N 67 25997

NASA CR 83958

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1055

A Study of Low-Thrust Guidance

G. R. Ash

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

April 15, 1967

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1055

A Study of Low-Thrust Guidance

G. R. Ash

Approved by:

G. E. Sweetnam, Manager
Spacecraft Control Section

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

April 15, 1967

TECHNICAL REPORT 32-1055

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

Contents

Figures

Figures (contd)

Abstract

The low-thrust guidance problem has been formulated. Approximate feedback solutions have been obtained using both minimum-time and least-squares criteria. Computer programs that simulate the resulting control systems are presented. Good performance was obtained with the minimum-time solution, and recommendations are made for future work on this problem.

The nonlinear, sequential estimation problem was considered, using the estimation equations obtained by Dr. R. Sridhar. A refinement of these equations was attempted, but the results have not been encouraging so far. The computer programs used are presented, and recommendations are also made for continued work in this area.

A Study of Low-Thrust Guidance

I. Introduction

In the past few years, much interest has been developed in the use of ion propulsion for space missions. The low-thrust ion engine will probably find its most important application in missions to the outer planets, where the retarding effect of the Sun's gravity will require a large space vehicle energy. Up to the present, all the energy (velocity) of a spacecraft has been provided by the launch vehicle. For high-energy missions, such as those to the outer planets, it seems desirable to use high-impulse, low-thrust engines to augment the energy supplied by the boost vehicle. These low-thrust devices would operate during the long flight time between launch and encounter, supplying a higher specific impulse than that available from the present chemical boosters.

If such a thrust vector were provided, it would be desirable to use the thrust to provide guidance to the spacecraft. The problem of guidance is to force the spacecraft to be at a certain place in space at a certain time and perhaps with a certain velocity. This is theoretically possible if a set of exact initial conditions and an exact thrust program are obtained in flight. In practice, such a scheme is clearly impossible, however, owing to initial energy dispersion (that is, the initial velocity vector not being obtained exactly) and also to random disturbances in flight. The guidance problem also involves choosing a method of guiding a vehicle that is "best" in some sense.

Obtaining guidance as described above is new, in that the guidance is *continuous*. At present, of course, guidance is obtained by one or several midcourse maneuvers. If one were to use low thrust for high-energy missions, there would appear to be little penalty in obtaining continuous control (guidance) and its many advantages—the main advantage being the ability to make trajectory corrections at any time during flight.

This report represents a study of the problem discussed above, including computer simulations. Recommendations are also made for future work.

II. Description of the Problem

To gain insight into any problem, one usually starts by making simplifying assumptions and then includes all practical considerations. This method of analysis will generally be followed in this report.

The first simplifying assumption is that the vehicle has been launched and is in heliocentric flight (i.e., Earth's gravity is neglected), and the second is that motion is constrained to one plane. The first assumption is based on the fact that the ion engine would not be turned on for about three days after launch, and therefore the spacecraft would be essentially out of the Earth's gravitational field. The second assumption is based on the statement in Ref. 1 that "performance loss incurred by the out-of-plane dynamics . . . will not exceed 5 percent in payload." A third simplifying assumption will be that the nominal thrust acceleration level over periods of time necessary for control is a constant (i.e., assuming constant thrust and neglecting changes in total vehicle mass).

The practical assumptions that are made concern the low-thrust vector. It will be assumed that the ion engine is fixed to the spacecraft. Since solar power will be necessary, and this implies pointing the vehicle at the Sun, the low-thrust vector will thus make a nominally constant angle with the Sun–vehicle line. A value of 90 deg is considered typical for this angle (Ref. 1) and will be used in this study (see Fig. 1). If control in two dimensions is to be obtained, one intuitively feels that it would be necessary to have independent control in two directions. One practical way of obtaining such control would be, first, to allow small attitude variations and, second, to allow the acceleration level to change slightly. The control scheme used in this study allows only 9 discrete states of the thrust vector, counting the nominal state (see Fig. 2). This scheme of control has the advantage of being both simple and highly realistic.

Because of initial energy dispersion and random effects during flight, the state (i.e., the position and velocity) of the vehicle will not be known exactly. Hence there is a need for state estimation, or orbit determination, if one is to obtain control of the vehicle. If the random disturbances on the vehicle have a Gaussian distribution and if certain conditions of system linearity are satisfied (i.e., if the "deterministic controller" is linear), the "separation theorem" states that the estimation problem and the control problem can be separated in an optimal sense. Unfortunately, our system will not turn out to be linear, but we will still separate the estimation and control problems (a suboptimal solution). Hence we will assume that the estimated state is available at all times for purposes of control. Work on the estimation problem appears in Section XI.

To solve a control problem, one mu[illegible] will judge each system—i.e., one must specify [illegible] or a performance index. Before that, however, one [illegible] specify exactly what it is that he wants to control. Most often, in the control of space vehicles, one wishes to obtain certain terminal conditions at planet encounter. To this end, one specifies a nominal (standard) trajectory that will be followed if the correct initial conditions and thrust programs are used, without outside disturbances (see Ref. 3). Actually, if at any time during flight the vehicle were put onto the nominal trajectory at the point in space with the velocity it would normally have at that

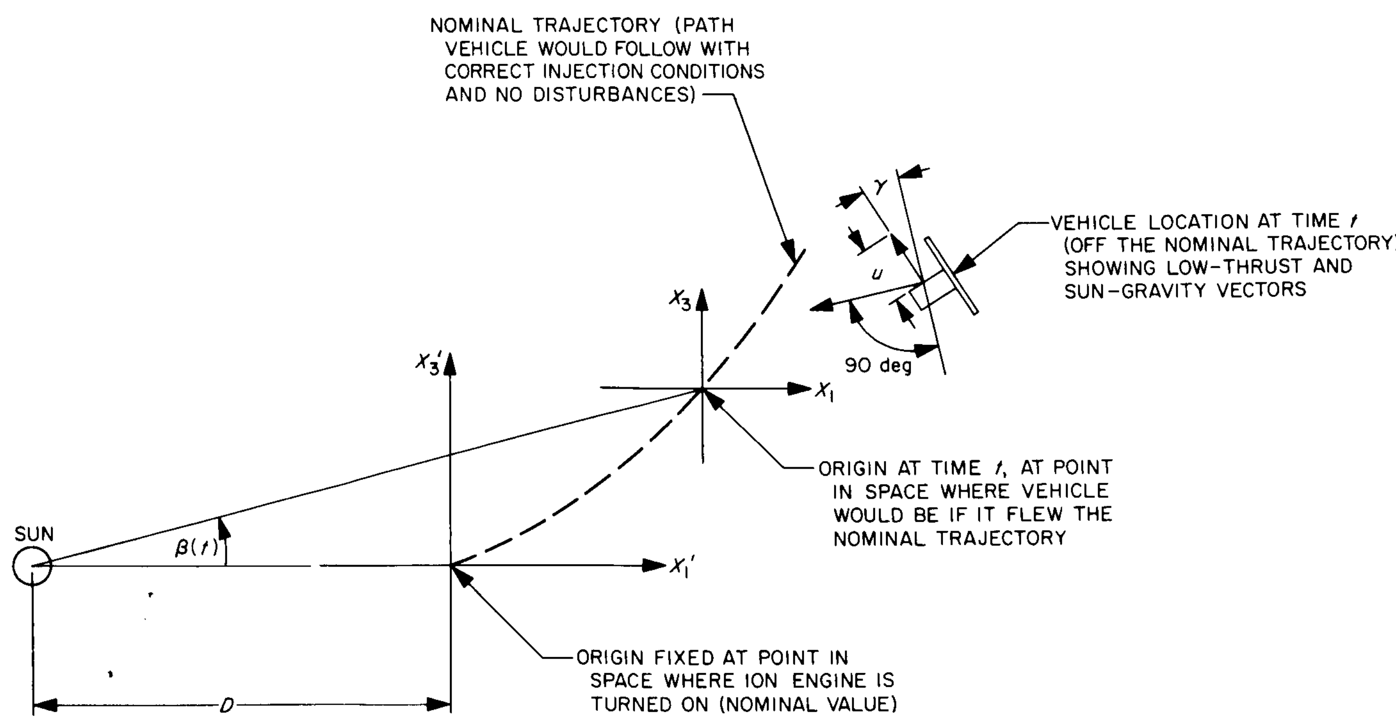


Fig. 1. Definition of the coordinate frames (x_1', x_3') and (x_1, x_3)

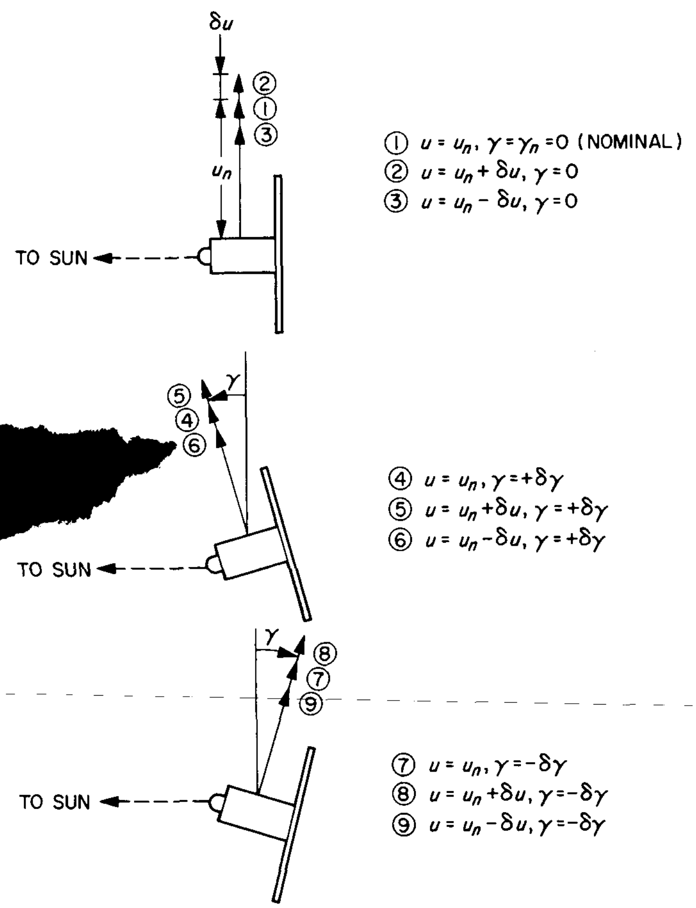


Fig. 2. The nine allowable states of the ion-engine thrust vector

particular time, the vehicle would, of course, fly the nominal trajectory and hence satisfy the right terminal conditions. Therefore, one method of controlling a spacecraft would be to force it to fly on the nominal, or design, trajectory. R. J. Parks points out (Ref. 4) that the "standard trajectory will be the result of many compromises between conflicting requirements such as propulsion efficiency (including drag losses), aerodynamic heating, guidance accuracy (including effects of ground station location limitations), tracking and telemetering considerations. Once this standard trajectory has been selected, it is the function of the guidance system to (1) cause the vehicle to approach the destination in the intended fashion . . . , and (2) to cause the vehicle to fly as closely as is practical to the standard trajectory at all times, so as to ensure the compromises chosen." In this way, also, a control system would be obtained that would be good for many missions; i.e., for many nominal trajectories.

The criterion we will use will be that of minimum time; that is, we will try to get the vehicle back onto the nominal trajectory in a minimum of time. This seems a good criterion for this problem, in that velocity errors will have less time to propagate. Also, the solution to the optimum minimum-time problem involves "bang-bang" control, or using discrete levels of control. Since we have constrained our thrust vector control to be discrete, an optimum minimum-time solution can be obtained for this problem. (For such small deviations of the thrust vector magnitude, minimizing fuel would tend to be less important than minimizing the time off the trajectory. However, for the purpose of choosing a nominal trajectory thrust program, a minimum-fuel problem would probably be considered.)

So far, we have described the control we have available, the state we want to obtain, and the performance index we wish to minimize. What remains is to translate this into mathematical language and attempt to obtain an exact solution to the problem.

III. Mathematical Statement of the Problem

The coordinate systems we will be considering appear in Fig. 1. The coordinate frame (x_1, x_3) is a frame whose origin at time t is at the point in space a vehicle on the nominal trajectory would be at time t, assuming flight begins at time $= 0$. The angle β is the angle the line connecting the origin of (x_1, x_3) and the Sun makes with the x'_1 axis of the fixed inertial reference frame (x'_1, x'_3). Hence β is a function of time only and is determined by the nominal trajectory desired. The equations of motion in the (x'_1, x'_3) frame are as follows (note that dots above variables represent derivatives with respect to time):[1]

$$\left.\begin{aligned}
\dot{x}'_1 &= x'_2 \triangleq F_1 \\
\dot{x}'_2 = \ddot{x}'_1 &= \frac{-GM_s(x'_1 + D)}{((x'_1 + D)^2 + (x'_3)^2)^{3/2}} \\
&\qquad - \frac{u(x'_3 \cos\gamma + (x'_1 + D)\sin\gamma)}{((x'_1 + D)^2 + (x'_3)^2)^{1/2}} \triangleq F_2 \\
\dot{x}'_3 &= x'_4 \triangleq F_3 \\
\dot{x}'_4 = \ddot{x}'_3 &= \frac{-GM_s(x'_3)}{((x'_1 + D)^2 + (x'_3)^2)^{3/2}} \\
&\qquad + \frac{u((x'_1 + D)\cos\gamma - x'_3 \sin\gamma)}{((x'_1 + D)^2 + (x'_3)^2)^{1/2}} \triangleq F_4
\end{aligned}\right\} \tag{1}$$

[1]Throughout this report, vectors are shown in lightface roman letters (e.g., x), matrices in boldface roman (**x**), and scalars in italics (*x*); the Hamiltonian is represented by $\mathcal{H}$.

Here G is the constant of gravitation, M_s is the mass of the Sun, u and γ have the same meaning as in Fig. 2, and D is defined in Fig. 1.

Using vector notation,

$$\begin{bmatrix} \dot{x}_1' \\ \dot{x}_2' \\ \dot{x}_3' \\ \dot{x}_4' \end{bmatrix} \triangleq \dot{X}', \quad \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \triangleq F, \quad \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \end{bmatrix} \triangleq X' \tag{2}$$

Then

$$\dot{X}' = F(u, \gamma, X') \tag{3}$$

where the independent variables of F have been indicated. Then the problem is as follows: given Eq. (3) and deviations from the nominal trajectory (remembering that (x_1, x_3) is fixed to the nominal trajectory) at time τ, that is,

$$\begin{bmatrix} x_1(\tau) \\ x_2(\tau) \\ x_3(\tau) \\ x_4(\tau) \end{bmatrix} \triangleq X(\tau) \tag{4}$$

(where $x_2(\tau)$ and $x_4(\tau)$ are defined as velocities in the x_1 and x_3 directions, respectively), find the controls

$$u(t), \gamma(t) \qquad \tau \leq t \leq T \tag{5}$$

such that at some time $T > \tau$

$$X(T) = 0$$

and the performance index

$$\int_\tau^T dt \tag{7}$$

is minimized (that is, T is minimized).

IV. First Solution of the Minimum-Time Problem

Referring to Fig. 3, consider the following coordinate transformation:

$$\left. \begin{aligned} x_1'' &= x_1' \cos\beta + x_3' \sin\beta \\ x_3'' &= -x_1' \sin\beta + x_3' \cos\beta \end{aligned} \right\} \tag{8}$$

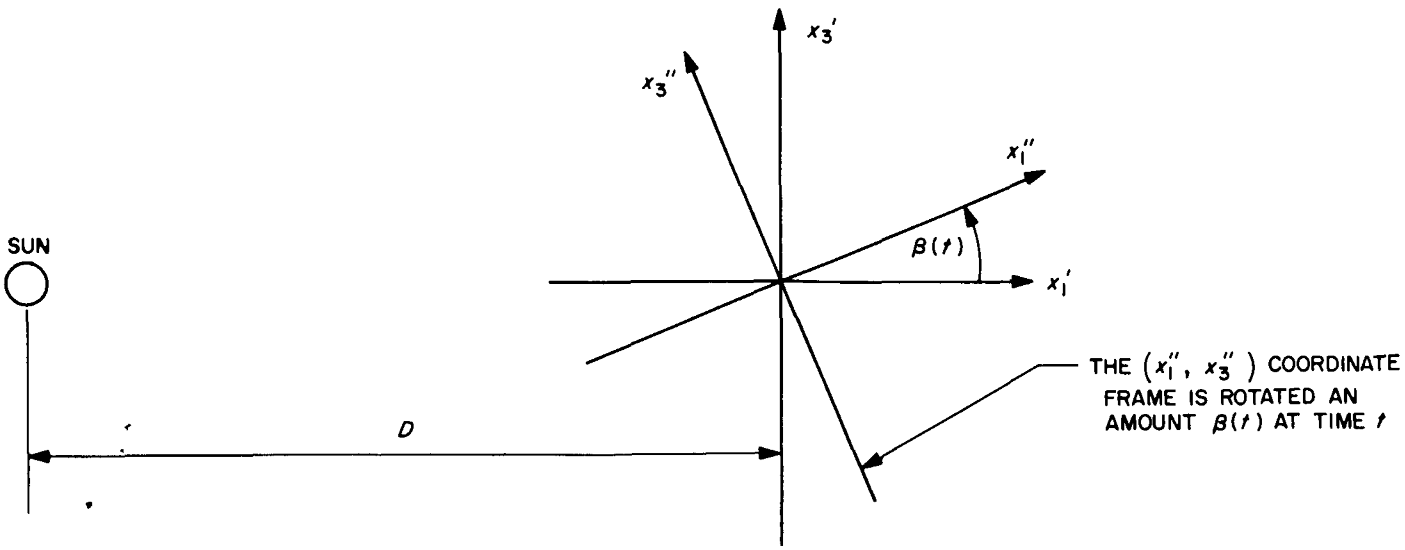


Fig. 3. Definition of the (x_1'', x_3'') coordinate frame

Then

$$
\left.\begin{aligned}
\dot{x}_1'' &= \dot{x}_1' \cos\beta + \dot{x}_3' \sin\beta - x_1'\dot{\beta}\sin\beta + x_3'\dot{\beta}\cos\beta \triangleq x_2'' \\
\ddot{x}_1'' &= \dot{x}_2'' = \ddot{x}_1' \cos\beta + \ddot{x}_3' \sin\beta - 2\dot{x}_1'\dot{\beta}\sin\beta - x_1'\dot{\beta}^2\cos\beta - x_1'\ddot{\beta}\sin\beta + 2\dot{x}_3'\dot{\beta}\cos\beta - x_3'\dot{\beta}^2\sin\beta + x_3'\ddot{\beta}\cos\beta \\
\dot{x}_3'' &= -\dot{x}_1'\sin\beta + \dot{x}_3'\cos\beta - x_1'\dot{\beta}\cos\beta - x_3'\dot{\beta}\sin\beta \triangleq x_4'' \\
\ddot{x}_3'' &= \dot{x}_4'' = -\ddot{x}_1'\sin\beta + \ddot{x}_3'\cos\beta - 2\dot{x}_1'\dot{\beta}\cos\beta + \dot{x}_1'\dot{\beta}^2\sin\beta - x_1'\ddot{\beta}\cos\beta - 2\dot{x}_3'\dot{\beta}\sin\beta - x_3'\dot{\beta}\cos\beta - x_3'\ddot{\beta}\sin\beta
\end{aligned}\right\} \quad (9)
$$

Let

$$
\mathbf{R}(t) \triangleq \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & \cos\beta & 0 & \sin\beta \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & -\sin\beta & 0 & \cos\beta \end{bmatrix}, \dot{X}'' \triangleq \begin{bmatrix} \dot{x}_1'' \\ \dot{x}_2'' \\ \dot{x}_3'' \\ \dot{x}_4'' \end{bmatrix}, X'' \triangleq \begin{bmatrix} x_1'' \\ x_2'' \\ x_3'' \\ x_4'' \end{bmatrix}
$$

$$
\mathbf{S}(t) \triangleq \begin{bmatrix} -\dot{\beta}\sin\beta & 0 & \dot{\beta}\cos\beta & 0 \\ -\ddot{\beta}\sin\beta - \dot{\beta}^2\cos\beta & -2\dot{\beta}\sin\beta & \ddot{\beta}\cos\beta - \dot{\beta}^2\sin\beta & 2\dot{\beta}\cos\beta \\ -\dot{\beta}\cos\beta & 0 & -\dot{\beta}\sin\beta & 0 \\ -\ddot{\beta}\cos\beta + \dot{\beta}^2\sin\beta & -2\dot{\beta}\cos\beta & -\ddot{\beta}\sin\beta - \dot{\beta}^2\cos\beta & -2\dot{\beta}\sin\beta \end{bmatrix}
$$

Then a shorthand notation for Eq. (9), using Eq. (3), is

$$
\dot{X}'' = \mathbf{R}(t)\,F(u, \gamma, X') + \mathbf{S}(t)\,X' \quad (10)
$$

If at time t the vehicle is off the nominal trajectory by an amount $X(t)$, there will be a difference between the nominal and actual states in all reference frames. Letting the subscript n denote the nominal values of variables at time t, the last statement can be written:

$$
\dot{X}_n'' + \delta\dot{X}'' = \mathbf{R}(t)\,F(u_n + \delta u, \gamma_n + \delta\gamma, X_n' + X) + \mathbf{S}(t)(X_n' + X) \quad (11)
$$

where $\delta\dot{X}''$, δu, and $\delta\gamma$ are deviations from their nominal values at time t. It should be pointed out that Eq. (11) is an exact equation. Now the quantities δu, $\delta\gamma$, and X are small in the sense that a first-order expansion of F about the nominal values will be a uniformly "good" approximation for all values of time. This statement is certainly true for the control deviations δu and $\delta\gamma$ (this has been mentioned before), and the spacecraft state deviations from nominal are not expected to go outside the region where linearity holds for any reasonable errors in initial conditions or disturbances en route. Hence, through this expansion, Eq. (11) becomes

$$
\begin{aligned}
\dot{X}_n'' + \delta\dot{X}'' = \mathbf{R}(t)\,(&F(u_n, \gamma_n, X_n') + F_{X'}(u_n, \gamma_n, X_n')\,X + F_u(u_n, \gamma_n, X_n')\,\delta u + F_\gamma(u_n, \gamma_n, X_n')\,\delta\gamma + (\text{higher-order terms})) \\
&+ \mathbf{S}(t)(X_n' + X) \quad (12)
\end{aligned}
$$

where the following definitions apply:

$$\mathbf{F}_{\mathbf{X}'} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1'} & \frac{\partial F_1}{\partial x_2'} & \frac{\partial F_1}{\partial x_3'} & \frac{\partial F_1}{\partial x_4'} \\ \frac{\partial F_2}{\partial x_1'} & \frac{\partial F_2}{\partial x_2'} & \frac{\partial F_2}{\partial x_3'} & \frac{\partial F_2}{\partial x_4'} \\ \frac{\partial F_3}{\partial x_1'} & \frac{\partial F_3}{\partial x_2'} & \frac{\partial F_3}{\partial x_3'} & \frac{\partial F_3}{\partial x_4'} \\ \frac{\partial F_4}{\partial x_1'} & \frac{\partial F_4}{\partial x_2'} & \frac{\partial F_4}{\partial x_3'} & \frac{\partial F_4}{\partial x_4'} \end{bmatrix}_{u_n, \gamma_n, \mathbf{X}_n} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ A & 0 & B & 0 \\ 0 & 0 & 0 & 1 \\ C & 0 & D & 0 \end{bmatrix}$$

where

$$A = \frac{-GM_s + u_n (x_{1n}' + D)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{3/2}} + \frac{3GM_s (x_{1n}' + D)^2}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{5/2}}$$

$$B = \frac{u_n ((x_{3n}')^2 - (x_{1n}' + D)^2 - (x_{3n}')^2)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{3/2}} + \frac{3GM_s (x_{1n}' + D)\, x_{3n}'}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{3/2}}$$

$$C = \frac{3GM_s\, x_{3n}' (x_{1n}' + D)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{5/2}} + \frac{u_n (2 (x_{1n}' + D)^2 + (x_{3n}')^2)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{3/2}}$$

$$D = \frac{GM_s (3(x_{3n}')^2 - (x_{1n}' + D)^2 - (x_{3n}')^2)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{5/2}} - \frac{u_n\, x_{3n}' (x_{1n}' + D)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{3/2}}$$

$$\mathbf{F}_u \triangleq \begin{bmatrix} \frac{\partial F_1}{\partial u} \\ \frac{\partial F_2}{\partial u} \\ \frac{\partial F_3}{\partial u} \\ \frac{\partial F_4}{\partial u} \end{bmatrix}_{u_n, \gamma_n, \mathbf{X}_n(t)} = \begin{bmatrix} 0 \\ \frac{-x_{3n}'}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{1/2}} \\ 0 \\ \frac{(x_{1n}' + D)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{1/2}} \end{bmatrix} = \begin{bmatrix} 0 \\ -\sin\beta(t) \\ 0 \\ \cos\beta(t) \end{bmatrix}$$

$$\mathbf{F}_\gamma \triangleq \begin{bmatrix} \frac{\partial F_1}{\partial \gamma} \\ \frac{\partial F_2}{\partial \gamma} \\ \frac{\partial F_3}{\partial \gamma} \\ \frac{\partial F_4}{\partial \gamma} \end{bmatrix}_{u_n, \gamma_n, \mathbf{X}_n(t)} = \begin{bmatrix} 0 \\ \frac{-u_n (x_{1n}' + D)}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{1/2}} \\ 0 \\ \frac{-u_n\, x_{3n}'}{((x_{1n}' + D)^2 + (x_{3n}')^2)^{1/2}} \end{bmatrix} = \begin{bmatrix} 0 \\ -u_n \cos\beta(t) \\ 0 \\ -u_n \sin\beta(t) \end{bmatrix}$$

Since γ can have only three values $(0, +\delta\gamma, -\delta\gamma)$, and similarly for $u\,(u_n, u_n + \delta u, u_n - \delta u)$, let us define

$$u_n\,\delta\gamma \overset{\Delta}{=} -u_1 \qquad \delta u \overset{\Delta}{=} u_2 \tag{13}$$

Hence, positive and negative u_1 correspond to rotating the low-thrust vector away from and toward the Sun respectively. Also, positive and negative u_2 correspond to increasing and decreasing thrust vector length respectively. If we neglect higher-order terms and use Eq. (10), Eq. (12) becomes

$$\delta\dot{\mathbf{X}}''(t) \doteq \mathbf{R}(t)\,\mathbf{F}_{\mathbf{X}'}(u_n, \gamma_n, \mathbf{X}'_n)\,\mathbf{X}(t) + \mathbf{R}(t)\,\mathbf{F}_u(u_n, \gamma_n, \mathbf{X}'_n)\,\delta u(t) + \mathbf{R}(t)\,\mathbf{F}_\gamma(u_n, \gamma_n, \mathbf{X}'_n)\,\delta\gamma(t) + \mathbf{S}(t)\,\mathbf{X}(t) \tag{14}$$

Now, using the definitions for $\mathbf{F}_{\mathbf{X}'}$, $\mathbf{R}(t)$, $\mathbf{F}_u$, $\mathbf{F}_\gamma$, and $\mathbf{S}(t)$, and Eq. (13), we have

$$\delta\dot{\mathbf{X}}''(t) = \begin{bmatrix} 0 & \cos\beta(t) & 0 & \sin\beta(t) \\ A\cos\beta(t) + C\sin\beta(t) & 0 & B\cos\beta(t) + D\sin\beta(t) & 0 \\ 0 & -\sin\beta & 0 & \cos\beta \\ -A\sin\beta(t) + C\cos\beta(t) & 0 & -B\sin\beta(t) + D\cos\beta(t) & 0 \end{bmatrix} \mathbf{X}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2(t)$$

$$+ \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_1(t) + \begin{bmatrix} -\dot\beta\sin\beta & 0 & \dot\beta\cos\beta & 0 \\ -\ddot\beta\sin\beta - \dot\beta^2\cos\beta & -2\dot\beta\sin\beta & \ddot\beta\cos\beta - \dot\beta^2\sin\beta & 2\dot\beta\cos\beta \\ -\dot\beta\cos\beta & 0 & -\dot\beta\sin\beta & 0 \\ -\ddot\beta\cos\beta + \dot\beta^2\sin\beta & -2\dot\beta\cos\beta & -\ddot\beta\sin\beta - \dot\beta^2\cos\beta & -2\dot\beta\sin\beta \end{bmatrix} \mathbf{X}(t) \tag{15}$$

To gain more insight into the problem, Eq. (15) will be simplified by neglecting small terms. The quantities A, B, C, and D are proportional to changes in the Sun's gravity and the angle β over a region in space (the region includes the deviations of the spacecraft from the nominal trajectory). These quantities are of the order of 10^{-12}, in mks units, and hence can be neglected. The same is true for the quantities $\ddot\beta$ and $\dot\beta^2$, which are of the order of 10^{-14} or less for the mission under consideration (i.e., a Mars mission—these quantities would be even smaller for missions to the outer planets). Finally, it will be assumed that the quantities $\dot\beta x_2$ and $\dot\beta x_4$ are negligible with respect to u_1 and u_2. Actually, typical values would be 10^{-6} for $\dot\beta x_2$ and $\dot\beta x_4$, and 10^{-4} for u_1 and u_2. Hence, although this is a good approximation, it is the one that would give by far the largest error. Note that

$$\mathbf{X}'' = \mathbf{X}''_n + \delta\mathbf{X}'' = \mathbf{Q}(t)\,(\mathbf{X}'_n + \mathbf{X})$$

where

$$\mathbf{Q}(t) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ -\dot\beta\sin\beta & \cos\beta & \dot\beta\cos\beta & \sin\beta \\ -\sin\beta & 0 & \cos\beta & 0 \\ -\dot\beta\cos\beta & -\sin\beta & -\dot\beta\sin\beta & \cos\beta \end{bmatrix}$$

Since

$$X''_n = \mathbf{Q}(t)\, X'_n$$

we find

$$\delta X''(t) = \mathbf{Q}(t)\, X(t)$$

Clearly

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \delta X''(t) = \begin{bmatrix} -\dot{\beta}\sin\beta & \cos\beta & \dot{\beta}\cos\beta & \sin\beta \\ 0 & 0 & 0 & 0 \\ -\dot{\beta}\cos\beta & -\sin\beta & -\dot{\beta}\sin\beta & \cos\beta \\ 0 & 0 & 0 & 0 \end{bmatrix} X(t)$$

If we use this and neglect the smaller terms mentioned, Eq. (15) becomes

$$\delta \dot{X}''(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \delta X''(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_1(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2(t) \qquad (16)$$

where $\delta X''(t)$ are deviations from nominal values of X'' at time t. It should be noted that our problem of reducing $X(t)$ to zero is equivalent to reducing $\delta X''(t)$ to zero as shown by Eqs. (15) and (16).

Examining Eq. (16), it becomes evident that our four-dimensional minimum-time problem has been reduced to two two-dimensional problems, since the $\delta \dot{x}_1''$ and $\delta \dot{x}_2''$ equations are decoupled from the $\delta \dot{x}_3''$ and $\delta \dot{x}_4''$ equations. Redefining $\delta X''$ and $\delta \dot{X}''$,

$$\begin{bmatrix} \delta x_1'' \\ \delta x_2'' \\ \delta x_3'' \\ \delta x_4'' \end{bmatrix} \triangleq \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}, \quad \begin{bmatrix} \delta \dot{x}_1'' \\ \delta \dot{x}_2'' \\ \delta \dot{x}_3'' \\ \delta \dot{x}_4'' \end{bmatrix} \triangleq \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \end{bmatrix}$$

Equation (16) becomes

$$\left.\begin{array}{ll} \dot{y}_1 = y_2 & \dot{y}_3 = y_4 \\ \dot{y}_2 = u_1 & \dot{y}_4 = u_2 \end{array}\right\} \qquad (17)$$

Using the results of Appendix A, we find the multiplier equations for y_1 and y_2 to be

$$\dot{\lambda}_1 = 0$$
$$\dot{\lambda}_2 = -\lambda_1$$

Solving these, we have

$$\lambda_1 = \lambda_1(0)$$
$$\lambda_2 = -\lambda_1(0)\,t + \lambda_2(0)$$

Since $u_1^* = -\text{sgn}(\lambda_2)$, we see that only one switching is possible.

