64-638

A GENERAL, EXPLICIT, OPTIMIZING GUIDANCE LAW FOR ROCKET-PROPELLED SPACEFLIGHT

by

GEORGE W. CHERRY
Massachusetts Institute of Technology
Cambridge, Massachusetts

AIAA Paper
No. 64-638

AIAA/ION ASTRODYNAMICS GUIDANCE AND CONTROL CONFERENCE

LOS ANGELES, CALIFORNIA AUGUST 24-26, 1964

A General, Explicit, Optimizing Guidance Law For Rocket-Propelled Spaceflight

by

GEORGE W. CHERRY
Staff Member
Massachusetts Institute of Technology
Instrumentation Laboratory
Cambridge, Massachusetts

Abstract

An explicit optimizing method for guiding rocket-propelled vehicles is derived and applied. The method provides a universal solution to the many kinds of boundary-value problems encountered in powered-flight guidance. The derivation of the steering laws is extremely simple and avoids the use of *difficult or specialized mathematics. The method is called E Guidance.* An essential feature of the scheme is the E matrix, which maps the separation between the current boundary conditions and the desired boundary conditions into thrust-allocation guidance coefficients. These coefficients determine the required allocation of thrust acceleration along controlled coordinate axes. The guidance laws can control final coordinates of position as well as final components of velocity, and control throttleable as well as fixed-thrust rockets. Because of the generality of E Guidance, the method is particularly applicable to many-faceted complex space missions. A universal powered-flight guidance program for such a mission is described. The program provides for each type of powered-flight guidance problem by linking the appropriate set of stored-program subroutines. The universal powered-flight guidance program is tailored to the peculiar powers of a digital computer by exploiting the machine's *switching, branching, and decision-making capabilities.*

The Nature of the Powered-Flight Guidance Problem

The synthesis of guidance laws for rocket-propelled flight has been called a multidimensional, nonlinear, boundary-valued, variational problem. It is worth analyzing this formidable-sounding description; for it serves as an introduction to the difficulties, challenges, and terminology of the synthesis problem. Each of the adjectives in the description describes an aspect of the problem which makes its solution far from trivial.

The adjective "multidimensional" refers to the fact that there are usually two or three spacecraft spatial coordinates (geometric "dimensions") which must be controlled. During a lunar landing at a specified landing site, there are three spatial coordinates (or dimensions) which must be controlled - for example, the three cartesian coordinates of physical space. This "multidimensionality" represents a synchronization problem, because the three coordinates of the position vector and the three components of the velocity vector, must simultaneously or "synchronously" achieve the desired values. The guidance equationist must, therefore, solve several problems simultaneously. He solves the control problem along any given axis or coordinate by allocating thrust acceleration along that axis. Since the component of thrust acceleration along any axis is obtained from a single thrust acceleration vector, the total thrust acceleration must be time-shared among the controlled axes. The program for time-sharing thrust-acceleration, the master schedule for allocating thrust acceleration along the controlled axes, must be carefully planned so that the solutions to the problems along the several controlled axes are realized simultaneously. A quantity T_{go} appears in all the guidance equations in this paper. This quantity is the remaining time of rocket-burning at any instant of the powered flight. The time-to-go is the synchronizing variable for ensuring simultaneous solutions of all dimensions of the total guidance problem.

When the rocket engine is throttleable there is a certain latitude in the choice of T_{go}; for there is then additional flexibility in the allocation of thrust acceleration. When the rocket-engine thrust magnitude is fixed, the solution T_{go} is unique and the computation of *the correct* T_{go} *constitutes a more serious problem.*

The adjective "nonlinear" refers to the fact that the differential equations of state of a rocket-propelled spacecraft are nonlinear in the state variables. For example, the differential equation of radial motion is

$$\ddot{r} = -\mu / r^2 + v_\theta^2 / r + a_T \sin\alpha \tag{1}$$

This differential equation is nonlinear in r. This nonlinearity usually makes more difficult the synthesis of a guidance law for controlling the terminal radius and radial rate of a spacecraft. The E Guidance technique conveniently avoids the difficulties usually caused by nonlinear terms such as gravitational acceleration and centrifugal acceleration, the first and second terms on the right-hand side of Eq.(1). The nonlinear terms are taken into account in a way that necessitates no approximations.

The adjective "boundary-valued" refers to the fact that in a powered-flight guidance problem certain boundary values of the position and velocity vectors are specified (are "valued"). These boundary-valued problems are "two-point" or "mixed" boundary-valued problems because components of the position and velocity vectors are specified at both ends of the trajectory. An example of a powered-flight, two-point, boundary- valued problem may be illuminating. At some instant, $t = t_o$, the navigation system of a lunar-landing spacecraft informs the computer of the current position and velocity of the vehicle.

$$\underline{r}(t_o) = [x(t_o) \quad y(t_o) \quad z(t_o)] \tag{2}$$

$$\underline{v}(t_o) = [\dot{x}(t_o) \quad \dot{y}(t_o) \quad \dot{z}(t_o)] \tag{3}$$

The desired terminal conditions are

$$\underline{r}(T) = (x_D \quad y_D \quad z_D) \tag{4}$$

$$\underline{v}(T) = (0 \quad 0 \quad 0) \tag{5}$$

The guidance law must steer the spacecraft from the current boundary conditions, Eqs. (2,3), to the desired boundary conditions, Eqs. (4,5).

Some powered-flight guidance problems do not require specification of the components of the position vector at thrust termination. Guidance laws for problems of this type can frequently be synthesized by defining an instantaneous velocity-to-be gained, $\underline{v}_g$.

$$\underline{v}_g = \underline{v}_r^* - \underline{v} \tag{6}$$

where $\underline{v}_r$ is the current velocity required and $\underline{v}$ is the current spacecraft velocity. The *steering* technique is to so orientate the thrust acceleration vector $\underline{a}_T$ that $|\underline{v}_g|$ eventually shrinks to zero. This can be accomplished by continually directing $\underline{a}_T$ along $\underline{v}_g$ or by directing $\underline{a}_T$ in a direction which causes $\dot{\underline{v}}_g$ to lie along $-\underline{v}_g$, or even some weighted combination of the two regimes. The clarity of the conception behind this steering technique, which is, intuitively, simply to point the thrust acceleration vector in a direction which causes $\underline{v}_g$ to shrink in length, has recommended it to many engineers working in the area of spacecraft guidance and control. Its limitation is that $\underline{v}_r$ and $\underline{v}_g$ cannot be defined in a useful way for many significant powered-flight boundary-value problems. If the boundary-value problem is of the type which continually permits a solution under a single impulsive change in the velocity vector, then the method is applicable. If the problem is essentially of the type whose solution requires two timed impulsive changes in the velocity vector, then the method is not applicable, i.e., $\underline{v}_r$ and $\underline{v}_g$ cannot be usefully defined. Many spaceflight boundary-value problems are of the two-impulse type: lunar landing, for example, (the first impulse to place the spacecraft on a freefall trajectory which passes through the landing site, the second impulse - applied exactly when the landing site is reached - to null $\underline{v}$); insertion into a circular orbit at a specified altitude (the first impulse to place the spacecraft on a trajectory which passes through the required orbital altitude, the second impulse - applied exactly at the instant of reaching the desired orbital altitude - to place the spacecraft in a circular orbit.)

It is interesting to understand where the real difficulty with the $\underline{v}_g$ - nulling method lies. It is apparently powerless to solve the synchronization problem of making the components of the position and velocity vectors become equal to specified values simultaneously.* It can be used to achieve a specified velocity vector (it can even make the specified velocity vector a function of time and position) but where the spacecraft is when the specified velocity vector is achieved is not under explicit control. If infinite thrust levels and hence impulsive (instantaneous) velocity changes were physically realizable, then the $\underline{v}_g$ - nulling method would be universal; for then the synchronization problem could be solved by waiting until the spacecraft has coasted to the correct position and then applying an infinite thrust pulse to instantaneously adjust the velocity vector. Because of the limitations of the $\underline{v}_g$ - nulling method, only approximately one-half of the powered-flight maneuvers of a complex space mission (by actual count on a typical mission of this type) can be handled by that technique. The E Guidance technique, on the other hand, can handle all the powered-flight boundary-value problems of such a mission.

*See Appendix B for a partial recantation of this statement.

The adjective "variational" refers to the fact that the solution to the multidimensional, nonlinear, boundary-valued guidance problem must be optimum according to some criterion. The two-point boundary-value problems of guided powered flight are difficult in their own right without the requirement to optimize some function of the state or control variables. Nevertheless, the need to optimize the guidance program, to minimize the fuel used during a powered maneuver, for example, is inescapable. The problem of optimization has given rise in this decade to a great deal of mathematical speculation, research, and discovery. Some effective sophisticated numerical optimization methods have been programmed (often in double precision arithmetic) on high-speed, large-memory digital computers. The programs construct a single optimum guided trajectory in approximately 10 to 20 minutes. These techniques will not be applicable for many years to real-time control of spacecrafts with compact, light-weight guidance and control computers. Attempts have been made to precompute one nominal optimum trajectory (or even hundreds of perturbed-from-nominal optimum trajectories) and fit thrust angle regimes in-flight to the precomputed optimum trajectories. These attempts often encounter serious numerical and control problems. Furthermore, these methods, which are inherently dependent on precomputed nominal trajectories, do not have the flexibility that an explicit, direct, in-flight solution to the equations of motion has.

