TRAJECTORY OPTIMIZATION

USING FAST-TIME REPETITIVE COMPUTATION

James S. Raby
Ames Research Center, NASA, Moffett Field, Calif.

and

Rodney C. Wingrove
Ames Research Center, NASA, Moffett Field, Calif.

ABSTRACT

A perturbation method of computing optimum trajectories is described. This method uses fast-time repetitive computations in determining control impulse response functions and requires only dynamic solutions of the state equations. The solution of additional linear adjoint equations is not required.

Both a hybrid and a digital computer mechanization of this impulse response method are described. Sample solutions for each computer mechanization are presented, using the same problem formulation in each case. The problem constraints were that a reentry vehicle travel a specified range, that the control remain within specified limits, and that the heat input to the vehicle be minimized. A comparison is made between the hybrid computer and the digital computer mechanizations, each computed near optimum trajectories in about 2 minutes and full optimum trajectories in about 5 minutes of computer time.

INTRODUCTION

Space vehicle trajectories must be near optimum in the sense that some parameter is either a maximum or a minimum; for example, in reentry the trajectory to desired terminal conditions is near optimum when the total aerodynamic heating is a minimum. Several perturbation methods,[1] such as the calculus of variations, applications of the maximum principle, and direct steepest descent, have been considered for determining the time histories of nonlinear controls that correspond to optimum trajectories.

The computations[2] in these previous optimization studies involved the dynamic solution of two sets of equations: (1) nonlinear state equations and (2) linear adjoint equations. An alternate perturbation computation technique - the impulse response method[3] - will be discussed here. This method differs from previous studies in that only the solution of the nonlinear state equations is used. The response of given functions (e.g., terminal error or quantity to be optimized) to a control impulse is determined along the trajectory by fast-time repetitive computations rather than by a solution of adjoint equations. This impulse response method enables the investigator to retain an intuitive understanding of the optimization process. Furthermore, since adjoint equations are not required, the state equations or cost functions need not be amenable to linearization. The impulse response method does require many solutions of the state equations; however, the programming is straightforward and the task of computing a large number of dynamic solutions is ideally suited to modern high-speed computers.

[redacted]

NOTATION

The following notation is used in the body of the text. Additional symbols are described as they are introduced.

L/D	control value of lift-drag ratio
n	number of storage points in control time history
t	time
t_f	final time
t_o	initial time
Δt	time increment of control impulse
u	control
Δu	height of control impulse
φ	cost at final time
$\Delta\varphi(t)$	change in cost at final time due to control impulses at time t
ψ	state value at final time
ψ_d	desired state value at final time
$\Delta\psi(t)$	change in state value at final time due to control impulses at time t

GENERAL OUTLINE OF METHOD

The impulse response method, as discussed in this report, uses the steepest descent optimization process.[4-7] The process commences with any nonoptimal trajectory from which a slightly improved one is derived. The improved trajectory is then used as a new nominal trajectory, and the procedure is repeated until the optimum or nearly optimum trajectory is found.

The iterative procedure is: (1) estimate a reasonable nominal control program; (2) determine impulse response functions that indicate the best method of making small changes in the control that will decrease the cost (the quantity to be minimized); (3) compute a new nominal control by adding this change in control to the previous nominal control (this results in a new trajectory with a decreased cost); (4) repeat step 2. This iterative process continues until the change in cost for each new trajectory is very small; the control is then very near a local optimum. If at any point along the trajectory a limit value of the control is reached before the cost is completely minimized, no further optimization is possible at that point. In this case, the process continues until at each point on the trajectory either a local optimum or the control limit is reached.

Computation of Impulse Response Functions

The technique by which the impulse response function is determined is the most important feature of the impulse response method. Figure 1 illustrates the manner in which the influence of small control changes on the cost are calculated. The equations of motion are first solved with a positive control impulse at time t superimposed upon the nominal control. During the next solution of the equations of motion, a negative control impulse of the same magnitude is inserted at time t. The impulse response, $\Delta\varphi$, is derived from these two solutions. In a similar manner, the impulse response is determined at successive times along the trajectory. The impulse response function, $\Delta\varphi(t)$, is the complete time history of $\Delta\varphi$. Its computation for the same control impulse at

Fig. 1

different times along the trajectory is defined as one iteration. This corresponds to previous optimization studies[4-7] that used one iteration of the adjoint equations to compute essentially the same impulse response function along the trajectory.

Calculation of Minimum Cost

When the cost is to be minimized and there is no terminal constraint, the impulse response function is used in the steepest descent technique to modify the control toward the optimum in the following manner:

$$\begin{pmatrix}\text{New}\\ \text{Nominal}\\ \text{Control}\end{pmatrix} = \begin{pmatrix}\text{Previous}\\ \text{Nominal}\\ \text{Control}\end{pmatrix} + K_\varphi \, \Delta\varphi(t) \tag{1}$$

The gain K_φ weights the impulse response function for the cost; its sign is negative to decrease the cost. The magnitude of K_φ is determined experimentally for each problem: too large a gain may cause instability in the convergence procedure, while too small a gain may extend the time of convergence.

A representative sample of what one may expect with this type of optimization procedure is sketched in figure 2. During the first iteration, the repetitive solutions determine the impulse response function, $\Delta\varphi(t)$. This $\Delta\varphi(t)$ is added to the nominal control with an appropriate gain K_φ, and the new nominal control time history, as shown in the center of figure 2, is obtained. The iterative process is repeated until the optimum control is reached. The optimum control may take on either or both of the properties illustrated in the final iteration of figure 2. In the region (A) the impulse response function is, for all practical

Fig. 2

purposes, zero. This implies that a small change in control in this region will not modify the cost; thus the control is at a local optimum. In region (B) the control is at the limiting constraints, and the impulse response function indicates that only control beyond the constraint will decrease the cost. Thus, on the constraint, the control is at a local optimum.

Minimum Cost With Terminal Constraint

When a terminal constraint (or destination) is to be reached, while minimizing the cost, the iteration procedure is performed as follows:

$$\begin{pmatrix}\text{New}\\ \text{Nominal}\\ \text{Control}\end{pmatrix} = \begin{pmatrix}\text{Previous}\\ \text{Nominal}\\ \text{Control}\end{pmatrix} + K_\varphi\, \Delta\varphi(t) + K_\psi\, \Delta\psi(t) \qquad (2)$$

where ψ denotes the state variable at the final time. The quantity $\Delta\psi(t)$ represents the change in the state variable due to control impulses, and is evaluated in the same manner as $\Delta\varphi(t)$.

Gains K_φ and K_ψ are constants for each iteration. Gain K_φ weights the impulse response function for cost; its sign is negative to decrease the cost. Gain K_ψ must be calculated for each iteration so that the term $K_\psi\, \Delta\psi(t)$ will account for terminal displacement due to the optimizing term, $K_\varphi\, \Delta\varphi(t)$, and correct any terminal displacement error from the previous iteration.

The equation for K_ψ is:

$$K_\psi = \underbrace{-K_\varphi \frac{\int_{t_o}^{t_f} \Delta\varphi(t)\Delta\psi(t)\,dt}{\int_{t_o}^{t_f} \Delta\psi^2(t)\,dt}}_{\text{Steepest descent optimization term}} + \underbrace{2\,\Delta u\,\Delta t \frac{\psi_d - \psi}{\int_{t_o}^{t_f} \Delta\psi^2(t)\,dt}}_{\text{Terminal error correction term}} \qquad (3)$$

The derivation of this equation can be found in appendix A. Δu is the height of each control impulse; Δt is the time interval of each control impulse; ψ_d is the desired end-point value; and $\psi_d - \psi$ represents any terminal displacement error from each previous iteration. This equation gives the general form of the steepest descent computations; the computer mechanization of this method will be discussed next.