Now, solving the y_1, y_2 equations for constant u_1, we have

$$y_1(t) = \frac{1}{2}u_1 t^2 + y_2(0)\,t + y_1(0)$$
$$y_2(t) = u_1 t + y_2(0)$$

Eliminating t from these equations, we find that

$$2u_1(y_1 - y_1(0)) = (y_2 - y_2(0))^2 + 2y_2(0)(y_2 - y_2(0)) \tag{18}$$

Equation (18) shows that the vehicle will follow a parabolic trajectory in the y_1, y_2 plane (see Fig. 4) for constant u_1. Coupling this fact with the fact that only one switching is optimal, the "switching boundary" is obtained, as shown in Fig. 4. A similar analysis is valid for y_3 and y_4, and the switching boundary is the same as for y_1 and y_2. The expected trajectory for a set of initial deviations from the nominal trajectory is also shown in Fig. 4.

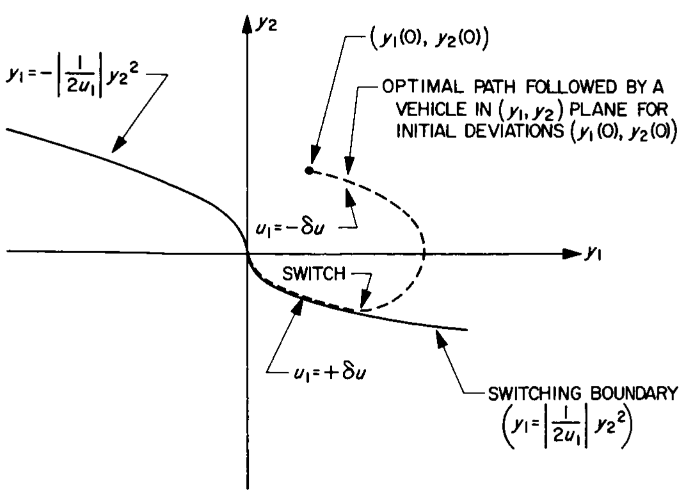


Fig. 4. Definition of the "switching boundary" in the (y_1, y_2) plane

V. Experimental Results of the First Solution

A computer program was written (see Appendix B) to simulate the flight of a space vehicle on a typical nominal trajectory. An initial velocity error of about 12 m/sec, considered to be typical (Ref. 1), was used. An initial position error that would result from 3 days of such an initial velocity error was used (assuming that the engine was turned on 3 days after launch). A nominal value of 10^{-3} m/sec^2 was used for the low-thrust acceleration, with the vehicle weight taken at 4,535 kg. Hence u_1 and u_2 were taken as 10% of u_n, or 10^{-4} m/sec^2.

It was found that the error incurred by neglecting $\dot{\beta} x_1$ and $\dot{\beta} x_3$ was large enough to require that the minimum-time solution be applied twice; that is, the large initial errors were reduced, and then the resulting errors were reduced. The trajectories obtained are shown in Figs. 5–10. A deviation from zero indicates a deviation from the nominal state.

Probably the most significant disturbance on a practical system will be attitude-control limit cycle operation causing attitude variation of the thrust vector. A sinusoidal disturbance with an amplitude of 1 deg (peak-to-peak) and a frequency of 1 cycle per 20 min was put into the control system. The resulting trajectories are shown in Figs. 11–16, and the performance is seen to be very good. More work is certainly needed in investigating the effects of other disturbances on this control system and the ones following.

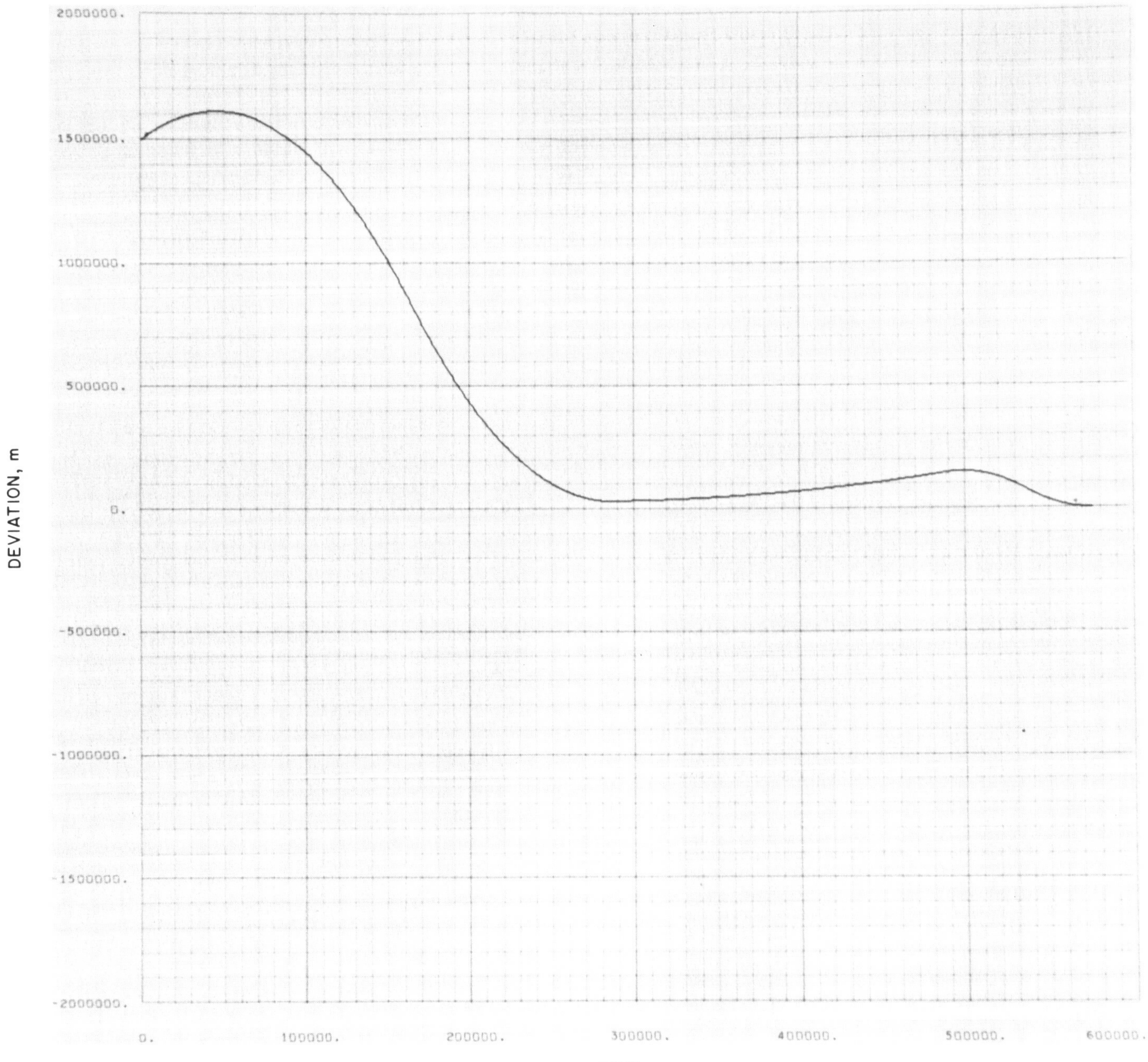


Fig. 5. The x_1 position deviation vs time for the first solution

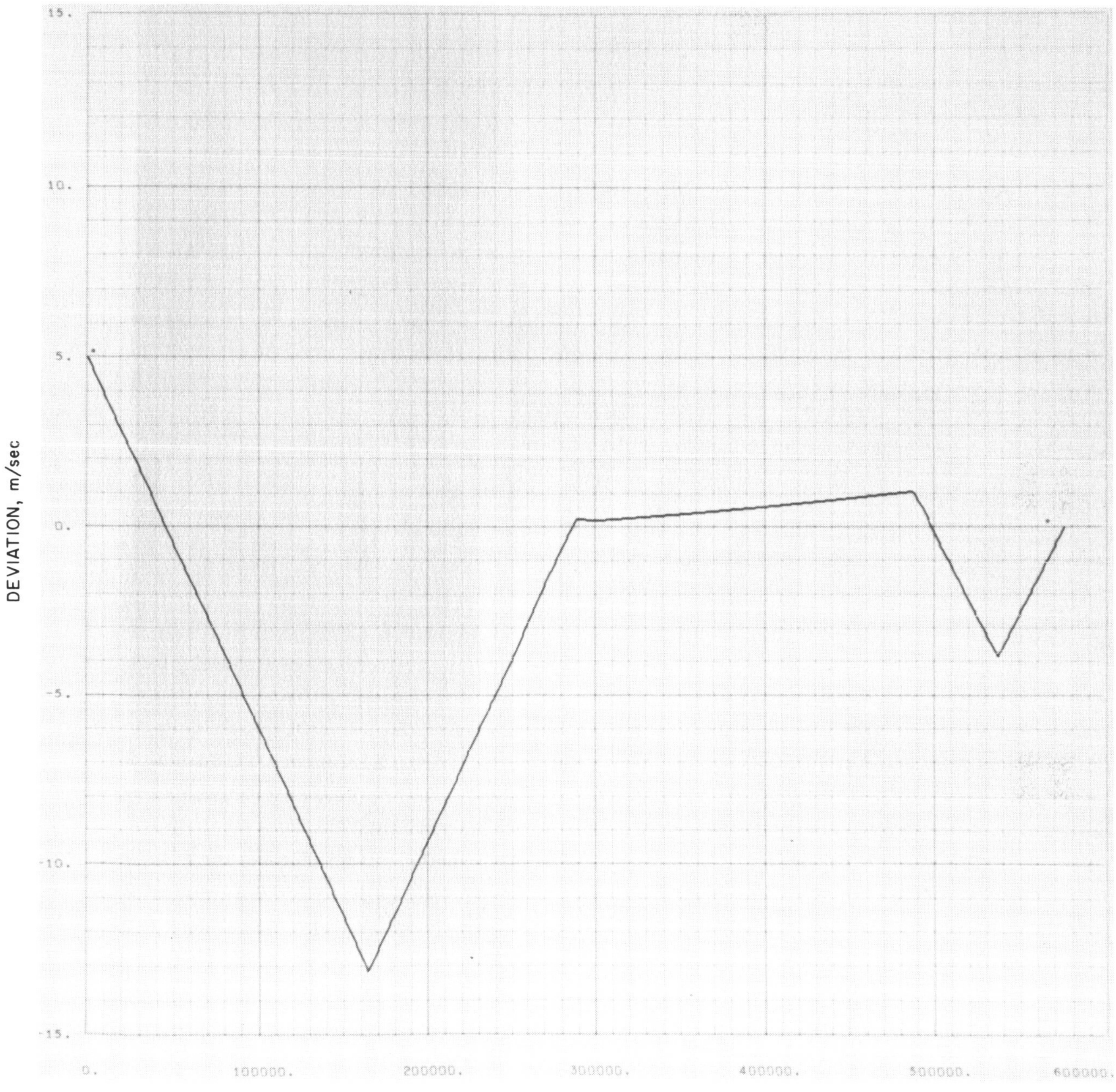


Fig. 6. The x_1 velocity deviation vs time for the first solution

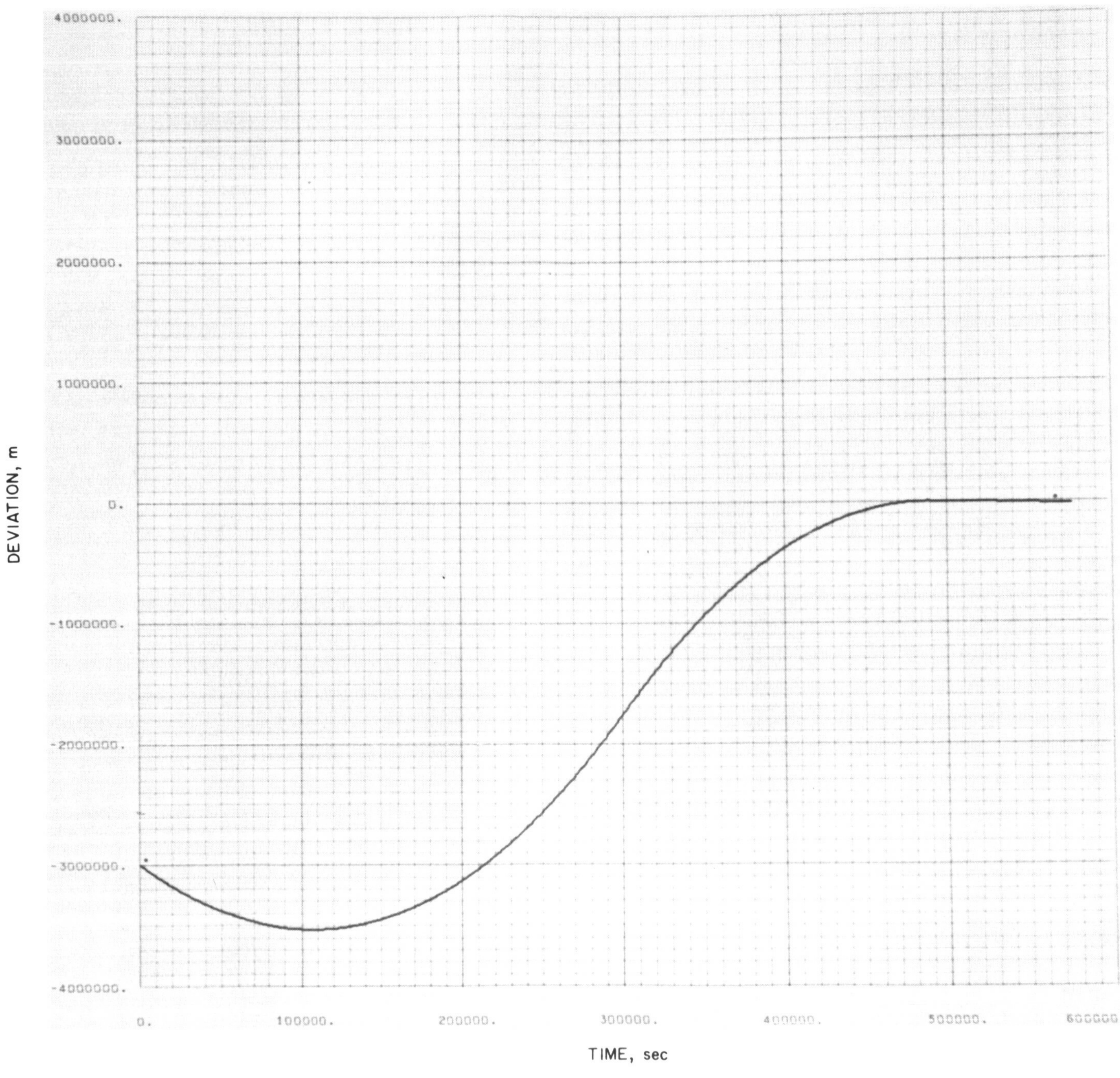


Fig. 7. The x_3 position deviation vs time for the first solution

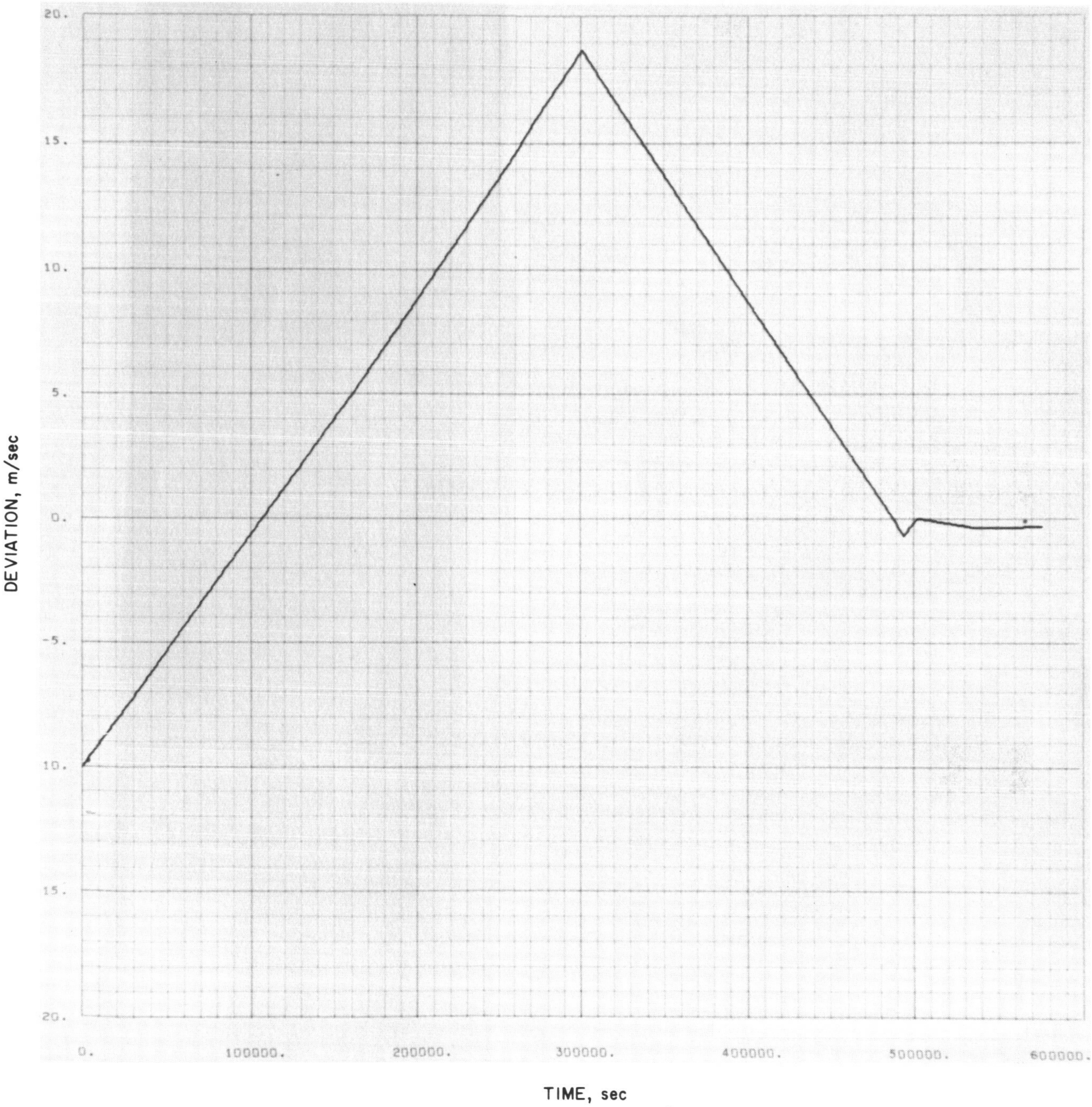


Fig. 8. The x_3 velocity deviation vs time for the first solution

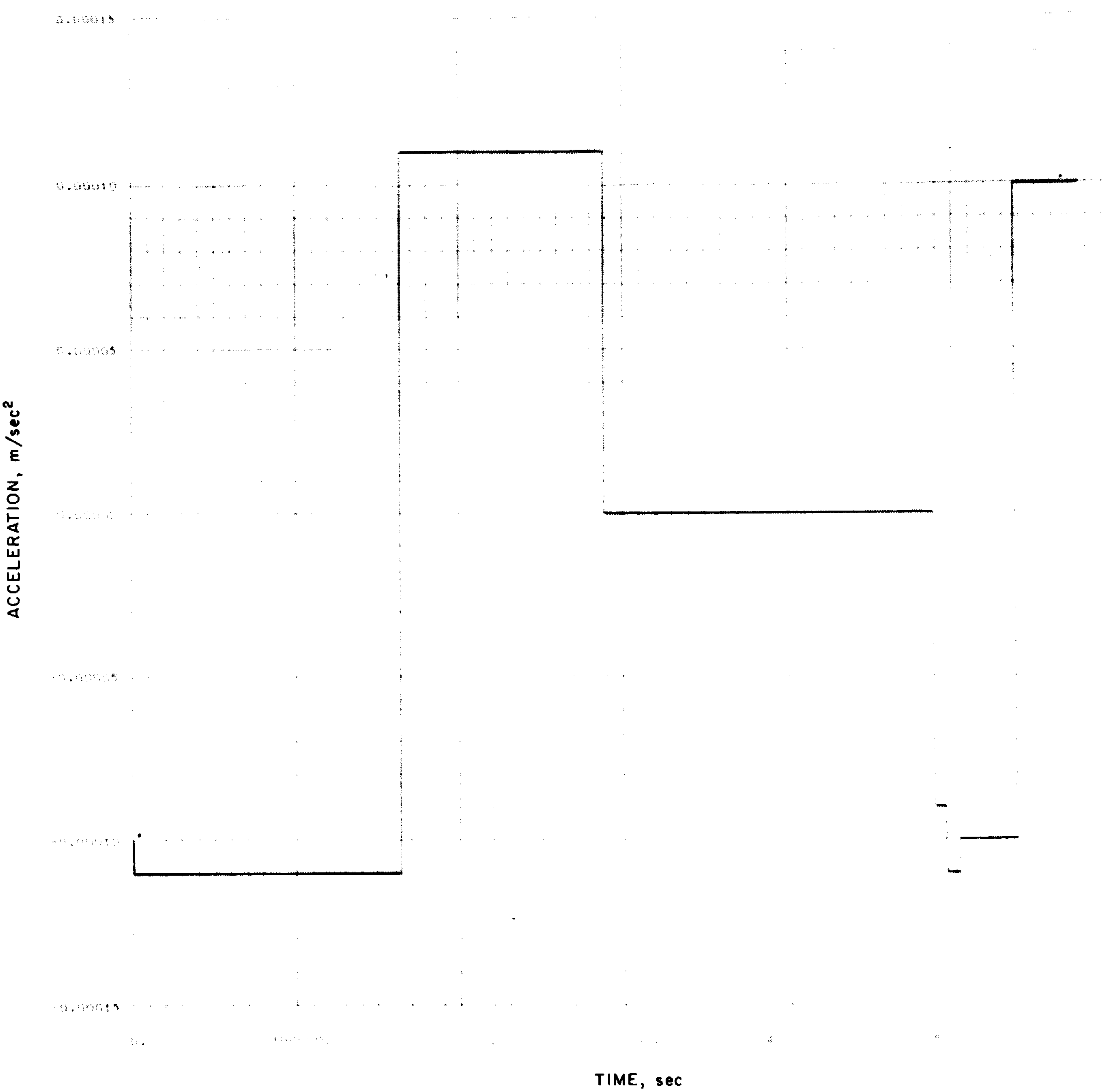


Fig. 9. The control variable u_1 vs time for the first solution

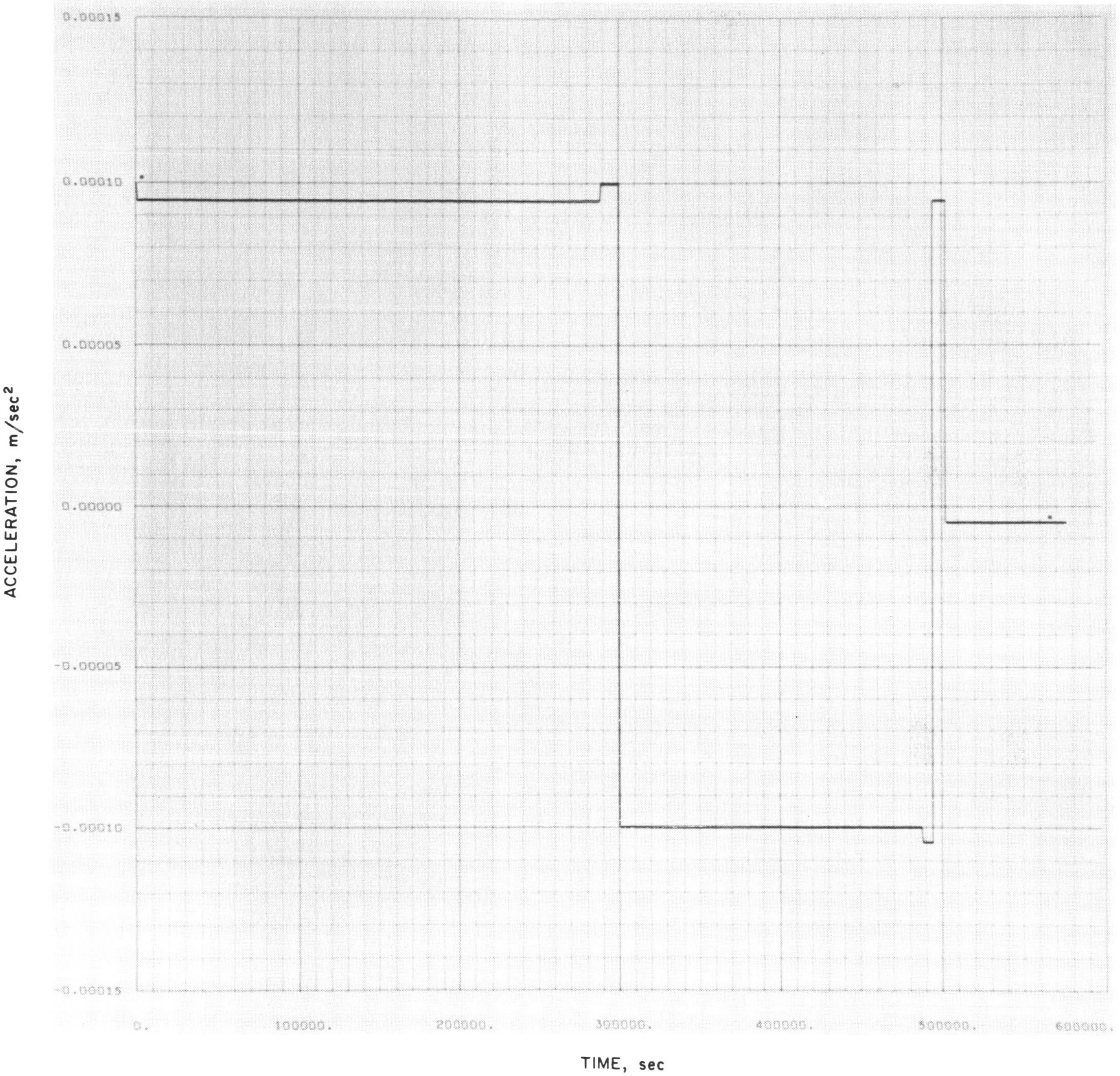


Fig. 10. The control variable u_2 vs time for the first solution

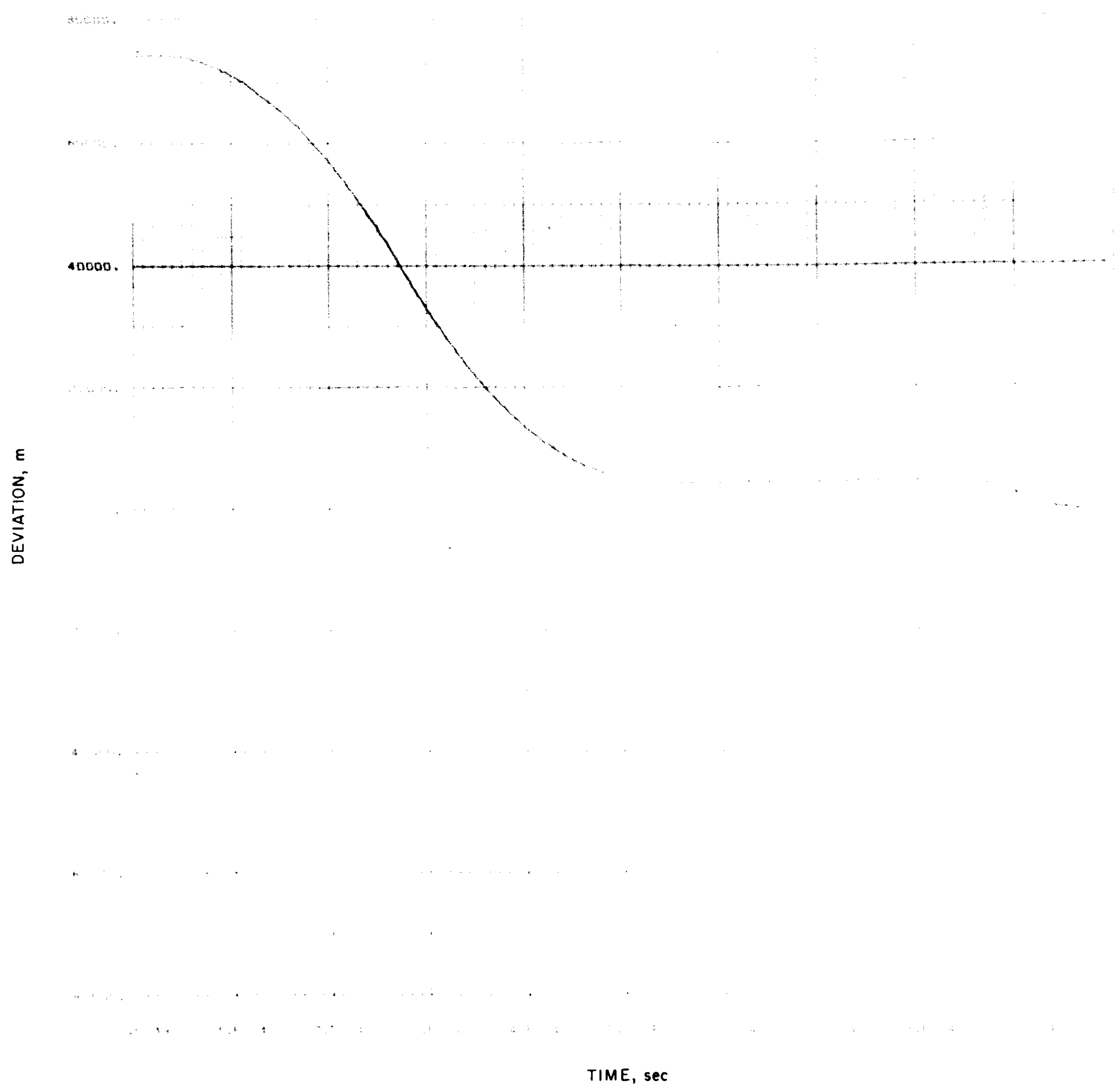


Fig. 11. The x_1 position deviation vs time for the first solution, with attitude variations