The numerical optimization methods, such as the method of steepest ascent, for example, are very useful for establishing target performance figures for more practical spacecraft guidance schemes. This does not mean that the sophisticated optimization computer programs consistently produce performance figures which are better than, or even as good as, the performance figures produced by the guidance law in this paper. The theoretical performance of the steepest ascent or gradient method is not realized by the digital computer programs which mechanize the algorithm. The programs are beset with numerical roundoff difficulties in computing the influence functions and gradient. It is probably for this reason that the E Guidance method frequently produces performance figures slightly superior to the theoretically optimum numerical optimization programs. Consequently, the E Guidance method can be recommended not only as a real-time spacecraft guidance law but also as a ground-based computer algorithm suitable for further research into the nature of the optimum guided trajectory.

Explicit guidance laws are laws which express the formulas for the steering commands directly in terms of the current and desired boundary values of the components of the position and velocity vectors. For the guidance laws to be truly explicit, that is valid for any values of the current and desired boundary conditions, the laws must be derived as direct solutions to the equations of motion. It is often possible to derive guidance laws which are exact as well as direct solutions to the equations of motion. This is a very satisfying accomplishment but not, of course, necessary. When it is not possible to derive explicit guidance laws which are exact, it is usually feasible to derive explicit equations which become better and better approximations as the current and desired boundary conditions approach each other closer and closer.

The alternative to explicit guidance laws are laws which are valid, or approximately valid, in the neighborhood of a pre-computed reference trajectory. This kind of implicit solution is useful for the unmanned single or double maneuver mission if the perturbations in the initial conditions, and perturbations in the thrust levels, are sufficiently low. But if the mission requires many different powered-flight maneuvers, and if the mission is manned and consequently requires provisions for the myriad possible abort and emergency maneuvers, the solution based on pre-computed reference trajectories becomes very unattractive. Whereas there is no a priori requirement for explicit guidance laws, the greater elegance, accuracy, flexibility, and generality of the explicit method makes it extremely desirable.

The programming of guidance equations for the spacecraft computer is an interesting and significant problem; it is also a problem which has far-reaching implications for the computer hardware and software designer. It is evident that the simpler the equationist can make his guidance laws, the smaller the computer designer can make his computer. But it would seem ounce-wise and pound-foolish to replace a fifty pound computer with a thirty-five pound computer when the consequence must be the utilization of inefficient steering equations which require a disproportionate increase in fuel expenditure. The implications which the guidance equations have for the software designer are equally obvious. Should he program an interpreter which permits the guidance programs to be written in powerful instructions (such as matrix and vector manipulations)? The existence of such an interpreter decreases the amount of stored program required for the guidance equations; but it generally increases the execution time of the guidance computations. Does this increase in computation time affect the accuracy and stability of the guidance and control loop? If not, well and good, program an interpreter. But the interpreter requires storage itself. Can the guidance equations be so trimmed and condensed that their coding in the machine language requires less storage than the interpreter and interpreter-coded equations together? If so, then no interpreter! These questions are fascinating; but before logical answers can be given the guidance equations should be optimized and unified as much as possible. The mission requirements, the fuel budget, the vehicle dynamics, and many other factors must all be kept in mind simultaneously. This paper does not even try to answer all the questions just raised. It does represent a beginning, it is hoped, because it proposes a system of powered-flight guidance equations which offer mission flexibility, fuel economy, and programming unity.

Introduction to E Guidance
(An Example Using a Throttleable Engine)

This section of the paper contains the derivation of a guidance law for steering throttleable rockets. This guidance law serves as a remarkably simple introduction to the E matrix and the method of E Guidance. The derivation assumes that the rocket engine is continuously throttleable between a minimum thrust level and a maximum thrust level. The guidance equations are particularly applicable to such powered phases of the mission as a lunar landing maneuver or a terminal rendezvous maneuver. Figure 1 is a simple block representation of the explicit steering equations. While this guidance law could be used for simpler maneuvers such as orbit insertion or intercept trajectory injection, it would be more appropriate and more economical to use the fixed-thrust guidance law (derived in a later section) for such problems.

The guidance equations derived in this section are exact. They can accomodate any gravitational field model. It will be seen that the throttleable-engine guidance law is a very simple algorithm.

When either the position vector of rendezvous, or the time of rendezvous, is chosen, the terminal position vector, terminal velocity vector, and terminal time, of the rendezvous maneuver, are all fixed. When the time of rendezvous is chosen, for example, the target vehicle's ephemeris yields the corresponding required terminal position and velocity vectors, $\underline{r}(T)$ and $\underline{v}(T)$. Thus, the rendezvous two-point boundary-value problem is: Given the spacecraft's current position and velocity vectors

$$\left.\begin{aligned} \underline{r}(t_o) &= (x_o \quad y_o \quad z_o) \quad (7) \\ \underline{v}(t_o) &= (\dot{x}_o \quad \dot{y}_o \quad \dot{z}_o) \quad (8) \end{aligned}\right\} \begin{array}{l}\text{Current position}\\ \text{and velocity}\\ \text{of spacecraft}\end{array}$$

find a thrust acceleration vector program for $t_o \leq t \leq T$

$$\underline{a}_T(t) = [a_{Tx}(t) \quad a_{Ty}(t) \quad a_{Tz}(t)] \quad . \tag{9}$$

such that at the specified terminal time, t = T

$$\left.\begin{aligned}\underline{r}(T) &= (x_D \quad y_D \quad z_D)\\ \underline{v}(T) &= (\dot{x}_D \quad \dot{y}_D \quad \dot{z}_D)\end{aligned}\right\}\text{From position and velocity of target vehicle at } t = T \tag{10, 11}$$

The coordinate system is a planet-centered, inertial, cartesian coordinate system. The "variational" aspect of this problem appears in the requirement that

$$\int_{t_o}^{T} \sqrt{\underline{a}_T \cdot \underline{a}_T}\, dt = \text{Min} \tag{12}$$

The differential equations to which the vehicle's motion are subject are

$$d^2\, \underline{r}/dt^2 = \underline{g} + \underline{a}_T \tag{13}$$

or, in scalar form

$$\ddot{x} = g_x + a_{Tx} \tag{14}$$

$$\ddot{y} = g_y + a_{Ty} \tag{15}$$

$$\ddot{z} = g_z + a_{Tz} \tag{16}$$

If the gravity field is spherical, as it is usually assumed to be, Eqs. (14, 15, 16) are nonlinear.

$$\ddot{x} = -\mu x/(x^2 + y^2 + z^2)^{3/2} + a_{Tx} \tag{14a}$$

$$\ddot{y} = -\mu y/(x^2 + y^2 + z^2)^{3/2} + a_{Ty} \tag{15a}$$

$$\ddot{z} = -\mu z/(x^2 + y^2 + z^2)^{3/2} + a_{Tz} \tag{16a}$$

The nonlinear components of gravitational acceleration will be handled in such a way that they cause no difficulty.

It is effective to adopt a "divide-and-conquer" attitude toward this problem, and consider only one axis initially. Consider, therefore, the x-axis boundary-value problem. It is given that

$$\ddot{x}(t) = g_x(t) + a_{Tx}(t) \tag{17}$$

and specified that at $t = t_o$

$$\left.\begin{aligned}x(t_o) &= x_o\\ \dot{x}(t_o) &= \dot{x}_o\end{aligned}\right\}\text{From navigation system at the current time, } t = t_o \tag{18}$$

and required that at t = T

$$\left.\begin{aligned}x(T) &= x_D\\ \dot{x}(T) &= \dot{x}_D\end{aligned}\right\}\text{From known position of target vehicle at } t = T\text{, the rendezvous time} \tag{19}$$

The allocation of thrust acceleration along the x-axis, $a_{Tx}(t)$, is, of course, the control function and must be chosen to effect the transfer of the spacecraft from the initial boundary conditions, Eqs. (18), to the required terminal boundary conditions, Eqs. (19). It is convenient to work *initially with the* <u>total</u> *required acceleration* $\ddot{x}(t)$, *however, rather than the required* allocation of thrust acceleration, $a_{Tx}(t)$. When a solution $\ddot{x}(t)$ is determined, it is obtained by adjusting $a_{Tx}(t)$ so that the sum of $g_x(t)$ and $a_{Tx}(t)$ is equal to the solution $\ddot{x}(t)$. This approach makes it far easier to compute a solution $a_{Tx}(t)$ and include the effect of the generally nonlinear term $g_x(t)$.

If $\ddot{x}(t)$ is integrated from the current time $t = t_o$ to a general time t = t, the result is

$$\dot{x}(t) - \dot{x}(t_o) = \int_{t_o}^{t} \ddot{x}(s)\, ds \tag{20}$$

The dummy variable s is introduced in Eq. (20) to prevent confusing the dummy variable of integration with the upper limit of integration. Equation (20) can be used to develop two equations of constraint which $\ddot{x}(t)$ must satisfy. The first of these, Eq. (21), is developed by simply substituting the terminal time T for t in Eq. (20); the second, Eq. (22), is developed by integrating Eq. (20) between the current time $t = t_o$ and the terminal time t = T. The resulting equations of constraint are

$$\dot{x}_D - \dot{x}_o = \int_{t_o}^{T} \ddot{x}(s)\, ds \tag{21}$$

$$x_D - x_o - \dot{x}(t_o)\, T_{go} = \int_{t_o}^{T} \left[\int_{t_o}^{t} \ddot{x}(s)\, ds \right] dt \tag{22}$$

where, in Eq. (22), the notation

$$T_{go} = T - t_o \quad , \tag{23a}$$

has been introduced. Equations (21, 22) are superior to Eqs. (17, 18, 19) in the sense that two integral equations, which inherently contain the four auxiliary or boundary conditions represented by Eqs. (18) and Eqs. (19), replace five equations, Eqs. (17, 18, 19). The bad side of the coin is that the solution functions $\ddot{x}(t)$ and $a_{Tx}(t)$ must be determined by solving a pair of simultaneous linear integral equations. It does not seem very helpful at first glance to have put the unknown function under integral signs. It will shortly be seen, however, that the two integral equations can be transformed into a pair of simultaneous linear algebraic equations in two unknowns.

