum condition. The one term program is superior because its shorter length results in a higher permissible data rate, with a corresponding reduction in achievable integrated error squared. In this particular example there is no penalty associated with the poorer transient response of the one term program, since the time response to a step input extends to infinity, and it is therefore the steady-state response that is being optimized. For this reason the one term program is superior for any finite a.

When b is not zero, so that the system output is exponentially damped, it is possible for the poorer transient response of the one term program to overshadow its higher data rate advantage, in which case the two term program will be superior.

In general, the shorter program has the advantages of a shorter delay before a change in the input is reflected in the output and a higher data rate, and the disadvantage of a poorer transient build-up to the optimum response condition.

The detailed procedures that have been described in this report pertain to the synthesis of linear systems employing linear digital computer programs, in which the minimization of mean squared error or integrated error squared is an effective optimization criterion. It is important to realize that many other classes of Computer Limited sampled data systems exist, and that a great deal of further work is required to extend the basic approach to the Computer Limited problem presented in this report to these problem areas. For example, the problems of simulating nonlinear systems, of utilizing nonlinear computer programs and of employing optimization criteria other than mean squared error minimization present considerable challenge. Many of the actual problems to be simulated by multiplexed real time digital computers are actually nonlinear or can, in specific instances, benefit from the utilization of nonlinear programs. Unfortunately, the sharpness of presently available mathematical tools seems to limit the generality of synthesis techniques applicable to these nonlinear problems.

The solutions presented in this report benefit from the analytic flexibility inherent in the analysis of linear systems. It is submitted that, pending the development of comprehensive nonlinear synthesis techniques, linear theory can provide basic insights into the fundamental problems associated with the synthesis of Computer Limited sampled data systems, and that this fundamental understanding can in turn serve as a guide in the synthesis of systems considerably more complex than those covered by the basic theory.

TABLE XV

Data Limited Solution to Problem Example Deterministic Input

$$C^*(Z) = \frac{2ae^{-bk}(1 - e^{-bT}e^{-aT})}{\gamma(a+b)(1 - e^{-2aT})}(1 - e^{-aT}z^{-1}) \qquad (15.1)$$

$$\int \epsilon^2 = \frac{1}{2b} - \frac{2ae^{-2bk}(1 - e^{-bT}e^{-aT})^2}{(a+b)^2(1 - e^{-2bT})(1 - e^{-2aT})} \qquad (15.2)$$

$$C^*(Z) = \frac{2ae^{-bk}(1 - e^{-2bNT_1}e^{-2aNT_1})}{\gamma(a+b)(1 - e^{-4aNT_1})}(1 - e^{-2aNT_1}) \qquad (15.3)$$

$$\int \epsilon^2 = \frac{1}{2b} - \frac{2ae^{-2bk}(1 - e^{-2bNT_1}e^{-2aNT_1})^2}{(a+b)^2(1 - e^{-4bNT_1})(1 - e^{-4aNT_1})} \qquad (15.4)$$

TABLE XVI

Computer Limited Solution to Problem Example Deterministic Input

$$C^*(Z) = \frac{2ae^{-bk}(1 - e^{-bT}e^{-aT})}{\gamma(a+b)(1 + e^{-bT}e^{-aT})} \qquad (16.1)$$

$$\int \epsilon^2 = \frac{1}{2b} - \frac{2ae^{-2bk}(1 - e^{-bT}e^{-aT})}{(a+b)^2(1 - e^{-2bT})(1 + e^{-bT}e^{-aT})} \qquad (16.2)$$

$$C^*(Z) = \frac{2ae^{-bk}(1 - e^{-bNT_1}e^{-aNT_1})}{\gamma(a+b)(1 + e^{-bNT_1}e^{-aNT_1})} \qquad (16.3)$$

$$\int \epsilon^2 = \frac{1}{2b} - \frac{2ae^{-2bk}(1 - e^{-bNT_1}e^{-aNT_1})}{(a+b)^2(1 - e^{-2bNT_1})(1 + e^{-bNT_1}e^{-aNT_1})} \qquad (16.4)$$

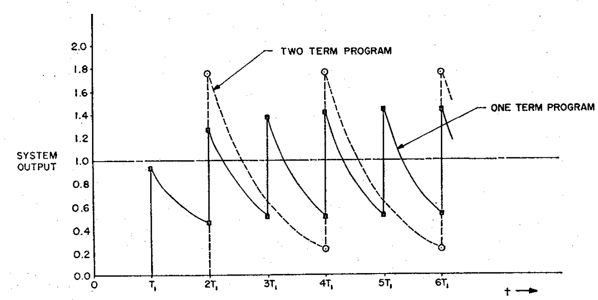

Fig. 9—Data Limited and Computer Limited response for single time constant-output filter—deterministic input.