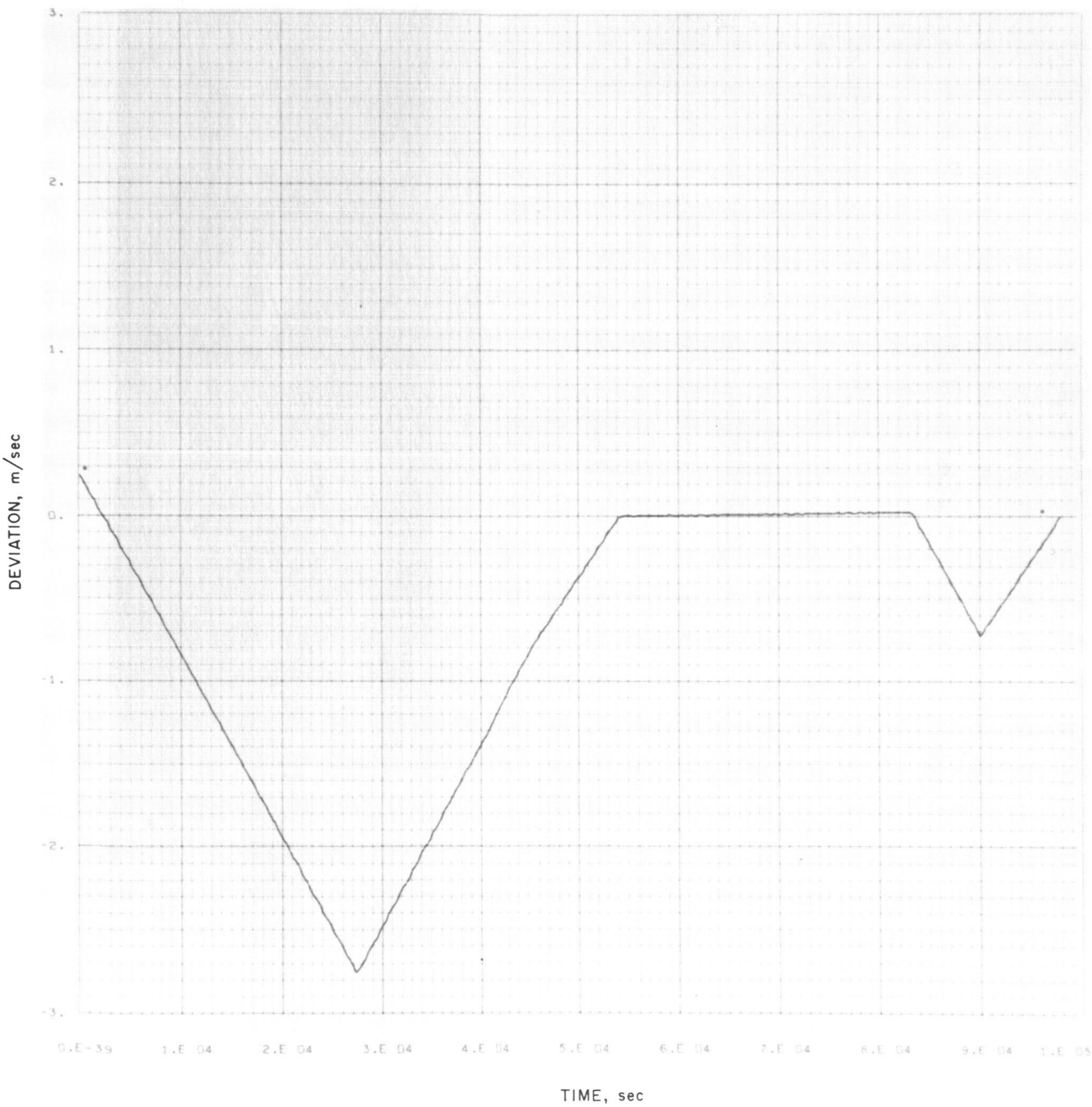


Fig. 12. The x_1 velocity deviation vs time for the first solution, with attitude variations

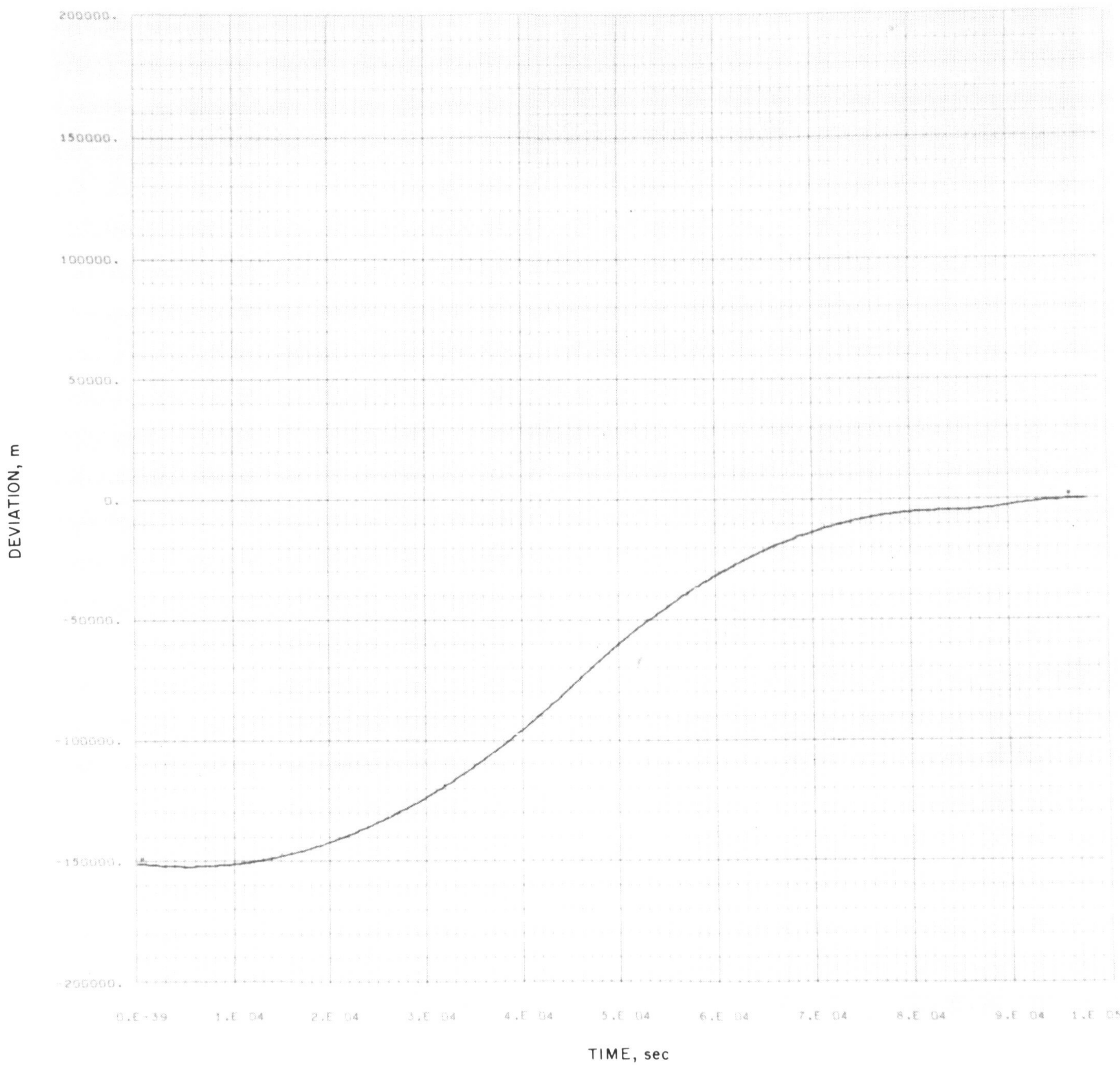


Fig. 13. The x_3 position deviation vs time for the first solution, with attitude variations

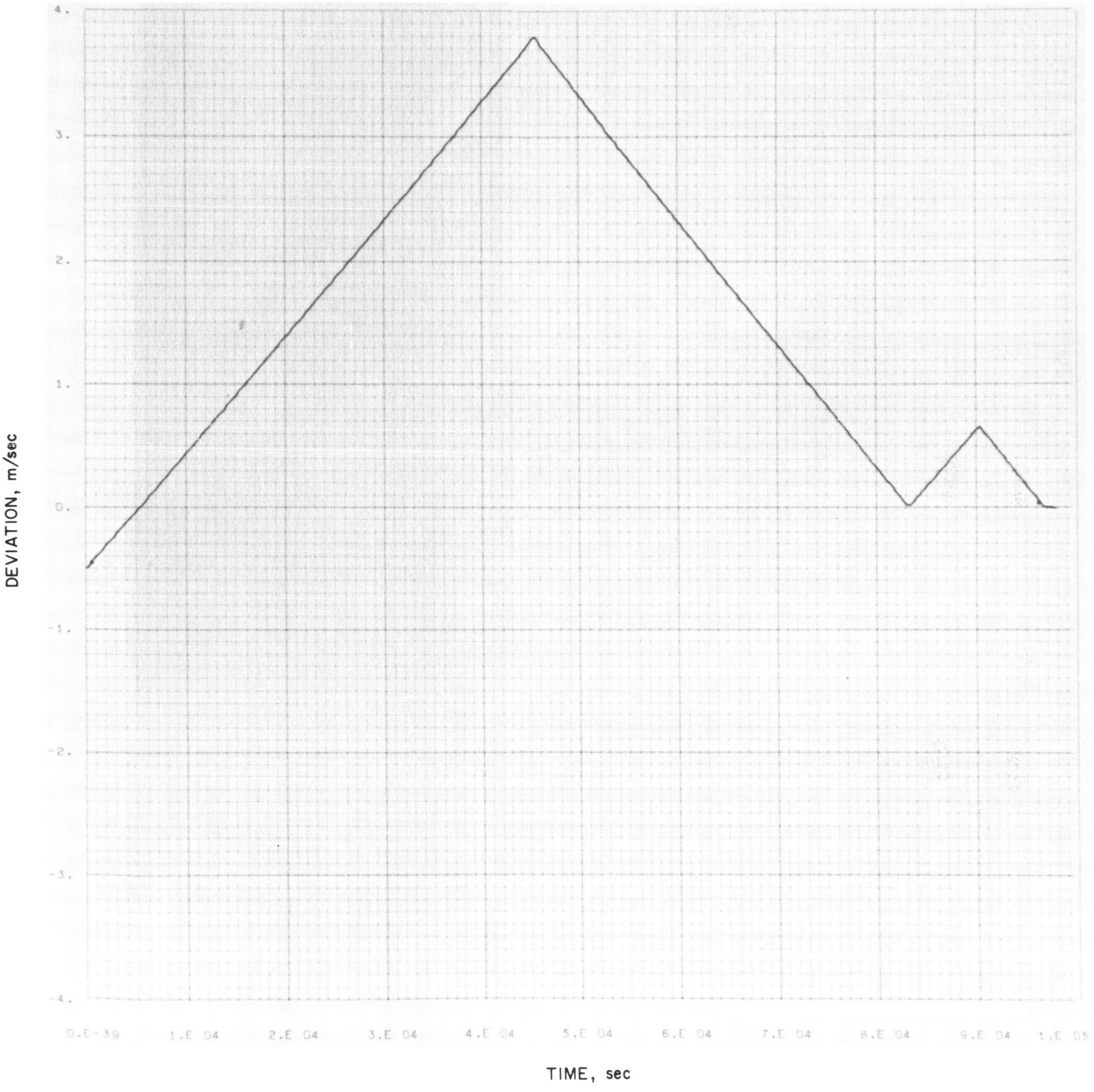


Fig. 14. The x_3 velocity deviation vs time for the first solution, with attitude variations

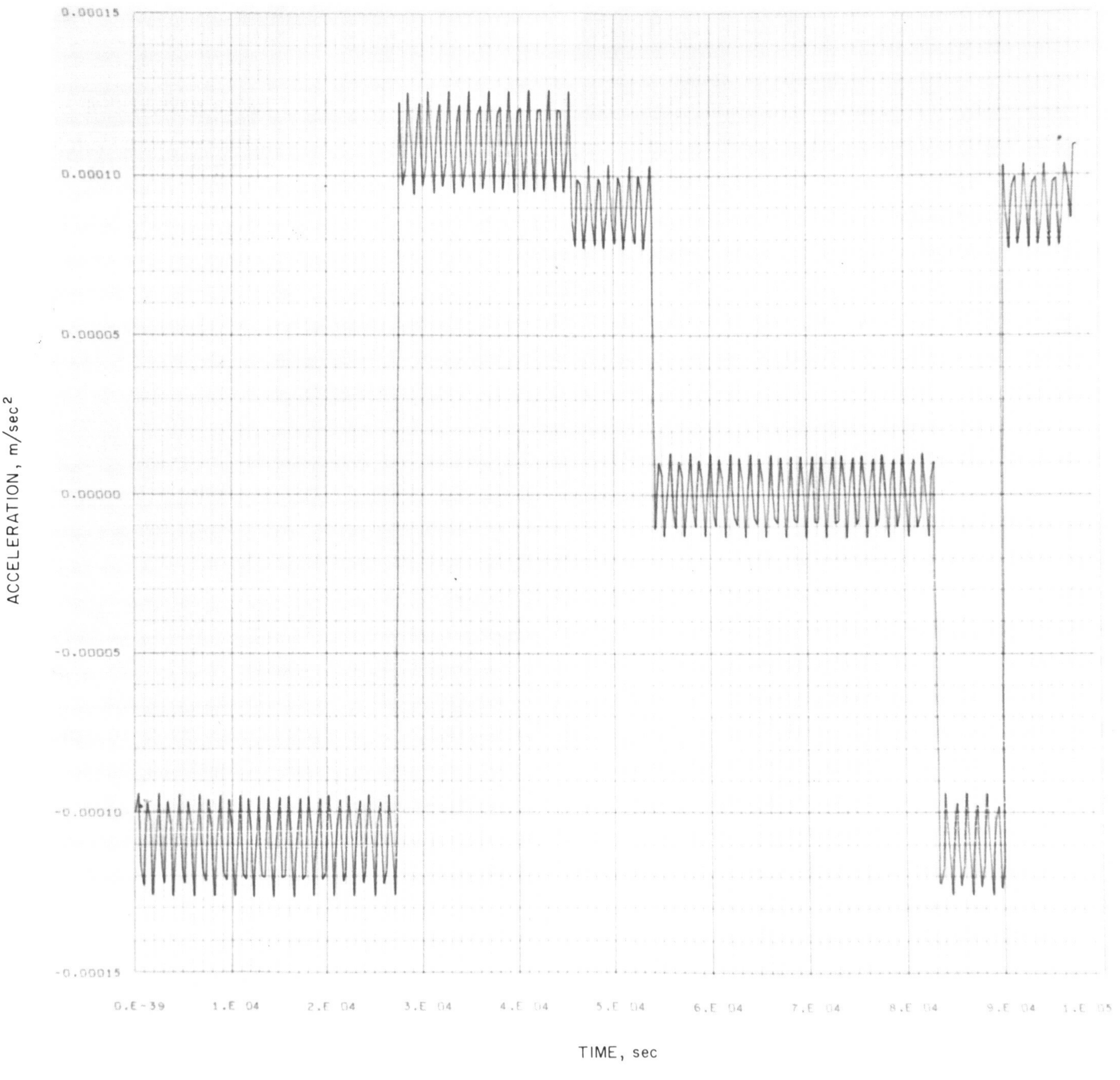


Fig. 15. The control variable u_1 vs time, with attitude variations

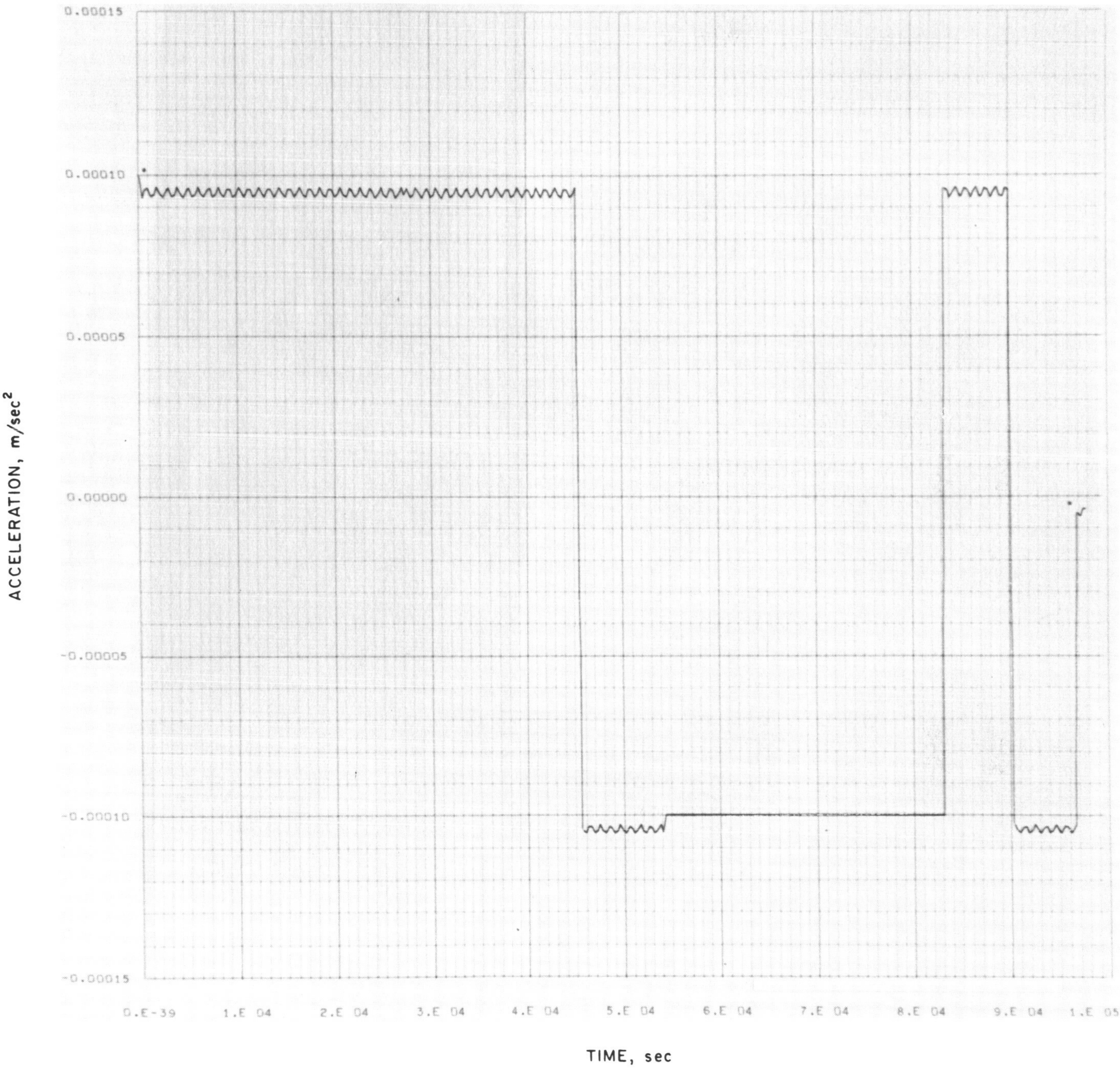


Fig. 16. The control variable u_2 vs time, with attitude variations

VI. Second Solution of the Minimum-Time Problem

With the aid of a digital computer, it may be possible to solve the four-dimensional minimum-time problem exactly —a difficult task in general. It is anticipated that by doing this, one may be able to reduce the number of switchings necessary and, consequently, the number of commands to be executed by the space vehicle.

First, we shall linearize Eqs. (1) as follows:

$$\dot{X}'(t) = \dot{X}'_n(t) + \dot{X}(t) = F(u_n + \delta u, \gamma_n + \delta\gamma, X'_n + X) = F(u_n, \gamma_n, X'_n) + F_u \delta_u + F_\gamma \delta\gamma + F_{X'} X + \text{(higher-order terms)} \tag{19}$$

As before, we neglect higher-order terms, and the terms A, B, C, and D in $F_{X'}$. Also, we use the definitions of u_1 and u_2 to obtain

$$\dot{X}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} X(t) + \begin{bmatrix} 0 \\ \cos\beta(t) \\ 0 \\ \sin\beta(t) \end{bmatrix} u_1(t) + \begin{bmatrix} 0 \\ -\sin\beta(t) \\ 0 \\ \cos\beta(t) \end{bmatrix} u_2(t) \tag{20}$$

As was pointed out previously, this linearization is an excellent approximation to the true differential equations. Using Appendix A, we can write the multiplier equations

$$\begin{bmatrix} \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \dot{\lambda}_3 \\ \dot{\lambda}_4 \end{bmatrix} = - \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix}$$

Solving this system, we obtain

$$\lambda_1 = \lambda_1(0)$$

$$\lambda_2 = -\lambda_1(0)\,t + \lambda_2(0)$$

$$\lambda_3 = \lambda_3(0)$$

$$\lambda_4 = -\lambda_3(0)\,t + \lambda_4(0)$$

Now the optimal controls are

$$\left.\begin{aligned} u_1^*(t) &= -\text{sgn}\,((-\lambda_1(0)\,t + \lambda_2(0))\cos\beta(t) + (-\lambda_3(0)\,t + \lambda_4(0))\sin\beta(t)) \\ u_2^*(t) &= -\text{sgn}\,((-\lambda_1(0)\,t + \lambda_2(0))(-\sin\beta(t)) + (-\lambda_3(0)\,t + \lambda_4(0))\cos\beta(t)) \end{aligned}\right\} \tag{21}$$

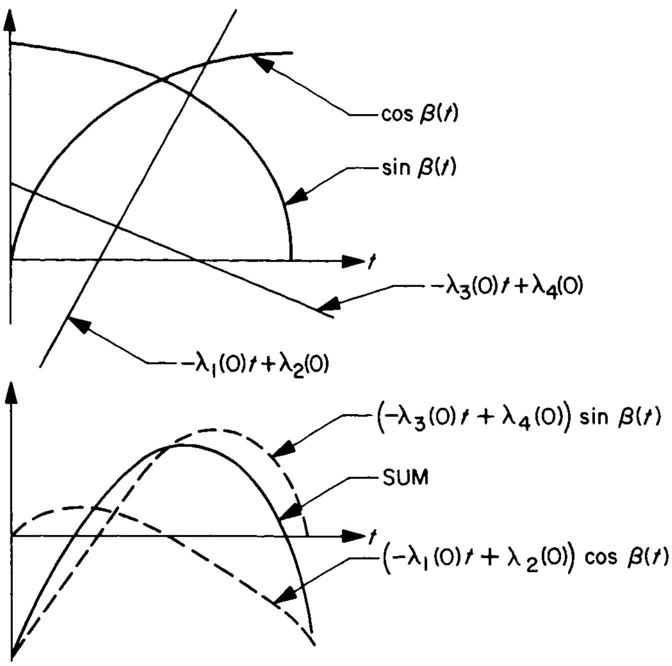
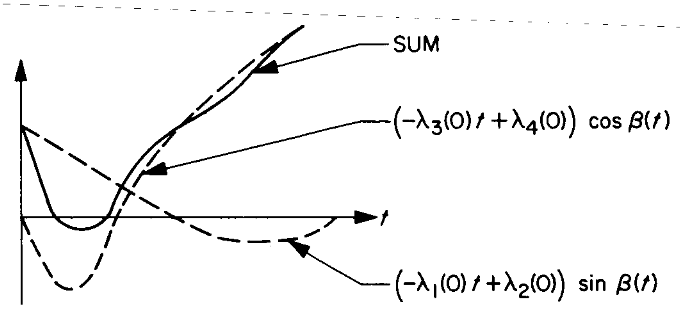


Fig. 17. Control function switchings for second solution



Some possible solutions of Eqs. (21) appear in Fig. 17. (Note that β is not expected to exceed 90 deg before nominal trajectory acquisition.) It seems intuitively reasonable, then, that each control would have a maximum of two switchings for $\beta(T)$ less than 90 deg.

Now, given the initial conditions on Eq. (20), we can write the explicit solution for $X(T)$, where T is nominal trajectory acquisition time. That is,

$$X(T) = \Phi(T,0)\,X(0) + \int_0^T \Phi(T,t) \begin{bmatrix} 0 \\ \cos\beta(t) \\ 0 \\ \sin\beta(t) \end{bmatrix} u_1(t)\,dt + \int_0^T \Phi(T,t) \begin{bmatrix} 0 \\ -\sin\beta(t) \\ 0 \\ \cos\beta(t) \end{bmatrix} u_2(t)\,dt \tag{22}$$

where $\Phi(t_2, t_1)$ is the fundamental matrix that satisfies the matrix differential equation

$$\dot{\Phi}(t_2, t_1) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Phi(t_2, t_1) \tag{23}$$

with

$$\Phi(t_1, t_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The solution of Eq. (23) is

$$\Phi(t_2, t_1) = \begin{bmatrix} 1 & (t_2 - t_1) & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & (t_2 - t_1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}$$

Since the absolute values of u_1 and u_2 are constant, only the sign of these quantities is needed inside the integrals of Eq. (22). If we designate $u_1(0)$ and $u_2(0)$ as the initial values of u_1 and u_2, t_1 and t_2 as the switching times of u_1, and t_3 and t_4 as the switching times of u_2 (recall that a maximum of two switchings is possible for each control variable), then Eq. (22) becomes

$$X(T) = \Phi(T, 0) X(0) + u_1(0)\left(\int_0^{t_1} - \int_{t_1}^{t_2} + \int_{t_2}^{T}\right) + u_2(0)\left(\int_0^{t_3} - \int_{t_3}^{t_4} + \int_{t_4}^{T}\right) \tag{25}$$

The integrals of Eq. (25) can be explicitly evaluated if we assume that β varies at a constant rate. This is an excellent approximation for the trajectories of interest. Hence, if we assume that

$$\beta(t) = wt \qquad w = \text{constant} \doteqdot \dot{\beta}$$

and if we define

$$\begin{aligned} I_1 = {} & u_1(0)\left(\frac{2t_1}{w}\sin wt_1 + \frac{2}{w^2}\cos wt_1 - \frac{2t_2}{w}\sin wt_2 - \frac{2\cos wt_2}{w^2} + \frac{T\sin wT}{w} + \frac{1}{w^2}\cos wT - \frac{1}{w^2}\right) \\ & - u_2(0)\left(-\frac{2t_3\cos wt_3}{w} + \frac{2\sin wt_3}{w^2} + \frac{2t_4\cos wt_4}{w} - \frac{2\sin wt_4}{w^2} - \frac{T\cos wT}{w} + \frac{\sin wT}{w^2}\right) \end{aligned}$$

$$I_2 = u_1(0)\left(\frac{2}{w}\sin wt_1 - \frac{2}{w}\sin wt_2 + \frac{1}{w}\sin wT\right) - u_2(0)\left(-\frac{2}{w}\cos wt_3 + \frac{2}{w}\cos wt_4 - \frac{1}{w}\cos wT + \frac{1}{w}\right)$$

$$\begin{aligned} I_3 = {} & u_1(0)\left(\frac{-2t_1\cos wt_1}{w} + \frac{2\sin wt_1}{w^2} + \frac{2t_2\cos wt_2}{w} - \frac{2\sin wt_2}{w^2} - \frac{T\cos wT}{w} + \frac{\sin wT}{w^2}\right) \\ & + u_2(0)\left(\frac{2t_3}{w}\sin wt_3 + \frac{2}{w^2}\sin wt_3 - \frac{2t_4}{w}\sin wt_4 - \frac{2\cos wt_4}{w^2} + \frac{T\sin wT}{w} + \frac{1}{w^2}\cos wT - \frac{1}{w^2}\right) \end{aligned}$$

$$I_4 = u_1(0)\left(\frac{-2\cos wt_1}{w} + \frac{2}{w}\cos wt_2 - \frac{1}{w}\cos wT + \frac{1}{w}\right) + u_2(0)\left(\frac{2}{w}\sin wt_3 - \frac{2}{w}\sin wt_4 + \frac{1}{w}\sin wT\right)$$

then Eq. (25) becomes

$$\left.\begin{aligned} x_1(T) &= x_1(0) + Tx_2(0) - I_1 + TI_2 \triangleq G_1(t_1, t_2, t_3, t_4, T) \\ x_2(T) &= x_2(0) + I_2 \triangleq G_2(t_1, t_2, t_3, t_4, T) \\ x_3(T) &= x_3(0) + Tx_4(0) - I_3 + TI_4 \triangleq G_3(t_1, t_2, t_3, t_4, T) \\ x_4(T) &= x_4(0) + I_4 \triangleq G_4(t_1, t_2, t_3, t_4, T) \end{aligned}\right\} \tag{26}$$

Equations (26) are four equations in five unknowns. Since it is desired that $\mathrm{X}(T) = 0$, the problem is now to find the minimum value of T for which Eqs. (26) can be satisfied. In order to solve Eqs. (26), we first define

$$\mathrm{t} \triangleq \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix}, \mathrm{G} \triangleq \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}$$

Then Eqs. (26) become

$$\mathrm{X}(T) = \mathrm{G}(\mathrm{t}, T) \tag{27}$$

One method of solving Eq. (27) is by a Newton–Raphson iterative technique. If we guess at the vector t for a fixed value of T, $\mathrm{X}(T)$ will in general not be zero, as desired, but some value that we shall designate $\mathrm{X}_\varepsilon(T)$. We wish to find a new vector $\mathrm{t} + \Delta\mathrm{t}$ such that

$$\mathrm{G}(\mathrm{t} + \Delta\mathrm{t}, T) = 0 \tag{28}$$

Making a first-order expansion of Eq. (28), we have

$$\mathrm{G}(\mathrm{t}, T) + \mathbf{G}_\mathrm{t}(\mathrm{t}, T)\,\Delta\mathrm{t} = 0$$

where

$$\mathbf{G}_\mathrm{t} = \left.\begin{bmatrix} \frac{\partial G_1}{\partial t_1} & \cdots & \frac{\partial G_1}{\partial t_4} \\ \vdots & & \vdots \\ \frac{\partial G_4}{\partial t_1} & \cdots & \frac{\partial G_4}{\partial t_4} \end{bmatrix}\right|_{\mathrm{t}, T} \tag{29}$$

and

$$\mathrm{G}(\mathrm{t}, T) = \mathrm{X}_\varepsilon(T)$$

Then, if $\mathbf{G}_t(t, T)$ is nonsingular,

$$\Delta t = -\mathbf{G}_t^{-1}(t, T)\, X_\varepsilon(T) \tag{30}$$

We use Eq. (30) in an iterative fashion to find, for each value of T, the values of t_1, t_2, t_3, and t_4 that make $X(T) = 0$.

Computer analysis indicates that the minimum value of T is achieved when $T = t_2$ or $T = t_4$. In most cases it is easy to guess which solution will prevail. Hence, one control will have one switching, and the other will have two switchings. It is usually an easy matter to determine $u_1(0)$ and $u_2(0)$, and hence Eqs. (26) can be solved for the minimum value of T and for the switching times of the control variables.

VII. Experimental Results of the Second Solution

The flight situation that was used to test the first solution was used on the "exact" solution. It was found that neglecting second-order effects in the control variables caused large errors in this solution. Since our motivation here is to obtain an exact solution, we shall account for the second-order effects by modifying Eq. (25). For the case when $T = t_4$, we have

$$X(T) = \Phi(t,0)\,X(0) + u_1(0)\left(\mathrm{FAC\,1}\int_0^{t_1} - \mathrm{FAC\,1}\int_{t_1}^{t_3} - \mathrm{FAC\,2}\int_{t_3}^{t_2} + \mathrm{FAC\,2}\int_{t_2}^{T}\right) + u_2(0)\left(\mathrm{FAC\,3}\int_0^{t_3} - \mathrm{FAC\,4}\int_{t_3}^{T}\right)$$

where FAC 1, FAC 2, FAC 3, and FAC 4 are the factors that account for the second-order effects. As a result of integrating these equations, one obtains answers very similar to Eqs. (26). The interested reader may find these integrals in the computer program in Appendix B.

Excellent performance was obtained using this modified solution, which includes second-order effects. The results appear in Figs. 18–23, and comparison with Figs. 5–10 shows that the "exact" solution (1) requires about 24 hours less time to acquire the nominal trajectory, and (2) requires 3 fewer commands (switchings) to be sent to the vehicle. It should also be noted that with this solution the relative sizes of u_1, u_2, and $\dot{\beta}\,(=\omega)$ are of no consequence. (This statement was checked using $u = 0.25 \times 10^{-3}$ m/sec^2 and $u_1 = u_2 = 0.25 \times 10^{-4}$ m/sec^2, and the results appears in Figs. 24–29.) The results using attitude variations appear in Figs. 30–35, and the same advantages over the first solution are obtained.