There is a difficulty with Eqs. (21, 22) which is more fundamental than that the *unknown function $\ddot{x}(t)$ appears under integral signs. Equations (21, 22) do not even uniquely* determine $\ddot{x}(t)$. The unknown function $\ddot{x}(t)$ can be thought of as being expressed in a generalized Fourier series; it has therefore an infinite number of undetermined coefficients or degrees of freedom. But Eqs. (21, 22) can determine only two of these undetermined coefficients or degrees of freedom. In other words, there are an infinite number of $\ddot{x}(t)$'s which satisfy Eqs. (21, 22). This is a rather ironical fact: in the midst of such a plenty of possible solutions, not a single one has so far been found. The typical mathematical sophisticate's way out of this wilderness of an infinite number of invisible solutions is to consider Eqs. (7-16) simultaneously. All these equations considered simultaneously - and Eq. (12) is particularly potent in this respect - do determine a unique solution. This "solution", unfortunately, is a formidable bunch of non-linear differential equations with mixed boundary conditions. The method considered in this paper is to defer simultaneous consideration of Eqs. (7-16), and to limit the number of degrees of freedom of $\ddot{x}(t)$ to two, exactly the number of degrees of freedom which Eqs. (21, 22) can determine. The function $\ddot{x}(t)$ is therefore defined to be

$$\ddot{x}(t) = c_1\, p_1(t) + c_2\, p_2(t) \tag{23}$$

where $p_1(t)$ and $p_2(t)$ are linearly independent, pre-specified functions of time, and c_1 and c_2 are coefficients which are chosen to satisfy Eqs. (21, 22). The two degree-of-freedom definition of $\ddot{x}(t)$ must be substituted into Eqs. (21, 22), and c_1 and c_2 chosen to satisfy these two equations of constraint. Substituting Eq. (23) in Eqs. (21, 22) yields

$$\dot{x}_D - \dot{x}_o = f_{11}\, c_1 + f_{12}\, c_2 \tag{24}$$

$$x_D - x_o - \dot{x}_o\, T_{go} = f_{21}\, c_1 + f_{22}\, c_2 \tag{25}$$

where

$$f_{11} = \int_{t_o}^{T} p_1(t)\, dt \tag{26}$$

$$f_{12} = \int_{t_o}^{T} p_2(t)\, dt \tag{27}$$

$$f_{21} = \int_{t_o}^{T} \left[\int^{t} p_1(s)\, ds \right] dt \tag{28}$$

$$f_{22} = \int_{t_o}^{T} \left[\int_{t_o}^{t} p_2(s)\, ds \right] dt \tag{29}$$

Assuming that $p_1(t)$ and $p_2(t)$ are integrable functions, the f_{ij}'s are simply algebraic functions of t_o and T. It is convenient to express the solution for c_1 and c_2 in matrix notation.

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_D - \dot{x}_o \\ x_D - (x_o + \dot{x}_o\, T_{go}) \end{bmatrix} \tag{30}$$

The E matrix, $E = [e_{ij}]$, is the inverse of the matrix $F = [f_{ij}]$. In terms of the f_{ij}, the elements of the E matrix are

$$e_{11} = f_{22}/\Delta \tag{31}$$

$$e_{12} = -f_{12}/\Delta \tag{32}$$

$$e_{21} = -f_{21}/\Delta \tag{33}$$

$$e_{22} = f_{11}/\Delta \tag{34}$$

where

$$\Delta = f_{11} f_{22} - f_{12} f_{21} \tag{35}$$

For example, suppose $p_1(t)$ and $p_2(t)$ are chosen as follows

$$p_1(t) = 1 \tag{36}$$

$$p_2(t) = T - t \tag{37}$$

These are linearly independent functions of time and therefore permit the solution of Eqs. (24, 25) for c_1 and c_2.

The E matrix for this particular choice of functions is very simple

$$\begin{bmatrix} c_1 \\ \\ c_2 \end{bmatrix} = \begin{bmatrix} 4/T_{go} & -6/T_{go}^2 \\ \\ -6/T_{go}^2 & 12/T_{go}^3 \end{bmatrix} \begin{bmatrix} \dot{x}_D - \dot{x}_o \\ \\ x_D - (x_o + \dot{x}_o T_{go}) \end{bmatrix} \tag{38}$$

The solution $\ddot{x}(t)$ is therefore

$$\ddot{x}(t) = c_1 + c_2 (T - t) \tag{39}$$

where c_1 and c_2 must be computed from Eq. (38). The required $\ddot{x}(t)$, Eq. (39), must be obtained by making the sum of the thrust acceleration and gravitational acceleration equal to the total required acceleration.

$$a_{Tx}(t) + g_x(t) = c_1 + c_2 (T - t) \tag{40}$$

or

$$a_{Tx}(t) = c_1 + c_2 (T - t) - g_x(t) \tag{41}$$

Thus, Eq. (41) yields the solution thrust acceleration profile.

The choice made for $p_1(t)$ and $p_2(t)$ in Eqs. (36, 37) leads to the particularly simple E matrix in Eq. (38). This choice of functions for $p_1(t)$ and $p_2(t)$ is not necessarily the optimum choice, although it is probably the simplest. It leads to quite a useful steering law, nevertheless, and results in the approximate minimization of the integral of the square of $a_T(t)$. It is better to postpone the discussion of optimizing the choice of $p_1(t)$ and $p_2(t)$, and complete the development of an E Guidance steering law. The E matrix in Eq. (38) exhibits the behavior and characteristics typical of all E matrices. Its simplicity is to the reader's advantage in gaining insight into the method of E Guidance.

It should be quite evident that the y-axis boundary-valued problem can be solved by a procedure identical to the one used to solve the x-axis boundary-valued problem. Indeed, the same E matrix could be used.

$$\begin{bmatrix} c_3 \\ \\ c_4 \end{bmatrix} = \begin{bmatrix} 4/T_{go} & -6/T_{go}^2 \\ \\ -6/T_{go}^2 & 12/T_{go}^3 \end{bmatrix} \begin{bmatrix} \dot{y}_D - \dot{y}_o \\ \\ y_D - (y_o + \dot{y}_o T_{go}) \end{bmatrix} \tag{42}$$

$$a_{Ty}(t) = c_3 + c_4 (T - t) - g_y(t) \tag{43}$$

Similar equations can be written to solve the z-axis boundary-valued problem. Thus, it is possible to compute all the components of a solution $\underline{a}_T(t)$. Of course, there is a constraining relationship between the length of $\underline{a}_T$ and the three components of $\underline{a}_T$. Because the engine is throttleable as well as gimballed, this constraint is easily satisfied. The throttle commands are issued to satisfy

$$|\underline{a}_T| = \sqrt{a_{Tx}^2 + a_{Ty}^2 + a_{Tz}^2} \tag{44}$$

The direction cosines of the desired thrust direction are simply

$$\cos \alpha = a_{Tx}/a_T \tag{45}$$

$$\cos \beta = a_{Ty}/a_T \tag{46}$$

$$\cos \gamma = a_{Tz}/a_T \tag{47}$$

Thus, a unit vector in the direction of the desired thrust vector is

$$\underline{1}\,a_T = (\cos \alpha \qquad \cos \beta \qquad \cos \gamma)$$

A discussion of programming these equations on a spacecraft computer which contains an interpreter for performing matrix and vector operations, is contained in a later section of the paper. The next section of the paper concerns optimizing $p_1(t)$ and $p_2(t)$. The balance of this section contains a discussion which is intended to give the reader physical insight into the E matrix and Eqs. (38-41).

The procedure for computing the allocation of thrust acceleration along the x-axis which will solve the x-axis boundary-valued problem, is to compute, at some instant $t = t_o$, Eq. (38), and, then, at any subsequent instant t, to compute $a_{Tx}(t)$ from Eq. (41).

Equation (38) should be recomputed periodically; but note that c_1 and c_2 will not change if the navigation system and the flight control system perform perfectly. The constancy of c_1 and c_2 under perfect conditions is due to the fact that when c_1 and c_2 are first computed a complete unique solution to the boundary-valued problem is given by $a_{Tx}(t)$ in Eq. (41). If x_o and $\dot{x}_o$ were correct when c_1 and c_2 were computed from Eq. (38), if $g_x(t)$ is correct for $t_o \leq t$, and if the flight control system generates $a_{Tx}(t)$ accurately, c_1 and c_2 remain the solution coefficients. At any subsequent time, if the navigation system is still operating perfectly, Eq. (38) will yield the same c_1 and c_2, and $a_{Tx}(t)$ will remain the same unique solution it was originally. It is not necessary, therefore, to compute Eq.(38) at a rate higher than that corresponding to the period at which flight control system errors produce a noticeable change in c_1 and c_2. If the navigation system information improves with increasing t, as it will, for example, during a lunar landing (because the landing radar data improves with decreasing altitude), then Eq. (38) should be recomputed at a rate corresponding to the period over which the improving radar data would make a meaningful change in c_1 and c_2.