COMPUTER MECHANIZATION AND RESULTS

The impulse response method has been mechanized on both a hybrid and a digital computer to determine the optimal time history of the lift-drag ratio (control L/D) that must be flown for a vehicle returning into the earth's atmosphere. The example problem requires that (1) the cost, φ, which is the heat input to the vehicle, be minimized, (2) the vehicle arrive at a terminal constraint, ψ_d, (destination), and (3) the control time history remain within specified limits. The solution to this particular problem is known a priori to be a bang-bang control; therefore, the final results can be verified.

The equations of motion are presented in appendix B. The vehicle characteristics and flight conditions were those of a manned capsule returning from earth orbit and having the following parameters:

Initial conditions:	altitude	250,000 ft
	horizontal velocity	25,000 fps
	vertical velocity	748 fps
	range to destination	1,000 miles
Stopping conditions:	altitude	100,000 ft
Control limits:	L/D	$0 \leq L/D \leq 0.5$

The computer systems used in the two mechanizations are described in appendix C.

Hybrid Computer Mechanization

The major elements of the hybrid computer consisted of: (1) an analog computer to solve the trajectory equations, (2) parallel digital logic units to control the computer program, (3) delay line memories to store the control time history, and (4) D-A and A-D converters to transfer the control time history between the analog computer and the delay line memory.

The L/D time history was stored in 64 word serial delay line memories with a resolution of 13 bits. The access time of the serial memory was 128 μsec. To permit a complete solution of the trajectory equations within the 128 μsec, the analog computer was time-scaled at 3750 to 1.

The mechanization of this problem on the hybrid computer is illustrated in figures 3 and 4: Figure 3 is the problem flow chart, and figure 4 illustrates the logic used in controlling the problem. Figs. 3 and 4 The serial memory unit is continuously driven by counter pulses (Logic No. 1). The output of the serial memory is the nominal control time history with n points. This time history is used, together with the appropriate control impulse, to solve the trajectory equations. These equations are started at the specified initial conditions with Logic No. 2, and stopped with Logic No. 3 when the trajectory reaches the specified end condition on altitude. The final values of the cost quantity (heat) and the state quantity (range) are stored at the end of each run as indicated by Logic Nos. 4 and 5. The positive or negative control impulse is added to the nominal control input with Logic Nos. 6 and 7, respectively. Logic No. 8 inserts the modifying control $(K_\varphi \, \Delta\varphi(t) + K_\psi \, \Delta\psi(t))$ into the serial memory. This procedure runs in essentially a continuous manner, that is, one point out of the n points in the nominal control history is updated after each two repetitive computations. After $2n$ repetitive computations (one iteration), every point in storage has been modified and the process is repeated. For each iteration, gains K_φ and K_ψ are held constant. As previously mentioned, gain K_φ determines the relative speed and stability of the convergence onto the optimum. The corresponding value of K_ψ to be used with each new iteration is calculated by equation (3) as a function of the terminal error from each previous iteration $(\psi_d - \psi)$ and the following two integrated quantities from each previous iteration:

$$\int_{t_o}^{t_f} \Delta\varphi(t)\Delta\psi(t)\,dt \tag{4}$$

and

$$\int_{t_o}^{t_f} \Delta\psi^2(t)\,dt \tag{5}$$

Time t_o was represented by a logic signal at the first repetitive computation in an iteration cycle and time t_f was represented by a logic signal at the last computation in an iteration cycle. It should be noted that during those parts of the trajectory when the control was at a constraint limit, no further optimization was possible and the integration of equations (4) and (5) was not carried out during those times.

Hybrid Computer Results

The results obtained from the hybrid simulation are illustrated in figures 5 and 6. Figure 5 shows a portion of one iteration, while figure 6 shows the convergence to the optimum control L/D. Figs. 5 and 6

In the upper trace of figure 5, the control impulses are superimposed upon the initial nominal control. Each control impulse had a magnitude of L/D = ±0.25 and a time increment of one clock pulse (0.002 sec). This control impulse was chosen because it gave variation in the final range and heat load on the order of ±5 percent. The integrated heat loads along each of the repetitive trajectories are presented in the next trace. The difference between the final quantities for each pair of subsequent runs is $\Delta\varphi$, and represents the heat load impulse response.

In figure 6, the first few iterations of the converging optimization procedure are illustrated together with the thirtieth iteration. In the upper trace the nominal control is recorded as it is read out of serial memory every 128 +1 counter pulse (with Logic No. 8). This gives a convenient time history to show the manner in which the control has been modified during each iteration. Notice that the control is limited within

$0 \leq L/D \leq 0.5$. This was achieved by simply limiting the output of the serial memory to these values. The modifying control shown in the lower trace of figure 6 is the sum, $K_\varphi \, \Delta\varphi(t) + K_\psi \, \Delta\psi(t)$. For this series of runs, a constant $K_\varphi = -2.5\times10^{-3}/\text{Btu/ft}^2$ permitted fairly rapid convergence while program stability was maintained. The value of K_ψ was calculated for each iteration by equation (3) to be that value which kept the final value of range near 1,000 miles.

As can be seen in figure 6, the optimum control variation for this particular example was a bang-bang control. With the steepest descent method, it was found that near-optimum control could be achieved in the first few iterations, but to "square up the corner" and achieve full optimum control required more iterations (20 to 30).

Digital Computer Mechanization

The major elements of the digital computer system consisted of: (1) a digital computer to solve the trajectory equations, perform the logical control of the program, and store the control L/D time histories; (2) a line printer to print hard copies of the results; and (3) D-A converters and a strip chart recorder for fast observation of trends.

The digital program was written in floating point symbolic language. Since the optimization technique requires repetitive computation of the trajectory, the choice of an integration routine was very important. A fast, stable, and fairly accurate routine was needed. These requirements conflict to some extent;[8,9] however, the fourth-order Adams-Bashford integration algorithm gave satisfactory results at a step size of 5 seconds, provided a satisfactory starter was used. The starter used the lower order Adams-Bashford algorithms with a step size of 1 second.

The program flow is as follows (see fig. 7): (1) A nominal control time history is used to calculate the nominal trajectory. (2) This trajectory is stored for use as the initial conditions for the repetitive computations of the trajectory. (3) At the initial point along the nominal trajectory, the control is perturbed with a positive pulse, and a new trajectory is calculated. (4) At this same point on the trajectory, the control is perturbed with a negative pulse and another new trajectory calculated. (5) From these two repetitive computations of the trajectory the heat impulse response, $\Delta\varphi$, and the range impulse response, $\Delta\psi$, are calculated. (6) The program is then advanced to new initial conditions along the nominal trajectory by the length of the integration step size. (7) Steps (3) through (6) are repeated until the initial altitude reaches the stopping condition (100,000 ft). (8) At this time, a new nominal control time history is computed using equations (2) and (3). (9) Steps (1) through (8) are repeated. This iterative computation continues until an optimum trajectory is reached. Fig. 7

Digital Computer Results

The first five iterations and the twentieth iteration of the digital simulation are illustrated in figure 8. The upper trace of figure 8 shows the control L/D time history. During the first iteration, the control L/D was a constant 0.25; at the end of this iteration it was modified by equations (2) and (3). By the fifth iteration the control L/D was approaching bang-bang and by the twentieth iteration it was essentially bang-bang. The pulse used to perturb the trajectory had a height of 0.25 L/D and a width equal to one integration Fig. 8

step size. For this pulse, a constant value of $K_{\varphi} = -7.5 \times 10^{-2}/\text{Btu/ft}^2$ permitted a fairly rapid convergence and the computation remained quite stable.

