A Numerical Method for Solving Control Differential Equations on Digital Computers

W. H. ANDERSON, R. B. BALL, J. R. VOSS

Bendix Aviation Corporation, Mishawaka, Indiana

Abstract. Frequently, as in missile control systems, linear differential equations are simultaneous with nonlinear but slower acting differential equations. The numerical solution of this type of system on a digital computer is significantly speeded up by approximating the forcing functions with polynomials, solving the linear equations exactly, and numerically integrating the nonlinear equations with Milne integration. Automatic interval adjustment is possible by comparing errors in the nonlinear integration. The interval selected is related to the shortest time constant of the nonlinear equations rather than the shortest of all the equations. With this system, both detailed transient response and steady state conditions are revealed with a minimum of machine time.

Introduction

The engineering analysis of a missile guidance system led to arrays of simultaneous ordinary differential equations, many of which were too complicated to be solved analytically. Analog computer simulation of these systems has been found quite useful but somewhat limited in some areas, especially where the necessary linearization is unrealistic or where the number of equations is too large.

The guidance system being studied consisted of an electronic and hydraulic control system with a fast response combined with an aerodynamic system which had an inherently slower response.

This system of equations was programmed for a digital computer to be integrated numerically. It was found that the time interval required to achieve reasonable accuracy was very small, making the computer time usage prohibitive for the number of solutions that were needed.

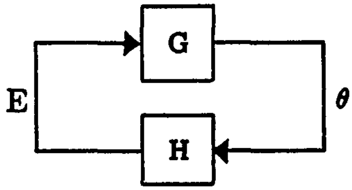
Examination of this equation system pointed out that the time interval required for numerical stability was of the order of the smallest time constant in the fast acting control system, whereas the nonlinearities which made analytic solutions impossible existed in the equations describing the slower acting aerodynamics.

In many cases, it is possible to approximate a control system with a simple time delay without adversely affecting the numerical solution. Replacing differential equations with fast response by a simple time delay allows the numerical integration to proceed at a time interval of the order of the shortest time constant of the aerodynamic part of the guidance system. However, the effect of transients in the control system is ignored by so doing and significant errors would have been made in our particular study.

The method herein described was developed to overcome these problems, allowing a complete transient analysis of the guidance system with a reasonable amount of computer time.

Description of Method

Typically, a missile guidance system can be represented by



where

H is a set of nonlinear ordinary differential equations which describe a controlled mechanism,

G is a set of linear ordinary differential equations with constant coefficients which describe a controlling device with a fast response in comparison with H,

E is the error function,

θ is the correction function.

The control system differential equations are of the form

$$A_j \overset{(j)}{\theta} + \cdots + A_0 \theta = F(E, \overset{(\bullet)}{E}, \cdots, \overset{(i)}{E})$$

Because the aerodynamic system is relatively slow acting, the function F can be accurately approximated by a low order polynomial in time, that is,

$$F\,(E, \overset{(\bullet)}{E}, \cdots, \overset{(i)}{E}) \approx p(t).$$

If the polynomial p is substituted for F as the forcing function, the differential equation can be solved analytically. Thus, the dependent variable θ can be found as

$$\theta = \Sigma B_j e^{-t/\tau_j} + P(t),$$

where the first term consists of a transient response and the second term is a polynomial in time. The coefficients of both terms can be calculated directly from the polynomial $p(t)$ and from the coefficients of the differential equation. In principle, then, the function θ can be derived analytically from the original differential equation and a suitable polynomial approximation to the forcing function. Further, it would be possible to feed the function θ into the nonlinear equations to solve for the forcing function F by means of numerical integration.

In this application, it was necessary to cause the analytic solution for θ to conform to the step-by-step evaluation required by the numerical integration of the nonlinear equations. By the same token, the information from which the forcing polynomial, p, must be derived is available at discrete points. With these simplifications it becomes possible to write the solution in the following form:

$$\theta_{n+1} = \alpha_1\theta_n + \alpha_2\theta_{n-1} + \cdots + \alpha_j\theta_{n-j+1} + \beta_1 E_{n+1} + \beta_2 E_n + \beta_3 E_{n-1} + \cdots + \beta_{k+1} E_{n-k+1}$$

where:

$\theta_{n+1} \equiv \theta(t_n + h)$, etc.,
j is the order of the differential equation,
k is the degree of the forcing polynomial,
h is the time interval between adjacent calculated points.