Insight can be gained into the E Guidance method by careful examination of Eq. (38, 39). Consider first the "vector" which the E matrix maps into c_1 and c_2. The first "component" of this vector is $(\dot{x}_D - \dot{x}_o)$. If $\ddot{x}(t)$ were identically zero for $t_o \leq t \leq T$, then $(\dot{x}_D - \dot{x}_o)$ would be the error in $\dot{x}$ at $t = T$. A useful viewpoint is to recognize that $(\dot{x}_D - \dot{x}_o)$ predicts the final error in $\dot{x}$. If $\ddot{x}(t)$ were identically zero, what would $x_D - x(T)$, the final error in x, be? Evidently, the final x would be $(x_o + \dot{x}_o T_{go})$, and, consequently, the correct prediction of the final error in x is $[x_D - (x_o + \dot{x}_o T_{go})]$, the second "component" of the "vector" in Eq. (38). The total x-acceleration profile $\ddot{x}(t)$ must be so chosen that the predicted final deficiencies do not occur. The E matrix maps the predicted final deficiencies into the coefficients c_1 and c_2. These coefficients determine the $\ddot{x}(t)$ program which eliminates the predicted final deficiencies. Thus, the E matrix method can be viewed as being essentially a final-value control scheme. The general versions of Eqs. (38, 39) are

$$\begin{bmatrix} c_1 \\ \\ c_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \text{predicted final speed error} \\ \\ \text{predicted final displacement error} \end{bmatrix} \tag{48}$$

$$\text{correction acceleration program} = c_1\, p_1(t) + c_2\, p_2(t) \tag{49}$$

Equations (48, 49) are a general solution to final-value control problems which require attaining specified values of speed and displacement of some general coordinate, q, whose differential equation of motion is

$$\ddot{q} = f\,(p,\ \dot{p},\ q,\ \dot{q},\ \ldots,\ t) + a_{Tq} \tag{50}$$

The variables $\dot{p}$, p, $\dot{q}$, q, etc., are the state variables. Their current values are measured by the navigation system. The function $f(p, \dot{p}, q, \dot{q}, \ldots, t)$ can be nonlinear in the state variables, of course. The term a_{Tq} is the control or choice variable; it must be chosen to force q(t) and $\dot{q}(t)$ to the specified values at $t = T$. Figure 2 contains a block diagram for this rather general solution to an important problem.

The elements of the E matrix in Eq. (38) increase without bound as T_{go} becomes vanishingly small. This behavior is typical of the elements of any E matrix, no matter what functions are used for $p_1(t)$ and $p_2(t)$. Under ideal conditions the predicted final errors in the boundary conditions vanish as T_{go} approaches zero. Thus, ideally, c_1 and c_2 do not "blow up" as T_{go} approaches zero. Under practical conditions, the predicted final errors in the boundary conditions become negligible; but they do not vanish as surely as time-to-go vanishes. For example, noisy navigation data and computation roundoff noise make the predicted final errors fluctuate. Consequently, as T_{go} becomes vanishingly small, the negligible but non-vanishing errors in the boundary conditions require an infinite acceleration to correct them, and c_1 and c_2 blow up. This undesirable behavior of the E matrix and c_1 and c_2 is avoided by simply not recomputing the E matrix and c_1 and c_2 during the last few seconds of powered flight.

Before leaving the subject of the E matrix, an illustration is presented to show how the choice of an appropriate $\ddot{x}(t)$ program can control the terminal values of $\dot{x}$ and x.

Consider the hypothetical boundary-valued problem where

$$\left.\begin{aligned} x(o) &= 1 \\ \dot{x}(o) &= 2 \end{aligned}\right\} \quad \text{Initial Boundary Conditions} \tag{51}$$

and

$$\left.\begin{aligned} x(T) &= 11 \\ \dot{x}(T) &= 0 \end{aligned}\right\} \quad \text{Terminal Boundary Conditions} \tag{52}$$

$$T = 10 \tag{53}$$

Suppose that

$$p_1(t) = 1 \tag{54}$$

$$p_2(t) = T - t = 10 - t \tag{55}$$

Then, Eq. (38) is applicable; it becomes

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0.4 & -0.06 \\ -0.06 & 0.012 \end{bmatrix} \begin{bmatrix} 0-2 \\ 11-(1+20) \end{bmatrix} \tag{56}$$

Then

$$c_1 = -0.2 \tag{57}$$

$$c_2 = 0 \tag{58}$$

In this case, the linear function of time postulated for $\ddot{x}(t)$,

$$\ddot{x}(t) = c_1 + c_2 (T - t) \tag{59}$$

degenerates to a constant

$$\ddot{x}(t) = -0.2 \tag{60}$$

The constant, small, negative acceleration, -0.2 ft/sec^2, causes the guided body to move from the initial boundary conditions, Eq. (51), to the desired terminal boundary conditions, Eqs. (52). Figure 3 plots $x(t)$, $\dot{x}(t)$, and $\ddot{x}(t)$ versus time for the duration of the control process.

Optimization of the Guidance Law

In the previous section of this paper a guidance law for throttleable rockets was derived, and the E matrix and method of E Guidance was introduced. In this section the optimization of $p_1(t)$ and $p_2(t)$ is discussed.

The development of the E matrix assumed very few restrictions on $p_1(t)$ and $p_2(t)$. To ensure that the E matrix exists (remember that the E matrix is the inverse of the F matrix) it was necessary and sufficient to assume that $p_1(t)$ and $p_2(t)$ were linearly independent over any non-zero interval $t_o \le t \le T$. To ensure that the elements of the E matrix were algebraic functions of t_o and T it was assumed that $p_1(t)$ and $p_2(t)$ possessed first and second integrals over the interval $t_o \le t \le T$. For practical reasons, it is desirable that these first and second integrals be simple algebraic expressions. Simple polynomials in T_{go} would be desirable, for example. Except for these restrictions, the choice of functions for $p_1(t)$ and $p_2(t)$ is quite arbitrary, and the opportunity to choose fuel-optimizing functions clearly exists.

Suppose that the following functions are chosen

$$p_1(t) = a_0 + a_1 (T - t) + a_2 (T - t)^2 + \ldots + a_n (T - t)^n \tag{61}$$

$$p_2(t) = p_1(t) (T - t) \tag{62}$$

Note that as long as at least one of the a_i's in Eq. (61) is non-zero, the two functions $p_1(t)$ and $p_2(t)$ are linearly independent. For example, suppose that $a_2 = 1$ and that $a_i = 0$ for $i \neq 2$; then

$$p_1(t) = (T - t)^2 \tag{63}$$

$$p_2(t) = (T - t)^3 \tag{64}$$

Quite evidently, $p_1(t)$ and $p_2(t)$ are linearly independent. Thus the first restriction - linear independence of $p_1(t)$ and $p_2(t)$ - is satisfied; and the choice of the a_i's is arbitrary except for the single restriction that $a_i \neq 0$ for at least one i.

Consider now the satisfaction of the second restriction placed on $p_1(t)$ and $p_2(t)$: that these functions lead to simple algebraic formulae for

$$f_{11} = \int_{t_o}^{T} p_1(t)\, dt \tag{65}$$

$$f_{12} = \int_{t_o}^{T} p_2(t)\, dt \tag{66}$$

$$f_{21} = \int_{t_o}^{T}\left[\int_{t_o}^{t} p_1(s)\, ds\right] dt \tag{67}$$

$$f_{22} = \int_{t_o}^{T}\left[\int_{t_o}^{t} p_2(s)\, ds\right] dt \tag{68}$$

The functions chosen for $p_1(t)$ and $p_2(t)$ in Eqs. (61-62) satisfy this requirement very well because

$$\int_{t_o}^{T} (T - t)^n\, dt = T_{go}^{\ n+1}/(n + 1) \tag{69}$$

$$\int_{t_o}^{T}\left[\int_{t_o}^{t} (T - s)^n\, ds\right] dt = T_{go}^{\ n+2}/(n + 2) \tag{70}$$

and, consequently, the expressions in Eqs. (65-68) become simply

$$f_{11} = a_o T_{go} + a_1 T_{go}^{\ 2}/2 + \ldots + a_n T_{go}^{\ n+1}/(n + 1) \tag{71}$$

$$f_{12} = a_o T_{go}^{\ 2}/2 + a_1 T_{go}^{\ 3}/3 + \ldots + a_n T_{go}^{\ n+2}/(n + 2) \tag{72}$$

$$f_{21} = f_{12} \tag{72a}$$

$$f_{22} = a_o T_{go}^{\ 3}/3 + a_1 T_{go}^{\ 4}/4 + \ldots + a_n T_{go}^{\ n+3}/(n + 3) \tag{73}$$

Now consider the opportunity which exists for optimizing the fuel consumption. Using the terminology introduced earlier in the discussion of the control of a general coordinate, the total acceleration along the q coordinate is

$$\begin{aligned}\ddot{q}(t) = {} & c_1[\, a_o + a_1 (T - t) + \ldots + a_n(T - t)^n] \\ & + c_2\, [\, a_o (T - t) + \ldots + a_n(T - t)^{n+1}]\end{aligned} \tag{74}$$

Thus $\ddot{q}(t)$ is a polynomial function of $(T - t)$; but any analytic function of time can be expressed as a Taylor's series expansion about T, and thus be expressed as a polynomial in $(T - t)$. The optimum form of $\ddot{q}(t)$ should thus be expressible in the form of Eq. (74). For any allowable selection of the a_i's (at least one non-zero) the E matrix provides the c_1 and c_2 which satisfy the boundary conditions. Thus, the a_i's can be systematically varied in preflight simulations until the fuel consumption is minimized with respect to the a_i's. (For each selection the boundary-valued problem is solved exactly). The system for optimizing the choice of the a_i,s is of no concern here. The method could be a bulky, inefficient, brute-force procedure. No matter. The procedure is a ground-based computation. Once a good set of a_i's is found, the spacecraft computer program is supplied with these coefficients to provide an efficient, closed-form, exact solution to the real-time, in-flight boundary-valued problem.

In a later section of this paper, an analytic choice of the a_i's is described. This choice has been found to be extremely efficient for most of the boundary-valued problems encountered in spaceflight.