The second trace of figure 8 shows the variation in heat from one iteration to the next. The heat, which is the cost in this example, decreases markedly during the first five iterations and nearly reaches its final value by the end of the fifth iteration. Tables I, II, and III give the results in tabular form. The range is shown to remain near 1,000 miles while the heat is reduced from 23,491 Btu/ft^2, at the end of iteration 1, to 21,517 Btu/ft^2 at the end of iteration 5. The major change during iterations 5 through 20 was to "square up" the L/D control and achieve the full optimum control. At the end of iteration 20, the final range achieved was 999.9 miles and the heat 20,966 Btu/ft^2. During the optimization procedure the range varied slightly about the desired value of 1,000 miles and the heat load was reduced about 10 percent.

Discussion of Hybrid and Digital Results

It was interesting to observe that both the hybrid and the digital simulations required approximately the same amount of computer time, approximately 2 minutes to obtain near optimum trajectories and approximately 5 minutes to obtain full optimum trajectories. However, it should be pointed out that no real attempt was made to minimize either of these computing times. There are several methods for reducing the computer time required to obtain optimum trajectories. One method would be to select the gain K_{φ} automatically for each iteration instead of using a constant value for the entire computing run. This would cause the solution to converge to an optimum in fewer iterations at the expense of complicating the computer program. Another method of decreasing the computation time would be to decrease the number of points used to store the control time history which would decrease the number of repetitive computations required for each iteration.

The results obtained by both the hybrid and the digital computer appear satisfactory for engineering purposes. The final values of range and heat computed by the two simulations agree to within approximately 1 percent and both simulations arrived at the same bang-bang control time histories.

One excellent feature of the digital simulation was the program documentation obtained by using the on-line typewriter and line printer. The typewriter documented every change made during the time the program was in the computer, and the line printer permitted the analysis of each variable at specific points along the trajectory. Equally valuable was the strip chart recording normally obtained in hybrid computation. It was obtained in the digital program by D-A conversion of the digital variables. This "quick look" capability made it possible to observe trends not readily apparent in numerical printouts.

The result of this test example was no surprise. In simulations that require complicated logic control of the program and a moderate amount of storage, there is a distinct advantage to using a digital computer. It proved reasonable to use a digital computer in this simulation because there was only a moderate number of simplified equations to be solved. If the number of equations were increased, the time to solve them on the digital computer would, of course, also increase.

COMPARISON WITH ADJOINT STEEPEST DESCENT

A current reentry optimization study at Ames Research Center is using both the impulse response method of this report and the standard adjoint steepest descent computing method. This study is of interest because the two methods have been programmed on the same computer (IBM 7094) and their ability to solve several identical problems has been compared.

Representative solutions obtained from the two methods are illustrated in figure 9. This particular example is for the same reentry vehicle and initial flight conditions used in the previous example of this report. However, the cost function is of the form: Fig. 9

$$\varphi = \int_{t_o}^{t_f} [(\text{Heat rate}) + (\text{Drag})^2]dt$$

and there is no terminal constraint. This was chosen in order to illustrate a problem formulation that does not represent a bang-bang optimal control result.

The results of the twentieth iteration are shown in figure 9. The upper curve shows that the control solutions are almost identical. In the lower curve the impulse response function $\Delta\varphi(t)$ has been normalized[3] for comparison with the corresponding results obtained by the adjoint solutions. Figure 9 demonstrates that the two methods arrive at essentially the same final solution.

For reentry problems similar to the one presented herein, it has been found that the computing time required with the adjoint method is about one order of magnitude less than that required by the impulse response method. Because the adjoint method uses less computer time, it has been the more desirable method for production runs that require a large number of optimized trajectories. However, because the impulse method is straightforward to program and because the engineer is able to retain an intuitive understanding of the optimization procedure, the impulse method has been the more desirable method for initial problem mechanization. Furthermore, adjoint equations require linearization and, therefore, cannot be used in some problem formulations. For example, in reentry problem formulations with complicated heat-balance equations,[10] rather than the simple heating expression shown in appendix B, the heat rate cannot be linearized. In this type of formulation, the impulse response method has provided the only practical solution.*

---

*Dynamic programming was also tried for this problem but the computer time was found to be excessive, one to two orders of magnitude greater than that required with the impulse response method.

CONCLUSIONS

This paper has described reentry trajectory optimization using the impulse response method. The method requires that the computer perform a large number of fast-time repetitive computations in solving the state equations and in determining impulse response functions. These repetitive computations are readily performed by both hybrid and digital computers.

The mechanization of the impulse response method on both hybrid and digital computers was found to be straightforward. Near optimum reentry trajectories were obtained in approximately 2 minutes and full optimum reentry trajectories in approximately 5 minutes of computer time. The solutions obtained from either mechanization agreed to within approximately 1 percent.

The impulse response method has been compared with the adjoint steepest descent method. The solutions obtained by either method were essentially identical. The adjoint method requires less computer time; however, the impulse response method does not require familiarization with or use of an auxiliary set of linear adjoint equations. Furthermore, for problem formulations that are not amenable to linearization, the impulse response method may be the only practical method.

APPENDIX A

DERIVATION OF EQUATION FOR K_ψ

Along a normal trajectory, small changes, $\delta\psi$, in the terminal state due to small changes, $\delta u(t)$, in control can be approximated by:

$$\delta\psi = \frac{1}{2\,\Delta u\,\Delta t}\int_{t_o}^{t_f} \delta u(t)\Delta\psi(t)\,dt \tag{A1}$$

where Δu is the height of each control impulse and Δt is the time interval of each control impulse. Substituting $K_\varphi\,\Delta\varphi(t) + K_\psi\,\Delta\psi(t)$ from equation (1) for $\delta u(t)$, we have:

$$\delta\psi = \frac{1}{2\,\Delta u\,\Delta t}\int_{t_o}^{t_f} [K_\varphi\,\Delta\varphi(t)\Delta\psi(t) + K_\psi\,\Delta\psi^2(t)]dt \tag{A2}$$

Solving for K_ψ and letting $-\delta\psi = \psi_d - \psi$ (the previous terminal error), we obtain:

$$K_\psi = \underbrace{-K_\varphi \frac{\int_{t_o}^{t_f} \Delta\varphi(t)\Delta\psi(t)\,dt}{\int_{t_o}^{t_f} \Delta\psi^2(t)\,dt}}_{\text{Steepest descent optimization term}} + \underbrace{2\,\Delta u\,\Delta t\,\frac{\psi_d - \psi}{\int_{t_o}^{t_f} \Delta\psi^2(t)\,dt}}_{\text{Terminal error correction term}} \tag{A3}$$

APPENDIX B

REENTRY TRAJECTORY EQUATIONS

The following simplified equations derived for flight within the atmosphere were used for the example problem herein. The primary assumptions include a spherical nonrotating earth, small flight-path angles, and a constant gravity term. The derivation of these equations and their applicability have been considered in a number of reports.[11]

$$\ddot{h} = -g + \frac{V^2}{r} + \left(\frac{C_D A}{m}\right) \frac{1}{2} \rho V^2 \left(\frac{L}{D} - \frac{\dot{h}}{V}\right)$$

$$\dot{V} = -\left(\frac{C_D A}{m}\right) \frac{1}{2} \rho V^2$$

$$\psi = \int_{t_o}^{t_f} V \, dt$$

$$\varphi = 1.7 \times 10^{-8} \int_{t_o}^{t_f} \sqrt{\rho}\, V^3 \, dt$$

where

$\frac{C_D A}{m}$	drag loading, 2.0 ft^2/slug
g	local gravitational acceleration, 32.2 ft/sec^2
h	altitude, ft
$\frac{L}{D}$	control value of lift-drag ratio
r	radius from earth center, 21.1×10^6 ft
V	horizontal velocity, fps
ρ	atmosphere density, $0.00237 \, e^{-h/23,500}$ slug/ft^2
φ	total heat input, Btu/ft^2
ψ	final range, ft

APPENDIX C

DESCRIPTION OF COMPUTER SYSTEMS

In order to make meaningful a comparison of the results obtained from the analog and digital simulations, it is necessary to very briefly describe the computer systems used.