In this expression the α's and β's depend on the coefficients of the differential equation and the interval h. That is, for a given interval of integration the α's and β's will be constant.

EXAMPLE. A simple control system might be represented by

$$\tau\dot{\theta} + \theta = E \tag{1}$$

(or $G = 1/(\tau S + 1)$ in Laplace notation).

Assume that E has been calculated through time t_{n+1} and that θ has been calculated through time t_n (where t_{n+1} is defined as $t_n + h$). Using the three latest values of E, a quadratic function of time can be derived to approximate E of the form

$$E = K_1 + K_2(t - t_n) + K_3(t - t_n)^2$$

where

$$K_1 = E_n\,,$$

$$K_2 = \frac{1}{2h}(E_{n+1} - E_{n-1})\,,$$

$$K_3 = \frac{1}{2h^2}(E_{n+1} - 2E_n + E_{n-1})\,.$$

Equation (1) becomes

$$\tau\dot{\theta} + \theta = K_1 + K_2(t - t_n) + K_3(t - t_n)^2 \tag{2}$$

which has the analytic solution

$$\theta = Ae^{-(t-t_n)/\tau} + C_1 + C_2(t - t_n) + C_3(t - t_n)^2. \tag{3}$$

Then, differentiating,

$$\dot{\theta} = -\frac{A}{\tau}e^{-(t-t_n)/\tau} + C_2 + 2C_3(t - t_n)\,. \tag{4}$$

To solve for the C's, equation (3) and equation (4) are substituted into equation (2) and like coefficients are equated. Thus:

$$C_1 + C_2\tau = K_1$$

$$C_2 + 2C_3\tau = K_2$$

$$C_3 = K_3$$

or

$$\left.\begin{aligned} C_1 &= K_1 - \tau K_2 + 2\tau^2 K_3 \\ C_2 &= K_2 - 2\tau K_3 \\ C_3 &= K_3 \end{aligned}\right\} \tag{5}$$

and

$$C_1 = E_n - \frac{\tau}{2h}(E_{n+1} - E_{n-1}) + \frac{\tau^2}{h^2}(E_{n+1} - 2E_n + E_{n-1})$$

$$C_2 = \frac{1}{2h}(E_{n+1} - E_{n-1}) - \frac{\tau}{h^2}(E_{n+1} - 2E_n + E_{n-1})$$

$$C_3 = \frac{1}{2h^2}(E_{n+1} - 2E_n + E_{n-1})$$

To obtain the coefficient A, substitute $t = t_n$ into equation (3) and obtain

$$\theta_n = A + C_1$$

or

$$A = \theta_n - E_n + \frac{\tau}{2h}(E_{n+1} - E_{n-1}) - \frac{\tau^2}{h^2}(E_{n+1} - 2E_n + E_{n-1}).$$

To get the next needed value for θ, that is, θ_{n+1}, the value t_{n+1} must be substituted into equation (3) with the values of the C's found in equation (5). The resulting expression will relate the value of θ_{n+1} to a combination of terms containing, θ_n, E_{n+1}, E_n, E_{n-1}.

$$\begin{aligned} \theta_{n+1} = \theta_n e^{-h/\tau} &+ E_{n+1}\left[-\left(\frac{\tau^2}{h^2} - \frac{\tau}{2h}\right)e^{-h/\tau} + \frac{\tau^2}{h^2} - \frac{\tau}{2h} + 1\right] \\ &+ E_n\left[\left(-1 + \frac{2\tau^2}{h^2}\right)e^{-h/\tau} - \frac{2\tau^2}{h^2} + \frac{2\tau}{h}\right] \\ &+ E_{n-1}\left[-\left(\frac{\tau^2}{h^2} + \frac{\tau}{2h}\right)e^{-h/\tau} + \frac{\tau^2}{h^2} - \frac{\tau}{2h}\right]. \end{aligned}$$

Generalization

In general, as in this example, the new value θ_{n+1} is a linear expression of the previous values of θ and the known values of the forcing variable where the coefficients of this linear expression depend only on the coefficients of the linear differential equation, the time interval h, and the degree of the forcing polynomial.