It has been found that the fuel efficiency of the guidance law is not very critical with respect to the choice of a_i's. A good choice of a_i's for a given powered flight boundary-valued problem, remains a good choice for a vastly different boundary-valued problem. Furthermore, the choice of $p_1(t)$ and $p_2(t)$ does not seem at all critical. Simulation of the guidance of a fixed-thrust rocket (the fixed-thrust rocket guidance law is discussed in a later section of the paper) from a launch site on the surface of the moon into a circular orbit was undertaken with different sets of $p_1(t)$ and $p_2(t)$. The first simulation used

$$p_1(t) = a_T(t) \tag{75}$$

$$p_2(t) = (T - t)\, a_T(t) \tag{76}$$

to find the solution $\ddot{r}(t)$ which resulted in $r(T)$ being equal to the desired orbital radius and $\dot{r}(T)$ being zero. The terminal time T was chosen to control $v_\theta(T)$. The second simulation used

$$p_1(t) = 1 \tag{77}$$

$$p_2(t) = T - t \tag{78}$$

(This second choice results in the same E matrix as in Eq. (38), of course.) The two simulations used negligibly different burning times to attain the circular orbit; the burning times differed by less than 0.05 percent.

Programming the Throttleable Rocket Guidance Law

Space guidance and navigation computations *involve a great many manipulations which* are naturally expressed in the language of matrix and vector analysis. While the computer is performing its guidance and navigation tasks, it is acting, from one viewpoint at least, as a "vector processor". The navigation programs for example propagate the state vector. During the midcourse phases the state vector is propagated by a linearized optimizing navigation procedure which involves a series of vector and matrix operations. During powered flight and midcourse corrections there are usually coordinate transformations required between navigation base axes and spacecraft body axes; these transformations are naturally performed by matrix multiplications. If all these computations were programmed as scalar equations in the digital machine's basic language, *a great deal of storage for* programs would be required. A system for reducing the storage required for programs is to provide the spacecraft computer with an interpreter. The interpretive system translates ("interprets") a powerful and convenient set of matrix and vector instructions into machine language at the time of machine execution of each instruction. The interpretive instructions allow the guidance and navigation algorithms to be programmed more economically. This section of the paper contains an interpreter-oriented formulation of the E Guidance throttleable engine steering law. Since the interpreter would be designed to process instructions for 3×3 matrices, the E and S matrices are defined as follows

$$E = \begin{bmatrix} 4/T_{go} & -6/T_{go}^2 & 0 \\ -6/T_{go}^2 & 12/T_{go}^3 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{79}$$

$$S = \begin{bmatrix} \text{-------}(\underline{v}_D - \underline{v}_o)\text{-------} \\ \text{---}(\underline{r}_D - \underline{r}_o - T_{go}\,\underline{v}_o)\text{---} \\ \text{-----------}0\text{-----------} \end{bmatrix} \tag{80}$$

The first row of the S matrix is the row vector (array) $\underline{v}_D$ minus the row vector $\underline{v}_o$. The last row of the S matrix is the zero vector

$$\underline{0} = (0 \quad 0 \quad 0) \tag{81}$$

(All the vectors in this section should be considered to be row arrays.) The following matrix product and row vectors are also defined

$$C = E\,S \tag{82}$$

$$\underline{p} = [1 \quad (T - t) \quad 0] \tag{83}$$

$$\underline{g} = (g_x \quad g_y \quad g_z) \tag{84}$$

In terms of the defined matrices and vectors, the solution thrust acceleration vector is

$$\underline{a}_{TD} = \underline{p}\,C - \underline{g} \tag{85}$$

Equation (85) can be verified by performing the vector-matrix product and vector addition on the right-hand side of this equation and comparing the result with Eqs. (38, 39) and Eqs. (42, 43). In particular, it is easy to verify that

$$C = \begin{bmatrix} c_1 & c_3 & c_5 \\ c_2 & c_4 & c_6 \\ 0 & 0 & 0 \end{bmatrix} \tag{86}$$

It is interesting to note that if the navigation system and flight control system were functioning perfectly, the matrix C would not change, even though recomputed, throughout the entire flight. Even with the errors associated with real hardware, the elements of C change very slowly. *Thus the C matrix can be computed at a relatively low computation rate.* The elements of the $\underline{g}$ vector also evolve slowly. Consequently, the desired thrust acceleration vector can be computed from Eq. (83) and Eq. (85) for many seconds without recomputation of C and $\underline{g}$. Consequently, major and minor computation loops can be set up in the guidance computer. During the major computation loop $\underline{r}$ and $\underline{v}$ are updated, and E, S, C, $\underline{p}$, $\underline{g}$ and $\underline{a}_{TD}$ are recomputed. *During the minor computation loop only the following computations* and the transformations to spacecraft body axes are performed

$$\underline{p} = [1 \quad (T - t) \quad 0] \tag{87}$$

$$\underline{a}_{TD} = \underline{p}C - \underline{g} \tag{88}$$

Figure 4 depicts the throttleable engine guidance law in block diagram form. Figure 5 is a flow diagram of the equations. The block labelled "Computation of Incremental Commands" contains the navigation base axes to spacecraft body axes conversion as well as the conversion of Δa_T commands to Δ thrust commands. *The detailed computations in this* block depend on the kind of hardware involved and consequently are not taken up in this paper. The computations in the block labelled "Gravity Computer and Accelerometer Integrator" are described in a later section of the paper.

The Fixed-Thrust Rocket Model

An excellent model for the behavior of many chemical rockets is an engine with constant massflow and constant effective exhaust velocity. Since the rocket thrust is the product of massflow and effective exhaust velocity, the thrust of this engine model is constant

$$v_e = g\, I_{sp} = \text{constant} \tag{89}$$

$$\dot{m} = \text{constant} \tag{90}$$

$$\dot{m} > o \tag{91}$$

$$F = \dot{m}\, v_e \tag{92}$$

Newton's second law for the acceleration due to thrust (the "thrust acceleration") becomes

$$\begin{aligned} m\, a_T &= F \\ m\, a_T &= \dot{m}\, v_e \end{aligned} \tag{93}$$

Since the massflow is constant, the mass of the rocket-propelled vehicle is a linear function of time

$$m = m_o - \dot{m}\, t \tag{94}$$

and the thrust acceleration is inversely proportional to a linear function of time

$$a_T = \dot{m}\, v_e / (m_o - \dot{m}\, t) \tag{95}$$

This expression for thrust acceleration is not useful for calculations aboard the spacecraft because m_o and $\dot{m}$ cannot be measured in flight. Dividing the numerator and denominator of the right-hand side of Eq. (95) by $\dot{m}$ yields

$$a_T = v_e / (\tau - t) \tag{96}$$

where

$$\tau \equiv m_o / \dot{m} \tag{97}$$

The quantity τ has the dimensions of time; and from its definition - total mass at $t = 0$, divided by massflow - it is seen that τ is the time at which a rocket-propelled vehicle which is all fuel (no structure) would completely consume itself. (Accordingly, the thrust acceleration increases very rapidly as t approaches τ.)

Fortunately, τ and v_e can be inferred from the accelerometer measurements. The process of inferring v_e and τ from the accelerometer measurements should be regarded as fitting, as accurately as possible, the constant thrust rocket model to the actual engine data. The fitting is done in flight and compensates, as far as guidance goes, for the rocket designer's erroneous predictions. Quite evidently, the two rocket-model parameters, v_e and τ, cannot be inferred from one datum, i.e., one accelerometer measurement. Nor will two accelerometer readings produce good estimation of v_e and τ - although two readings make estimation just mathematically possible. Since most accelerometers and their associated pulse storage registers, perform one integration, it is $\underline{\Delta V}$ which is actually available. The vector $\underline{\Delta V}$ must be numerically differentiated (divided) to obtain the thrust acceleration vector, from which the magnitude of the thrust acceleration can be obtained. The quantization of the acceleration outputs, the vehicle vibration, and finally, the numerical differentiation process all result in the thrust acceleration measurement being somewhat noisy.

The fixed-thrust rocket guidance law uses the quantities a_T, τ, and v_e in the explicit calculations of the required thrust angle regime and time of thrust termination. One of these quantities, v_e, is usually given to the equations designer with a fair degree of accuracy and reliability. However, it is possible to make the guidance law in this paper quite independent of any a priori information about v_e. While it is not the present purpose to describe in detail an efficient numerical process for estimating a_T, τ, and v_e from the accelerometer readings, it is appropriate to demonstrate how such an estimation could be performed and clearly establish its mathematical feasibility. Consider Eq. (96) which, upon inversion of each side, becomes:

$$1/a_T = \tau/v_e - (1/v_e)\, t \tag{98}$$

Now consider taking a series of accelerometer measurements in time, reciprocating each measurement.

$$\left.\begin{aligned} \alpha_1 &= 1/a_T(t_1) \\ \alpha_2 &= 1/a_T(t_2) \\ &\ \ \vdots \\ \alpha_n &= 1/a_T(t_n) \end{aligned}\right\} \begin{array}{l} \text{Series of} \\ \text{reciprocated} \\ \text{accelorometer} \\ \text{measurements} \end{array}$$

Accordingly, an overdetermined system of equations for τ/v_e and $(1/v_e)$ can be formed

$$\alpha_1 = \tau / v_e - (1/v_e)\, t_1 \tag{99}$$

$$\alpha_2 = \tau / v_e - (1/v_e)\, t_2$$

$$\vdots \tag{100}$$

$$\alpha_n = \tau / v_e - (1/v_e)\, t_n \tag{101}$$

These equations can be solved by the method of least squares, resulting in smoothed estimates of the unknown quantities. Clearly, the mathematical feasibility of the smoothing and estimation procedure has been established. It would of course be desirable to use a recursive estimation technique; in fact, an optimum linear filter, identical in concept to the midcourse navigation procedure of estimating the state vector, could be applied.

Introduction to the Fixed-Thrust Guidance Law

In the previous section a model of the fixed-thrust rocket was described. Even if this model is violated by the details of the real engine's operation, the model is still adequate because the guidance law is computed periodically, and a loop is closed, so to speak, around the incorrect model.