The analog computer was an EAI 231R-V equipped with electronic mode control of the amplifiers. The logic element of the hybrid simulation was an EAI DOS 350. The DOS 350 has a patchboard which permits one to combine logical elements, such as AND gates, flip-flops, shift registers, counters, etc., into complicated logic systems. It also has several delay line memories of various lengths as well as A-D and D-A converters for communicating between the DOS 350 and the analog computers.

The digital computer was an EAI 8400 mode 0 computer which had a 2 μsec memory access time, an average floating point add time of approximately 13 μsec, an average floating point multiply time of approximately 15 μsec, and a floating point word size of 32 bits.

REFERENCES

[1]Leitmann, G., ed., Optimization Techniques (Academic Press, 1962).

[2]Balakrishnan, A. V., and Neustadt, L. W., eds., Computing Methods in Optimization Problems (Academic Press, 1964).

[3]Wingrove, R. C., Raby, J. S., and Crane, D. F., A Method of Trajectory Optimization by Fast-Time Repetitive Computations. NASA TN D-3404, 1966.

[4]Bryson, A. E., and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems." Raytheon Rep. BR 1303, 1961. Also J. Appl. Mech., vol. 29, no. 2, June 1962, pp. 247-257.

[5]Kelley, H. J., "Gradient theory of optimal flight paths," ARS J., 30, 10, (Oct. 1960), pp. 947-954.

[6]Bryson, A. E., Denham, W. F., Carroll, F. J., and Mikami, K., "Determination of lift or drag programs that minimize reentry heating," J. Aerospace Sci., 29, 4, (April 1962), pp. 420-430.

[7]Blanton, H., ed., Three-Dimensional Trajectory Optimization Study. Pt. 1 - Optimum Programming Formulation. NASA CR-57030, 1964. (Supersedes Aero. Sys. Div. Rep. ASD-TDR-62-295 and Raytheon Rep. BR-1759-1)

[8]Hamming, R. W., Numerical Methods for Scientists and Engineers. (McGraw-Hill, 1962).

[9]Ralston, A., A First Course in Numerical Analysis (McGraw-Hill, 1965)

[10]Lundell, J. H., Wakefield, R. M., and Jones, W. J., Experimental Investigation of a Charring Ablative Material Exposed to Combined Convective and Radiative Heating in Oxidizing and Nonoxidizing Environments. Proc. AIAA Entry Technology Conf., Oct. 1964, pp. 216-227.

[11]Chapman, D. R., An Approximate Analytical Method for Studying Entry Into Planetary Atmospheres. NASA TR R-11, 1959.

TABLE I. - ALTITUDE TIME HISTORIES

Time, sec	Altitude, 10^3 ft, Iteration 1	2	10	20
0	250	250	250	250
60	207	206	197	197
120	180	184	160	160
180	182	184	212	209
240	158	165	196	195
300	125	132	142	142
360			127	127
400			104	110

TABLE II. - CONTROL TIME HISTORIES

Time, sec	Control L/D, Iteration 1	2	10	20
0	0.250	0.212	0	0
60	.250	.192	0	0
120	.250	.291	.500	.500
180	.250	.290	.500	.500
240	.250	.294	.447	.500
300	.250	.294	.500	.500
360		.250	.347	.475
400			.254	.277

TABLE III. - TERMINAL CONDITIONS

	Iteration 1	2	5	10	20
Time, sec	344	358	389	407	414
Altitude, 10^3 ft	99.3	99.9	99.8	99.5	98.8
Range, miles	997.7	1001.5	1003.7	1002.8	999.9
Heat, Btu/ft^2	23491	23197	21517	21025	20966

FIGURE CAPTIONS

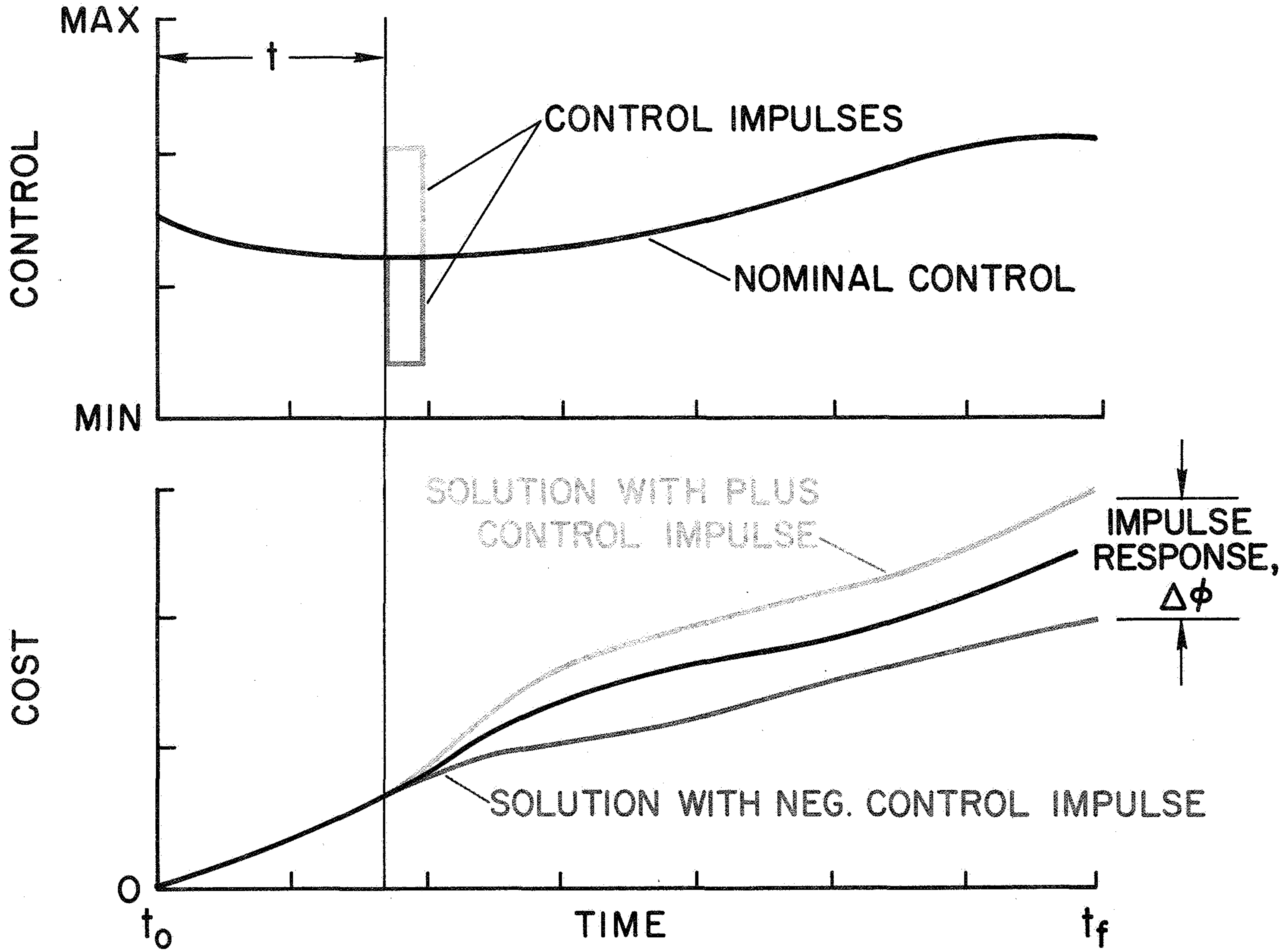

Figure 1.- Computation of the impulse response.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

