Control System Digital Computer Transfer Function Simulation

Jens J. Jonson

and

$$c = e^{-20t}(12.5t - 2.754) + 1 \text{ radian, for } t > 0.07 \text{ sec.} \quad (17)$$

This equation is represented by curve V in Fig. 2. For $t \leq 0.07$, the output is given by (15), which is represented by curve III in Fig. 2.

Defining the "response time" as the time required for the output to reach 95 per cent of the ultimate value in response to a position step-function input, it is seen by comparing curve I with curves III and V in Fig. 2, that the response time of the switching system is 60 per cent of the response time of the linear negative-feedback system. The reduction of the response time does not require the introduction of oscillations. Further reduction of response time is possible as explained below.

Comments

As was pointed out above, a rigorous study of the system can be made by the use of the phase space. This study shows the generality of the improvement in response time obtained by using the type of system under discussion. The study provides a simple method to establish the optimum switching instant with respect to a prescribed criterion without trial and error. A common criterion is to obtain the fastest possible response without overshoot. The optimum switching condition in this case is given by a linear relationship between the error and the derivative of the error, independent of the input signal magnitude. This provides for physically simple realizable switching circuitry. The study also shows that the switching system response time can be shortened further by providing less than critical damping during the positive-feedback mode of operation, and more than critical damping during the negative-feedback mode.

Finally, it should be noted that the switching system can be driven into saturation, which results in maximum exploitation of the components. The behavior of the system under this condition can be studied quite easily by the phase-space method, but can also be demonstrated by the differential equation approach used in the present paper by substituting a constant value for the torque in (2).

Control System Digital Computer Transfer Function Simulation*

JENS J. JONSSON†, SENIOR MEMBER, IRE

Summary—**The z transform of simple delay, integrator, lead-lag, and differentiator networks are suggested as the basic building blocks for simulating complex control systems on a digital computer. The system parameters retain their explicit identity, thus facilitating analysis and design of the system as a function of specific parameter changes.**

Introduction

FROM A PRACTICAL standpoint the engineer has found it expedient to avoid the expression of a control system in the form of a differential equation for purposes of design and analysis. Rather he prefers to use the block diagram and the useful tools of root-locus plotting or the Bode presentation because the effect of changing a given system constant remains reasonably apparent; in contrast, investigation of system performance as a function of some parameter through the differential equation is generally an arduous task since the parameter becomes lost in the differential equation constants.

The desire to use digital computers to analyze and design automatic control systems has led to the use of the z transform as a basic mode of investigation. However, there seems to be a propensity on the part of authors on this subject to lump together major portions of the control system (for example, the entire feedforward path or even the feedforward path together with its feedback path) to obtain the z transform of a given system without regard to the desirability of retaining the identity of important parameters.

It is the purpose of this paper to present basic building blocks (wherein the system parameters are quite apparent) which may be interconnected so as to form a complex control system amenable to implementation on the digital computer.

Basic Transfer Function Simulation

Conventional nomenclature, as fairly well established in the sampled-data field, will be employed and therefore

* Received by the PGE, April 13, 1961.
† Visiting Research Professor at the Polytechnic Institute of Brooklyn, N.Y.; on sabbatical leave from Brigham Young University, Provo, Utah.

will not be redefined here.[1-3]

Four basic building blocks, together with their appropriate recursion equations, are presented below. Each is assumed to include a conventional triangular hold circuit with a transfer function

$$H_\Delta(s) = \frac{\epsilon^{Ts}(1 - \epsilon^{-Ts})^2}{Ts^2}. \qquad (1)$$

The Simple Time Delay

Fig. 1(a) shows the transfer function relating $E(t)$ and $C(t)$ for a simple time delay, and Fig. 1(b) is the sampled equivalent.

The z transformation of $H_\Delta(s)$ and $K/(s + \beta K)$ produce the equation

$$C(z) = DC(z^{-1}) + BE(z^{-1}) + AE(z) \qquad (2)$$

which may be written as a function of time as

$$C(nT) = DC(n-1)T + BE(n-1)T + AE(nT), \qquad (3)$$

where

$$A = \left[\frac{1}{\beta} - \frac{1}{T\beta^2 K}(1 - \epsilon^{-K\beta T})\right]$$

$$B = \left[\frac{1}{T\beta^2 K}\left\{1 - (1 + TK\beta)\epsilon^{-K\beta T}\right\}\right]$$

$$D = \epsilon^{-K\beta T}.$$

In words (3) reads as follows:

Current Output at time (nT)
= D[Previous Output at time $(n-1)T$]
+ B[Previous Input at time $(n-1)T$] (4)
+ A[Current Input at time (nT)].