What kind of boundary-valued problems are usually presented to the fixed-thrust rocket system?

The problems belong to two general classifications:

1. A desired terminal velocity vector, $\underline{v}_D$, is specified. There are no burnout position constraints. The most important problem which can be solved this way is intercept trajectory injection. In general, $\underline{v}_D$ will be a function of burnout position and burnout time. Thus

$$\underline{v}_D = \underline{v}_D(\underline{r}_{BO}, T) \tag{102}$$

   This class of problems is precisely the one which can be solved by $\underline{v}_g$ - nulling methods. It is a sub-class of the class of problems solved by the E Guidance method.

2. A desired terminal velocity vector is specified. The altitude, or radius, is also specified. In addition, the plane of the trajectory may be specified. An example is the insertion of a vehicle into a circular orbit of specified altitude which lies in a prescribed plane.

Figure 6 is a block diagram representation of the fixed-thrust rocket guidance law. This guidance program has two major segments of computation. The block labeled "Radial Coordinate Control" computes a thrust angle regime, $\alpha(t)$, which will result in acquisition of the desired radial rate and desired radius (if specified) by the specified time of thrust termination T. The thrust angle $\alpha(t)$ is measured with respect to the local horizontal. This angle is illustrated in Fig. 7. For any set of radial boundary conditions and specified time of thrust termination (or, equivalently, time-to-go) the "Radial Coordinate Control" block can compute a solution thrust angle regime. (This statement is not precisely true; if too short a time-to-go is specified the allocation of thrust acceleration along the radial coordinate, $a_T \sin\alpha$, may be required to exceed a_T, a condition which results in the angle $\alpha(t)$ being imaginary since $\sin\alpha$ must exceed one.)

Consider the effect of varying time-to-go for some given set of radial boundary conditions. Evidently, as time-to-go is increased the terminal value of the horizontal component of velocity also increases. Since the terminal value of the horizontal speed is a function of T or time-to-go, it is feasible to select time-to-go so that the burnout horizontal speed is equal to the desired value. The block labeled "Horizontal Speed Control" adjusts time-to-go so that the predicted horizontal speed is equal to the desired value.

Incrementing time-to-go changes the terminal value of the horizontal speed through two effects. The first and most important is that a longer time of rocket burning simply results in thrust acceleration being applied to the spacecraft for a longer time. The second and secondary effect is that the thrust angle regime is a function of time-to-go. The thrust acceleration "lost" is therefore a function of time-to-go. Because the horizontal speed attained is affected by the precise form of $\alpha(t)$, the "Horizontal Speed Control" block uses $\alpha(t)$ as an input to predict the horizontal speed resulting from a given time-to-go. Figure 6 illustrates the computation loop through the two blocks "Radial Coordinate Control" and "Horizontal Speed Control". The circuit through this loop can be described as follows: time-to-go is guessed; $\alpha(t)$ corresponding to this time-to-go is computed in the upper block; using $\alpha(t)$ and time-to-go, the lower block computes the resulting final horizontal speed; time-to-go is adjusted so that final horizontal speed will be equal to the desired value. Initally, several passes through this loop may be made to attain the consistent values of $\alpha(t)$ and time-to-go which result in the satisfaction of the specified boundary conditions. Subsequent to this initial determination, a single cycle through this loop every second or so results in very accurate control of the terminal boundary values. The next two sections of this paper discuss in detail the contents of the "Radial Coordinate Control" and "Horizontal Speed Control" blocks.

Radial Coordinate Control

When both radius and radial rate are controlled, a solution $\ddot{r}(t)$ is obtained by use of the E matrix formulation exactly as in the first example in this paper. The only detailed difference between the illustration in the Introduction to E Guidance section and the present section, is that the sum of the "effective gravity" (combined effect of gravitational acceleration and centrifugal acceleration) and the radial component of thrust acceleration must be equal to the solution $\ddot{r}(t)$. Figure 2 and the discussion of this figure on page 14 can be particularized to the radial coordinate control problem, as illustrated in Fig. 8.

When control of the terminal radius is not required, the solution in Fig. 8 is modified somewhat. When the burnout radius is unconstrained, there is only one equation of constraint for the radial coordinate. Consequently, a one degree-of-freedom $\ddot{r}(t)$ is appropriate

$$\ddot{r}(t) = c_1\, p_1(t) \tag{103}$$

The equation of constraint which the total radial acceleration must satisfy is

$$\dot{r}_D - \dot{r}_o = \int_{t_o}^{T} \ddot{r}(t)\, dt \tag{104}$$

Substituting the single degree-of-freedom definition of the total radial acceleration into Eq. (104) yields

$$\dot{r}_D - \dot{r}_o = c_1 \int_{t_o}^{T} p_1(t)\, dt \tag{105}$$

Note that the integral in Eq. (105) has been defined in Eq. (26) as f_{11}. Hence, the appropriate c_1 for solving the radial rate boundary-valued problem is

$$c_1 = (\dot{r}_D - \dot{r}_o)/f_{11} \tag{106}$$

and the solution total acceleration program is

$$\ddot{r}(t) = c_1\, p_1(t) \tag{107}$$

The solution thrust angle program is obtained from

$$a_T \sin\alpha + g_{eff} = c_1\, p_1(t) \tag{108}$$

which can be solved for $\alpha(t)$.

$$\alpha(t) = \sin^{-1}\{[c_1\, p_1(t) - g_{eff}]/a_T\} \tag{109}$$

When the final radius is not controlled, it is still desirable, and often even necessary, to predict the burnout radius. The predicted burnout radius is required, to give some examples, for intercept trajectory injection (this problem is discussed in detail later); for insertion into a circular orbit of unspecified altitude; and, more generally, for insertion onto a conic trajectory of specified major axis and eccentricity, but unspecified orientation of the line of apsides. It is simple to develop an expression for the uncontrolled terminal radius from Eq. (107). Integration of this expression from the current time to general time t yields

$$\dot{r}(t) - \dot{r}_o = c_1 \int_{t_o}^{t} p_1(s)\, ds \tag{110}$$

Integrating this expression from the current time to the predicted terminal time yields

$$r(T) - r_o - \dot{r}_o T_{go} = c_1 \int_{t_o}^{T} \left[\int_{t_o}^{t} p_1(s)\, ds\right] dt \tag{111}$$

The double integral on the right-hand side of Eq. (111) has already been defined in Eq.(28). *Solving Eq. (111) for r(T) and using the definition in Eq. (28)* yields

$$r_{BO} = r(T) = r_o + \dot{r}_o T_{go} + c_1 f_{21} \tag{112}$$

Radial Coordinate Control Computer Program

It is desirable to write a radial coordinate control computer program which uses computer logical branching to select the constrained burnout radius mode or the unconstrained radius mode. Such a program follows. The assignment statements (the "equations" are really instructions assigning values to the symbols on the left-hand sides of the "equality" signs) are given equation numbers which are used as labels for logical branching. For example, (123) is the label for the instruction which sets c-sub-two to zero. In the following "program" advantage is taken of the symmetry of the F and E matrix.

Do state vector update at $t = t_o$ (113)

Do time-to-go computation (114)

$$T = t_o + T_{go} \tag{115}$$

$$r_o = \sqrt{\underline{r}_o \cdot \underline{r}_o} \tag{116}$$

$$\dot{r}_o = (\underline{r}_o \cdot \underline{v}_o)/r_o \tag{117}$$

If $T_{go} < K$, go to (135) (118)

$$f_{11} = a_o T_{go} + a_1 T_{go}^2/2 + a_2 T_{go}^3/3 \tag{119}$$

$$f_{21} = a_o T_{go}^2/2 + a_1 T_{go}^3/3 + a_2 T_{go}^4/4 \tag{120}$$

If r_{BO} is constrained, go to (126) (121)

$$c_1 = (\dot{r}_D - \dot{r}_o)/f_{11} \tag{122}$$

$$c_2 = 0 \tag{123}$$

$$r_{BO} = r_o + \dot{r}_o T_{go} + c_1 f_{21} \tag{124}$$

Go to (135) (125)

$$f_{22} = a_o T_{go}^3/3 + a_1 T_{go}^4/4 + a_2 T_{go}^5/5 \quad (126)$$

$$\det = f_{11} f_{22} - f_{21} f_{21} \quad (127)$$

$$e_{11} = + f_{22}/\det \quad (128)$$

$$e_{21} = - f_{21}/\det \quad (129)$$

$$e_{22} = + f_{11}/\det \quad (130)$$

$$\Delta\dot{r} = \dot{r}_D - \dot{r}_o \quad (131)$$

$$\Delta r = r_D - r_o - \dot{r}_o T_{go} \quad (132)$$

$$c_1 = e_{11} \Delta\dot{r} + e_{21} \Delta r \quad (133)$$

$$c_2 = e_{21} \Delta\dot{r} + e_{22} \Delta r \quad (134)$$

$$g_{eff} = -\mu/r^2 + v_\theta^{\ 2}/r \quad (135)$$

Observe spacecraft clock (136)

$$p_1(t) = a_o + a_1(T - t) + a_2 (T - t)^2 \quad (137)$$

$$p_2(t) = (T - t)\, p_1(t) \quad (138)$$

$$\ddot{r}(t) = c_1\, p_1(t) + c_2\, p_2(t) \quad (139)$$

$$\alpha(t) = \sin^{-1} \{ [\ddot{r}(t) - g_{eff}] / a_T(t) \} \quad (140)$$

This "program" deserves several comments.

The computation in Eqs. (133, 134) could, of course, be handled as a matrix multiplication, just as in the throttleable engine guidance law. In fact, the computation of the f_{ij}, s, the e_{ij}, s and the E matrix could be a subroutine called by either the throttleable engine guidance law or the fixed-thrust rocket guidance law.