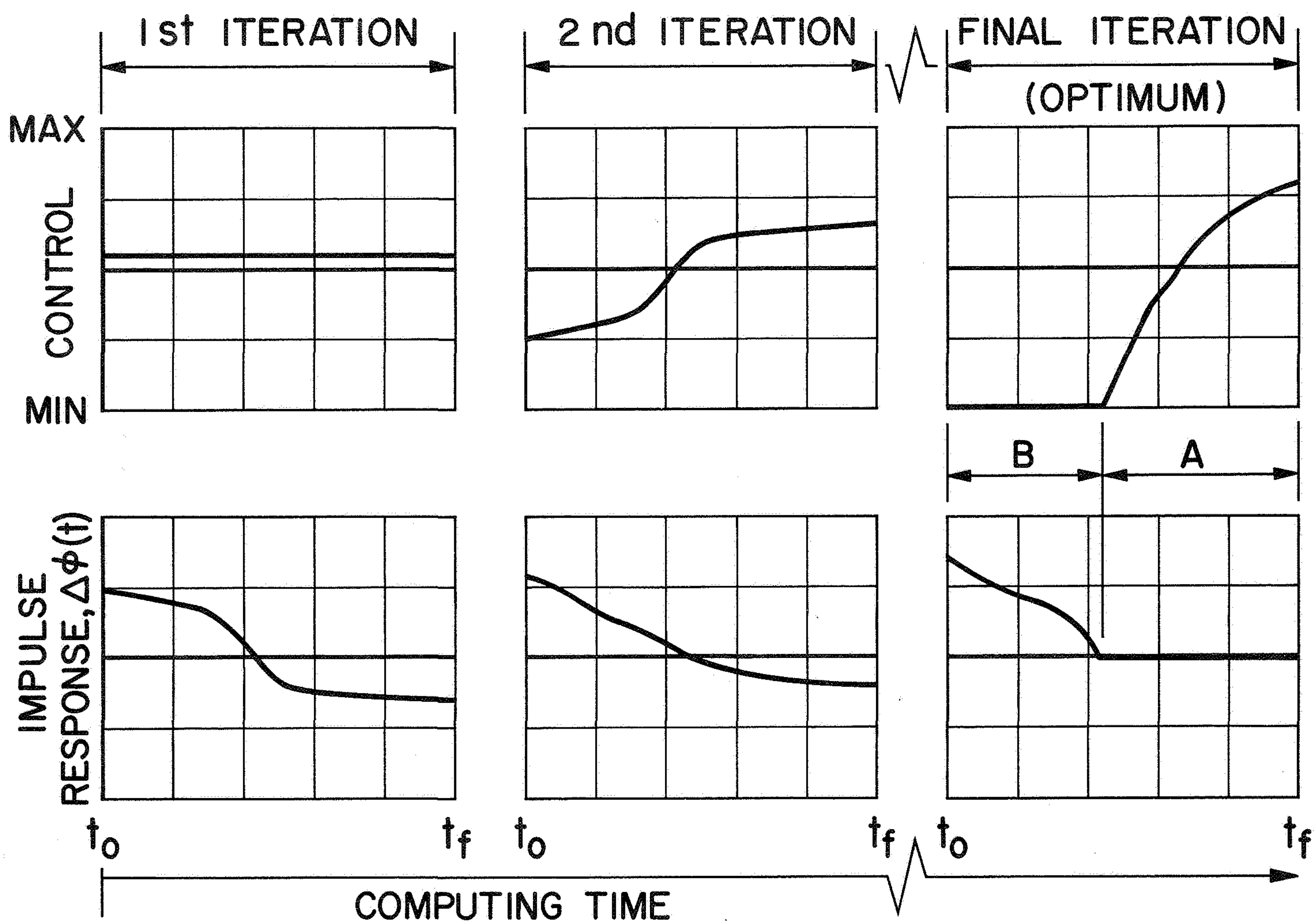

Figure 2.- Computation of the optimal control.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

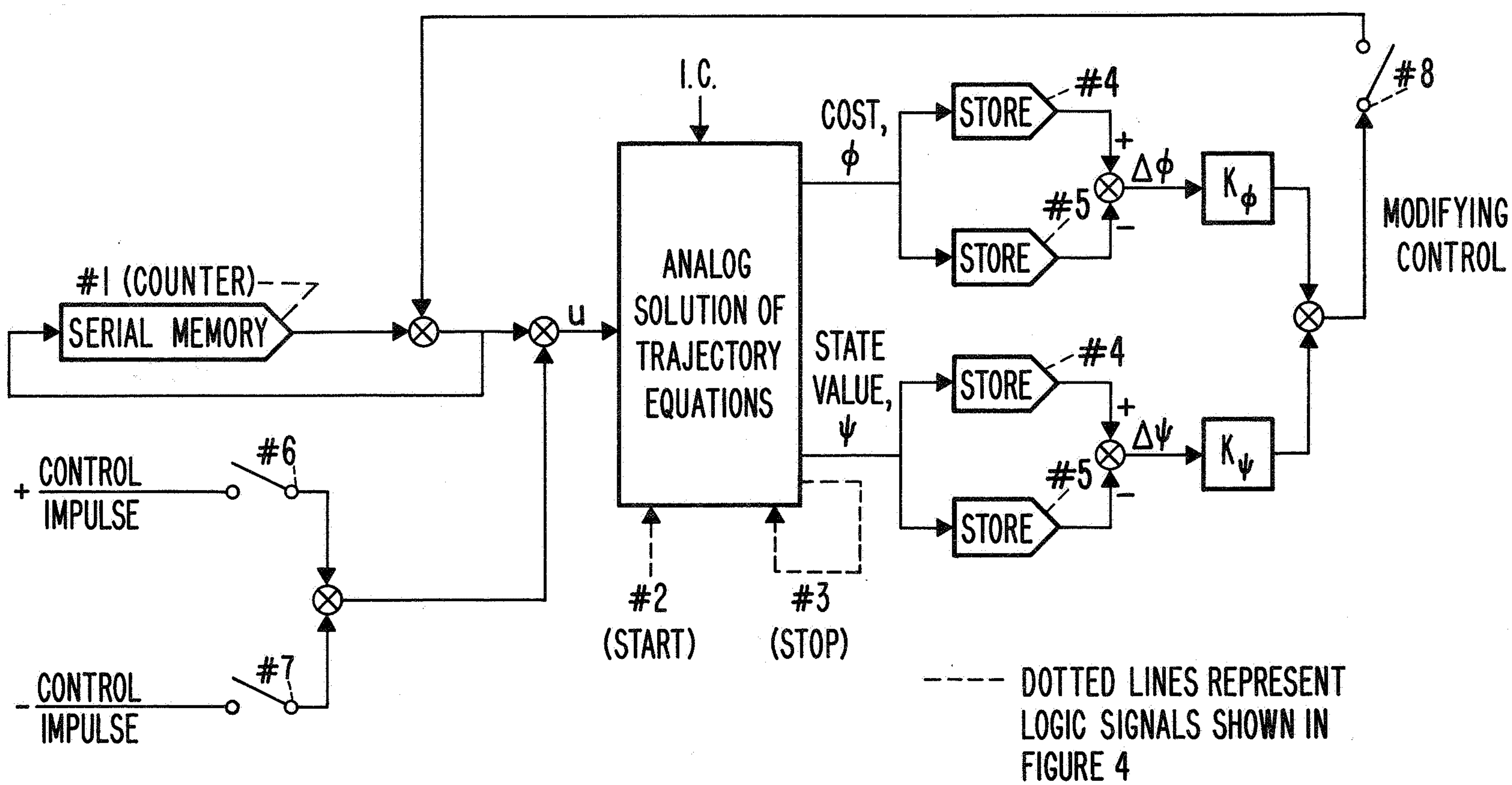

Figure 3.- Hybrid computer flow diagram.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