It is important to note that while the coefficients in the linear expression are functions of the differential equation coefficients, the accuracy of the solution depends only on how well E can be fit by the polynomial in time. In the example, if E is a quadratic from t_{n-1} to t_{n+1}, then the solution is exact. During transient periods a small interval is used to insure a good fit by the polynomial.

This method may be applied to a linear differential equation with constant coefficients of any order, and the degree of the polynomial fit can be varied. However, the complexity of the derivation and evaluation of the coefficients increases considerably as the order of the equation increases. Fortunately, this type of equation can be broken up into a series of second order equations where the numerical solution of one equation serves as the forcing function of the next. The selection of the degree of the polynomial depends on the accuracy required; a quadratic polynomial will suffice in most cases.

The solution of the equation

$$L_1\ddot{\theta} + L_2\dot{\theta} + \theta = A_0E + B_0\dot{E} + C_0\ddot{E}$$

takes the form

$$\theta_{n+1} = \alpha_1\theta_n + \alpha_2\theta_{n-1} + \beta_1E_{n+1} + \beta_2E_n + \beta_3E_{n-1} \tag{6}$$

where

$$\left.\begin{aligned}\alpha_1 &= \left[\exp\left\{-h\Big/\left(\frac{L_2}{2} + \frac{1}{2}\sqrt{L_2^2 - 4L_1}\right)\right\}\right] \\ &\qquad + \left[\exp\left\{-h\Big/\left(\frac{L_2}{2} - \frac{1}{2}\sqrt{L_2^2 - 4L_1}\right)\right\}\right] \\ \alpha_2 &= -\left[\exp -h\Big/\left(\frac{L_2}{2} + \frac{1}{2}\sqrt{L_2^2 - 4L_1}\right)\right\}\Bigg] \\ &\qquad \cdot\left[\exp\left\{-h\Big/\left(\frac{L_2}{2} - \frac{1}{2}\sqrt{L_2^2 - 4L_1}\right)\right\}\right]\end{aligned}\right\}, \quad L_2^2 \geqq 4L_1$$

$$\left.\begin{aligned}\alpha_1 &= 2\left[\exp\left\{-hL_2/2L_1\left(\cos\frac{h\sqrt{4L_1 - L_2^2}}{2L_1}\right)\right\}\right] \\ \alpha_2 &= -[\exp(-hL_2/2L_1)]^2\end{aligned}\right\}, \quad L_2^2 \geqq 4L_1$$

$$m_3 = 1 - \alpha_1 - \alpha_2$$
$$m_4 = 1 + \alpha_2$$
$$m_5 = 1 - \alpha_2$$

$$\beta_1 = \frac{m_3}{2h}\left\{(B_0 - L_2A_0) + \frac{2}{h}[C_0 - L_2(B_0 - L_2A_0) - L_1A_0]\right\} + m_4\left\{\frac{A_0}{2} + \frac{1}{h}(B_0 - L_2A_0)\right\} + m_5\left\{\frac{A_0}{2}\right\}$$

$$\beta_2 = m_3\left\{A_0 - \frac{2}{h^2}[C_0 - L_2(B_0 - L_2A_0) - L_1A_0]\right\} + m_4\left\{-\frac{2}{h}(B_0 - L_2A_0)\right\} + m_5\{-A_0\}$$

$$\beta_3 = \frac{m_3}{2h}\left\{-(B_0 - L_2A_0) + \frac{2}{h}[C_0 - L_2(B_0 - L_2A_0) - L_1A_0]\right\} + m_4\left\{-\frac{A_0}{2} + \frac{1}{h}(B_0 - L_2A_0)\right\} + m_5\left\{\frac{A_0}{2}\right\}$$

Method of Application

This method lends itself nicely to digital computer applications. The nonlinear equations must be separated from the linear equations and solved by a regular numerical integration. Since the coefficients of equation (6) require successively evenly spaced intervals, Milne Integration, a compatible system, was selected to integrate the nonlinear differential equations. To achieve the desired result of large enough intervals of integration to keep the computer time reasonable as well as small enough intervals to accurately calculate transient conditions, it was necessary to use a varying time interval selected automatically by means of an error check in the Milne Integration. The coefficients from equation (6) required for the variety of time intervals that may be used during the course of the integration can be calculated initially and stored in the computer for use as needed.