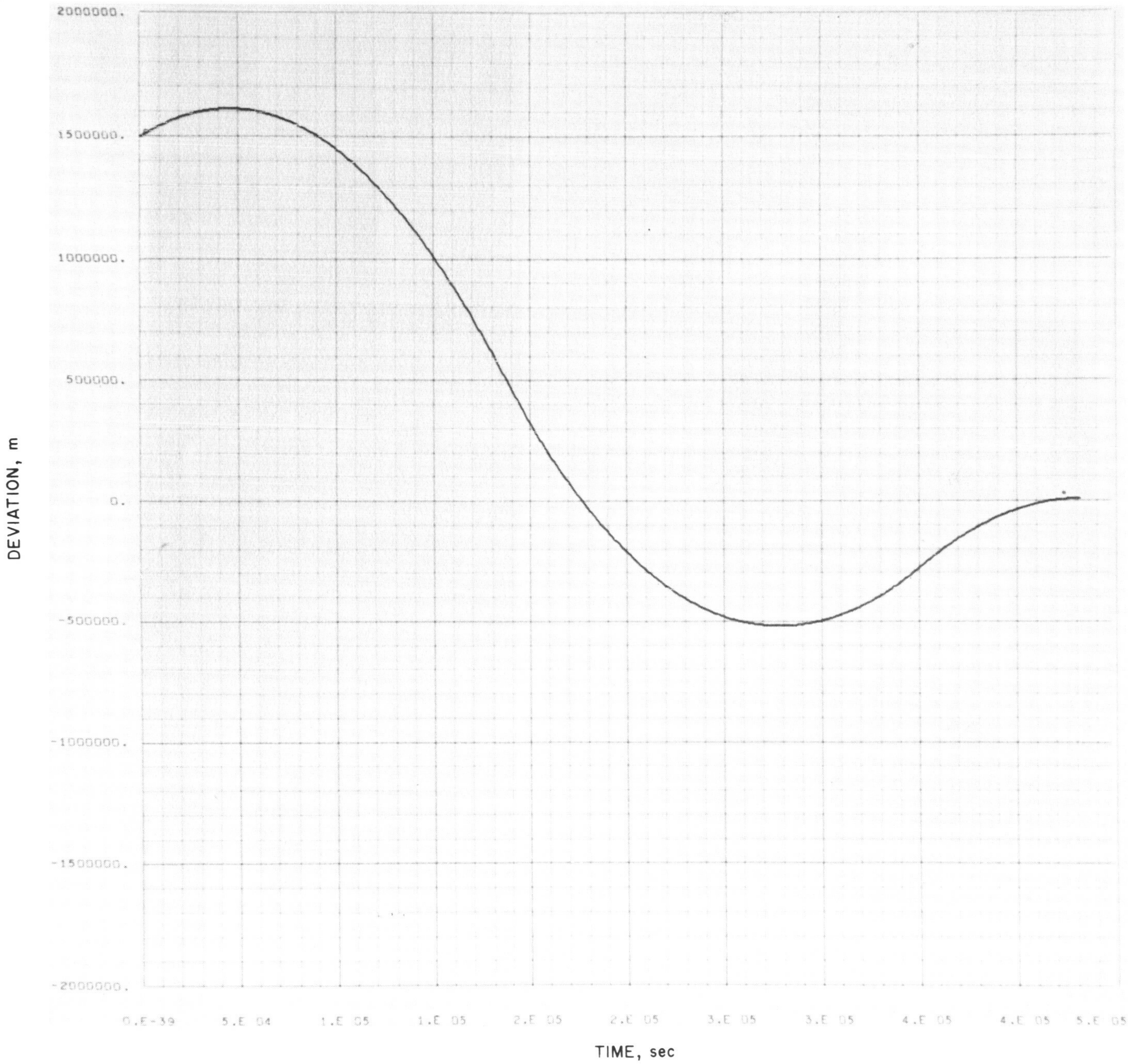


Fig. 18. The x_1 position deviation vs time for the second solution

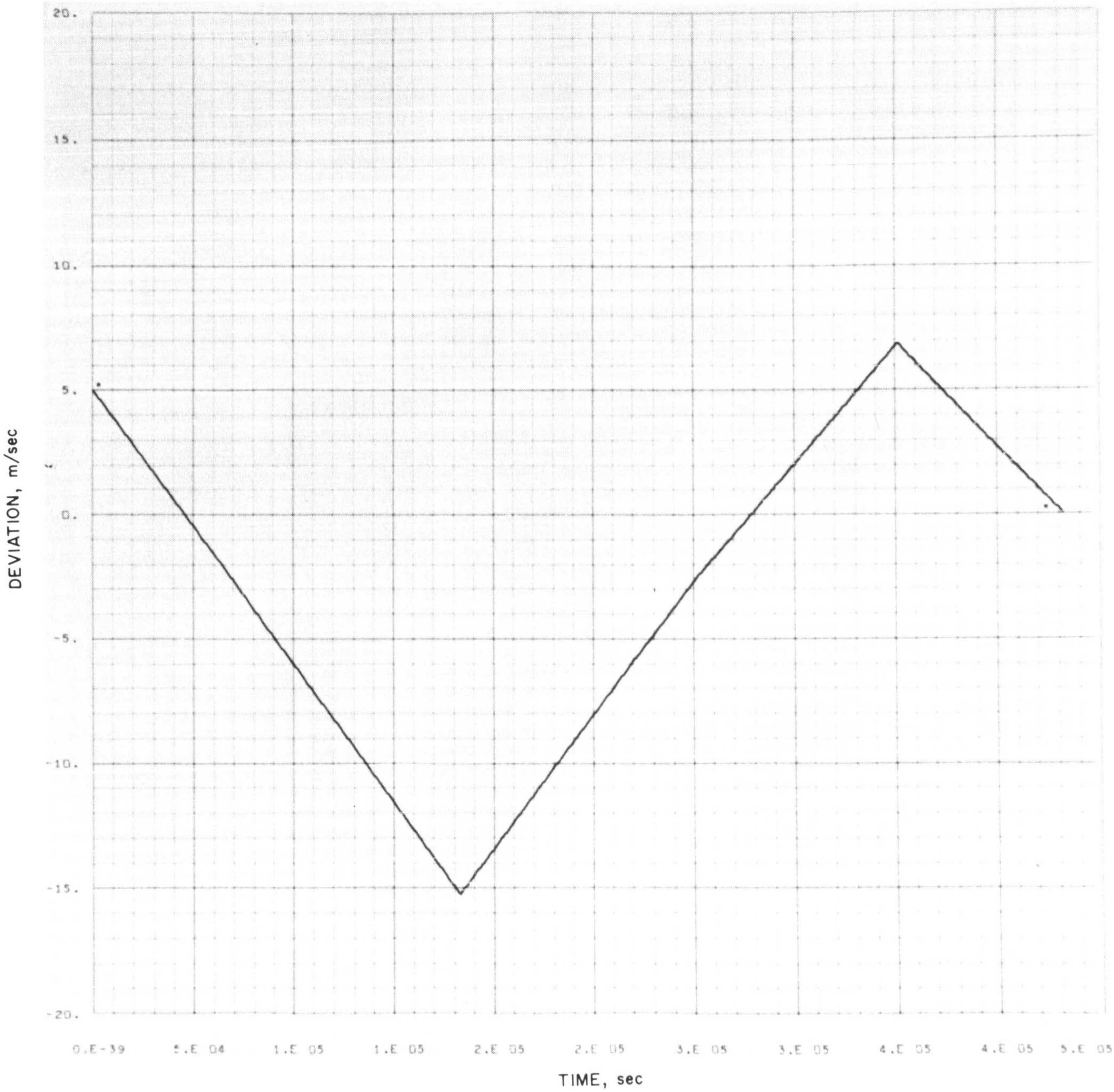


Fig. 19. The x_1 velocity deviation vs time for the second solution

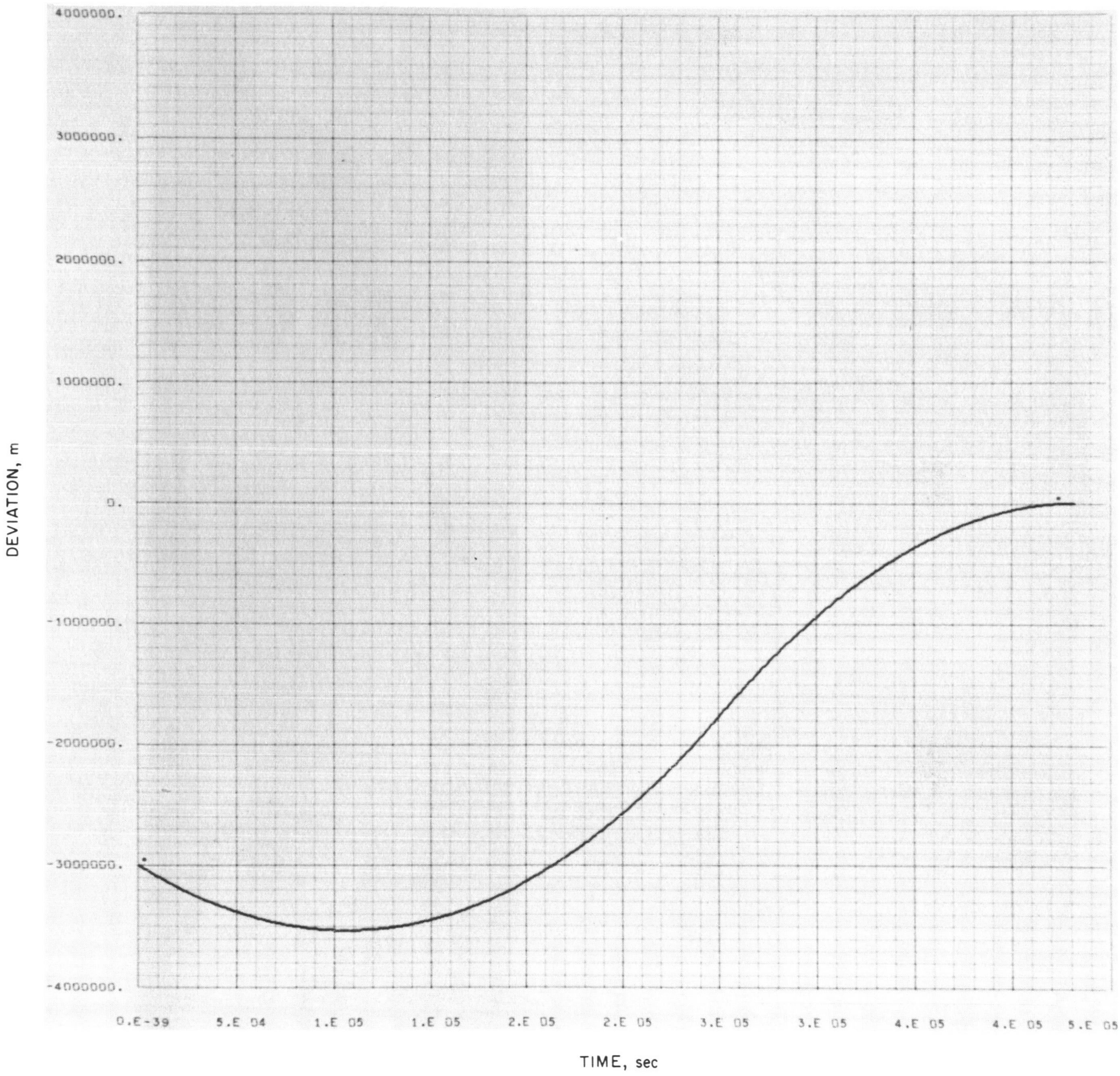


Fig. 20. The x_3 position deviation vs time for the second solution

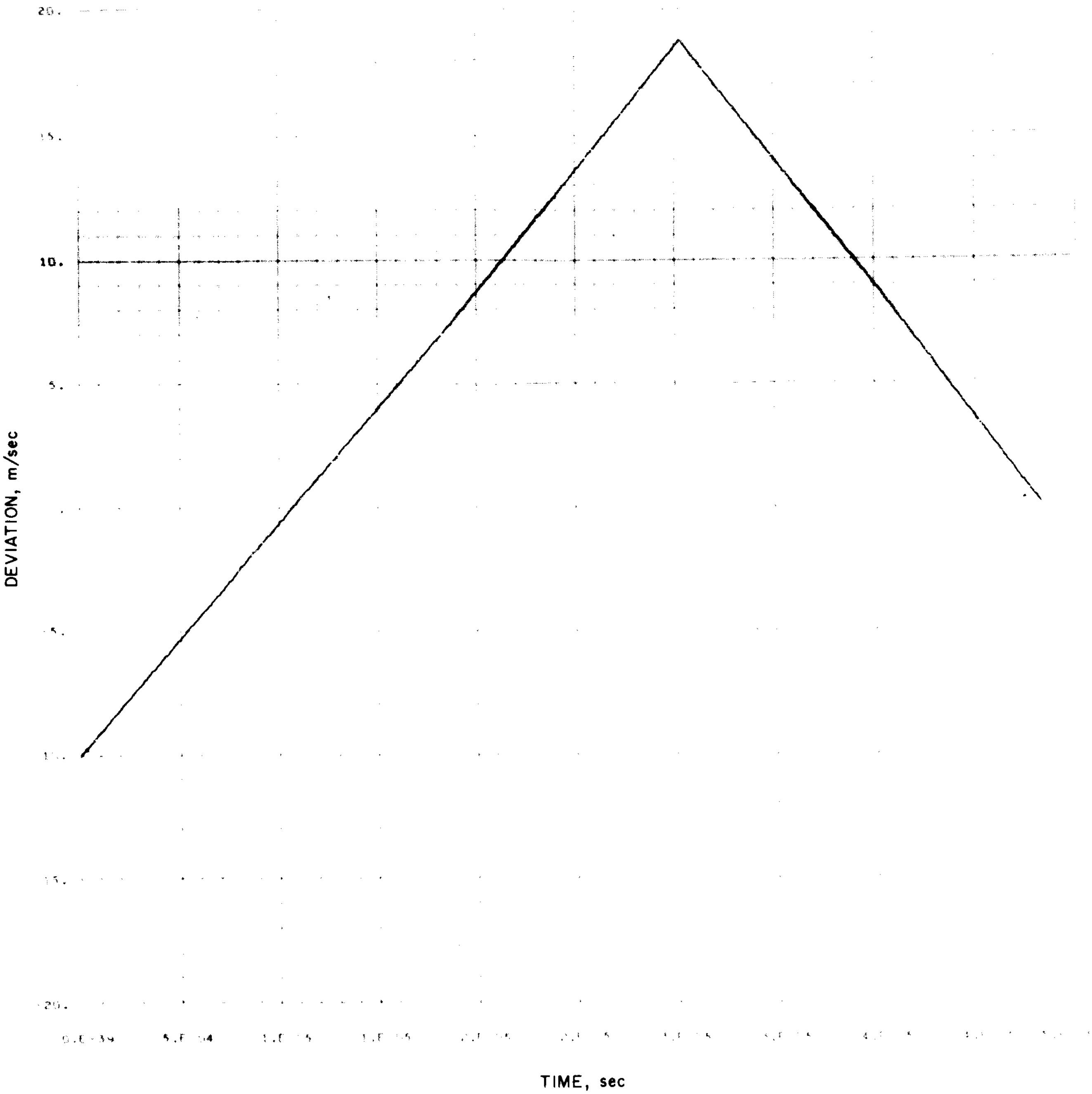


Fig. 21. The x_3 velocity deviation vs time for the second solution

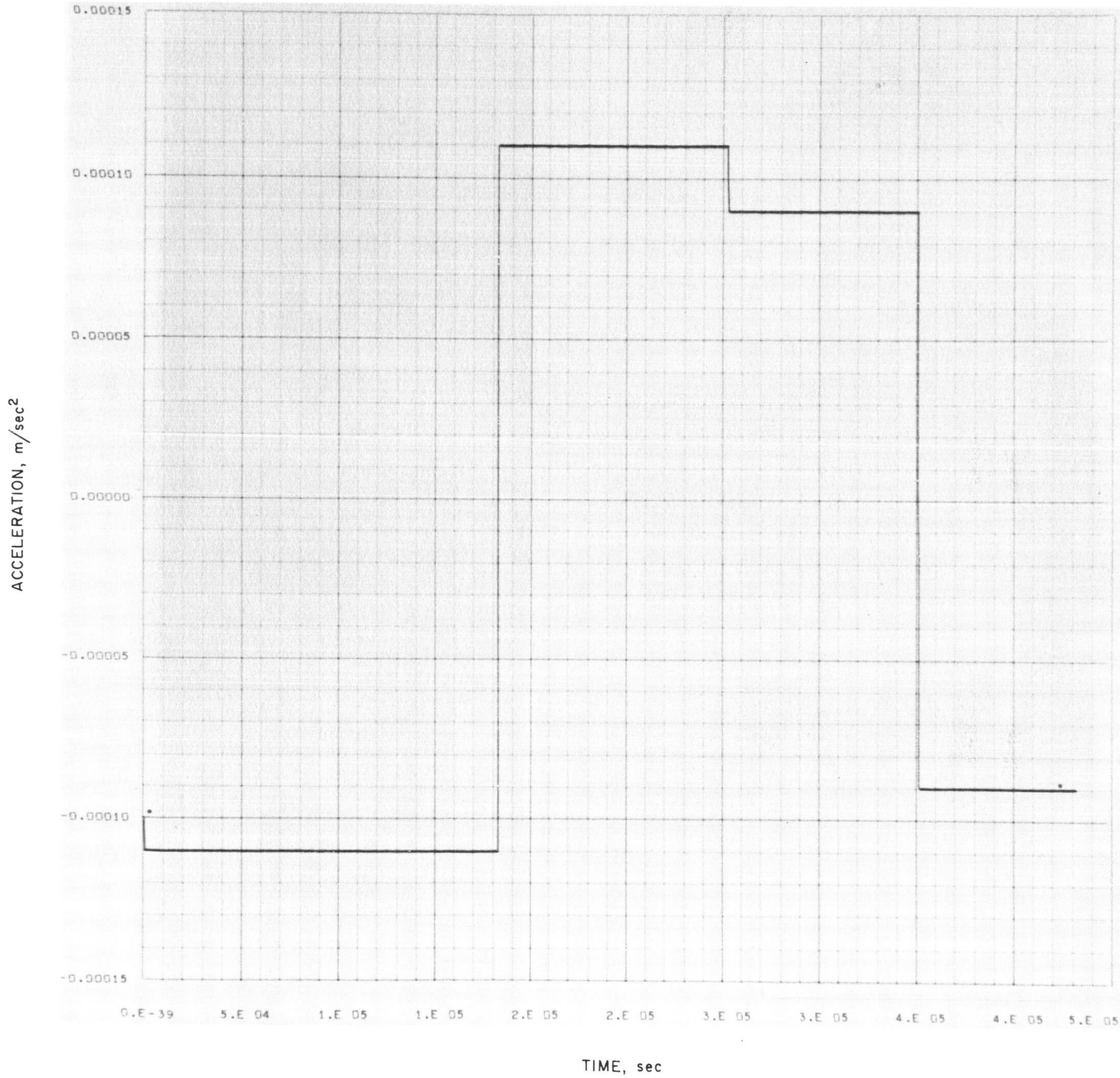


Fig. 22. The control variable u_1 vs time for the second solution

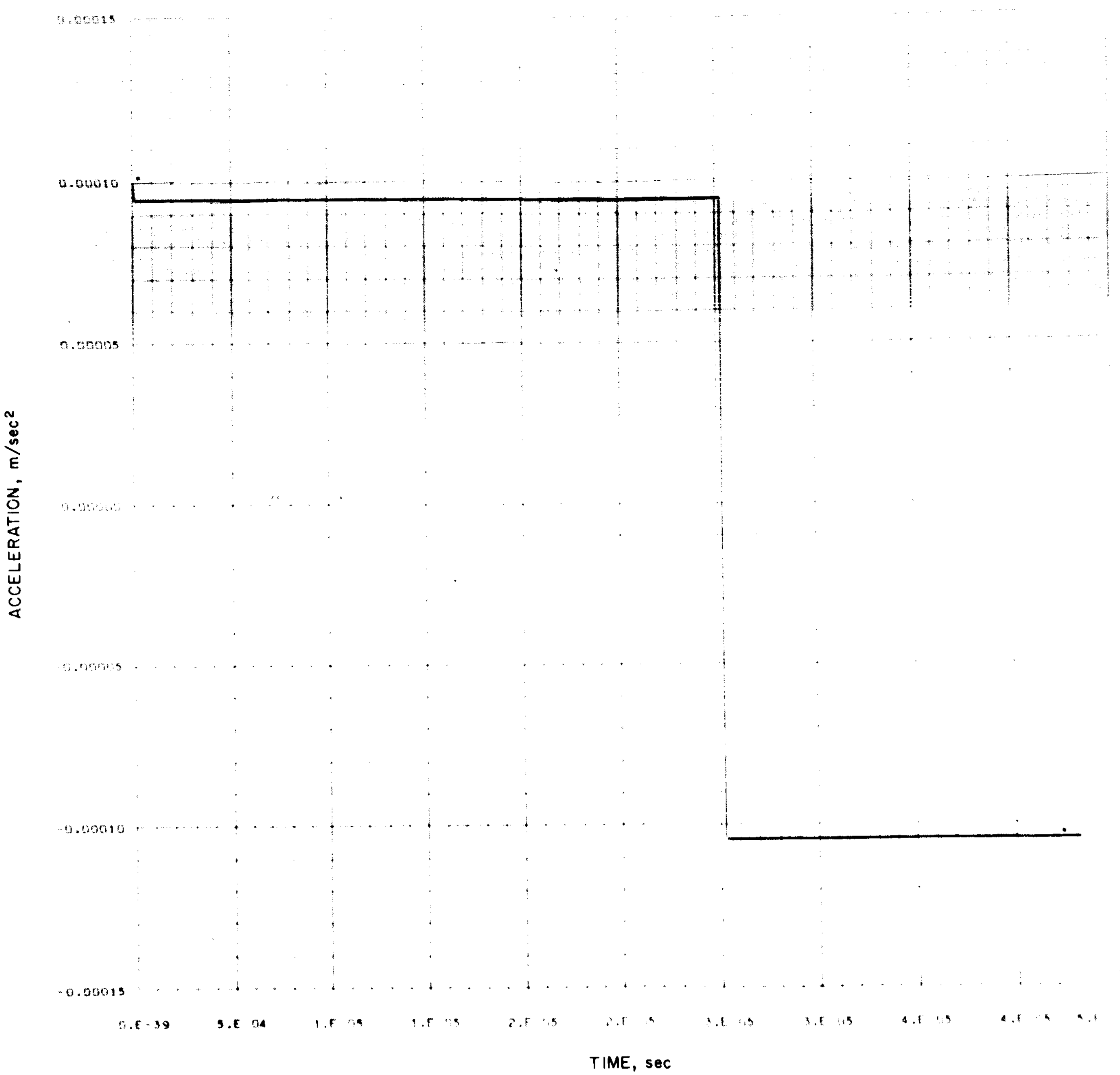


Fig. 23. The control variable u_2 vs time for the second solution

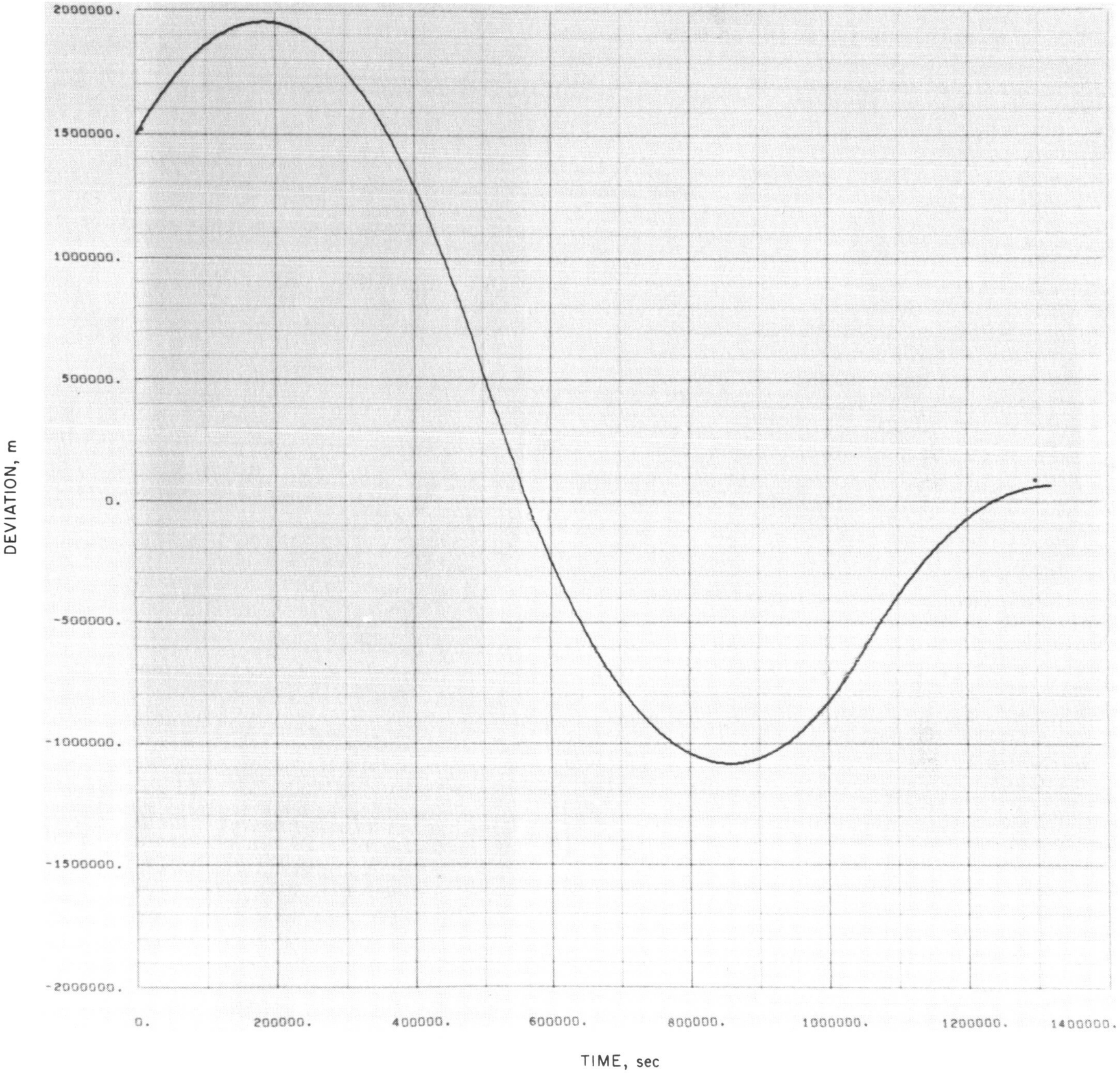


Fig. 24. The x_1 position deviation vs time for the second solution, using smaller u

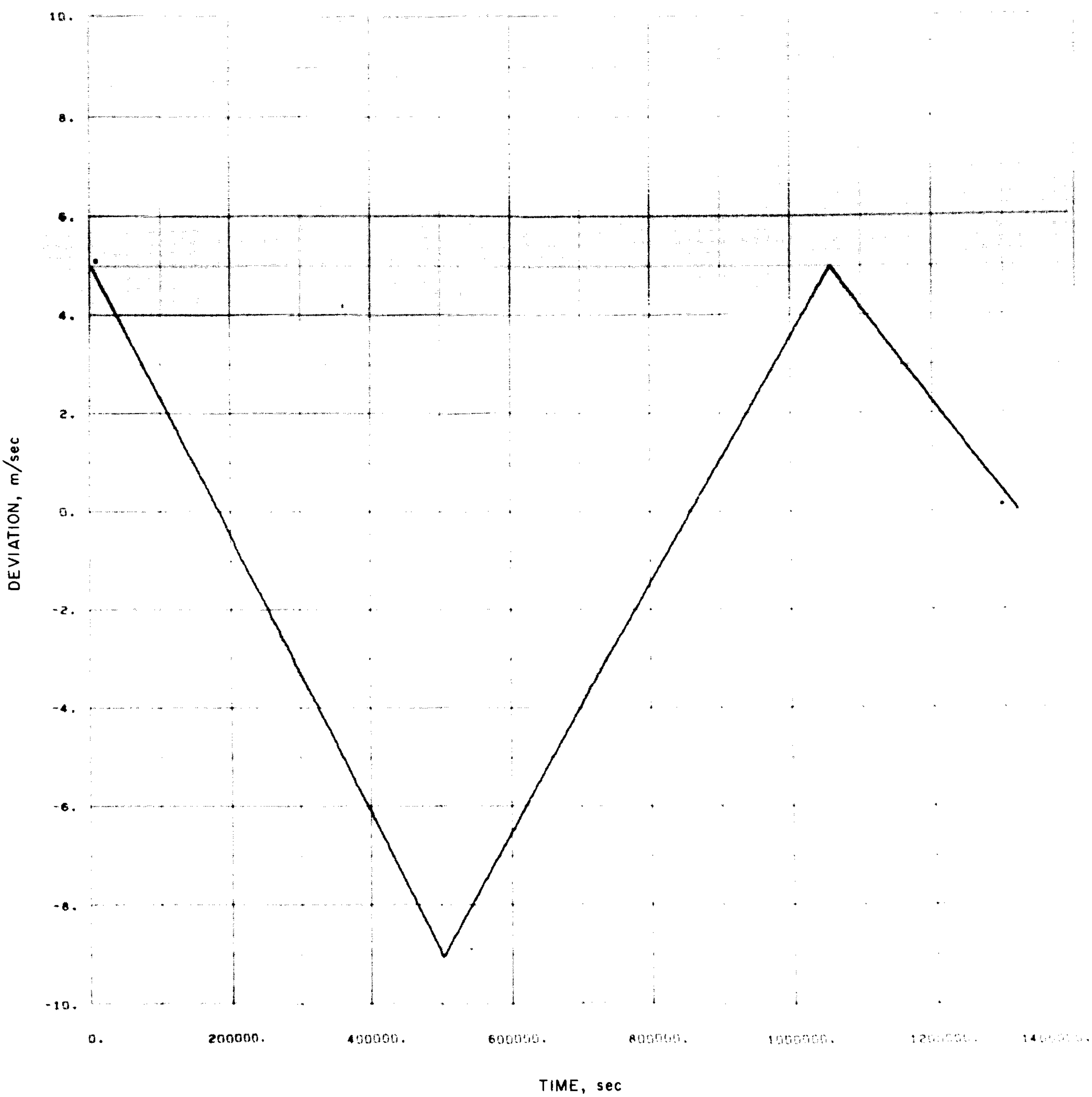


Fig. 25. The x_1 velocity deviation vs time for the second solution, using smaller u