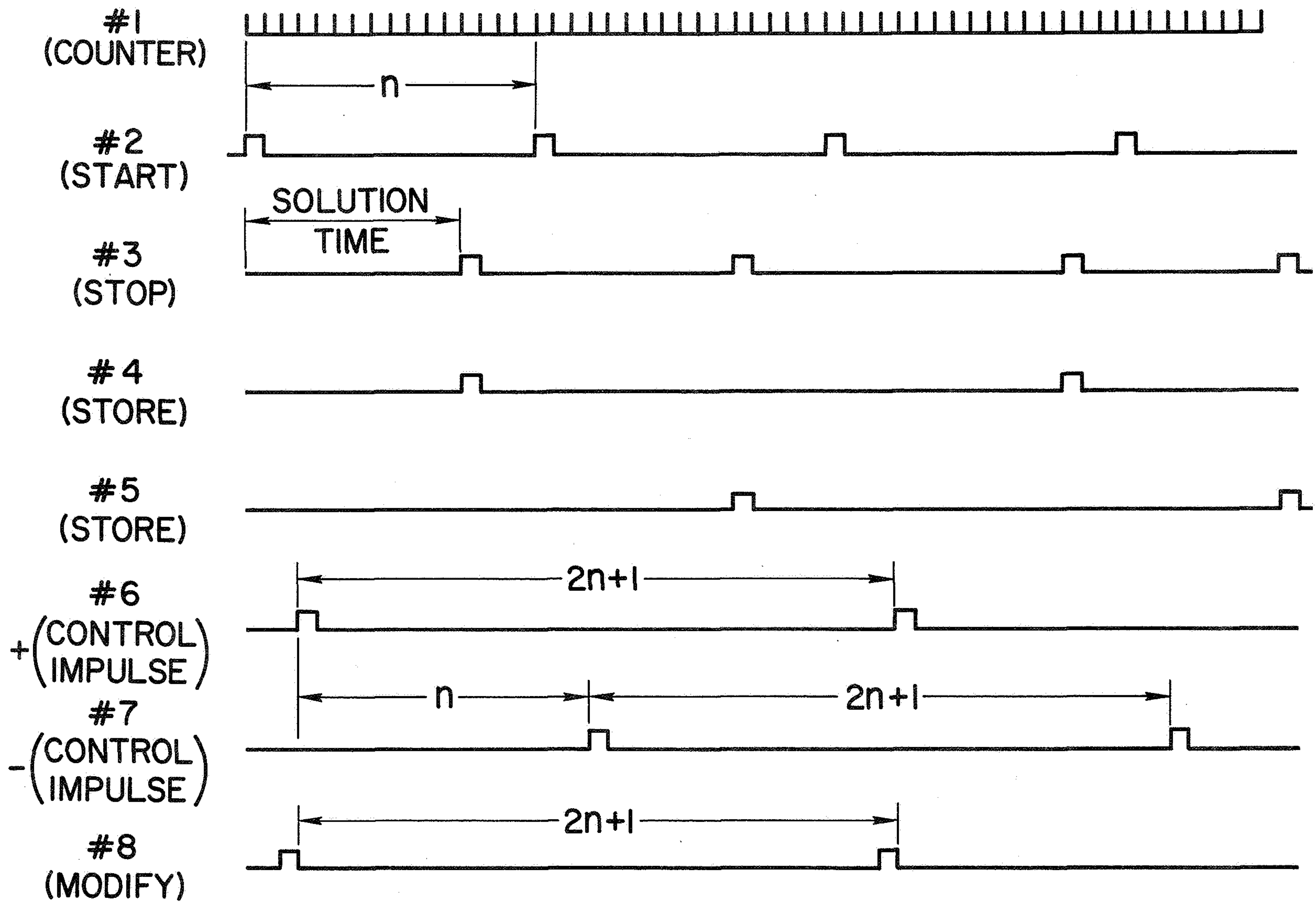

Figure 4.- Hybrid computer program logic.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

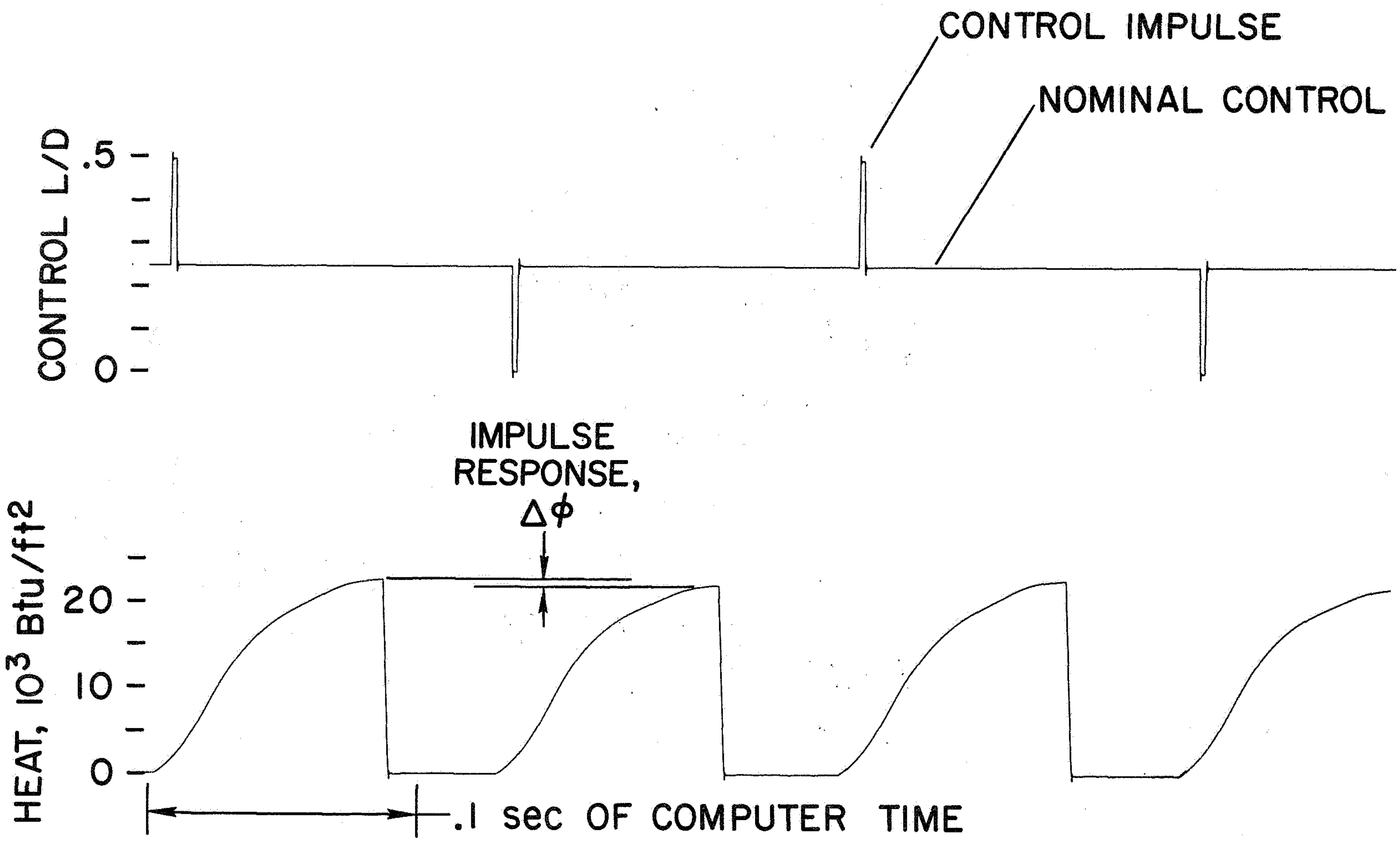

Figure 5.- Hybrid repetitive computations.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