The function $p_1(t)$ is assumed to be quadratic in time; consequently, $p_2(t)$ and $\ddot{r}(t)$ are cubic in t. Simulations have shown that there is very little to be gained by postulating an $\ddot{r}(t)$ which is a higher order polynomial.

Most of the simulations have used K = 5.

A somewhat subtle question of timing has not been explicitly faced in the "program". Equation (135), the computation of "effective gravity", should be based on the current state, $\underline{r}(t)$ and $\underline{v}(t)$, not the past state $\underline{r}(t_o)$ and $\underline{v}(t_o)$. If a great deal of time is expended in performing the computations between Eq. (113) and Eq. (135), it might be desirable to update the state vector between Eq. (134) and Eq. (135).

Another problem which has not been explicitly faced in the program is that the a_i, s which are optimum when controlling both final radius and final radial rate are not necessarily optimum when controlling final radial rate alone. It is a simple matter to insert assignment statements (between Eq. (118) and Eq. (119), for example) which select a_i, s appropriate to the problem.

Because the thrust acceleration's capability to change the vehicle's altitude becomes greatly diminished as time-to-go becomes small, and because very precise control of altitude is normally not required, it may be desirable to abandon radius control when time-to-go becomes smaller than some preset value. Abandoning radius control late in the flight usually results in more accurate control of radial rate, since all the radial coordinate control effort is then exerted to solve the radial rate problem only.

After $\alpha(t)$ is obtained, Eq. (140), a commanded thrust angle orientation which can be interpreted by the flight control system must be computed. This computation involves routine coordinate transformations and is consequently omitted.

The Determination of T_{go} for Fixed-Thrust Rockets

As explained in earlier sections, time-to-go (or equivalently T) is chosen so that the terminal horizontal speed is equal to the desired value. The differential equation for the horizontal component of velocity is

$$\dot{v}_\theta = a_T \cos\alpha - \dot{r}\, v_\theta / r \quad (141)$$

(The second expression on the right-hand side of Eq. (141) is a Coriolis-like term.) It would certainly be helpful if this differential equation were simpler, say, if it were

$$\dot{v}_\theta = a_T = v_e/(\tau - t) \quad (142)$$

It is sometimes helpful to pursue wishful thinking and solve a simpler problem; for the solution to the simpler, manageable problem may suggest a useful attack on the real problem. Suppose that Eq. (142) is integrated between the present time and the <u>correct</u> terminal time.

$$v_{\theta D} - v_{\theta o} = v_e \int_{t_o}^{T} dt/(\tau - t) \quad (143)$$

Note that the following fact has been used

$$v_\theta(T) = v_{\theta D}$$

i.e., if the correct T is chosen the terminal value of the horizontal component of velocity is equal to the desired value. Carrying out the integration on the right-hand side of Eq. (143) and performing a few manipulations yields

$$v_{\theta D} - v_{\theta o} = -v_e \ln[1 - T_{go}/(\tau - t_o)] \quad (144)$$

Solving the above equation for time-to-go yields

$$T_{go} = \tau_o \{1 - \exp[-(v_{\theta D} - v_{\theta o}) / v_e]\} \quad (145)$$

where

$$\tau_o \equiv \tau - t_o \quad (146)$$

Equation (145) is the correct formula for time-to-go when the differential equation of horizontal speed is the simple expression in Eq. (142). Except for the lamentable fact that the formula in Eq. (145) is not correct for the real problem, Eq. (141), this expression in Eq. (145) is very attractive for computing time-to-go. The quantities v_e and τ_o are yielded by the smoothing process on the reciprocated thrust acceleration measurements. The expression $v_{\theta D} - v_{\theta o}$ is simply the horizontal speed-to-be-gained; i.e., the desired horizontal speed minus the current horizontal speed. Thus, all the quantities required to evaluate the expression are readily available, and the expression itself is fairly simple.

One suspects that for the real problem, Eq. (141), the time-to-go computed from Eq. (145) would be optimistically short. Equation (142) assumes that all the thrust acceleration is allocated along the local horizontal, i.e., that $\alpha(t)$ is zero and cos α is identically one for all t. But the radial coordinate boundary-valued problem requires the allocation of some thrust acceleration along the radius vector, diverting thrust acceleration from the local horizontal. Another effect arises from the Coriolis-like term which usually diminishes the effect of the thrust acceleration that is allocated along the horizontal coordinate.

How can Eq. (145) be modified so that it is the correct expression for time-to-go? Let Eq. (141), the correct expression for the rate-of-change of horizontal speed, be rewritten as follows:

$$\dot{v}_\theta = a_T - [(1 - \cos\alpha)a_T + \dot{r}\, v_\theta / r] \quad (147)$$

Note that Eq. (147) is exactly equivalent to Eq. (141). Conceptually, however, a significant change has been wrought in rewriting Eq. (141) as Eq. (147): the rate-of-change of horizontal speed has been conceived as being equal to the thrust acceleration minus a "thrust-acceleration-loss" term. Therefore, define

$$a_L = (1 - \cos\alpha)\, a_T + \dot{r}\, v_\theta / r \quad (148)$$

where the subscript "L" denotes "Loss". Using this notation, Eq. (141) becomes

$$\dot{v}_\theta = a_T - a_L \quad (149)$$

If Eq. (149) is integrated between the current time and the desired terminal time and the resulting equation solved for time-to-go, the following equation results

$$T_{go} = \tau_o \{1 - \exp[-(v_{\theta D} - v_{\theta o} + \Delta v_{\theta L})/v_e]\} \quad (150)$$

where

$$\Delta v_{\theta L} = \int_{t_o}^{T} a_L(t)\, dt \quad (151)$$

Equation (150) is a correct expression for time-to-go. Equation (150) is not, however, as it stands, a useable expression for computing time-to-go because the evaluation of $\Delta v_{\theta L}$ requires the evaluation of the integral on the right-hand side of Eq. (151), and the direct evaluation of this integral is quite difficult. A computationally simple procedure for estimating $\Delta v_{\theta L}$ and time-to-go will not be outlined. (Note that when $\Delta v_{\theta L}$ is known, the computation of time-to-go from Eq. (150) is routine and assured). It is important to understand the concept underlying this procedure before examining the mathematical details. Figure 9 is an aid to understanding this computational procedure. Although the motivation for this procedure is primarily mathematical, the present discussion is solely concerned with gaining a physical insight into the process. Figure 9 represents an iterative loop for computing time-to-go (or, its equivalent, $\Delta v_{\theta L}$). Suppose the value of $\Delta v_{\theta L}$ is estimated. (The first estimate or guess could be zero.) This guess is used in the upper left-hand block of Fig. 9 to compute the corresponding, but probably incorrect, time-to-go. Using this time-to-go the radial coordinate control subroutine is called to compute the thrust angle regime which will satisfy the radial boundary conditions by the predicted (initially incorrect) time of thrust termination. Because the thrust-angle regime, remaining time of rocket burning, and thrust acceleration model, are all available, it is possible to predict (the upper right-hand block in Fig. 9) the terminal horizontal speed which would result. If this resulting terminal horizontal speed is not equal to the desired value, the loss in horizontal speed (due to a_L) must have been incorrectly estimated. Consequently, the computed deficiency in terminal horizontal speed should be added to the first guess of the loss in horizontal speed to obtain a better estimate of the horizontal speed loss. The general equation for computing the improved estimate of the horizontal speed loss is

$$\Delta v_{\theta L, n+1} = v_{\theta D} - v_{\theta F, n} + \Delta v_{\theta L, n} \quad (152)$$

The better estimate of horizontal speed loss is used to go through the loop again. This procedure is repeated until

$$\left| v_{\theta D} - v_{\theta F, n} \right| < \epsilon \quad (153)$$

where ϵ is equal to the tolerable guidance scheme error. (The scheme error is usually considered satisfactory if it is an order of magnitude smaller than the errors resulting from the guidance instrumentation error.) The described procedure converges to the desired level of accuracy in very few passes through the iteration loop.

Only one major new algorithm must be developed to implement this procedure: namely, the block required for predicting the terminal horizontal speed which would result from a given $T_{go,n}$ and $\alpha_n(t)$.

For various computational reasons it is desirable to predict $H_{F,n}$ rather than $v_{\theta F,n}$. The function H(t), the terminal value of which must be predicted in the scheme proposed here, is

$$H(t) = \exp\{-[v_\theta(t) - v_{\theta o}]/v_e\} \tag{154}$$

This function is nearly linear; and, consequently, lends itself to accurate and simple prediction.