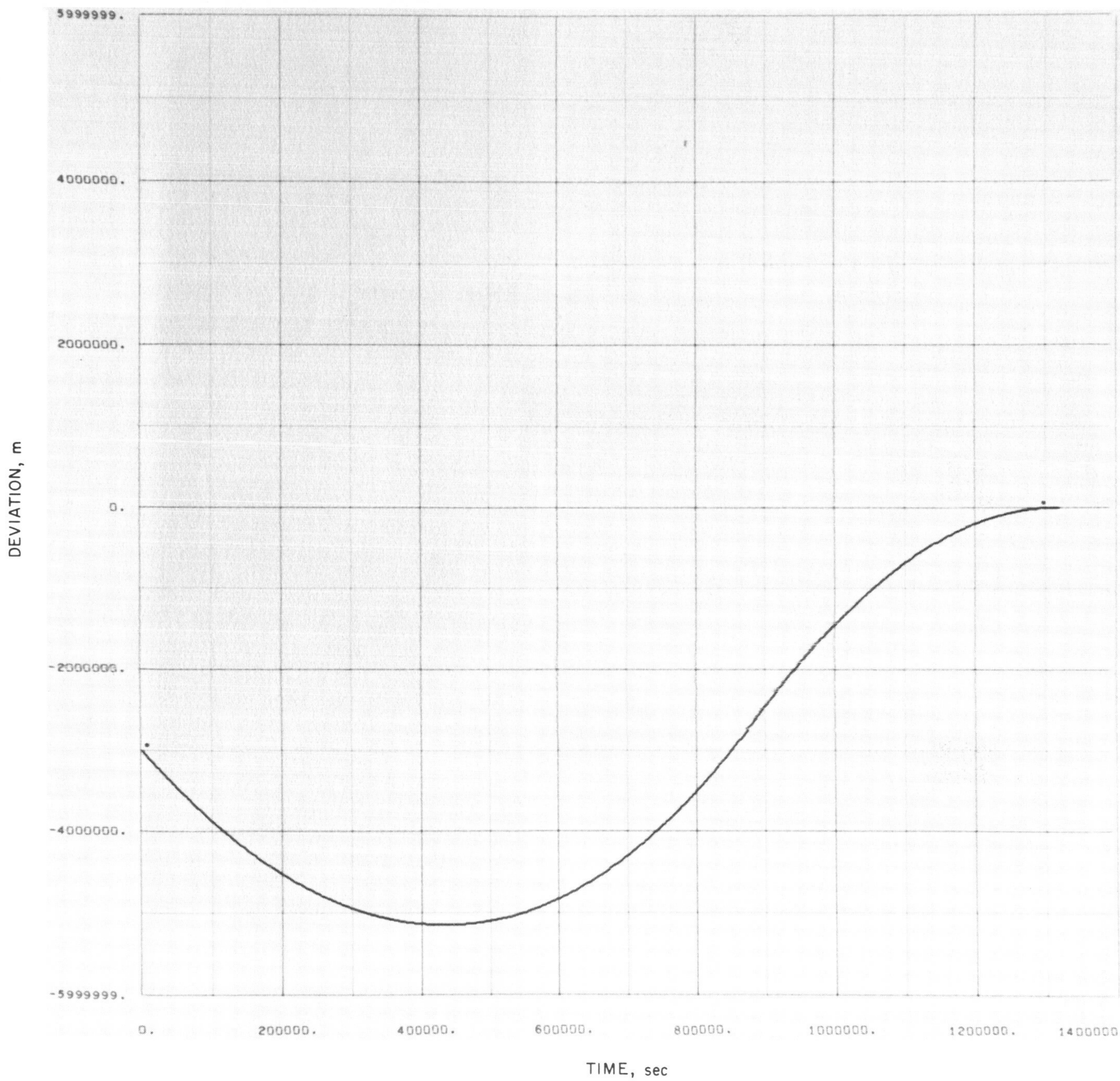


Fig. 26. The x_3 position deviation vs time for the second solution, using smaller u

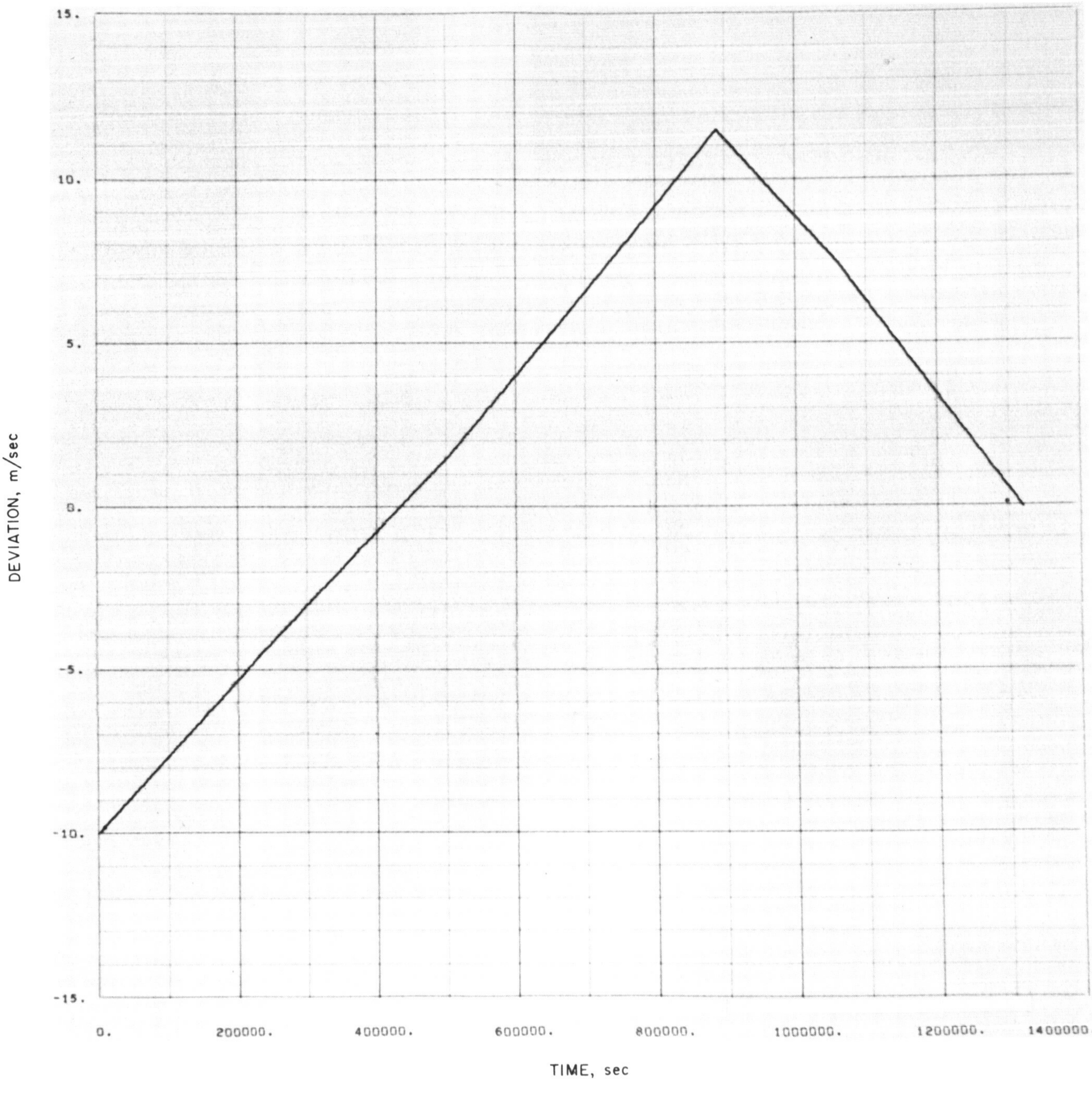


Fig. 27. The x_3 velocity deviation vs time for the second solution, using smaller u

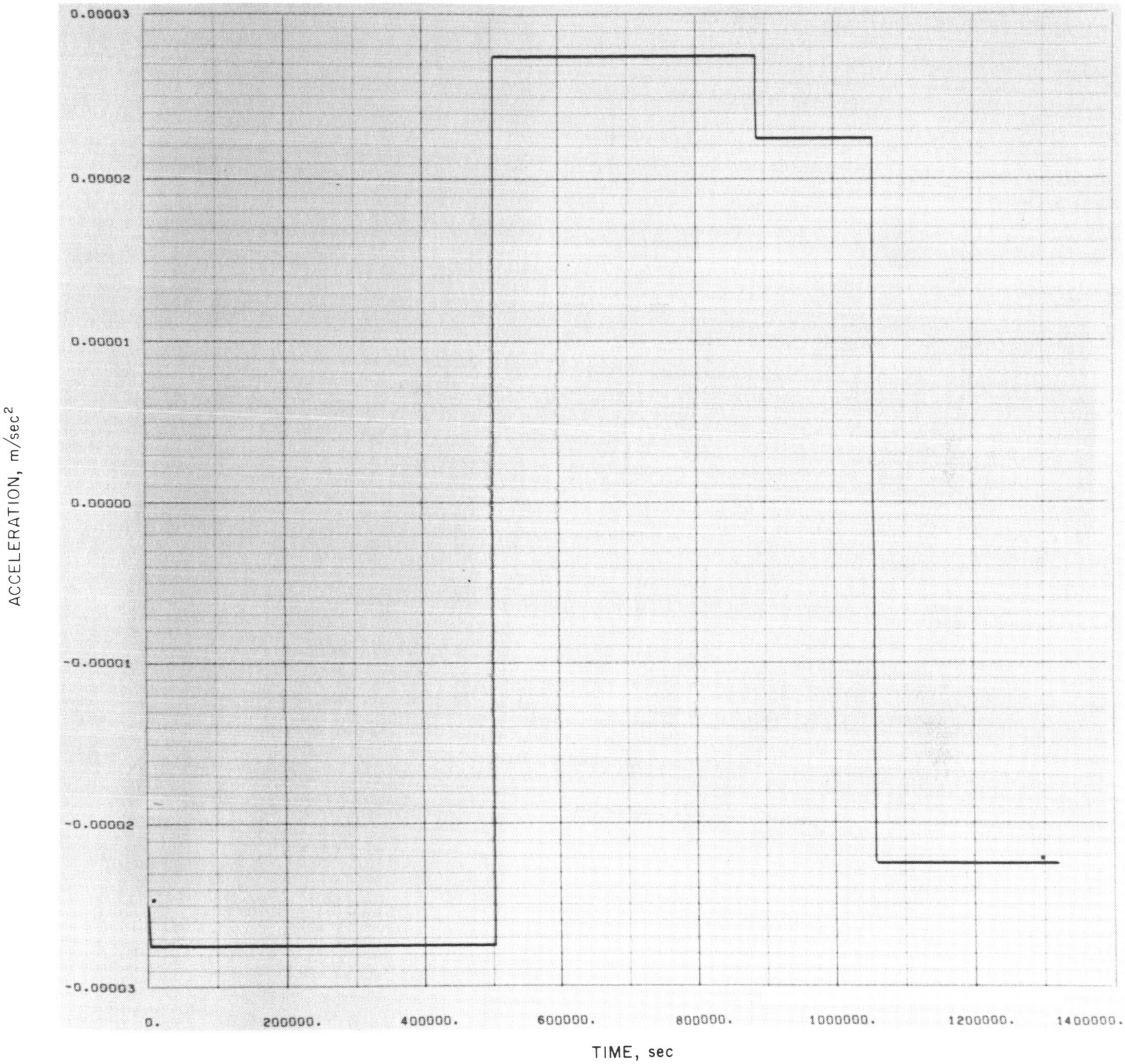


Fig. 28. The control variable u_1 vs time for the second solution, using smaller u

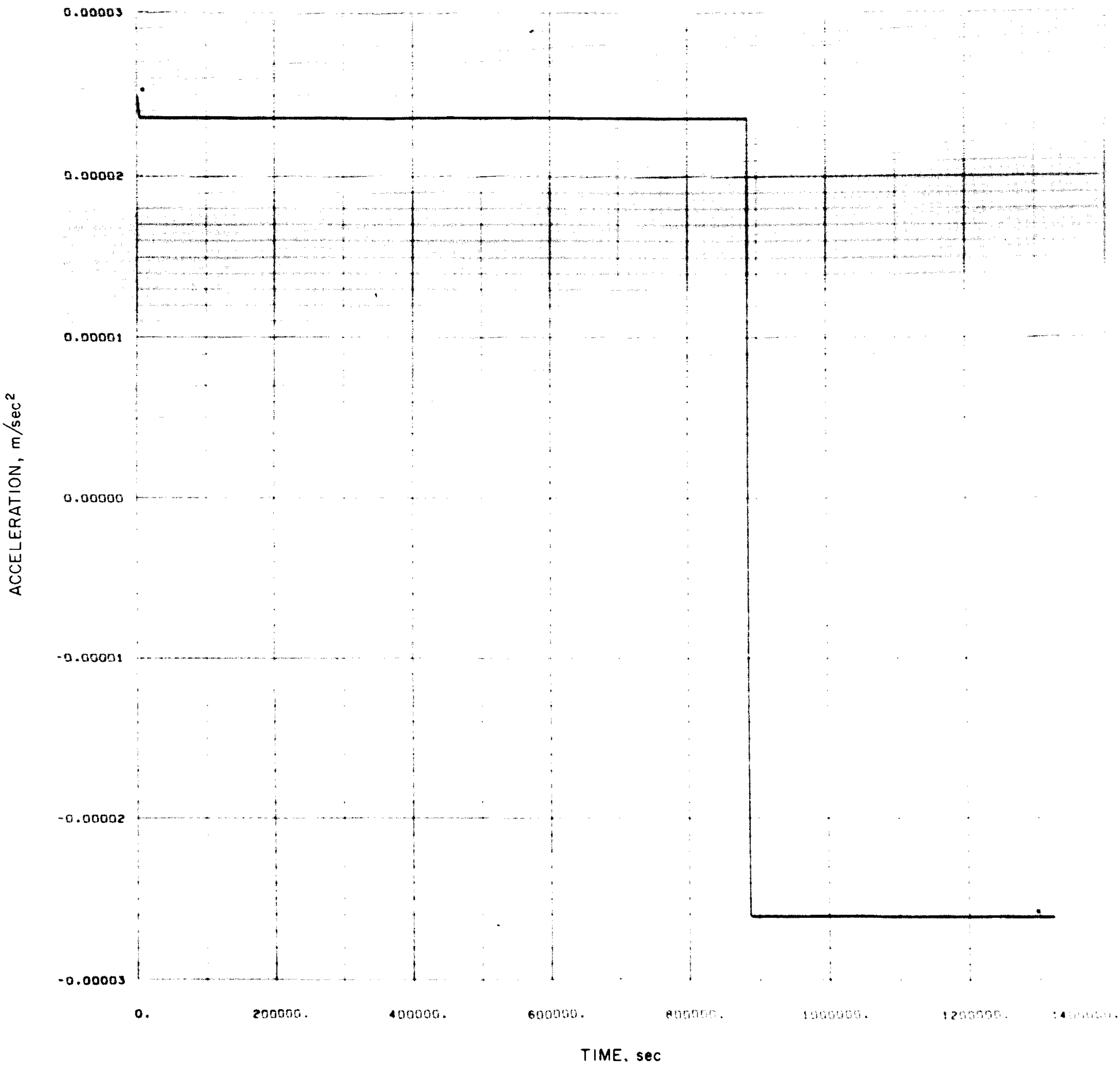


Fig. 29. The control variable u_2 vs time for the second solution, using smaller u

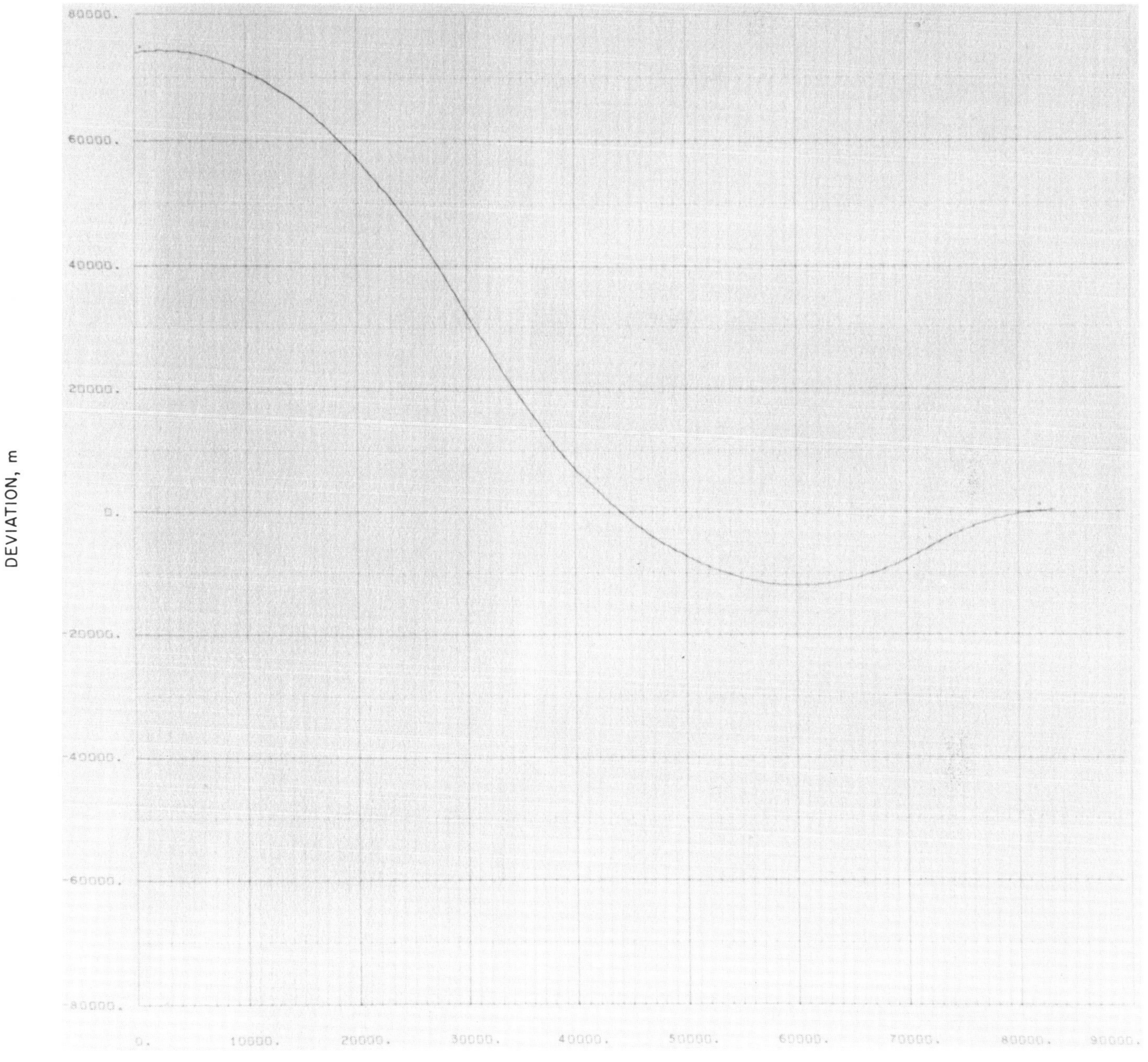


Fig. 30. The x_1 position deviation vs time for the second solution, with attitude variations

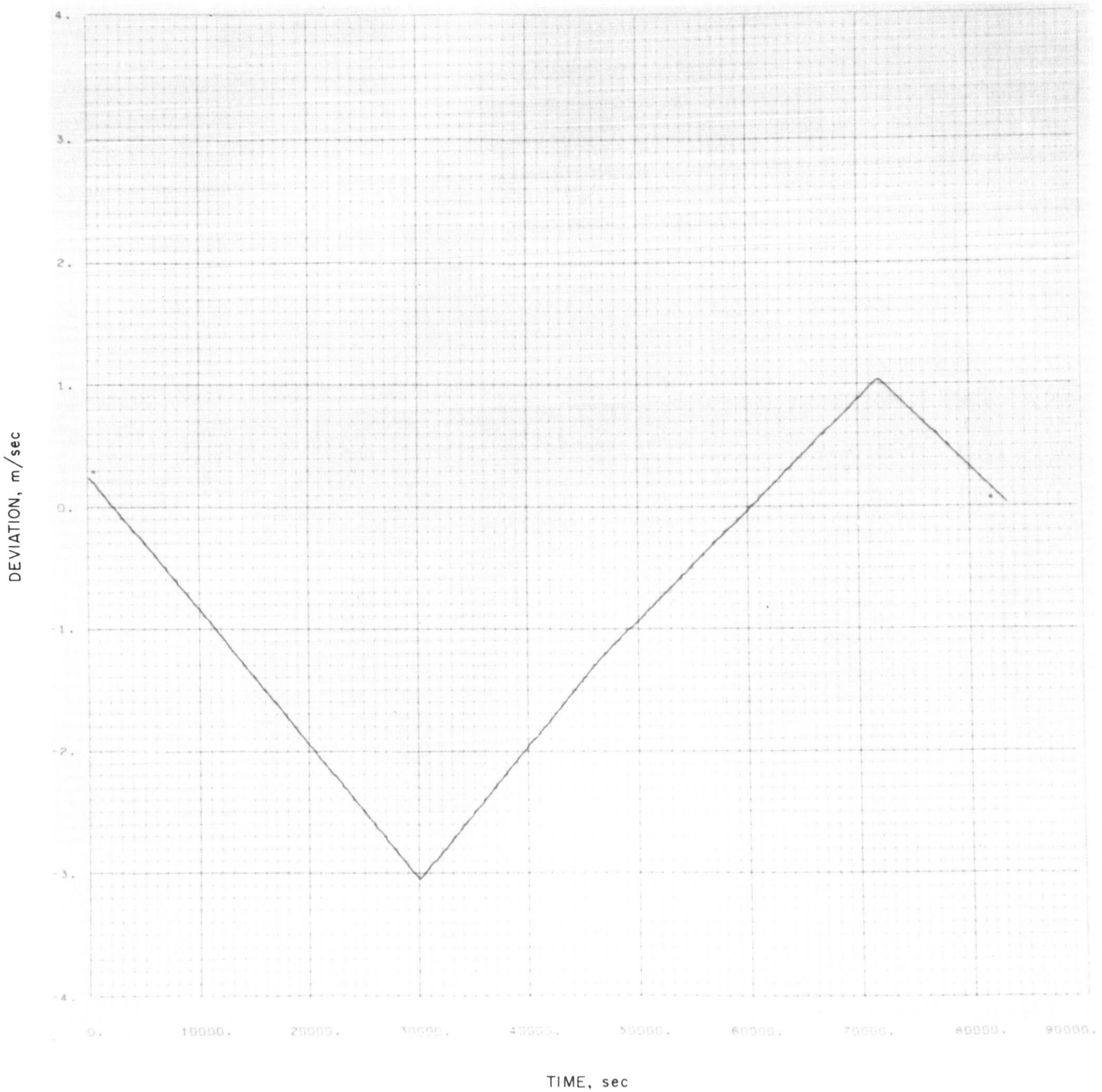


Fig. 31. The x_1 velocity deviation vs time for the second solution, with attitude variations

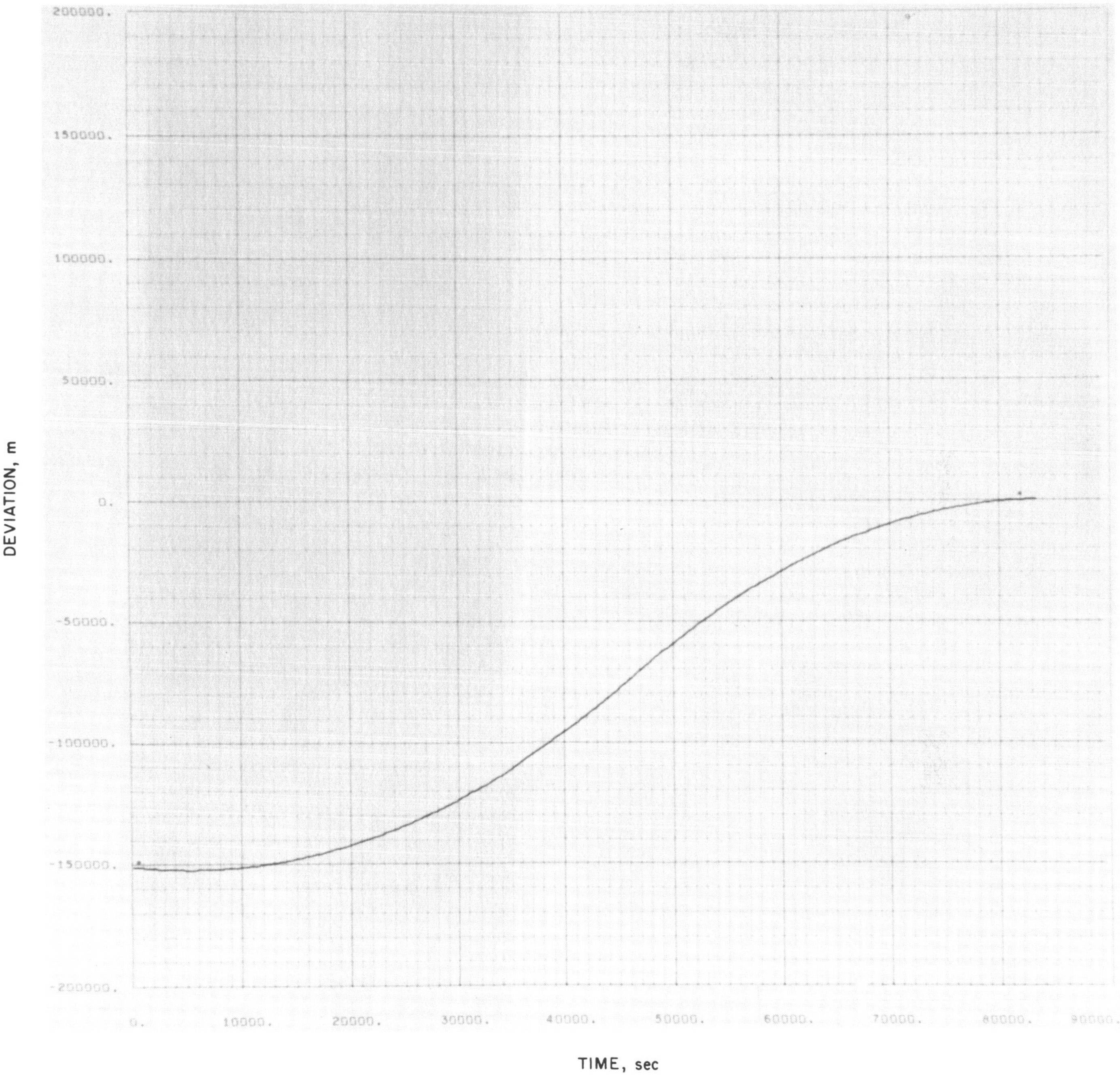


Fig. 32. The x_3 position deviation vs time for the second solution, with attitude variations

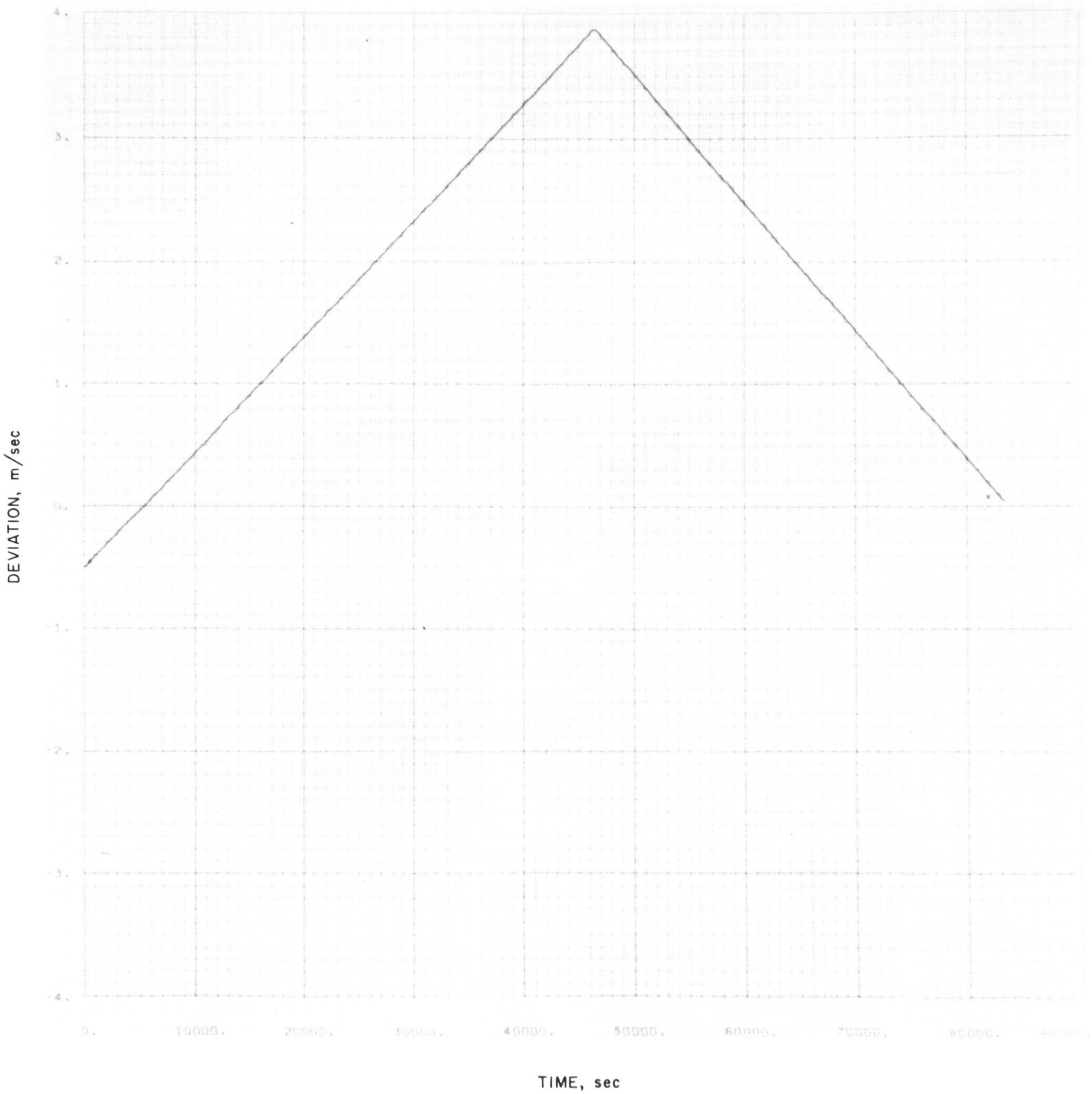


Fig. 33. The x_3 velocity deviation vs time for the second solution, with attitude variations

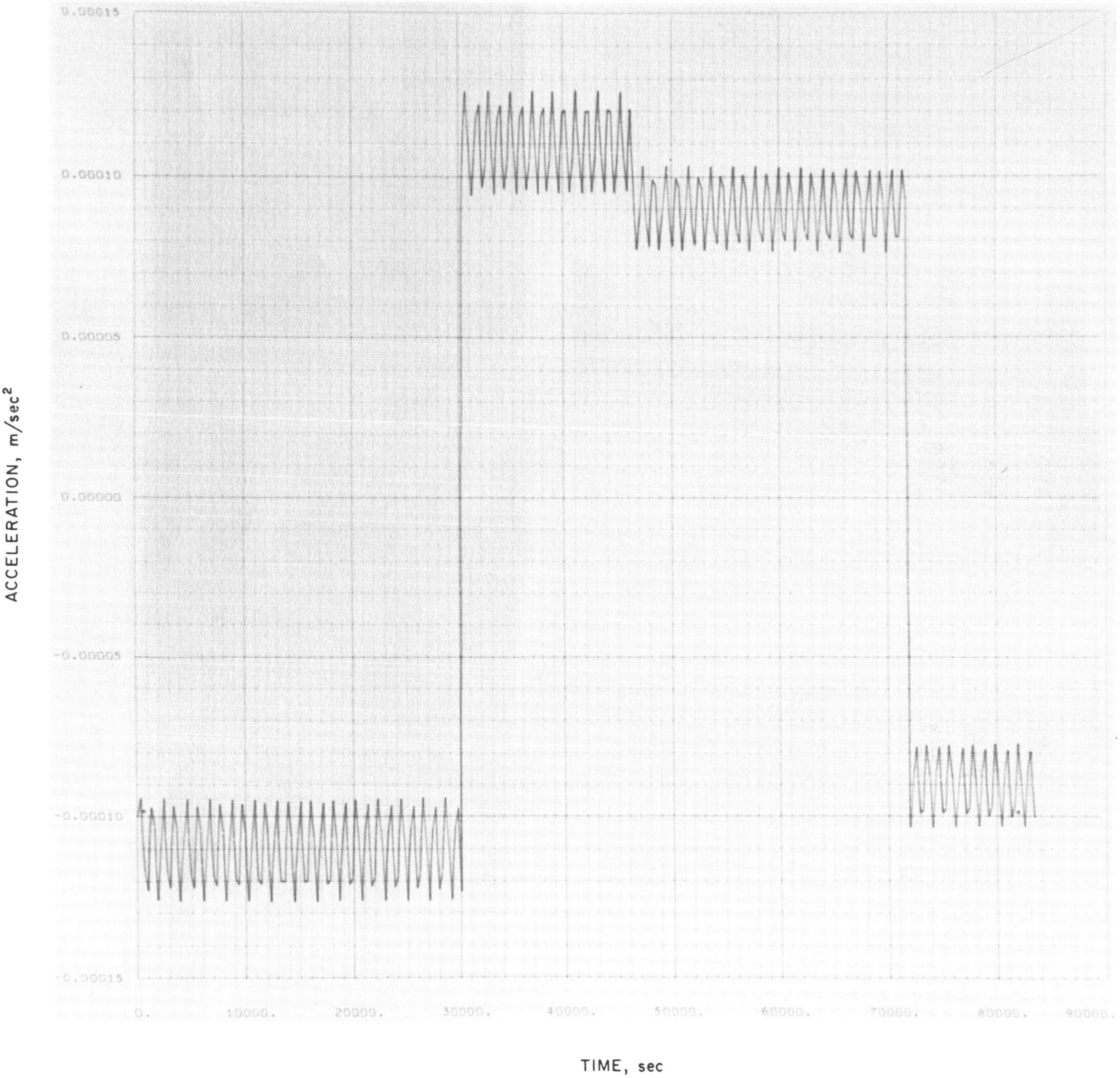


Fig. 34. The control variable u_1 vs time for the second solution, with attitude variations