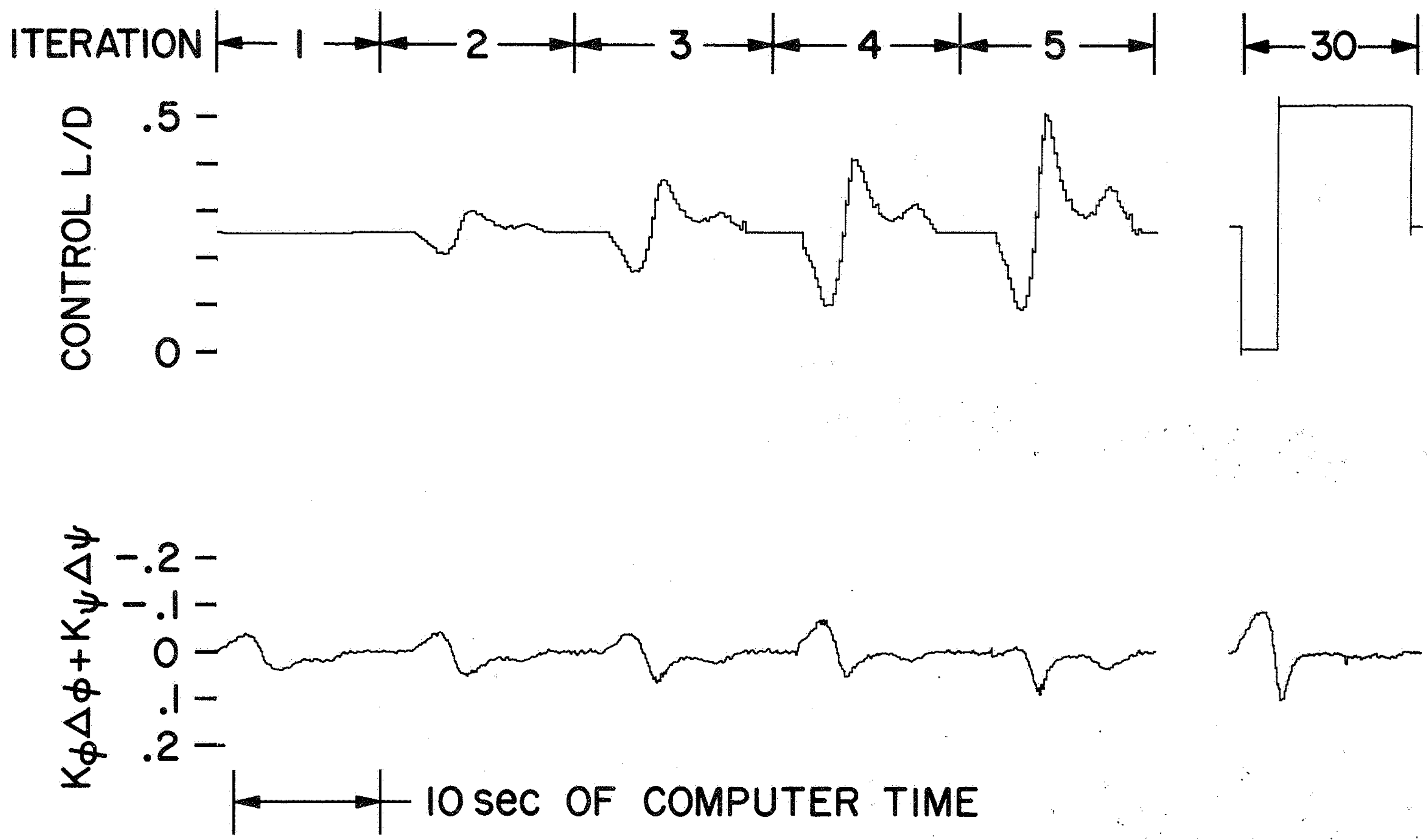

Figure 6.- Hybrid computation of the optimal control.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

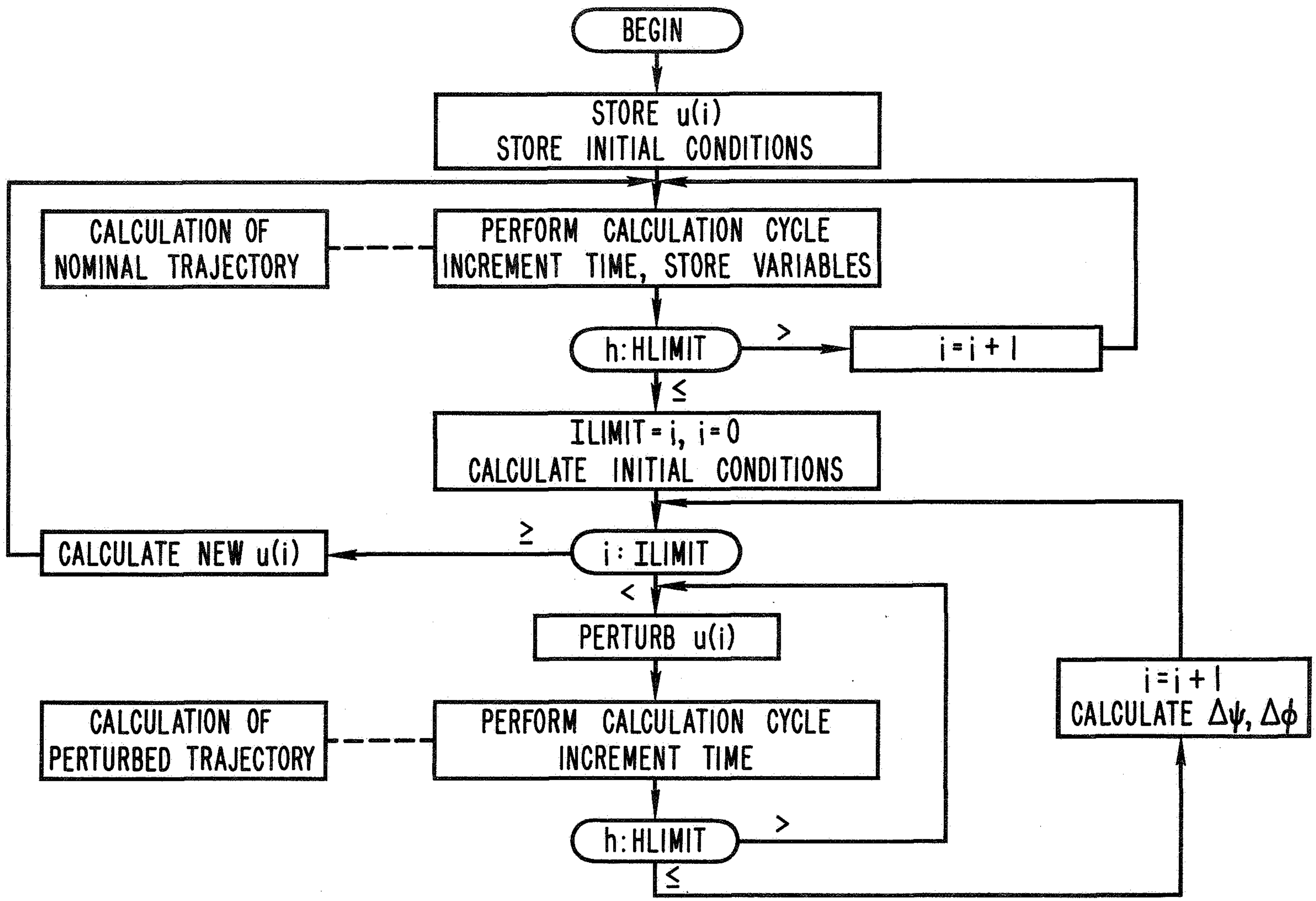

Figure 7.- Digital computer flow chart.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