It is evident that this recursion equation lends itself to digital computer programming.

The Integrator

Fig. 2 shows the ideal integrator and its sampled equivalent. It is seen that the circuit is equivalent to $\beta = 0$ for Fig. 1, however, it is not advisable to allow the simple time delay circuit to be reduced in this fashion since it will generate an indeterminate form for constants A and B (a situation which the computer may have trouble evaluating).

The recursion equation is

$$C(nT) = C(n-1)T + B\{E(n-1)T + E(nT)\} \qquad (5)$$

where

$$B = \frac{KT}{2}.$$

[1] E. I. Jury, "Sampled-Data Control System," John Wiley and Sons, Inc., New York, N.Y.; 1958.
[2] J. R. Ragazzini and G. F. Franklin, "Sampled-Data Control Systems," McGraw-Hill Book Co., Inc., New York, N.Y.; 1958.
[3] J. T. Tou, "Digital and Sampled-Data Control Systems," McGraw-Hill Book Co., Inc., New York, N.Y.; 1959.

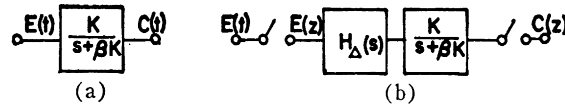


Fig. 1—The simple time delay. (a) Time constant, (b) Sampled equivalent.

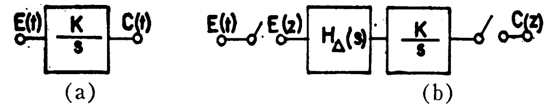


Fig. 2—The integrator. (a) Integrator. (b) Sampled equivalent.

The Lead-Lag or Lag-Lead Network

A zero and pole in combination is shown in Fig. 3, where the recursion equation is

$$C(nT) = DC(n-1)T + JE(n-1)T + IE(nT), \qquad (6)$$

$$I = \left\{\alpha T + \left(1 - \frac{\alpha}{\beta K}\right)\left(1 - \epsilon^{-K\beta T}\right)\right\} \Big/ \beta T$$

$$J = \left\{-\alpha T + \left(-1 + \alpha T + \frac{\alpha}{\beta K}\right)\left(1 - \epsilon^{-K\beta T}\right)\right\} \Big/ \beta T$$

$$D = \epsilon^{-K\beta T}.$$

The constants for the simple time delay, integrator, and foregoing network are seen to be a function of the sampling time T. It may be advisable to include the evaluation of such constants (A, B, D, I, etc.) as an explicit part of the computer program in order that the effect of computing error (*i.e.*, the difference between the exact real-time solution and the sampled digital computer solution) may be investigated as a function of the sampling time. For the pole-zero network under consideration it can readily be demonstrated that the maximum unit error U.E. always occurs at time equal to zero for a unit step input. The unit error is defined as

$$\text{U.E.}(t) = \frac{\text{Exact Real-Time Solution} - C(nT)}{\text{Exact Real-Time Solution}}. \qquad (7)$$

In particular if $E(t)$ is a unit step at zero time the initial (and maximum) unit error is

$$\text{U.E.}(0) = \frac{K - C(0)}{K} = 1 - \frac{I}{K}. \qquad (8)$$

As an aid in selecting the appropriate sampling time the relationship between U.E.(0) and $K\beta T$ is shown in Fig. 4, with the parameter α as a variable. For example, if an initial unit error of 0.045 (4.5 per cent) were acceptable it is necessary that $K\beta T \leq 0.01$. With K and β specified the required T follows axiomatically.

As a general rule a ratio of 30 to 50 is required between T and the shortest system time constant for an accuracy in the order of 1 or 2 per cent.