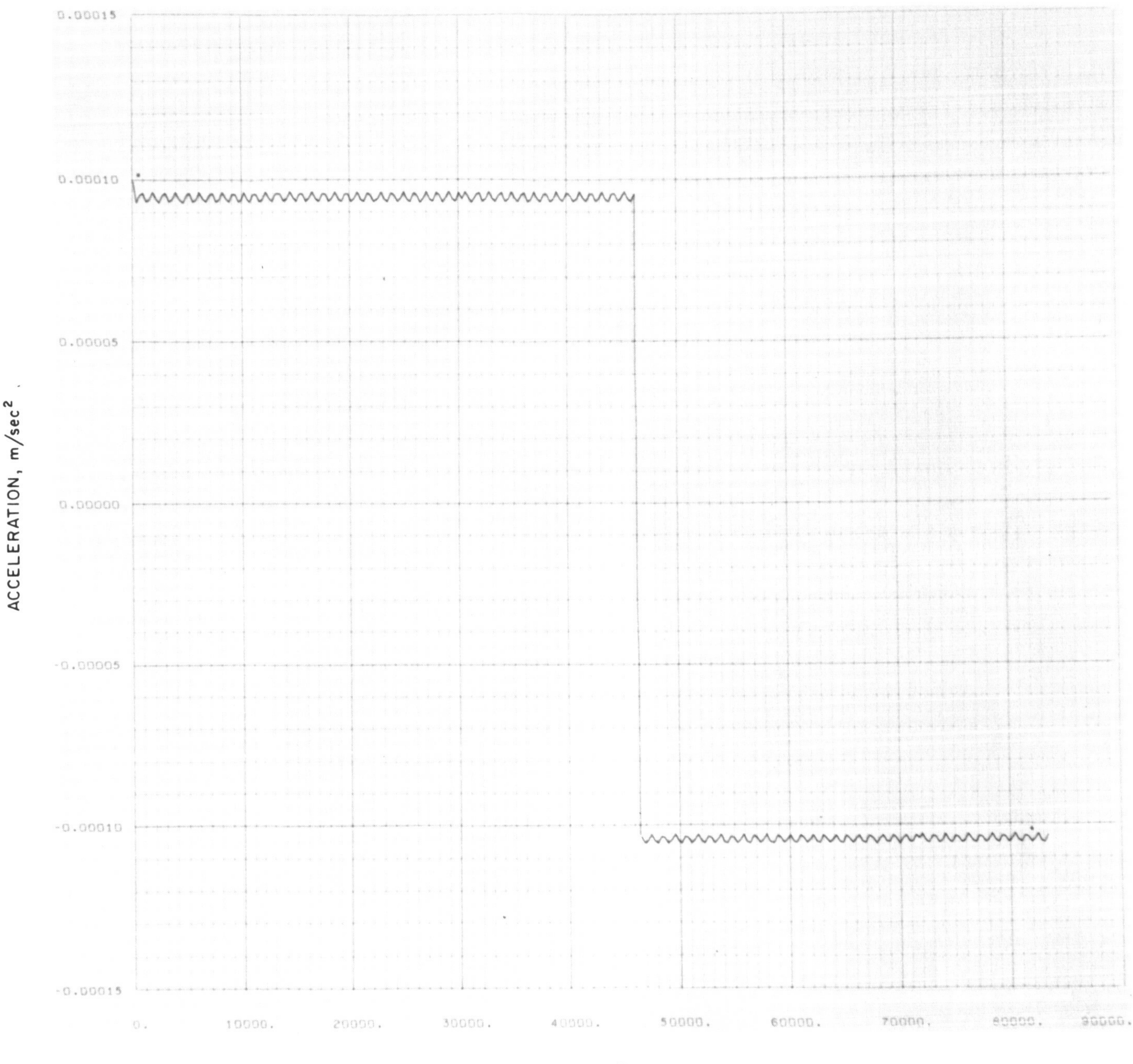


Fig. 35. The control variable u_2 vs time for the second solution, with attitude variations

VIII. Linear Regulator Formulation

It would be of interest to investigate other control configurations and performance indices. One further assumption we could make on available control would be to assume

$$|u_1| \leq k_1 \qquad |u_2| \leq k_2 \tag{31}$$

where k_1, k_2 are constants. This, of course, would be much more difficult to implement than the discrete control configuration (Fig. 2). A performance index that is often considered is that of least squares:

$$\int_0^\infty (\langle \mathrm{X}, \mathbf{Q}(t)\mathrm{X}\rangle + \langle \mathrm{u}, \mathbf{R}(t)\mathrm{u}\rangle)\, dt \tag{32}$$

where $\mathbf{Q}(t)$ and $\mathbf{R}(t)$ are weighting matrices, and $\langle \cdot \rangle$ is the inner product operator.

One control system that is suboptimal, but often yields good results, is the saturating unbounded solution (i.e., the Letov solution). That is, one solves for the optimum control functions, neglecting (31), and assumes that u_1 and u_2 can take on any values. Then condition (31) is imposed on the optimum solution. This will become clearer in the following discussion. First we shall solve the unbounded control problem.

The dynamical equations to be considered are

$$\dot{\mathrm{X}} = \mathbf{A}\mathrm{X} + \mathbf{b}(t)\mathrm{u} \tag{33}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{b}(t) = \begin{bmatrix} 0 & 0 \\ \cos\beta(t) & -\sin\beta(t) \\ 0 & 0 \\ \sin\beta(t) & \cos\beta(t) \end{bmatrix}, \mathrm{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \mathrm{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \dot{\mathrm{X}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}$$

The Hamiltonian (Ref. 2) for this problem is

$$\mathcal{H}(t, \mathrm{X}, \mathrm{u}, \lambda) = \langle \mathrm{X}, \mathbf{Q}(t)\mathrm{X}\rangle + \langle \mathrm{u}, \mathbf{R}(t)\mathrm{u}\rangle + \langle \lambda, \mathbf{A}\mathrm{X}\rangle + \langle \lambda, \mathbf{b}(t)\mathrm{u}\rangle \tag{34}$$

According to optimal control theory, the optimum $\mathrm{u}(=\mathrm{u}^*)$ is that control which minimizes the Hamiltonian at each instant of time. Hence

$$\mathrm{u}^* = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{b}^T(t)\lambda \tag{35}$$

Substituting this into Eq. (34),

$$\mathcal{H}^*(t, \mathrm{X}^*, \lambda) = \mathcal{H}(t, \mathrm{X}^*, \mathrm{u}, \lambda)\Big|_{\mathrm{u}^*} = \langle \mathrm{X}^*, \mathbf{Q}\mathrm{X}^*\rangle - \frac{1}{4}\langle \lambda, \mathbf{b}\mathbf{R}^{-1}\mathbf{b}^T\lambda\rangle + \langle \lambda, \mathbf{A}\mathrm{X}^*\rangle$$

where X^* is the optimum trajectory and $\mathcal{H}^*$ is the extremal Hamiltonian.

The equations of motion (Ref. 2) are

$$\dot{\mathbf{X}}^* = \mathcal{H}^*_\lambda = \mathbf{A}\mathbf{X}^* - \frac{1}{2}\mathbf{b}\mathbf{R}^{-1}\mathbf{b}^T\lambda \tag{36a}$$

$$\dot{\lambda} = -\mathcal{H}^*_X = -\mathbf{A}^T\lambda - 2\mathbf{Q}\mathbf{X}^* \tag{36b}$$

These equations can be solved exactly, and it is known that $\lambda(t)$ is of the form

$$\lambda(t) = \mathbf{P}(t)\mathbf{X}^* \tag{37}$$

Then

$$\dot{\lambda} = \dot{\mathbf{P}}\mathbf{X}^* + \mathbf{P}\dot{\mathbf{X}}^*$$

Using Eqs. (36) and (37), this becomes

$$\dot{\lambda}(t) = \left(\dot{\mathbf{P}} + \mathbf{P}\mathbf{A} - \frac{1}{2}\mathbf{P}\mathbf{b}\mathbf{R}^{-1}\mathbf{b}^T\mathbf{P}\right)\mathbf{X}^* \tag{38}$$

Also, from Eq. (36b) and Eq. (37),

$$\dot{\lambda}(t) = (-\mathbf{A}^T\mathbf{P} - 2\mathbf{Q})\mathbf{X}^* \tag{39}$$

Comparing Eqs. (38) and (39), we find that

$$\dot{\mathbf{P}}(t) = -(\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P}) + \frac{1}{2}\mathbf{P}\mathbf{b}\mathbf{R}^{-1}\mathbf{b}^T\mathbf{P} - 2\mathbf{Q} \tag{40}$$

Also, using Eq. (37) and Eq. (35),

$$\mathbf{u}^* = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{b}^T\mathbf{P}\mathbf{X}^* \tag{41}$$

If a problem is time-independent (i.e., if the dynamical equations and the performance index are not dependent on explicit time), then we may solve Eq. (40) with $\dot{\mathbf{P}} = 0$ (the stationary solution). This is not the case here, since the **b** matrix is time-dependent. But note that

$$\mathbf{b}\mathbf{b}^T = \begin{bmatrix} 0 & 0 \\ \cos\beta & -\sin\beta \\ 0 & 0 \\ \sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} 0 & \cos\beta & 0 & \sin\beta \\ 0 & -\sin\beta & 0 & \cos\beta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{42}$$

which is indeed independent of time. So if **R** and **Q** are time-independent, we will be able to solve Eq. (40) as a set of algebraic equations (i.e., set $\dot{\mathbf{P}} = 0$). Let

$$\mathbf{R} = C_1\mathbf{I} \qquad \mathbf{Q} = C_2\mathbf{I}$$

where C_1 and C_2 are scalars and $\mathbf{I}$ is the identity matrix. Then

$$\mathbf{R}^{-1} = \frac{1}{C_1}\mathbf{I} \tag{43}$$

Now we partition the matrices $\mathbf{A}$ and $\mathbf{bb}^T$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1 \end{bmatrix}, \mathbf{A}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{bb}^T = \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{44}$$

We can use the symmetry of these matrices to see that the $\mathbf{P}$ matrix will be of the form

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 \end{bmatrix} \tag{45}$$

where $\mathbf{P}_1$ is 2×2.

Using Eqs. (43), (44), and (45) in Eq. (40), and letting $\dot{\mathbf{P}} = 0$, we have

$$\begin{bmatrix} \mathbf{P}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 \end{bmatrix}\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1^T \end{bmatrix} + \begin{bmatrix} \mathbf{A}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1^T \end{bmatrix}\begin{bmatrix} \mathbf{P}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1^T \end{bmatrix} - \frac{1}{2C_1}\begin{bmatrix} \mathbf{P}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 \end{bmatrix}\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}\begin{bmatrix} \mathbf{P}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 \end{bmatrix} + 2C_2\mathbf{I} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{46}$$

When Eq. (46) is simplified, we find that $\mathbf{P}_1$ must satisfy

$$\mathbf{P}_1\mathbf{A}_1 + \mathbf{A}_1^T\mathbf{P}_1 - \frac{1}{2C_1}\mathbf{P}_1\mathbf{B}\mathbf{P}_1 + 2C_2\mathbf{I} = \mathbf{0} \tag{47}$$

The solution to Eq. (47) is easily obtained as

$$\mathbf{P}_1 = \begin{bmatrix} \frac{2}{C_1}(2C_1^2C_2 + (C_1C_2)^{3/2})^{1/2} & 2(C_1C_2)^{1/2} \\ 2(C_1C_2)^{1/2} & 2(2C_1(C_1C_2)^{1/2} + C_1C_2)^{1/2} \end{bmatrix} \tag{48}$$

Using Eq. (48) in Eq. (41), we finally obtain

$$\left.\begin{aligned} u_1^* = -\frac{1}{2C_1}&(\cos\beta(t)(2(C_1C_1)^{1/2}x_1 + 2(2C_1(C_1C_2)^{1/2} + C_1C_2)^{1/2}x_2) \\ &+ \sin\beta(t)(2(C_1C_2)^{1/2}x_3 + 2(2C_1(C_1C_2)^{1/2} + C_1C_2)^{1/2}x_4)) \\ u_2^* = -\frac{1}{2C_1}&(-\sin\beta(t)(2(C_1C_2)^{1/2}x_1 + 2(2C_1(C_1C_2)^{1/2} + C_1C_2)^{1/2}x_2) \\ &+ \cos\beta(t)(2(C_1C_2)^{1/2}x_3 + 2(2C_1(C_1C_2)^{1/2} + C_1C_2)^{1/2}x_4)) \end{aligned}\right\} \tag{49}$$

Recall that this is the solution for unbounded u_1 and u_2. The suboptimal (Letov) solution we will use is

$$\left.\begin{aligned} u_1 &= \begin{cases} u_1^* & |u_1^*| < k_1 \\ k_1 \operatorname{sgn}(u_1^*) & |u_1^*| \geq k_1 \end{cases} \\ u_2 &= \begin{cases} u_2^* & |u_2^*| < k_2 \\ k_2 \operatorname{sgn}(u_2^*) & |u_2^*| \geq k_2 \end{cases} \end{aligned}\right\} \qquad (50)$$

IX. Results of Saturating Linear Regulator

Initial computer runs using the control law (Eq. 50) indicate that there may be some problem with convergence of the state vector to zero (i.e., reducing deviations from nominal position and velocity to zero). It is possible that an investigation of the stability of this closed-loop system will yield enough insight into the problem to enable one to choose optimum values for C_1 and C_2, and also to find other compensatory measures that may exist. The methods of determining stability that should be used are Liapunov's direct method, describing function techniques, and the Popov criterion. These methods will have to be extended to include systems with multiple inputs, which are of interest in our problem. The computer program used to simulate the linear regulator solution appears in Appendix B.

X. Control Problem Summary and Future Work

We saw that the minimum time problem was solved to an excellent first approximation (i.e., the second solution). It would not be hard from that point to implement a program that would solve the nonlinear two-point boundary value problem for the exact switching times. It also seems it would be quite simple to extend this solution to three dimensions, where a new control (u_3) would be needed (this would correspond to roll axis deviations of the thrust vector). It has been pointed out that the effects of noise (e.g., solar pressure and thrust vector magnitude variations) have not been fully considered and that more work is needed in this area. Thrust vector orientation variations due to attitude-control deadband have been considered, but more sophisticated models for this effect should be used. Variations in the thrust vector magnitude that are due to variable vehicle mass and distance from the Sun (which affects power available for a solar-powered spacecraft) should also be considered.

The linear regulator feedback coefficients were obtained, and the Letov solution was tried. It is evident that more analysis of the stability and performance of this configuration is needed. This would, in part, involve extending the existing techniques, as has already been pointed out.

In each solution to the control problem, the knowledge of all the state variables (position and velocity vectors) is assumed. Hence the problem of state estimation, i.e., orbit determination, is of fundamental importance to these solutions. Accurate orbit determinations are, of course, already being made. At present, however, non-sequential estimation is being used. That is, each time an orbit determination is made, all the observed data up to that time are considered. This method has been satisfactory, although it is very "slow." To achieve continuous control, as we have formulated the problem, the estimator must "keep up" with the spacecraft. For these reasons it becomes clear that sequential estimation is mandatory. The sequential estimation problem was considered (see Section XI), but more work is obviously needed. Finally, owing to the problem of communication time lag between the spacecraft and the Earth, an orbit prediction will become necessary. Work in this area should also be considered.

XI. The Sequential Estimation Problem

The problem of state estimation, or orbit determination, as it is better known, is simply stated as follows: given observations (e.g., range and range-rate) on the spacecraft, determine the best guess (estimate) of the position and velocity (i.e., the state) of the spacecraft in space. A *sequential* estimator considers only the current observation and makes use of the present "best estimate"; hence a sequence of state estimates is generated. Estimator equations exist via Kalman (linear), and Sridhar and Detchmendy (nonlinear). The following discussion considers Sridhar's equations with both linear and nonlinear dynamics and observations.

The detailed derivation of the following equations is given in Reference 2. First we modify Eq. (3) as follows:

$$\dot{X}' = \mathbf{F}(u, \gamma, X') + \mathbf{k}(t, X')\,u' \tag{51}$$

where $\mathbf{k}(t, X')$ is an $n \times p$ matrix, and u' is a p-vector. The term $\mathbf{k}(t, X')\,u'$ gives Eq. (3) a new degree of freedom to account for unknown dynamics. Our observations are

$$y(t) = h(t, X') + (\text{observation error}) \tag{52}$$

where y and h are m-vectors (i.e., $m = 2$ for our problem, since we measure range and range-rate). Define the residual errors

$$e_1(t) = y - h(t, \overline{X})$$

$$e_2(t) = \dot{\overline{X}} - F(u, \gamma, \overline{X})$$

where $\overline{X}$ indicates the guessed state for $0 \leq t \leq T$. The criterion used is that of least squares. We wish to minimize

$$\int_0^T (\langle e_1(t), \mathbf{Q}e_1(t)\rangle + \langle e_2(t), \mathbf{W}e_2(t)\rangle)\,dt \tag{53}$$

where $\mathbf{Q}$ and $\mathbf{W}$ are weighting matrices. Defining $\overline{X}(T) = \hat{X}(T)$ as the best current estimate (at time T) and with

$$\mathbf{V}(t, \overline{X}) = \mathbf{k}^T(t, \overline{X})\,\mathbf{W}(t, \overline{X})\,\mathbf{k}(t, \overline{X})$$

the minimization of (53) yields the following estimation equations:

$$\left.\begin{aligned} \frac{d\hat{X}}{dT} &= \mathbf{F}(u(T), \gamma(T), \hat{X}(T)) + 2\mathbf{P}(T)\,\mathbf{H}(T, \hat{X})\,\mathbf{Q}(y - h(T, \hat{X})) \\ \frac{d\mathbf{P}}{dT} &= \mathbf{F}_{\hat{X}}(u(T), \gamma(T), \hat{X}(T))\,\mathbf{P} + \mathbf{P}\mathbf{F}_{\hat{X}}^T(u(T), \gamma(T), \hat{X}(T)) \\ &\quad + 2\mathbf{P}(\mathbf{H}\mathbf{Q}(y - h(T, \hat{X})))_{\hat{X}}\,\mathbf{P} + \frac{1}{2}\mathbf{k}(T, \hat{X})\,\mathbf{V}^{-1}(T, \hat{X})\,\mathbf{k}^T(T, \hat{X}) \end{aligned}\right\} \tag{54}$$

where the j, ith elements of $\mathbf{H}$ and $\mathbf{F}_{\hat{X}}$ are

$$\mathbf{H}(T, \hat{X})\Big|_{ji} = \left(\frac{\partial h_i}{\partial \hat{x}_j}\right) \qquad i = 1, 2, \cdots, m \qquad j = 1, 2, \cdots, n$$

$$\mathbf{F}_{\hat{X}}\Big|_{ji} = \left(\frac{\partial F_i}{\partial \hat{x}_j}\right) \qquad i = 1, 2, \cdots, n \qquad j = 1, 2, \cdots, n$$

For our problem

$$\left.\begin{aligned} h_1(X') &= \frac{x'_1 x'_2 + x'_3 x'_4}{((x'_1)^2 + (x'_3)^2)^{1/2}} && \text{(range-rate)} \\ h_2(X') &= ((x'_1)^2 + (x'_3)^2)^{1/2} && \text{(range)} \end{aligned}\right\} \tag{55}$$

In the computer program (Appendix B), both the linear and nonlinear cases were simulated. The linear dynamics are

$$\dot{X}(T) = \mathbf{F}_{X'}(u_n(T), \gamma_n(T), X'_n(T))\,X + \mathbf{F}_u\,\delta u + \mathbf{F}_\gamma\,\delta\gamma$$

where the subscript n indicates nominal values and X, as usual, is the deviation of the state from nominal. The linear observations are

$$\delta \mathrm{h}(X) = \mathbf{H}^T(T, X'_n)\,X$$

where δh indicates the deviation of observations from those that would be obtained on the nominal trajectory.

Efforts so far have failed to produce adequate convergence of Eqs. (54) to the true state (in a simulated flight with unknown initial conditions and simulated noise). All the possible ways of helping convergence have by no means been exhausted, and, owing to the importance of the problem, it would be very desirable to continue work in this area. In the effort to obtain convergence, the filter equations (Eqs. 54) were refined by including higher-order terms that were neglected in the original derivation (Ref. 2). This work appears in Appendix C.

References

1. *Solar Powered Electric Propulsion Spacecraft Study*, SSD-50094R (Final Report, JPL Contract 951144), Hughes Aircraft Company, El Segundo, California, December 1965.
2. Sridhar, R., et al., *Investigation of Optimization of Attitude Control Systems*, TR-EE65-3, NASA-CR-62195 (JPL Contract 950670), Purdue University, Lafayette, Indiana, January 1965. (Note: this is an excellent general reference on optimal control and filtering theory. It also contains a complete list of references in these areas.)
3. Melbourne, W. G., and Sauer, C. G., Jr., *Optimum Thrust Programs for Power Limited Propulsion Systems*, Technical Report No. 32-118, Jet Propulsion Laboratory, Pasadena, California, June 15, 1961.
4. *Handbook of Astronautical Engineering*, edited by H. H. Koelle, McGraw-Hill Book Co., Inc., New York, 1961.

Appendix A

Solution of the Minimum-Time Problem

Consider the dynamical equations

$$\dot{X} = \mathbf{A}X + \mathbf{b}(t)\,u \qquad |u| \leq k \tag{A-1}$$

where X is an n-vector, $\mathbf{b}$ is an $n \times p$ matrix, and k and u are p-vectors. Also

$$\left.\begin{aligned} X(t_0) &= X_0 \\ X(T) &= 0 \qquad (T \text{ is minimum}) \end{aligned}\right\} \tag{A-2}$$

The Hamiltonian for this problem is

$$\mathcal{H}(t, X, u, \lambda) = \langle \lambda, \mathbf{A}X \rangle + \langle \lambda, \mathbf{b}(t)\,u \rangle$$

The optimal u minimizes the Hamiltonian. Hence

$$u^* = \mathbf{K}(-\mathrm{sgn}(\mathbf{b}^T(t)\,\lambda)), \qquad \mathbf{K} = \begin{bmatrix} k_1 & 0 & \cdots & 0 \\ 0 & k_2 & & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & & \cdot\,\cdot \\ 0 & \cdot & \cdots & k_p \end{bmatrix}$$

where the sgn function is defined as

$$\mathrm{sgn}(y) \triangleq \begin{cases} +1 & y \geq 0 \\ -1 & y < 0 \end{cases}$$

Applied to a vector, the sgn function acts on each component. Thus

$$\mathcal{H}^*(t, X, \lambda) = \langle \lambda, \mathbf{A}X \rangle + \langle \lambda, \mathbf{b}(t)\,(-\mathbf{K}\,\mathrm{sgn}(\mathbf{b}^T(t)\,\lambda)) \rangle$$

The equations of motion are

$$\left.\begin{aligned} \dot{X} &= \mathcal{H}^*_\lambda = \mathbf{A}X - \mathbf{b}(t)\,\mathbf{K}\,\mathrm{sgn}(\mathbf{b}^T(t)\,\lambda) \\ \dot{\lambda} &= -\mathcal{H}^*_X = -\mathbf{A}^T\lambda \end{aligned}\right\} \tag{A-3}$$

The transversality condition yields

$$|\langle \lambda(T), \mathbf{b}(T)\mathbf{k}\rangle| = 1 \qquad \text{(A-4)}$$

Equations (A-3), with conditions (A-2) and (A-4), yield a two-point boundary-value problem that must be solved in order to obtain the optimal control.

Appendix B

Computer Simulation Programs

Program 1. State-estimation program, including nonlinear, two-term filter and linearized filter

Program 2. Simulation program for minimum-time solution No. 1

Program 3. Simulation program for minimum-time solution No. 2

Program 4. Computation of switching times for control system No. 2, according to Newton–Raphson technique