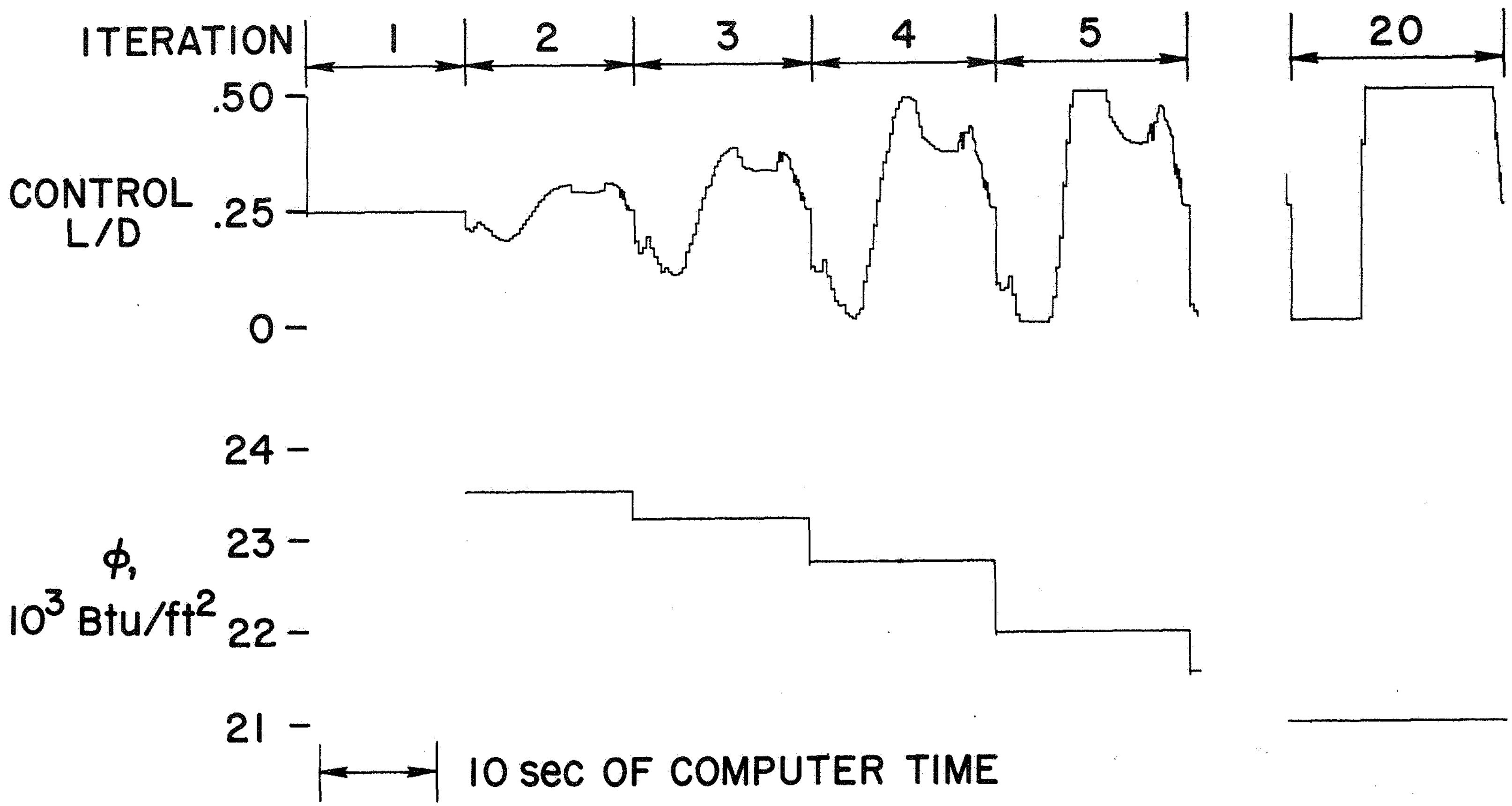

Figure 8.- Digital computation of the optimal control.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA

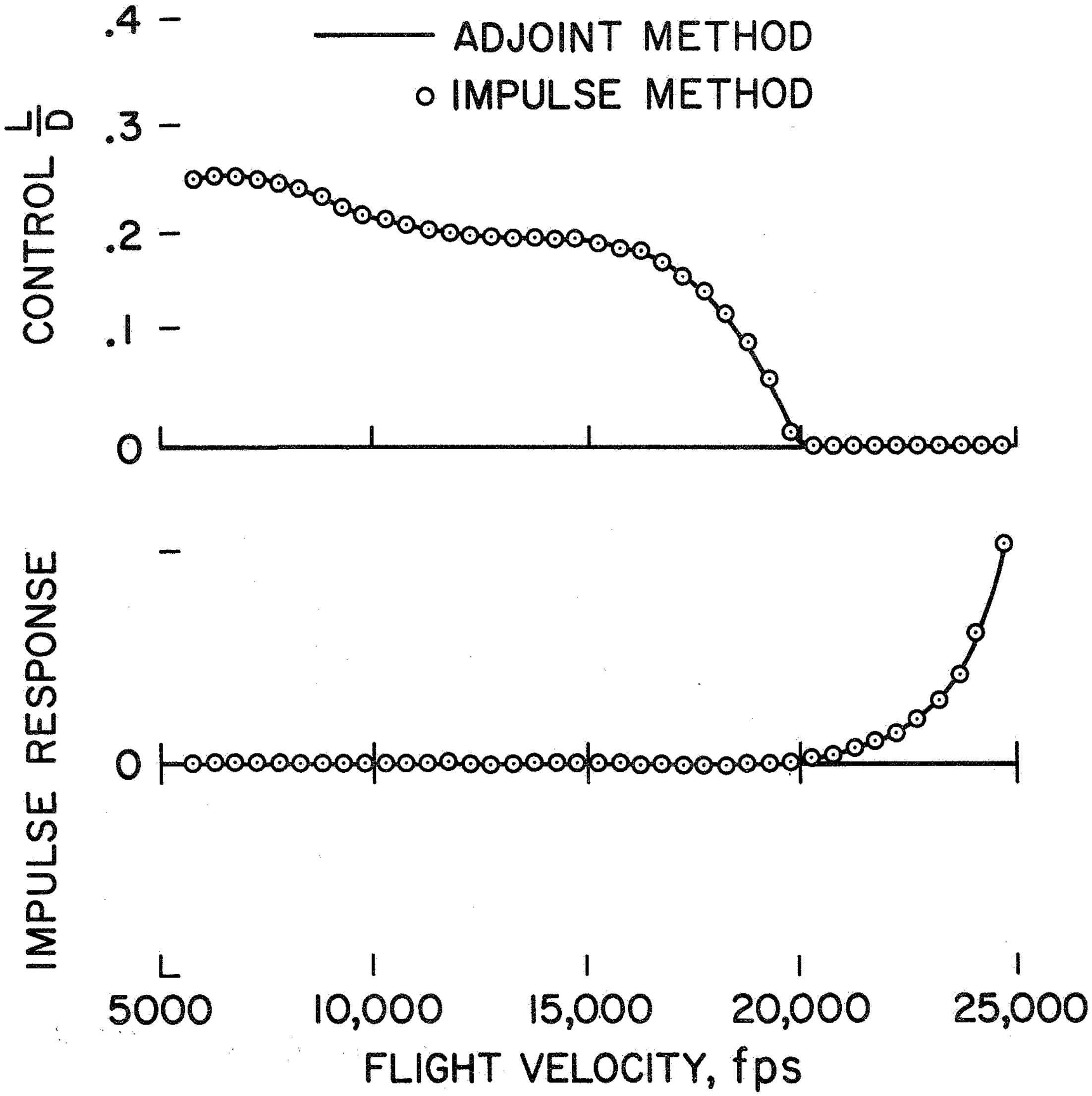

Figure 9.- Comparison of the impulse response and adjoint steepest descent methods.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER, MOFFETT FIELD, CALIFORNIA