To start the integration, a very small interval is used and then is successively doubled until a desirable interval is reached as indicated by the error check. Although it is straightforward to increase the interval by doubling in both Milne Integration and equation (6) by just dropping out alternate values, it is quite difficult to cut the interval down. When the detected error becomes too large, the program restarts from the last accurate point.

The stability of this overall system depends only on the stability of the nonlinear equations integrated by Milne Integration. To stabilize these equations it was found helpful to iterate when the errors were detected as significant.

Results

This method of numerical integration has been used successfully in a major engineering guidance system study. To mathematically simulate the control system it was necessary to include drastic discontinuities in the control system differential equations. For engineering convenience, there were some parameter discontinuities in the nonlinear differential equations. The system of numerical integration selected large intervals for integration while the variables were proceeding smoothly, but restarted when the integration became more difficult. Since restarts were always initiated when discontinuities occurred in either type of equations even though the errors were detected only in the nonlinear equations, it was felt that transients occurring for any reason were detected by this system. Through the use of this method, it was possible to reduce the computer running time from fifty hours to one hour retaining all of the transient characteristics. An IBM 650 Computer System was used to perform the calculations.

As a by-product of this effort, it was found useful to use equation (6) to determine the response of a given control system to any given forcing function. This system replaced the less accurate Tustin's method formerly used by this group.

Conclusions

On the basis of the success of this system on this application, it is planned to program it in a general way to extend it to other sets of differential equations.

The method has room for improvement by increasing the stability of the Milne Integration, c.f. [1]. In particular, if iteration could be eliminated, the computer time would be cut down even further. The accuracy criterion appeared to be adequate to insure the accuracy of the solution regardless of problems of stability or discontinuities.

Acknowledgement

The method described here is an extension of "Reddy's Method" as described in his paper [2], which had very limited circulation.

Summary

A typical control system might well consist of a high speed electronic controlling mechanism working with a slower nonlinear mechanical system. A mathematical simulation of this system usually consists of a set of simultaneous differential equations. Because of their nature, high speed control systems can usually be represented as linear differential equations with constant coefficients. Frequently, the mechanical system cannot be linearized so that the entire set of differential equations must be considered nonlinear. If size, accuracy and nonlinear limitations prohibit the use of an analog computer, a digital simulation is required. However, a normal digital computer solution which would include the fast acting linear control equations might lead to a prohibitive solution time. Thus, many digital simulations neglect the effect of high speed transfer functions, in effect assuming that their transient terms are negligible. This paper describes a method of including these high speed transfer functions in a digital simulation with only a small increase in computer time.

Linear control system differential equations can be written:

$$A_j \overset{(j)}{\theta} + \cdots + A_0 \theta = F(E, \overset{(\cdot)}{E}, \cdots, \overset{(i)}{E})$$

where the A's are constant.

This can be solved exactly for θ if F is in a simple form. If we approximate the forcing function F by a polynomial in time, θ can be solved analytically. This analytical solution can be manipulated into a step-by-step procedure which is compatible with other numerical methods of integrating differential equations.

Milne Integration is used to integrate the nonlinear differential equations. In this solution the stability of the differential equations is the basic limitation on the time interval. To stabilize the Milne Integration, an iteration technique was used and found to be quite profitable. In order to start the Milne Integration without special starting formulas, it is required that the first few time intervals be very small. As enough intervals are calculated and as transient conditions disappear, the time interval is successively doubled until it is of the order of the time constants in the nonlinear equations. Since Milne Integration allows the truncation error to be estimated, it is possible to establish an error criterion and automatically adjust the interval of integration so that the error is kept within predetermined bounds. It is important to note that the integration interval selected

will depend on the short time constants of the linear equations only during transient periods. During most of the calculations, the time constants in the nonlinear equations will determine the integration interval. However, when the error criterion indicates that the time interval is too large, a restart is initiated at a smaller time interval. Therefore, the larger time intervals which result from using this method do not detract from the accuracy of the solution.

This technique has been programmed and applied to a specific missile control problem with the result that a solution which previously required 50 hours of machine time now is completed in less than one hour.

REFERENCES

1. W. E. MILNE AND R. R. REYNOLDS, *Stability of a numerical solution of differential equations*, *J. Assoc Comp. Mach.* *6* (1959), 196–203.
2. Memo from R. B. REDDY to A. G. Carlton, The numerical integration of differential equations containing small time constants, JHU Applied Physics Laboratory, April 6, 1954