Program 5. Simulation program for solution to linear regulator problem

Program 1. State-estimation program, including nonlinear, two-term filter and linearized filter

```
100   FORMAT(5E18.8)
      DIMENSION TS(1019),EU(112),EL(112),VAR(113),RAN(6),DIR(113)
      COMMON VAR,RAN,DIR
      EXTERNAL DER
      VAR(1)=0.
      VAR(2)=1000.
      VAR(3)=10.3E3
      VAR(4)=1000.
      VAR(5)=17.82E3
      VAR(6)=1010.
      VAR(7)=10.31E3
      VAR(8)=990.
      VAR(9)=17.81E3
      VAR(10)=0.
      VAR(11)=0.
      VAR(12)=0.
      VAR(13)=0.
      VAR(30)=1000.
      VAR(31)=10.3E3
      VAR(32)=1000.
      VAR(33)=17.82E3
      DO 700 I=1,16
      II=13+I
      JJ=33+I
      VAR(II)=5.
700   VAR(JJ)=5.
      DO 701 I=1,4
      II=13+4*(I-1)+I
      JJ=33+4*(I-1)+I
      VAR(II)=50.
701   VAR(JJ)=50.
      DO 600 I=1,64
      II=49+I
600   VAR(II)=5.
      DO 601 I=1,4
      II=16*(I-1)
      DO 601 J=1,4
      JJ=49+II+4*(J-1)+J
601   VAR(JJ)=50.
      DO 20 I=1,1019
20    TS(I)=0.
      DIR(1)=.001
      CALL AMRKS(VAR,DIR,DER,112,0,EU,EL,.01,.001,TS,0)
25    CONTINUE
      DO 26 I=1,6
      CALL PRN(RN,0)
26    RAN(I)=RN
      IF(VAR(1)-10000.) 31,31,30
31    T=VAR(1)
28    CONTINUE
      DO 27 I=1,112
      EU(I)=.0001*ABS(VAR(I+1))+10.
27    EL(I)=.000001*ABS(VAR(I+1))+.1
      D1=VAR(10)+VAR(2)
      D2=VAR(11)+VAR(3)
      D3=VAR(12)+VAR(4)
      D4=VAR(13)+VAR(5)
      WRITE(6,100) D1,D2,D3,D4
      WRITE(6,100) VAR(30),VAR(31),VAR(32),VAR(33)
```

```
      WRITE(6,100)VAR(26),VAR(27),VAR(28),VAR(29)
      WRITE(6,100)VAR(34),VAR(35),VAR(36),VAR(37)
      WRITE(6,100)VAR(38),VAR(39),VAR(40),VAR(41)
      WRITE(6,100)VAR(42),VAR(43),VAR(44),VAR(45)
      WRITE(6,100)VAR(46),VAR(47),VAR(48),VAR(49)
      DO 900 I=1,16
      II=4*(I-1)
      I1=50+II
      I2=51+II
      I3=52+II
      I4=53+II
900   WRITE(6,100)VAR(I1),VAR(I2),VAR(I3),VAR(I4)
      CALL DER
      CALL AMRK
      IF(ABS(T+.01-VAR(1)).LT..000001) GO TO 25
      IF(T+.01-VAR(1)-DIR(1)) 50,50,51
50    DIR(1)=T+.01-VAR(1)
51    CONTINUE
      GO TO 28
30    STOP
      END
      SUBROUTINE DER
      DIMENSION H(8),V1(4,4),V2(4,4),V(4),U(2),V3(4),DZ(2),Y(4,4),
     2RX(4,4,4),         VX1(4,4,4),VX2(4,4,4),VX3(4,4,4),VX4(4,4,4),GX(4
     1,4),V21(4,4),DP(4,4),VAR(113),RAN(6),DIR(113),RI(4,4,4)
      DIMENSION H7(4,4,2),H8(4,4,4,2),H6(4,4),H9(4,4,4),H10(4,4,4),H11(4
     1,4,4),H12(4,4,4),H14(4,4,4),H15(4,4,4),H16(4,4,4),VX5(4,4,4)
      COMMON VAR,RAN,DIR
      FACT=1.
      A=(VAR(2)+1.5E11)**2+(VAR(4))**2
      B=SQRT(A)
      C=A*B
      FAC=1.
      DIR(2)=VAR(3)
      DIR(3)=(-1.325E20*(VAR(2)+1.5E11))/C-1.409E-4*VAR(4)/B
      DIR(4)=VAR(5)
      DIR(5)=(-1.325E20*VAR(4)/C+1.409E-4*(VAR(2)+1.5E11)/B)
      DIR(6)=DIR(2)+RAN(1)
      DIR(7)=DIR(3)+RAN(2)
11    DIR(8)=DIR(4)+RAN(3)
12    DIR(9)=DIR(5)+RAN(4)
13    D=SQRT(VAR(6)**2+VAR(8)**2)
      HH=VAR(2)*VAR(3)+VAR(4)*VAR(5)
14    Z=(VAR(6)*VAR(7)+VAR(8)*VAR(9))/D+RAN(5)
      ZX=D+RAN(6)
16    E=VAR(2)**2+VAR(4)**2
17    F=SQRT(E)
18    G=E*F
19    H(5)=VAR(2)/F
20    H(6)=0.
21    H(7)=VAR(4)/F
22    H(8)=0.
23    H(1)=(E*VAR(3)-HH*VAR(2))/G
24    H(2)=VAR(2)/F
25    H(3)=(E*VAR(5)-HH*VAR(4))/G
26    H(4)=VAR(4)/F
27    DZ(1)=Z-HH/F
29    DZ(2)=ZX-F
30    Q=(-1.325E20+1.4090E-4*(VAR(2)+1.5E11))/C
```

```
      1+3.*1.325E20/(((VAR(2)+1.5E11+VAR(4)*VAR(4)/(VAR(2)+1.5E11))**2)*B
      2)
31     R=(-A+VAR(4)**2)*1.409E-4/C
      1+3.*(1.325E20/C)*(VAR(2)+1.5E11)*(VAR(4)/A)
34     S=(A+(VAR(2)+1.5E11)**2)*1.409E-4/C
      1+3.*(1.325E20*(VAR(2)+1.5E11)/C)*(VAR(4)/A)
35     T=(-1.325E20-1.409E-4*(VAR(2)+1.5E11)*VAR(4))/C
      1+3.*(1.325E20/C)*((VAR(4)**2)/C)
36     U(2)=0.
37     U(1)=0.
38     DO 1 K=1,2
40     DO 1 I=1,4
41     J=9+I
42     II=4*(K-1)+I
1      U(K)=U(K)+H(II)*VAR(J)
       DO 2 I=1,4
43     V(I)=0.
44     V3(I)=0.
45     DO 2 J=1,4
46     V1(I,J)=0.
2      V2(I,J)=0.
49     DO 100 I=1,4
50     DO 100 J=1,2
51     K=4*(J-1)+I
100    V3(I)=V3(I)+H(K)*(DZ(J)-U(J))*2.*FAC
52     DO 3 I=1,4
53     II=(I-1)*4+13
54     DO 3 J=1,4
55     K=II+J
3      V(I)=V(I)+VAR(K)*V3(J)
57     DIR(10)=VAR(11)+V(1)
58     DIR(11)=Q*VAR(10)+R*VAR(12)+V(2)
59     DIR(12)=VAR(13)+V(3)
60     DIR(13)=S*VAR(10)+T*VAR(12)+V(4)
61     DO 4 I=1,4
62     II=13+(I-1)*4
63     DO 4 J=1,4
64     JJ=J+4
65     DO 4 K=1,4
66     L=II+K
       KK=K+4
4      V1(I,J)=V1(I,J)-VAR(L)*(H(J)*H(K)+H(JJ)*H(KK))
67     DO 5 I=1,4
68     DO 5 J=1,4
69     LL=13+J
70     DO 5 K=1,4
71     L=LL+4*(K-1)
5      V2(I,J)=V2(I,J)+V1(I,K)*VAR(L)*FAC
       DIR(14)=VAR(18)+VAR(15)+2.*V2(1,1)+.5*FACT
73     DIR(15)=VAR(19)+Q*VAR(14)+R*VAR(16)+2.*V2(1,2)
75     DIR(16)=VAR(20)+VAR(17)+2.*V2(1,3)
76     DIR(17)=VAR(21)+S*VAR(14)+T*VAR(16)+2.*V2(1,4)
77     DIR(18)=DIR(15)
78     DIR(19)=Q*(VAR(15)+VAR(18))+R*(VAR(23)+VAR(20))+2.*V2(2,2)+.5*FACT
79     DIR(20)=Q*VAR(16)+R*VAR(24)+VAR(21)+2.*V2(2,3)
80     DIR(21)=Q*VAR(17)+R*VAR(25)+S*VAR(18)+T*VAR(20)+2.*V2(2,4)
81     DIR(22)=DIR(16)
82     DIR(23)=DIR(20)
83     DIR(24)=VAR(28)+VAR(25)+2.*V2(3,3)+.5*FACT
```

```
 84   DIR(25)=VAR(29)+S*VAR(22)+T*VAR(24)+2.*V2(3,4)
 85   DIR(26)=DIR(17)
 86   DIR(27)=DIR(21)
 87   DIR(28)=DIR(25)
 88   DIR(29)=S*(VAR(17)+VAR(26))+T*(VAR(25)+VAR(28))+2.*V2(4,4)+.5*FACT
 89   E=VAR(30)**2+VAR(32)**2
 90   F=SQRT(E)
 91   G=F*E
 121  HH=VAR(30)*VAR(31)+VAR(32)*VAR(33)
 92   H(1)=(E*VAR(31)-HH*VAR(30))/G
 93   H(2)=VAR(30)/F
 94   H(3)=(E*VAR(33)-HH*VAR(32))/G
 95   H(4)=VAR(32)/F
 99   H(5)=H(2)
 101  H(7)=H(4)
 102  DZ(1)=2.*(Z-HH/F)
 103  DZ(2)=2.*(ZX-F)
 104  DO 6 I=1,4
 105  V(I)=0.
 106  V3(I)=0.
      DO 6 J=1,4
 107  V1(I,J)=0.
      V2(I,J)=0.
      V21(I,J)=0.
 6    DP(I,J)=0.
      DO 7 I=1,4
      DO 7 J=1,2
      K=4*(J-1)+I
 7    V3(I)=V3(I)+H(K)*DZ(J)
      DO 8 I=1,4
      II=(I-1)*4+33
      DO 8 J=1,4
      K=II+J
 8    V(I)=V(I)+VAR(K)*V3(J)*FAC
 110  Q=(VAR(30)+1.5E11)**2+VAR(32)**2
 111  R=SQRT(Q)
 112  S=R*Q
 113  DIR(30)=VAR(31)+V(1)
 114  DIR(31)=-1.325E20*(VAR(30)+1.5E11)/S-1.409E-4*VAR(32)/R+V(2)
 115  DIR(32)=VAR(33)+V(3)
 116  DIR(33)=-1.325E20*VAR(32)/S+1.409E-4*(VAR(30)+1.5E11)/R+V(4)
 117  W2=E
 118  W1=F
 119  W3=W1*W2
 1111 W5=W3*W2
 1106 W4=W2*W2
 120  W6=W3*W3
      W7=W6*W1
 122  H2=HH**2
      X1=VAR(30)
 130  X2=VAR(31)
 131  X3=VAR(32)
 135  X4=VAR(33)
 140  X12=X1**2
 141  X32=X3**2
 142  Y(1,1)=-Z*X2*X1/W3-Z*(W2*(HH+X1*X2)-X12*HH*3.)/W5+(W2*(H2+X1*2.*HH
     1*X2)-H2*X12*4.)/W6+(-W2*X2*X2+X2*HH*2.*X1)/W4+ZX*(W2-X12)/W3-1.
 150  Y(2,1)=Z*(W2-X12)/W3-(W2*(HH+X1*X2)-X12*2.*HH)/W4
 151  Y(3,1)=-Z*X4*X1/W3-Z*(W2*X2*X3-X3*HH*3.*X1)/W5+(W2*X3*2.*HH*X2-H2*
```

```
      1X3*4.*X1)/W6-(W2*X2*X4-X4*HH*2.*X1)/W4-ZX*X3*X1/W3
153   Y(4,1)=-Z*X3*X1/W3-(W2*X2*X3-X3*HH*2.*X1)/W4
      Y(1,2)=Y(2,1)
160   Y(2,2)=-X12/W2
      Y(3,2)=-Z*X1*X3/W3+2.*HH*X1*X3/W4-X1*X4/W2
180   Y(4,2)=-X3*X1/W2
181   Y(1,3)=Y(3,1)
182   Y(2,3)=Y(3,2)
183   Y(3,3)=-Z*X4*X3/W3-Z*(W2*(HH+X3*X4)-HH*X32*3.)/W5+(W2*(H2+X3*X4*2.
      1*HH)-H2*X32*4.)/W6-(W2*X4*X4-X4*HH*2.*X3)/W4+ZX*(W2-X32)/W3-1.
185   Y(4,3)=Z*(W2-X32)/W3-(W2*(HH+X3*X4)-X32*HH*2.)/W4
186   Y(1,4)=Y(4,1)
187   Y(2,4)=Y(4,2)
188   Y(3,4)=Y(4,3)
189   Y(4,4)=-X32/W2
190   DO 9 I=1,4
192   II=33+(I-1)*4
193   DO 9 J=1,4
194   DO 9 K=1,4
      L=II+K
9     V1(I,J)=V1(I,J)+VAR(L)*Y(K,J)
      DO 10 I=1,4
      DO 10 J=1,4
      LL=33+J
      DO 10 K=1,4
      L=LL+4*(K-1)
      V2(I,J)=V2(I,J)+V1(I,K)*VAR(L)*FAC
10    V21(I,J)=V21(I,J)+Y(I,K)*VAR(L)
      QQ=(-1.325E20+1.409E-4*(VAR(30)+1.5E11))/S
      2+3.*1.325E20/(((VAR(30)+1.5E11+VAR(32)*VAR(32)/(VAR(30)+1.5E11))**
      32)*R)
      RR=(-Q+VAR(32)**2)*1.409E-4/S
      1+3.*(1.325E20/S)*(VAR(30)+1.5E11)*(VAR(32)/Q)
      SS=(Q+(VAR(30)+1.5E11)**2)*1.409E-4/S
      1+3.*(1.325E20*(VAR(30)+1.5E11)/S)*(VAR(32)/Q)
      TT=(-1.325E20-1.409E-4*(VAR(30)+1.5E11)*VAR(32))/S
      1+3.*(1.325E20/S)*((VAR(32)**2)/S)
      DIR(34)=VAR(38)+VAR(35)+2.*V2(1,1)+.5*FACT
      DIR(35)=VAR(39)+QQ*VAR(34)+RR*VAR(36)+2.*V2(1,2)
      DIR(36)=VAR(40)+VAR(37)+2.*V2(1,3)
      DIR(37)=VAR(41)+SS*VAR(34)+TT*VAR(36)+2.*V2(1,4)
      DIR(38)=DIR(35)
      DIR(39)=QQ*(VAR(35)+VAR(38))+RR*(VAR(43)+VAR(40))+2.*V2(2,2)+.5
      1*FACT
      DIR(40)=QQ*VAR(36)+RR*VAR(44)+VAR(41)+2.*V2(2,3)
      DIR(41)=QQ*VAR(37)+RR*VAR(45)+SS*VAR(38)+TT*VAR(40)+2.*V2(2,4)
      DIR(42)=DIR(36)
      DIR(43)=DIR(40)
      DIR(44)=VAR(48)+VAR(45)+2.*V2(3,3)+.5*FACT
      DIR(45)=VAR(49)+SS*VAR(42)+TT*VAR(44)+2.*V2(3,4)
      DIR(46)=DIR(37)
      DIR(47)=DIR(41)
      DIR(48)=DIR(45)
      DIR(49)=SS*(VAR(37)+VAR(46))+TT*(VAR(45)+VAR(48))+2.*V2(4,4)+.5
      1*FACT
      DO 500 I=1,4
      DO 500 J=1,4
      GX(I,J)=0.
      DO 500 K=1,4
```

```
      VX1(I,J,K)=0.
      VX2(I,J,K)=0.
      VX3(I,J,K)=0.
      H9(I,J,K)=0.
      H10(I,J,K)=0.
      H11(I,J,K)=0.
      H12(I,J,K)=0.
      H15(I,J,K)=0.
      H16(I,J,K)=0.
      VX5(I,J,K)=0.
500   VX4(I,J,K)=0.
      GX(1,2)=1.
      GX(2,1)=QQ
      GX(2,3)=RR
      GX(3,4)=1.
      GX(4,1)=SS
      GX(4,3)=TT
      DO 505 I=1,4
      II=16*(I-1)
      DO 505 J=1,4
      JJ=4*(J-1)
      DO 505 K=1,4
      KK=49+II+JJ+K
505   RX(I,J,K)=VAR(KK)
      DO 501 I=1,4
      DO 501 J=1,4
      DO 501 K=1,4
      DO 501 L=1,4
      RI(I,J,L)=.5*(RX(I,J,L)+RX(I,L,J))
      VX1(I,J,K)=RI(I,J,L)*GX(K,L)*2.+VX1(I,J,K)
      VX2(I,J,K)=VX2(I,J,K)+V1(I,L)*RX(L,J,K)*2.
      VX3(I,J,K)=VX3(I,J,K)+RI(I,J,L)*V21(L,K)*4.
501   VX4(I,J,K)=VX4(I,J,K)+GX(I,L)*RX(L,J,K)
      H7(1,1,1)=-X2*X1/W3-(HH+X1*X2)/W3+3.*HH*X12/W5
      H7(1,2,1)=1./W1-X12/W3
      H7(1,3,1)=-X4*X1/W3-X3*X2/W3+3.*HH*X3*X1/W5
      H7(1,4,1)=-X3*X1/W3
      H7(2,1,1)=H7(1,2,1)
      H7(2,2,1)=0.
      H7(2,3,1)=-X1*X3/W3
      H7(2,4,1)=0.
      H7(3,1,1)=H7(1,3,1)
      H7(4,1,1)=H7(1,4,1)
      H7(4,2,1)=0.
      H7(4,3,1)=H7(3,4,1)
      H7(4,4,1)=0.
      H8(1,1,1,1)=-X2/W3+3.*X2*X12/W5-2.*X2/W3+(6.*HH*X1
     1+3.*X12*X2)/W5-15.*HH*X12*X1/W7+3.*X1*(HH+X1*X2)/W5
      H8(1,1,2,1)=-X1/W3-2.*X1/W3+3.*X12*X1/W5
      H8(1,1,3,1)=-X4/W3+3.*X4*X12/W5+X3*X2*3.*X1/W5+(3.*HH*X3+3.*X1*X3*
     1X2)/W5-15.*HH*X3*X12/W7
      H8(1,1,4,1)=-X3/W3+3.*X3*X12/W5
      H7(3,2,1)=H7(2,3,1)
      H7(3,3,1)=-X4*X3/W3-(HH+X3*X4)/W3+3.*HH*X32/W5
      H7(3,4,1)=1./W1-X32/W3
      H8(2,1,1,1)=H8(1,1,2,1)
      H8(2,1,2,1)=0.
      H8(2,1,3,1)=-X3/W3+3.*X12*X3/W5
      H8(2,1,4,1)=0.
```

```
      H8(3,1,1,1)=H8(1,1,3,1)
      H8(3,1,2,1)=H8(2,1,3,1)
      H8(3,1,3,1)=3.*X4*X3*X1/W5-X2/W3+3.*(HH+X3*X4)*X1/W5+3.*X32*X2/W5-
     115.*HH*X32*X1/W7
      H8(3,1,4,1)=-X1/W3+3.*X32*X1/W5
      H8(4,1,1,1)=-X3/W3+3.*X12*X3/W5
      H8(4,1,2,1)=0.
      H8(4,1,3,1)=H8(3,1,4,1)
      H8(4,1,4,1)=0.
      H8(2,2,1,1)=0.
      H8(2,2,2,1)=0.
      H8(2,2,3,1)=0.
      H8(2,2,4,1)=0.
      H8(3,2,1,1)=H8(2,1,3,1)
      H8(3,2,2,1)=0.
      H8(3,2,3,1)=-X1/W3+3.*X1*X32/W5
      H8(3,2,4,1)=0.
      H8(4,2,1,1)=0.
      H8(4,2,2,1)=0.
      H8(4,2,3,1)=0.
      H8(4,2,4,1)=0.
      H8(3,3,1,1)=-X2/W3+3.*X2*X32/W5+3.*X1*X4*X3/W5+(3.*HH*X1+3.*X3*X1*
     1X4)/W5-15.*HH*X1*X32/W7
      H8(3,3,2,1)=-X1/W3+3.*X1*X32/W5
      H8(3,3,3,1)=-X4/W3+3.*X4*X32/W5-2.*X4/W3+3.*(HH+X3*X4)*X3/W5+(H
     1H*6.*X3+3.*X32*X4)/W5-15.*HH*X32*X3/W7
      H8(3,3,4,1)=-X3/W3-2.*X3/W3+3.*X32*X3/W5
      H8(4,4,1,1)=0.
      H8(4,4,2,1)=0.
      H8(4,4,3,1)=0.
      H8(4,4,4,1)=0.
      DO 670 I=1,4
      H7(I,1,2)=H7(I,2,1)
      H7(I,2,2)=0.
      H7(I,3,2)=H7(I,4,1)
  670 H7(I,4,2)=0.
      DO 671 I=1,4
      H8(2,1,I,1)=H8(1,2,I,1)
      H8(3,1,I,1)=H8(1,3,I,1)
      H8(3,2,I,1)=H8(2,3,I,1)
      H8(4,1,I,1)=H8(1,4,I,1)
      H8(4,2,I,1)=H8(2,4,I,1)
  671 H8(4,3,I,1)=H8(3,4,I,1)
      DO 672 I=1,4
      DO 672 J=1,4
      H8(I,J,1,2)=H8(I,J,2,1)
      H8(I,J,2,2)=0.
      H8(I,J,3,2)=H8(I,J,4,1)
  672 H8(I,J,4,2)=0.
      DO 673 I=1,4
      DO 673 J=1,2
      II=4*(J-1)+I
  673 H6(I,J)=H(II)
      DO 674 I=1,4
      DO 674 J=1,4
      DO 674 K=1,4
      DO 674 L=1,2
      H9(I,J,K)=H9(I,J,K)-H7(I,J,L)*H6(K,L)
      H10(I,J,K)=-H6(J,L)*H7(I,K,L)+H10(I,J,K)
```

```
      H11(I,J,K)=H11(I,J,K)+H8(K,I,J,L)*DZ(L)
  674 H12(I,J,K)=H12(I,J,K)-H7(K,J,L)*H6(I,L)
      DO 675 I=1,4
      DO 675 J=1,4
      DO 675 K=1,4
  675 H14(I,J,K)=H9(I,J,K)+H10(I,J,K)+H11(I,J,K)/2.+H12(I,J,K)
      DO 676 I=1,4
      DO 676 J=1,4
      DO 676 K=1,4
      KK=33+K
      DO 676 L=1,4
      LL=KK+(L-1)*4
  676 H15(I,J,K)=H15(I,J,K)+2.*VAR(LL)*H14(I,L,K)
      DO 677 I=1,4
      DO 677 J=1,4
      DO 677 K=1,4
      KK=33+K
      DO 677 L=1,4
      LL=KK+4*(L-1)
  677 H16(I,J,K)=H16(I,J,K)+H15(I,J,L)*VAR(LL)
      DO 678 I=1,4
      II=33+(I-1)*4
      DO 678 J=1,4
      DO 678 K=1,4
      DO 678 L=1,4
      LL=II+L
  678 VX5(I,J,K)=VX5(I,J,K)+VAR(LL)*H16(L,J,K)
      DO 502 I=1,4
      II=16*(I-1)
      DO 502 J=1,4
      JJ=4*(J-1)
      DO 502 K=1,4
      KK=49+II+JJ+K
  502 DIR(KK)=VX1(I,J,K)+VX2(I,J,K)+VX3(I,J,K)+VX4(I,J,K)+VX5(I,J,K)
      DO 503 I=1,4
      DO 503 J=1,4
      DO 503 K=1,4
  503 DP(I,J)=DP(I,J)+V3(K)*RI(I,J,K)
      DO 504 I=1,4
      II=4*(I-1)
      DO 504 J=1,4
      K=33+II+J
  504 DIR(K)=DIR(K)+DP(I,J)
      RETURN
      END
```

Program 2. Simulation program for minimum-time solution No. 1

```
      DIMENSION VAR(9),DIR(9),TS(83),EU(8),EL(8),Y(4),RAN(2)
      DIMENSION XY1(7,1500),XY2(2,1500),YNAME2(10),YNAME3(10),XNAME(14),
     1TITLE1(14),TITLE2(14),TITLE3(14),TITLE4(14),YNAME1(10),TITLE5(14),
     2TITLE6(14)
      DOUBLE PRECISION VAR,DIR,U1,U2,DIRT1,TS,DIRT2,DIRT3,DIRT4,BDOT,
     1U10,U20,RAN
      DOUBLE PRECISION U2A,U2B,U2C,UX
      DOUBLE PRECISION DU1,DU2
      COMMON VAR,DIR,U1,U2,CO,SI
      EXTERNAL DER
      DATA YNAME1(1)/60HMETERS
     1          /
      DATA YNAME3(1)/60HMETERS PER SECOND SQUARED
     1          /
      DATA SYMBOL/1H./
      DATA YNAME2(1)/60HMETERS PER SECOND
     1          /
      DATA TITLE1(1)/84H
     1                                X POSITION/
      DATA TITLE2(1)/84H
     1                                X VELOCITY/
      DATA TITLE3(1)/84H
     1                                Y POSITION/
      DATA TITLE4(1)/84H
     1                                Y VELOCITY/
      DATA TITLE5(1)/84H
     1                                          U1/
      DATA TITLE6(1)/84H
     1                                          U2/
      DATA XNAME(1)/84H
     1                                  SECONDS/
      UX=DSQRT(.99D0)
      U2A=1.1D-3*UX-1.D-3
      U2B=.9D-3*UX-1.D-3
      U2C=1.D-3*UX-1.D-3
      BDOT=.11678565E-6
      II=1
      U20=1.D-4
      U2=1.D-4
      U10=-1.D-4
      U1=-1.D-4
      RAN(1)=0.D0
      RAN(2)=0.D0
      FLAG3=0.
      FLAG1=0.
      FLAG2=0.
      FLAG4=0.
      CO=1.D0
      SI=0.D0
      VAR(1)=0.D0
      DELT=300.
      VAR(2)=75000.
      VAR(3)=10300.25D0
      VAR(4)=-150000.D0
      VAR(5)=17819.5D0
      B=2.*3.141592654
      VAR(6)=100.D1
      VAR(7)=1030.D1
      VAR(8)=100.D1
```

```
      VAR(9)=1782.D1
      DIRT1=VAR(2)-VAR(6)
      DIRT2=VAR(3)-VAR(7)
      DIRT3=VAR(4)-VAR(8)
      DIRT4=VAR(5)-VAR(9)
      DO 1 I=1,83
1     TS(I)=0.
      DIR(1)=.001
      CALL AMRKS(VAR,DIR,DER,8,0,EU,EL,100.,.001,TS,1)
25    CONTINUE
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      XY1(1,II)=VAR(2)-VAR(6)
      XY1(2,II)=VAR(3)-VAR(7)
      XY1(3,II)=VAR(4)-VAR(8)
      XY1(4,II)=VAR(5)-VAR(9)
      XY1(5,II)=U1
      XY1(6,II)=U2
      XY1(7,II)=VAR(1)
      II=II+1
31    T=VAR(1)
28    CONTINUE
      AX=6.E-3*VAR(1)
      AA=AMOD(AX,B)
      BB=.015*SIN(AA)
      DU1=(U2+1.D-3)*BB
      DU2=U1*BB
      Y(1)=VAR(2)-VAR(6)
      Y(2)=VAR(3)-VAR(7)
      Y(3)=VAR(4)-VAR(8)
      Y(4)=VAR(5)-VAR(9)
      IF(U10.EQ.0..AND.U20.EQ.0.) GO TO 300
      IF(U10.GT.0..AND.U20.GT.0.)GO TO 2000
      IF(U10.GT.0..AND.U20.LT.0.)GO TO 2002
      IF(U10.LT.0..AND.U20.GT.0.)GO TO 2006
      IF(U10.LT.0..AND.U20.LT.0.)GO TO 2008
      IF(U10.LT.0..AND.U20.EQ.0.)GO TO 2007
      IF(U10.EQ.0..AND.U20.LT.0.)GO TO 2005
      IF(U10.EQ.0..AND.U20.EQ.0.)GO TO 2004
      IF(U10.EQ.0..AND.U20.GT.0.)GO TO 2003
      IF(U10.GT.0..AND.U20.EQ.0.)GO TO 2001
2009  CONTINUE
872   CONTINUE
      DO 20 I=1,8
      X=VAR(I+1)
      Z=ABS(X)
      EU(I)=1.E-10*Z+1.E-12
20    EL(I)=1.E-12*Z+1.E-14
      CALL DER
      CALL AMRK
      IF(ABS(T+DELT-VAR(1)).LT..000001)GO TO 25
      IF(T+DELT-VAR(1)-DIR(1))60,60,61
60    DIR(1)=T+DELT-VAR(1)
61    CONTINUE
      DIRT1=(VAR(2)-VAR(6))*CO+(VAR(4)-VAR(8))*SI
      DIRT2=(VAR(3)-VAR(7))*CO+(VAR(5)-VAR(9))*SI-(VAR(2)-VAR(6))*SI*B
     1DOT+(VAR(4)-VAR(8))*CO*BDOT
      DIRT3=-(VAR(2)-VAR(6))*SI+(VAR(4)-VAR(8))*CO
      DIRT4=-(VAR(3)-VAR(7))*SI+(VAR(5)-VAR(9))*CO-(VAR(2)-VAR(6))*CO*
     1BDOT-(VAR(4)-VAR(8))*SI*BDOT
```

```
      IF(1.-FLAG1)21,21,40
21    IF(DIRT2*U10.GT.0.) U10=0.D0
40    IF(DIRT2*U10.GT.0..AND.2.*U1*DIRT1+DIRT2**2.GE.0.)GO TO 200
50    IF(1.-FLAG2)51,51,23
51    IF(DIRT4*U20.GT.0.) U20=0.D0
23    IF(DIRT4*U20.GT.0..AND.2.*U2*DIRT3+DIRT4**2.GE.0.)GO TO 201
22    GO TO 28
871   FLAG1=0.
      FLAG2=0.
      U10=-1.D-4
      U20=-1.D-4
      FLAG4=1.
      GO TO 872
200   FLAG1=FLAG1+1.
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      U10=-U10
      GO TO 50
201   FLAG2=FLAG2+1.
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      U20=-U20
      GO TO 22
2000  U1=1.1D-4+DU1
      U2=U2A+DU2
      GO TO 2009
2001  U1=1.D-4+DU1
      U2=U2C+DU2
      GO TO 2009
2002  U1=.9D-4+DU1
      U2=U2B+DU2
      GO TO 2009
2003  U1=0.+DU1
      U2=1.D-4+DU2
      GO TO 2009
2004  U1=0.+DU1
      U2=0.+DU2
      GO TO 2009
2005  U1=0.+DU1
      U2=-1.D-4+DU2
      GO TO 2009
2006  U1=-1.1D-4+DU1
      U2=U2A+DU2
      GO TO 2009
2007  U1=-1.D-4+DU1
      U2=U2C+DU2
      GO TO 2009
2008  U1=-.9D-4+DU1
      U2=U2B+DU2
      GO TO 2009
300   WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      WRITE(6,1000)VAR(1)
      IF(FLAG4.EQ.0.)GO TO 871
1000  FORMAT(4E18.8)
      II=II-1
      DO 860 I=1,II
      XY2(1,I)=XY1(1,I)
860   XY2(2,I)=XY1(7,I)
```

```
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE1,XNAME,YNAME1,3)
      DO 861 I=1,II
861   XY2(1,I)=XY1(2,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE2,XNAME,YNAME2,3)
      DO 862 I=1,II
862   XY2(1,I)=XY1(3,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE3,XNAME,YNAME1,3)
      DO 863 I=1,II
863   XY2(1,I)=XY1(4,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE4,XNAME,YNAME2,3)
      DO 864 I=1,II
864   XY2(1,I)=XY1(5,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE5,XNAME,YNAME3,3)
      DO 865 I=1,II
865   XY2(1,I)=XY1(6,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE6,XNAME,YNAME3,3)
30    STOP
      END
      SUBROUTINE DER
      DIMENSION VAR(9),DIR(9)
      COMMON VAR,DIR,U1,U2,CO,SI
      DOUBLE PRECISION  VAR,DIR,A,B,C,U1,U2,CO,SI,BDOT
      A=(VAR(2)+1.5D11)**2+(VAR(4))**2
      B=DSQRT(A)
      C=A*B
      SI=VAR(4)/B
      CO=(VAR(2)+1.5D11)/B
      DIR(2)=VAR(3)
      DIR(3)=(-1.325D20*(VAR(2)+1.5D11))/C-1.D-3   *SI+U1*CO-U2*SI
      DIR(4)=VAR(5)
      DIR(5)= -1.325D20*VAR(4)/C+1.D-3   *CO+U1*SI+U2*CO
      A=(VAR(6)+1.5D11)**2+(VAR(8))**2
      B=DSQRT(A)
      C=A*B
      SI=VAR(8)/B
      CO=(VAR(6)+1.5D11)/B
      DIR(6)=VAR(7)
      DIR(7)=(-1.325D20*(VAR(2)+1.5D11))/C-1.D-3   *VAR(4)/B
      DIR(8)=VAR(9)
      DIR(9)= -1.325D20*VAR(8)/C+1.D-3   *(VAR(2)+1.5D11)/B
      RETURN
      END
```