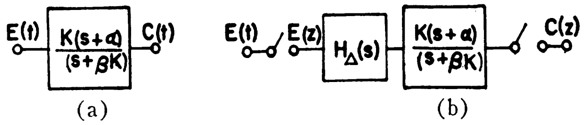
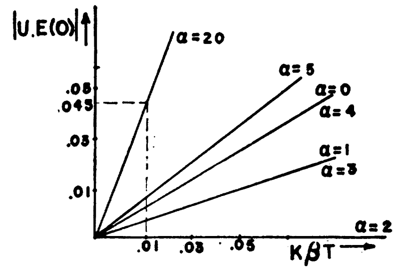


Fig. 3—The pole-zero network. (a) Pole-zero network. (b) Sampled equivalent.



for $0 \lesssim \alpha \lesssim 2$ U.E.(0) is positive
for $\alpha > 2$ U.E. (0) is negative

Fig 4—Normalized unit error vs sampling time (for $\beta K = 2.0$).

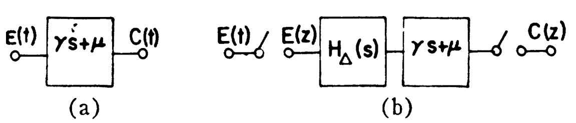


Fig. 5—The differentiator. (a) Differentiator and feedthrough. (b) Sampled equivalent.

The Differentiator

Differentiation is represented by the "γs" term in Fig. 5; simultaneously, the "μ" term indicates a feedthrough path of gain μ. (Such a circuit may represent both positional ($\mu = 1$) and tachometer feedback for a control system.)

The recursion equation is

$$C(nT) = PE(n-1)T + NE(nT), \tag{9}$$

where

$$N = \mu + \frac{\gamma}{T}$$

$$P = -\frac{\gamma}{T}.$$

An Application

Consider the transient closed-loop motor performance of the system shown in Fig. 6. Positional and velocity feedback is assumed; a unit step constitutes the input.

The simulated system is shown in Fig. 7, and the corresponding recursion equations are

$$E(nT) = R(nT) - F(n-1)T$$

$$E_1(nT) = D_1E_1(n-1)T + B_1E(n-1)T + A_1E(nT)$$

$$E_2(nT) = D_2E_2(n-1)T + B_2E_1(n-1)T + A_2E_1(nT)$$

$$C(nT) = D_3C(n-1)T + B_3\{E_2(n-1)T + E_2(nT)\}$$

$$F(nT) = PC(n-1)T + NC(nT), \tag{10}$$

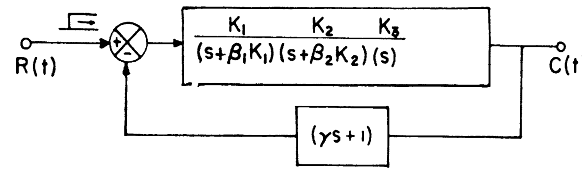


Fig. 6—Exact system.

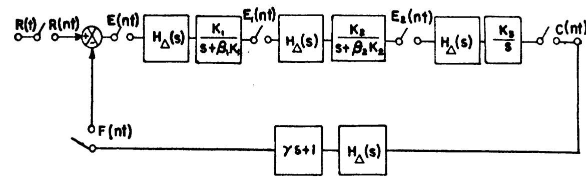


Fig. 7—Simulated system.

where the constants are defined in the foregoing presentation. The system parameters K_1, K_2, K_3, β_1, β_2, and γ are presented to the digital computer as data. Once a complete program is written, the ease with which performance may be investigated as a function of the tachometer gain γ, a forward gain K, or a time constant β is apparent since only a single entry of data (which is identical to the parameter under consideration) need be changed as preparation for a rerun.

Conclusion

The use of basic building blocks is a familiar technique in identifying specific individual functions of a more complex system. If the z transforms of the basic blocks are performed in such a manner that the system parameters retain their identity, design and analysis of a complex system by means of digital computer simulation may be performed with considerable dispatch.

Selection of the sampling time will depend on the accuracy requirements. The system of Fig. 7 has been simulated on an IBM 650 using the indicated rule for determining T with a deviation of 1-2 per cent between the exact real-time solution and the simulated solution, even though the parameters were adjusted to give 50-80 per cent overshoot with a step input.

This approach to control system design should appear attractive not only to the practicing engineer who may be interested in a specific response but it also may be employed as a very effective pedagogic tool in the teaching of automatic control theory. Once a program for a particular system is written by the instructor or student, the effect of inserting any of the various compensating networks can be seen immediately. This would seem to be rather desirable since compensating networks are usually designed in the frequency domain with the expectation that its effect on the real-time response will be understood either in rather general terms or on an intuitive basis. This implies an ability which may not be enjoyed by the novice, who might find it very instructive to see the detailed real-time response.