Program 3. Simulation program for minimum-time solution No. 2

```
      DIMENSION VAR(9),DIR(9),TS(83),EU(8),EL(8),Y(4),RAN(2)
      DIMENSION XY1(7,1500),XY2(2,1500),YNAME2(10),YNAME3(10),XNAME(14),
     1TITLE1(14),TITLE2(14),TITLE3(14),TITLE4(14),YNAME1(10),TITLE5(14),
     2TITLE6(14)
      DOUBLE PRECISION VAR,DIR,U1,U2,DIRT1,TS,DIRT2,DIRT3,DIRT4,BDOT,
     1U10,U20,RAN
      DOUBLE PRECISION U2A,U2B,U2C,UX
      DOUBLE PRECISION DU1,DU2
      COMMON VAR,DIR,U1,U2,CO,SI
      EXTERNAL DER
      DATA YNAME1(1)/60HMETERS
     1          /
      DATA YNAME3(1)/60HMETERS PER SECOND SQUARED
     1          /
      DATA SYMBOL/1H./
      DATA YNAME2(1)/60HMETERS PER SECOND
     1          /
      DATA TITLE1(1)/84H
     1                                  X POSITION/
      DATA TITLE2(1)/84H
     1                                  X VELOCITY/
      DATA TITLE3(1)/84H
     1                                  Y POSITION/
      DATA TITLE4(1)/84H
     1                                  Y VELOCITY/
      DATA TITLE5(1)/84H
     1                                          U1/
      DATA TITLE6(1)/84H
     1                                          U2/
      DATA XNAME(1)/84H
     1                                     SECONDS/
      UX=DSQRT(.99D0)
      U2A=1.1D-3*UX-1.D-3
      U2B=.9D-3*UX-1.D-3
      U2C=1.D-3*UX-1.D-3
      DELT=300.
      B=2.*3.141592654
      U10=-1.D-4
      U20=1.D-4
      U1=-1.D-4
      U2=1.D-4
      BDOT=.11678565E-6
      RAN(1)=0.D0
      RAN(2)=0.D0
      FLAG3=0.
      FLAG4=0.
      FLAG5=0.
      FLAG6=0.
      TS1=30004.524
      T1=71848.025
      TS2=46393.812
      T=83634.808
      II=1
      VAR(1)=0.D0
      VAR(2)=75000.
      VAR(3)=10300.25D0
      VAR(4)=-150000.D0
      VAR(5)=17819.5D0
      VAR(6)=100.D1
```

```
      VAR(7)=1030.D1
      VAR(8)=100.D1
      VAR(9)=1782.D1
      DO 1 I=1,83
1     TS(I)=0.
      DIR(1)=.001
      CALL AMRKS(VAR,DIR,DER,8,0,EU,EL,100.,.001,TS,1)
25    CONTINUE
      A=VAR(1)
      XY1(1,II)=VAR(2)-VAR(6)
      XY1(2,II)=VAR(3)-VAR(7)
      XY1(3,II)=VAR(4)-VAR(8)
      XY1(4,II)=VAR(5)-VAR(9)
      XY1(5,II)=U1
      XY1(6,II)=U2
      XY1(7,II)=VAR(1)
      II=II+1
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      FLAG7=1.
28    CONTINUE
      AX=6.E-3*VAR(1)
      AA=AMOD(AX,B)
      BB=.015*SIN(AA)
      DU1=(U2+1.D-3)*BB
      DU2=U1*BB
      Y(1)=VAR(2)-VAR(6)
      Y(2)=VAR(3)-VAR(7)
      Y(3)=VAR(4)-VAR(8)
      Y(4)=VAR(5)-VAR(9)
      DO 20 I=1,8
      X=VAR(I+1)
      Z=ABS(X)
      EU(I)=1.E-10*Z+1.E-12
20    EL(I)=1.E-12*Z+1.E-14
      IF(U10.GT.0..AND.U20.GT.0.)GO TO 2000
      IF(U10.GT.0..AND.U20.LT.0.)GO TO 2002
      IF(U10.LT.0..AND.U20.GT.0.)GO TO 2006
      IF(U10.LT.0..AND.U20.LT.0.)GO TO 2008
      IF(U10.LT.0..AND.U20.EQ.0.)GO TO 2007
      IF(U10.EQ.0..AND.U20.LT.0.)GO TO 2005
      IF(U10.EQ.0..AND.U20.EQ.0.)GO TO 2004
      IF(U10.EQ.0..AND.U20.GT.0.)GO TO 2003
      IF(U10.GT.0..AND.U20.EQ.0.)GO TO 2001
2009  CONTINUE
      IF(FLAG7)759,759,757
759   CONTINUE
      CALL DER
      CALL AMRK
757   IF(FLAG3)704,704,705
704   IF(VAR(1)+DIR(1).GE.TS1) GO TO 700
705   IF(FLAG4)706,706,707
706   IF(VAR(1)+DIR(1).GE.T1) GO TO 701
707   IF(FLAG5)708,708,709
708   IF(VAR(1)+DIR(1).GE.TS2) GO TO 702
709   IF(FLAG6)710,710,30
710   IF(VAR(1)+DIR(1).GE.T) GO TO 703
758   CONTINUE
      FLAG7=0.
      IF(ABS(A+DELT-VAR(I)).LT..000001)GO TO 25
```

```
      IF(A+DELT-VAR(1)-DIR(1))60,60,61
 60   DIR(1)=A+DELT-VAR(1)
 61   CONTINUE
 22   GO TO 28
 700  IF(A+DELT.LE.TS1)GO TO 758
      FLAG3=1.
      DIR(1)=TS1-VAR(1)
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      U10=-U10
      GO TO 28
 701  IF(A+DELT.LE.T1)GO TO 758
      FLAG4=1.
      DIR(1)=T1-VAR(1)
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      U10=-U10
      GO TO 28
 702  IF(A+DELT.LE.TS2)GO TO 758
      FLAG5=1.
      DIR(1)=TS2-VAR(1)
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      U20=-U20
      GO TO 28
 703  IF(A+DELT.LE.T)GO TO 758
      FLAG6=1.
      DIR(1)=T-VAR(1)
      WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      GO TO 28
 2000 U1=1.1D-4+DU1
      U2=U2A+DU2
      GO TO 2009
 2001 U1=1.D-4+DU1
      U2=U2C+DU2
      GO TO 2009
 2002 U1=.9D-4+DU1
      U2=U2B+DU2
      GO TO 2009
 2003 U1=0.+DU1
      U2=1.D-4+DU2
      GO TO 2009
 2004 U1=0.+DU1
      U2=0.+DU2
      GO TO 2009
 2005 U1=0.+DU1
      U2=-1.D-4+DU2
      GO TO 2009
 2006 U1=-1.1D-4+DU1
      U2=U2A+DU2
      GO TO 2009
 2007 U1=-1.D-4+DU1
      U2=U2C+DU2
      GO TO 2009
 2008 U1=-.9D-4+DU1
      U2=U2B+DU2
      GO TO 2009
 1000 FORMAT(4E18.8)
```

```
30    WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
      WRITE(6,1000) VAR(2),VAR(3),VAR(4),VAR(5)
      WRITE(6,1000)VAR(1)
      II=II-1
      DO 860 I=1,II
      XY2(1,I)=XY1(1,I)
860   XY2(2,I)=XY1(7,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE1,XNAME,YNAME1,3)
      DO 861 I=1,II
861   XY2(1,I)=XY1(2,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE2,XNAME,YNAME2,3)
      DO 862 I=1,II
862   XY2(1,I)=XY1(3,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE3,XNAME,YNAME1,3)
      DO 863 I=1,II
863   XY2(1,I)=XY1(4,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE4,XNAME,YNAME2,3)
      DO 864 I=1,II
864   XY2(1,I)=XY1(5,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE5,XNAME,YNAME3,3)
      DO 865 I=1,II
865   XY2(1,I)=XY1(6,I)
      CALL KCPLOT(XY2,2,1,2,1,II,1,SYMBOL,TITLE6,XNAME,YNAME3,3)
      STOP
      END
      SUBROUTINE DER
      DIMENSION VAR(9),DIR(9)
      COMMON VAR,DIR,U1,U2,CO,SI
      DOUBLE PRECISION  VAR,DIR,A,B,C,U1,U2,CO,SI,BDOT
      A=(VAR(2)+1.5D11)**2+(VAR(4))**2
      B=DSQRT(A)
      C=A*B
      SI=VAR(4)/B
      CO=(VAR(2)+1.5D11)/B
      DIR(2)=VAR(3)
      DIR(3)=(-1.325D20*(VAR(2)+1.5D11))/C-1.D-3    *SI+U1*CO-U2*SI
      DIR(4)=VAR(5)
      DIR(5)= -1.325D20*VAR(4)/C+1.D-3    *CO+U1*SI+U2*CO
      A=(VAR(6)+1.5D11)**2+(VAR(8))**2
      B=DSQRT(A)
      C=A*B
      SI=VAR(8)/B
      CO=(VAR(6)+1.5D11)/B
      DIR(6)=VAR(7)
      DIR(7)=(-1.325D20*(VAR(2)+1.5D11))/C-1.D-3    *VAR(4)/B
      DIR(8)=VAR(9)
      DIR(9)= -1.325D20*VAR(8)/C+1.D-3    *(VAR(2)+1.5D11)/B
      RETURN
      END
```

Program 4. Computation of switching times for control system No. 2, according to Newton–Raphson technique

```
      DIMENSION  A(4,4),DT(4),S1(12),S2(4),B(4)
      FAC1=1.1
      FAC2=.9
      FAC3=11.*SQRT(.99)-10.
      FAC4=10.-9.*SQRT(.99)
      W=(ATAN(.85479761E10/(1.5E11+.42321837E10)))/.47408336E6
      ST=(FAC3+FAC4)/2.
      WRITE(6,100)W
      W2=W**2
      FAC=.5
      DETERM=0.
      U1=-1.E-4
      U2=1.E-4
      X10=74000.
      X20=.25
      X30=-151000.
      X40=-.5
      T=25000.
      TS2=14000.
      TS1=9000.
      T1=20000.
4     S1=SIN(W*T)
      C1=COS(W*T)
      T2=T
      D1=X10+T*X20+U1*(T*S1*FAC2/W+(C1*FAC2-FAC1)/W2)-U2*FAC4*(S1/W2-T*C
     11/W)
      D2=X20+U1*FAC2*S1/W+U2*(C1*FAC4-FAC3)/W
      D3=X30+T*X40+U1*FAC2*(S1/W2-T*C1/W)+U2*(T*FAC4*S1/W+(C1*FAC4-FAC3)
     1/W2)
      D4=X40+U1*(FAC1-C1*FAC2)/W+U2*S1*FAC4/W
3     S2=SIN(W*TS1)
      C2=COS(W*TS1)
      S3=SIN(W*T1)
      C3=COS(W*T1)
      S4=SIN(W*TS2)
      C4=COS(W*TS2)
      S5=S1
      C5=C1
      X1T=D1+U1*(2.*(TS1*S2*FAC1-T1*S3*FAC2)/W+2.*(C2*FAC1-C3*FAC2)/W2+(
     1FAC2-FAC1)*(TS2*S4/W+C4/W2))-U2*(2.*(FAC4*T2*C5-TS2*ST*C4)/W+2.*(S
     14*ST-S5*FAC4)/W2)
      X2T=D2+U1*(2.*(S2*FAC1-S3*FAC2)/W+(FAC2-FAC1)*S4/W)-U2*(2.*(C5*FAC
     14-C4*ST)/W)
      X3T=D3+U1*(2.*(T1*C3*FAC2-TS1*C2*FAC1)/W+2.*(S2*FAC1-S3*FAC2)/W2
     3+(FAC2-
     1FAC1)*(S4/W2-TS2*C4/W))+U2*(2.*(TS2*ST*S4-T2*FAC4*S5)/W+2.*(C4*ST-
     2C5*FAC4)/W2)
      X4T=D4+U1*(2.*(C3*FAC2-C2*FAC1)/W-(FAC2-FAC1)*(C4/W))+U2*(2.*(S4*S
     1T-S5*FAC4)/W)
      X1T=-X1T+2.*X10+T*X20+T*X2T
      X3T=-X3T+2.*X30+T*X40+T*X4T
      A(2,1)=2.*U1*C2*FAC1
      A(1,1)=A(2,1)*TS1
      A(4,1)=2.*U1*S2*FAC1
      A(3,1)=A(4,1)*TS1
      A(2,2)=-2.*U1*C3*FAC2
      A(1,2)=A(2,2)*T1
      A(4,2)=-2.*U1*S3*FAC2
      A(3,2)=A(4,2)*T1
```

```
      A(2,3)=-U2*2.*ST*S4-(FAC2-FAC1)*C4*U1
      A(1,3)=A(2,3)*TS2
      A(4,3)=2.*ST*U2*C4-(FAC2-FAC1)*S4*U1
      A(3,3)=A(4,3)*TS2
      A(2,4)=U2*S5*FAC4
      A(1,4)=A(2,4)*T2+X20
      A(4,4)=-U2*C5*FAC4
      A(3,4)=A(4,4)*T2+X40
      A(1,1)=-A(1,1)+T*A(2,1)
      A(1,2)=-A(1,2)+T*A(2,2)
      A(1,3)=-A(1,3)+T*A(2,3)
      A(1,4)=-A(1,4)+T*A(2,4)+X2T
      A(3,1)=-A(3,1)+T*A(4,1)
      A(3,2)=-A(3,2)+T*A(4,2)
      A(3,3)=-A(3,3)+T*A(4,3)
      A(3,4)=-A(3,4)+T*A(4,4)+X4T
      CALL MATINV(4,A,4,B,0,DETERM,S1,S2)
      WRITE(6,100)TS1,T1,TS2,T2
      WRITE(6,100)X1T,X2T,X3T,X4T
      DO 2 I=1,4
2     DT(I)=-(A(I,1)*X1T+A(I,2)*X2T+A(I,3)*X3T+A(I,4)*X4T)*FAC
      WRITE (6,100)DT(1),DT(2),DT(3),DT(4),DETERM
      TS1=TS1+DT(1)
      T1=T1+DT(2)
      TS2=TS2+DT(3)
      T=T+DT(4)
      IF(ABS(T1).GE.1.E9)GO TO 30
      IF(X1T**2+X2T**2+X3T**2+X4T**2.LE..1)    GO TO 30
      GO TO 4
100   FORMAT(6E18.8)
30    STOP
      END
```

Program 5. Simulation program for solution to linear regulator problem

```
      DIMENSION VAR(9),DIR(9),TS(83),EU(8),EL(8),Y(4),RAN(2),P(4,4),UC1(
     14),UC2(4)
      DOUBLE PRECISION VAR,DIR,U1,U2,DIRT1,TS,DIRT2,DIRT3,DIRT4,BDOT,
     1U10,U20,RAN
      COMMON VAR,DIR,U1,U2,CO,SI
      EXTERNAL DER
      BDOT=.1188049D-6
      DELT=300.
      II=0
      CO=1.D0
      SI=0.D0
      U1=-1.D-4
      U2=1.D-4
      VAR(1)=0.D0
      VAR(2)=1100.D0
      VAR(3)=10305.D0
      VAR(4)=900.D0
      VAR(5)=17810.D0
      VAR(6)=100.D1
      VAR(7)=1030.D1
      VAR(8)=100.D1
      VAR(9)=1782.D1
      P(1,2)=200.
      P(1,4)=0.
      P(2,2)=20000.
      P(2,3)=0.
      P(2,4)=0.
      P(3,4)=200.
      P(4,4)=20000.
      P(2,1)=P(1,2)
      P(4,1)=P(1,4)
      P(4,2)=P(2,4)
      P(4,3)=P(3,4)
      XN1=ABS((VAR(2)-VAR(6))*1000.)
      XN2=ABS((VAR(3)-VAR(7))*2.)
      XN3=ABS((VAR(4)-VAR(8))*1000.)
      XN4=ABS((VAR(5)-VAR(9))*2.)
      DO 1 I=1,83
1     TS(I)=0.
      DIR(1)=.001
      CALL AMRKS(VAR,DIR,DER,8,0,EU,EL,100.,.001,TS,1)
25    CONTINUE
31    T=VAR(1)
      Y(1)=(VAR(2)-VAR(6))/XN1
      Y(2)=(VAR(3)-VAR(7))/XN2
      Y(3)=(VAR(4)-VAR(8))/XN3
      Y(4)=(VAR(5)-VAR(9))/XN4
      CALL JPLT3(1.,-1.,0.,DELT,4,II,Y)
      WRITE(6,1000)U1,U2
      II=1
28    CONTINUE
      DO 20 I=1,8
      X=VAR(I+1)
      Z=ABS(X)
      EU(I)=1.E-10*Z+1.E-12
20    EL(I)=1.E-12*Z+1.E-14
      IF((VAR(2)-VAR(5))**2+(VAR(4)-VAR(7))**2.LE.1.)GO TO 30
      DO 700 I=1,4
```

```
700    UC1(I)=-.5*(P(2,I)*CO+P(4,I)*SI)
       UV1=0.
       UV2=0.
       DO 701 I=1,4
701    UC2(I)=-.5*(P(2,I)*(-SI)+P(4,I)*CO)
       DO 702 I=1,4
       J=I+1
       JJ=5+I
       UV1=UV1+UC1(I)*(VAR(J)-VAR(JJ))
702    UV2=UV2+UC2(I)*(VAR(J)-VAR(JJ))
       IF(ABS(UV1).GE.1.E-4) GO TO 703
       U1=UV1
704    IF(ABS(UV2).GE.1.E-4) GO TO 705
       U2=UV2
706    CONTINUE
       CALL DER
       CALL AMRK
       IF(ABS(T+DELT-VAR(1)).LT..000001)GO TO 25
       IF(T+DELT-VAR(1)-DIR(1))60,60,61
60     DIR(1)=T+DELT-VAR(1)
61     CONTINUE
22     GO TO 28
703    IF(UV1.GT.0.)U1=1.D-4
       IF(UV1.LT.0.)U1=-1.D-4
       GO TO 704
705    IF(UV2.GT.0.)U2=1.D-4
       IF(UV2.LT.0.)U2=-1.D-4
       GO TO 706
30     WRITE(6,1000)Y(1),Y(2),Y(3),Y(4)
1000   FORMAT(4E18.8)
       STOP
       END
       SUBROUTINE DER
       DIMENSION VAR(9),DIR(9)
       COMMON VAR,DIR,U1,U2,CO,SI
       DOUBLE PRECISION  VAR,DIR,A,B,C,U1,U2,CO,SI,BDOT
       A=(VAR(2)+1.5D11)**2+(VAR(4))**2
       B=DSQRT(A)
       C=A*B
       SI=VAR(4)/B
       CO=(VAR(2)+1.5D11)/B
       DIR(2)=VAR(3)
       DIR(3)=(-1.325D20*(VAR(2)+1.5D11))/C-1.D-3    *SI+U1*CO-U2*SI
       DIR(4)=VAR(5)
       DIR(5)= -1.325D20*VAR(4)/C+1.D-3    *CO+U1*SI+U2*CO
       A=(VAR(6)+1.5D11)**2+(VAR(8))**2
       B=DSQRT(A)
       C=A*B
       SI=VAR(8)/B
       CO=(VAR(6)+1.5D11)/B
       DIR(6)=VAR(7)
       DIR(7)=(-1.325D20*(VAR(2)+1.5D11))/C-1.D-3    *VAR(4)/B
       DIR(8)=VAR(9)
       DIR(9)= -1.325D20*VAR(8)/C+1.D-3    *(VAR(2)+1.5D11)/B
       RETURN
       END
```

Appendix C

The Two-Term Nonlinear Filter Equations

The notation used here will be the same as that of Appendix F of Ref. 2. Equations designated by "F" will refer to that reference.

First we augment Eq. (F.15) to be

$$\mathbf{r}(\mathrm{C}, T) = \hat{\mathrm{X}}(T) + \mathbf{P}(T)\,\mathrm{C} + \begin{bmatrix} \mathrm{C}^{\mathrm{T}}\mathbf{R}_1\mathrm{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{C}^{\mathrm{T}}\mathbf{R}_n\mathrm{C} \end{bmatrix} \tag{C-1}$$

where the matrices $\mathbf{R}_1$ through $\mathbf{R}_n$ are $n \times n$. Equation (F.16) becomes

$$\frac{d\mathbf{P}}{dT}\mathrm{C} + \frac{d\hat{\mathrm{X}}}{dT} + \begin{bmatrix} \mathrm{C}^{\mathrm{T}}\dfrac{d\mathbf{R}_1}{dT}\mathrm{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{C}^{\mathrm{T}}\dfrac{d\mathbf{R}_n}{dT}\mathrm{C} \end{bmatrix} - \left(\mathbf{P}(T) + 2\begin{bmatrix} \mathrm{C}^{\mathrm{T}}\mathbf{R}_1' \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{C}^{\mathrm{T}}\mathbf{R}_n' \end{bmatrix}\right)\frac{\partial\mathcal{H}^*}{\partial r}\left(T, \mathbf{P}\mathrm{C} + \hat{\mathrm{X}} + \begin{bmatrix} \mathrm{C}^{\mathrm{T}}\mathbf{R}_1\mathrm{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{C}^{\mathrm{T}}\mathbf{R}_n\mathrm{C} \end{bmatrix}, \mathrm{C}\right) =$$

$$\frac{\partial\mathcal{H}^*}{\partial \mathrm{C}}\left(T, \hat{\mathrm{X}} + \mathbf{P}\mathrm{C} + \begin{bmatrix} \mathrm{C}^{\mathrm{T}}\mathbf{R}_1\mathrm{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathrm{C}^{\mathrm{T}}\mathbf{R}_n\mathrm{C} \end{bmatrix}, \mathrm{C}\right) \tag{C-2}$$

where $\mathbf{R}_1' = 1/2\,(\mathbf{R}_1 + \mathbf{R}_1^{\mathrm{T}})$, $\mathbf{R}_2' = 1/2\,(\mathbf{R}_2 + \mathbf{R}_2^{\mathrm{T}})$, etc.

Since we will consider terms of order C^2, we must include higher-order terms in the expansion of $\partial\mathcal{H}^*/\partial \mathbf{r}$ and $\partial\mathcal{H}^*/\partial \mathbf{C}$. Thus we have

$$\frac{\partial\mathcal{H}^*}{\partial \mathbf{r}}\left(T, \mathbf{PC} + \hat{\mathbf{X}} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}, \mathbf{C}\right) \doteq \frac{\partial\mathcal{H}^*}{\partial \mathbf{r}}(T, \hat{\mathbf{X}}, \mathbf{C}) + \frac{\partial^2\mathcal{H}^*}{\partial \mathbf{r}^2}(T, \hat{\mathbf{X}}, \mathbf{C})\left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right)$$

$$+ \frac{1}{2}\begin{bmatrix} \left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right)^T \frac{\partial}{\partial r_1}\frac{\partial^2\mathcal{H}^*}{\partial \mathbf{r}^2}\left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right) \\ \cdot \\ \cdot \\ \cdot \\ \left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right)^T \frac{\partial}{\partial r_n}\frac{\partial^2\mathcal{H}^*}{\partial \mathbf{r}^2}\left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right) \end{bmatrix} \tag{C-3}$$

and

$$\frac{\partial\mathcal{H}^*}{\partial \mathbf{C}}\left(T, \mathbf{PC} + \hat{\mathbf{X}} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}, \mathbf{C}\right) \doteq \frac{\partial\mathcal{H}^*}{\partial \mathbf{C}}(T, \hat{\mathbf{X}}, \mathbf{C}) + \frac{\partial^2\mathcal{H}^*}{\partial \mathbf{r}\,\partial \mathbf{C}}(T, \hat{\mathbf{X}}, \mathbf{C})\left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right)$$

$$+ \frac{1}{2}\begin{bmatrix} \left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right)^T \frac{\partial}{\partial r_1}\frac{\partial^2\mathcal{H}^*}{\partial \mathbf{r}\,\partial \mathbf{C}}\left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right) \\ \cdot \\ \cdot \\ \cdot \\ \left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right)^T \frac{\partial}{\partial r_n}\frac{\partial^2\mathcal{H}^*}{\partial \mathbf{r}\,\partial \mathbf{C}}\left(\mathbf{PC} + \begin{bmatrix} \mathbf{C}^T \mathbf{R}_1 \mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T \mathbf{R}_n \mathbf{C} \end{bmatrix}\right) \end{bmatrix} \tag{C-4}$$

The added terms in Eqs. (C-3) and (C-4) each contribute only one term that we shall consider. In Eq. (C-3), it is

$$\frac{1}{2}\begin{bmatrix} C^T P^T ((-2HQ(y-h))_{\hat{x}})_{\hat{x}_1} PC \\ \vdots \\ C^T P^T ((-2HQ(y-h))_{\hat{x}})_{\hat{x}_n} PC \end{bmatrix} \qquad \text{(C-5)}$$

and in Eq. (C-4), it is

$$\frac{1}{2}\begin{bmatrix} C^T P^T (g_{\hat{x}})_{\hat{x}_1} PC \\ \vdots \\ C^T P^T (g_{\hat{x}})_{\hat{x}_n} PC \end{bmatrix} \qquad \text{(C-6)}$$

(H was defined in connection with Eq. (54), and g is the plant dynamics vector as used in Appendix F of Ref. 2.) Now, including the terms of (C-5) and (C-6), Eq. (C-2) becomes

$$\frac{dP}{dT}C + \frac{d\hat{X}}{dT} + \begin{bmatrix} C^T \dfrac{dR_1}{dT} C \\ \vdots \\ C^T \dfrac{dR_n}{dT} C \end{bmatrix} - (P(T) + 2\begin{bmatrix} C^T R_1' \\ \vdots \\ C^T R_n' \end{bmatrix})\Bigg(-2HQ(y-h) + g_{\hat{x}}^T C - \frac{1}{4}(C^T kV^{-1}k^T C)_{\hat{x}} + \Bigg(-2(HQ(y-h))_{\hat{x}}$$

$$+ (g_{\hat{x}}^T C)_{\hat{x}} - \frac{1}{4}(C^T (kV^{-1}k^T) C)_{\hat{x}\hat{x}}\Bigg)(PC + \begin{bmatrix} C^T R_1 C \\ \vdots \\ C^T R_n C \end{bmatrix}) - \frac{1}{2}\begin{bmatrix} C^T P^T ((2HQ(y-h))_{\hat{x}})_{\hat{x}_1} PC \\ \vdots \\ C^T P^T ((2HQ(y-h))_{\hat{x}})_{\hat{x}_n} PC \end{bmatrix}\Bigg) =$$

$$g - \frac{1}{2}(kV^{-1}k^T)C + \left(g_{\hat{x}} - \frac{1}{2}(kV^{-1}k^T C)_{\hat{x}}\right)(PC + \begin{bmatrix} C^T R_1 C \\ \vdots \\ C^T R_n C \end{bmatrix}) + \frac{1}{2}\begin{bmatrix} C^T P^T (g_{\hat{x}})_{\hat{x}_1} PC \\ \vdots \\ C^T P^T (g_{\hat{x}})_{\hat{x}_n} PC \end{bmatrix} \qquad \text{(C-7)}$$

Collecting terms in Eq. (C-7) of order zero:

$$\frac{d\hat{X}}{dT} + 2\mathbf{P}(T)\,\mathbf{HQ}(\mathrm{y} - \mathrm{h}) = \mathrm{g} \tag{C-8}$$

Of order 1:

$$\frac{d\mathbf{P}}{dT}\mathbf{C} + 2\begin{bmatrix} \mathbf{C}^T\mathbf{R}_1' \\ \cdot \\ \vdots \\ \cdot \\ \mathbf{C}^T\mathbf{R}_n' \end{bmatrix}\mathbf{HQ}(\mathbf{y}-\mathbf{h}) - \mathbf{P}(T)\,\mathbf{g}_{\hat{\mathbf{x}}}^T\mathbf{C} + 2\mathbf{P}(\mathbf{HQ}(\mathbf{y}-\mathbf{h}))_{\hat{\mathbf{x}}}\mathbf{PC} = -\frac{1}{2}(\mathbf{kV}^{-1}\mathbf{k}^T)\,\mathbf{C} + \mathbf{g}_{\hat{\mathbf{x}}}\mathbf{PC} \tag{C-9}$$

Of order 2:

$$\begin{bmatrix} \mathbf{C}^T\frac{d\mathbf{R}_1}{dT}\mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T\frac{d\mathbf{R}_n}{dT}\mathbf{C} \end{bmatrix} + \frac{1}{4}\mathbf{P}(T)(\mathbf{C}^T\mathbf{kV}^{-1}\mathbf{k}^T\mathbf{C})_{\hat{\mathbf{x}}} - 2\begin{bmatrix} \mathbf{C}^T\mathbf{R}_1'\,\mathbf{g}_{\hat{\mathbf{x}}}^T\mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T\mathbf{R}_n'\,\mathbf{g}_{\hat{\mathbf{x}}}^T\mathbf{C} \end{bmatrix} + \mathbf{P}(T)(2\mathbf{HQ}(\mathbf{y}-\mathbf{h}))_{\hat{\mathbf{x}}}\begin{bmatrix} \mathbf{C}^T\mathbf{R}_1\mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T\mathbf{R}_n\mathbf{C} \end{bmatrix} - \mathbf{P}(T)(\mathbf{g}_{\hat{\mathbf{x}}}\mathbf{C})_{\hat{\mathbf{x}}}\mathbf{PC}$$

$$+ 4\begin{bmatrix} \mathbf{C}^T\mathbf{R}_1'(\mathbf{HQ}(\mathbf{y}-\mathbf{h}))_{\hat{\mathbf{x}}}\mathbf{PC} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T\mathbf{R}_n'(\mathbf{HQ}(\mathbf{y}-\mathbf{h}))_{\hat{\mathbf{x}}}\mathbf{PC} \end{bmatrix} + \frac{1}{2}\mathbf{P}(T)\begin{bmatrix} \mathbf{C}^T\mathbf{P}^T((2\mathbf{HQ}(\mathbf{y}-\mathbf{h}))_{\hat{\mathbf{x}}})_{\hat{x}_1}\mathbf{PC} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T\mathbf{P}^T((2\mathbf{HQ}(\mathbf{y}-\mathbf{h}))_{\hat{\mathbf{x}}})_{\hat{x}_n}\mathbf{PC} \end{bmatrix} =$$

$$\mathbf{g}_{\hat{\mathbf{x}}}\begin{bmatrix} \mathbf{C}^T\mathbf{R}_1\mathbf{C} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T\mathbf{R}_n\mathbf{C} \end{bmatrix} - \frac{1}{2}(\mathbf{kV}^{-1}\mathbf{k}^T\mathbf{C})_{\hat{\mathbf{x}}}\mathbf{PC} + \frac{1}{2}\begin{bmatrix} \mathbf{C}^T\mathbf{P}^T(\mathbf{g}_{\hat{\mathbf{x}}})_{\hat{x}_1}\mathbf{PC} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{C}^T\mathbf{P}^T(\mathbf{g}_{\hat{\mathbf{x}}})_{\hat{x}_n}\mathbf{PC} \end{bmatrix} \tag{C-10}$$

Letting $\mathbf{P} = -\mathbf{P}$, and eliminating C from Eqs. (8), (9), and (10), we finally obtain

$$\frac{d\hat{\mathrm{X}}}{dT} = \mathrm{g} + 2\mathbf{PHQ}(\mathrm{y} - \mathrm{h})$$

$$\frac{d\mathbf{P}}{dT} = 2\begin{bmatrix} (\mathrm{y}-\mathrm{h})^{T}\mathbf{Q}^{T}\mathbf{HR}_1' \\ \cdot \\ \cdot \\ \cdot \\ (\mathrm{y}-\mathrm{h})^{T}\mathbf{Q}^{T}\mathbf{HR}_n' \end{bmatrix} + \mathbf{P}(T)\mathrm{g}_{\hat{\mathrm{X}}}^{T} + \mathrm{g}_{\hat{\mathrm{X}}}\mathbf{P} + 2\mathbf{P}(\mathbf{HQ}(\mathrm{y}-\mathrm{h}))_{\hat{\mathrm{X}}}\mathbf{P} + \frac{1}{2}(\mathbf{kV}^{-1}\mathbf{k}^{T})$$

$$\begin{bmatrix} \dfrac{d\mathbf{R}_1}{dT} \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{d\mathbf{R}_n}{dT} \end{bmatrix} = \frac{1}{4}\mathbf{P}\bigstar\begin{bmatrix} (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_1} \\ \cdot \\ \cdot \\ \cdot \\ (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_n} \end{bmatrix} + 2\begin{bmatrix} \mathbf{R}_1'\mathrm{g}_{\hat{\mathrm{X}}}^{T} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{R}_n'\mathrm{g}_{\hat{\mathrm{X}}}^{T} \end{bmatrix} + 2(\mathbf{P}(\mathbf{HQ}(\mathrm{y}-\mathrm{h}))_{\hat{\mathrm{X}}})\bigstar\begin{bmatrix} \mathbf{R}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{R}_n \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_1}(1,1) & \cdots & \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_n}(1,1) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_1}(1,n) & \cdots & \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_n}(1,n) \end{bmatrix}\mathbf{P} \\ \cdot \\ \cdot \\ \cdot \\ \begin{bmatrix} \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_1}(n,1) & \cdots & \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_n}(n,1) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_1}(n,n) & \cdots & \mathbf{P}(\mathrm{g}_{\hat{\mathrm{X}}}^{T})_{\hat{x}_n}(n,n) \end{bmatrix}\mathbf{P} \end{bmatrix}$$

$$+ 4\begin{bmatrix} \mathbf{R}_1'(\mathbf{HQ}(\mathrm{y}-\mathrm{h}))_{\hat{\mathrm{X}}}\mathbf{P} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{R}_n'(\mathbf{HQ}(\mathrm{y}-\mathrm{h}))_{\hat{\mathrm{X}}}\mathbf{P} \end{bmatrix} + \mathrm{g}_{\hat{\mathrm{X}}}\bigstar\begin{bmatrix} \mathbf{R}_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{R}_n \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \begin{bmatrix} (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_1}(1,1) & \cdots & (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_n}(1,1) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_1}(1,1) & \cdots & (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_n}(1,n) \end{bmatrix}\mathbf{P} \\ \cdot \\ \cdot \\ \cdot \\ \begin{bmatrix} (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_1}(n,1) & \cdots & (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_n}(n,1) \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_1}(n,n) & \cdots & (\mathbf{kV}^{-1}\mathbf{k}^{T})_{\hat{x}_n}(n,n) \end{bmatrix}\mathbf{P} \end{bmatrix}$$

$$+ \mathbf{P}(T)\bigstar\begin{bmatrix} \mathbf{P}^{T}((\mathbf{HQ}(\mathrm{y}-\mathrm{h}))_{\hat{\mathrm{X}}})_{\hat{x}_1}\mathbf{P} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{P}^{T}((\mathbf{HQ}(\mathrm{y}-\mathrm{h}))_{\hat{\mathrm{X}}})_{\hat{x}_n}\mathbf{P} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \mathbf{P}^{T}(\mathrm{g}_{\hat{\mathrm{X}}})_{\hat{x}_1}\mathbf{P} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{P}^{T}(\mathrm{g}_{\hat{\mathrm{X}}})_{\hat{x}_n}\mathbf{P} \end{bmatrix}$$

where the star indicates a matrix multiplication defined by

$$\mathbf{A} \bigstar \begin{bmatrix} \mathbf{B}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{B}_n \end{bmatrix} = \begin{bmatrix} (A(1,1)\,\mathbf{B}_1 + A(1,2)\,\mathbf{B}_2 + \cdots + A(1,n)\,\mathbf{B}_n) \\ \cdot \\ \cdot \\ \cdot \\ (A(n,1)\,\mathbf{B}_1 + A(n,2)\,\mathbf{B}_2 + \cdots + A(n,n)\,\mathbf{B}_n) \end{bmatrix}$$

where $\mathbf{A}, \mathbf{B}_1 \cdots \mathbf{B}_n$ are $n \times n$ matrices.

The computer program that simulated the two-term filter is given in Appendix B. So far, no significant improvement has been noted over the one-term filter, probably because of the difficulty in choosing initial conditions on the **R** matrices. It is felt, however, that more work in this area would prove valuable.