Before developing the mathematical details of predicting $H(T_n)$ or $H_{F,n}$, the following paragraphs illustrate how $H(T_n)$ is used to implement the procedure diagrammed in Fig. 9. The primary mathematical motivation of this implementation is the avoidance of the evaluation of exponential functions during the iteration. Successive passes through the loop will not require the evaluation of any exponential functions, and the speed of the iteration process is correspondingly enhanced. The derivation of the algorithm requires the introduction and definition of certain quantities. Define

$$Q_n = \exp(-\Delta v_{\theta L,n}/v_e) \tag{155}$$

and note that

$$H(T_n) = H_{F,n} = \exp[-(v_{\theta F,n} - v_{\theta o})/v_e] \tag{156}$$

Because of Eq. (155), the expression for time-to-go is simply

$$T_{go,n} = \tau_o\{1 - \exp[-(v_{\theta D} - v_{\theta o})/v_e]\,Q_n \tag{157}$$

Note that the expression in Eq. (157) has the virtue of isolating the variable which will be iterated, Q, from the known exponential function of the horizontal-speed-to-be gained. The problem now is to make an algorithm which will make the following step

$$Q_{n+1} \leftarrow Q_n;$$

i.e., a procedure must be developed for performing the following step

$$Q_{n+1} = \exp(-\Delta v_{\theta L,n+1}/v_e) = \exp[-(v_{\theta D} - v_{\theta F,n} + \Delta v_{\theta L,n})/v_e] \tag{158}$$

(Remember Eq. (152), which, it was agreed, is the correct way to improve the estimate of the horizontal speed loss.) The right-hand side of Eq. (158) can be dissected (using the laws of multiplication and division of exponentials) to obtain

$$Q_{n+1} = \exp[-(v_{\theta D} - v_{\theta o})/v_e]\exp(-\Delta v_{\theta L,n}/v_e)/\exp[-(v_{\theta F,n} - v_{\theta o})/v_e] \tag{159}$$

With the previously defined symbols in Eqs. (155 and 156), Eq. (159) becomes

$$Q_{n+1} = \exp[-(v_{\theta D} - v_{\theta o})/v_e]\,Q_n/H(T_n) \tag{160}$$

The quantities Q_n and $H(T_n)$ are computed directly; they are not obtained by evaluating exponential functions of variables which are first computed. The advantage of predicting the terminal value of H(t) rather than the terminal value of the horizontal speed is thus twofold 1) H(t) is nearly linear in time ($v_\theta(t)$ is not; and, consequently, H(T) is easier to predict) 2) Obtaining H(T) directly by prediction avoids the necessity of computing an exponential function. Figure 10 illustrates the computation loop, which is equivalent to the one in Fig. 9, but uses the Q and H functions to advoid the need for the successive evaluation of exponential functions.

The algorithm for predicting the terminal value of H(t) must now be developed. The prediction of $H(T_n)$ is accomplished by simply expanding H(t) in a Taylor's series about $t = t_o$ and then evaluating the expansion for T_n

$$H(t) = H(t_o) + \dot{H}(t_o)(t - t_o) + H^{(2)}(t_o)(t - t_o)^2/2 + \ldots \tag{161}$$

$$H(T_n) = H(t_o) + \dot{H}(t_o)\,T_{go,n} + H^{(2)}(t_o)\,T_{go,n}^2/2 + \ldots \tag{162}$$

Note that

$$T_{go,n} = T_n - t_o \tag{163}$$

has been used. Because H(t) is nearly a linear function of time, the truncation of the Taylor's series results in little error. It remains, then, only to evaluate the derivatives of H(t) at $t = t_o$. From the definition of H(t) in Eq. (154), the derivatives are easily evaluated

$$\dot{H}(t) = H(t)\,[-\dot{v}_\theta(t)/v_e] \tag{164}$$

$$H^{(2)}(t) = H(t)\,[-\dot{v}_\theta(t)/v_e]^2 + H(t)\,[-v_\theta^{(2)}(t)/v_e] \tag{165}$$

$$H^{(3)}(t) = H(t)[-\dot{v}_\theta(t)/v_e]^3 + 3H(t)[-\dot{v}_\theta(t)/v_e][-\dot{v}_\theta^{(2)}(t)/v_e] + H(t)\,[-v_\theta^{(3)}(t)/v_e] \tag{166}$$

It is a routine matter to develop higher derivatives of H(t). (Note the welcome repetition of factors in the expressions for the derivatives.) These derivatives and H(t) are evaluated at $t = t_o$

$$H(t_o) = 1 \tag{167}$$

$$\dot H(t_o) = -\dot v_{\theta o}/v_e \tag{168}$$

$$H^{(2)}(t_o) = (-\dot v_{\theta o}/v_e)^2 - v_{\theta o}^{(2)}/v_e \tag{169}$$

$$H^{(3)}(t_o) = (-\dot v_{\theta o}/v_e)^3 + 3(-\dot v_{\theta o}/v_e)(v_{\theta o}^{(2)}/v_e) - v_{\theta o}^{(3)}/v_e \tag{170}$$

The derivatives of the horizontal speed must now be evaluated

$$\dot v_\theta(t) = a_T \cos\alpha(t) - \dot r v_\theta/r \tag{171}$$

$$\ddot v_\theta(t) \approx (a_T^2/v_e)\cos\alpha(t) - a_T \sin\alpha(t)\,\dot\alpha(t) - (\dot r\,\dot v_\theta + \ddot r\,v_\theta)/r \tag{172}$$

The computation of the second derivative of the horizontal speed assumed that r is constant; this is an excellent simplifying assumption since r changes by a very small percentage during a powered maneuver. This approximation is not necessary, however, and all the terms of the expansion can be computed exactly; but it would be inappropriate to include terms which are orders of magnitude smaller than the effects of thrust and guidance hardware uncertainties

$$v_\theta^{(3)} \approx 2(a_T^3/v_e^2)\cos\alpha(t) - 2(a_T^2/v_e)\sin\alpha(t)\,\dot\alpha(t) - a_T\cos\alpha(t)\,\dot\alpha^2(t) \tag{173}$$

Another reasonable approximation is to assume that

$$\ddot\alpha(t) \approx o \tag{174}$$

(It is frequently possible to omit the Coriolis-like term and its derivatives, particularly if the thrust acceleration is large compared to the Coriolis acceleration. However, the approximation has not been made here.)

Evaluating the derivatives of the horizontal speed for $t = t_o$ yields

$$\dot v_{\theta o} = a_{To}\cos\alpha_o - \dot r_o v_{\theta o}/r_o$$

$$\ddot v_{\theta o} \approx a_{To}\cos\alpha_o/\tau_o - a_{To}\sin\alpha_o\,\dot\alpha_o - (\dot r_o\,\dot v_{\theta o} + \ddot r_o\,v_{\theta o})/r_o \tag{176}$$

$$v_{\theta o}^{(3)} \approx 2a_{To}\cos\alpha_o/\tau_o^2 - 2a_{To}\sin\alpha_o\,\dot\alpha_o/\tau_o - a_{To}\cos\alpha_o\,\dot\alpha_o^2 - (2\ddot r_o\,\dot v_{\theta o} + \dot r_o\,\ddot v_{\theta o} + r_o^{(3)}v_{\theta o})/r_o \tag{177}$$

where

$$\tau_o = v_e/a_{To}$$

Only two problems remain: to compute the derivatives of $\alpha(t)$ and $\dot r(t)$. The derivatives of $\dot r$ evaluated at $t = t_o$ are

$$\ddot r_o = c_1 p_1(t_o) + c_2 p_2(t_o) \tag{178a}$$

$$r_o^{(3)} = c_1 \dot p_1(t_o) + c_2 \dot p_2(t_o) \tag{178b}$$

where

$$\dot p_1(t_o) = -a_1 - 2a_2 T_{go} \tag{179}$$

$$\dot p_2(t_o) = -a_o - 2a_1 T_{go} - 3a_2 T_{go}^2 \tag{180}$$

(Equations (137 and 138) were used for the definitions of $p_1(t)$ and $p_2(t)$.) Note that

$$T_{go} = T - t_o \tag{181}$$

was used in Eqs. (179, 180). Differentiating each side of

$$a_T \sin\alpha(t) = \ddot r(t) - g_{eff}$$

yields

$$(a_T^2/v_e)\sin\alpha(t) + a_T\cos\alpha(t)\,\dot\alpha(t) = r^{(3)}(t) - \dot g_{eff} \tag{182}$$

Solving Eq. (182) for $\dot\alpha(t)$ and evaluating the expression for $t = t_o$ yields

$$\dot\alpha_o = [r_o^{(3)} - \dot g_{eff}(t_o) - a_{To}\sin\alpha_o/\tau_o]/a_{To}\cos\alpha_o \tag{183}$$

Finally, evaluating the derivative of "g effective",

$$g_{eff}(t) = -\mu/r^2 + v_\theta^2/r \tag{184}$$

$$\dot g_{eff}(t_o) = +2\mu\,\dot r_o/r_o^3 + 2v_{\theta o}\,\dot v_{\theta o}/r_o - v_{\theta o}^2\,\dot r_o/r_o^2 \tag{185}$$

or, rearranging and simplifying

$$\dot g_{eff}(t_o) = \{2v_{\theta o}\,\dot v_{\theta o} - \dot r_o[g_{eff}(t_o) + g_o]\}/r_o \tag{186}$$

where

$$g_o = -\mu/r_o^2 \tag{187}$$

The equations developed here for predicting time-to-go are probably more accurate than most situations require. If the expansion for $H(t)$ were truncated to the first-order term, there could be considerable error in predicted time-to-go when time-to-go is large; but the error for small time-to-go would be acceptable and the terminal error in the boundary conditions would be tolerable. Why, then, construct a more accurate, more complicated time-to-go predictor? Although an accurate time-to-go prediction is not necessary for the attainment of the specified boundary conditions, it might be useful as a basis for monitoring and decision-making. For example, a precision time-to-go prediction early in a powered-flight maneuver could warn an astronaut that he is coming dangerously close to depleting his estimated fuel supply. Another possible use of the accurate long-range time-to-go prediction is in the optimization of engine ignition time. For example, imagine a spacecraft approaching a planet on a high-energy trajectory. Suppose the mission objective is insertion into a reconnaissance orbit around the planet. By computer or human monitoring of the predicted time-to-go for the maneuver, the engine ignition time could be chosen when the time required for the orbital insertion passes through a minimum.

Programming the Fixed-Thrust Rocket Guidance Law

The derivation of guidance equations does not always clarify exactly what sequence of computational steps is performed in the guidance computer. This section attempts a clarification by explicitly listing the order of computation and some of the logical branches which occur in the guidance program. No attempt has been made to "optimize" the "program" because the optimization would be heavily dependent on the particular hardware and software involved. If the reader has perused the previous sections of this paper, the following "program" should be fairly self-explanatory. Equations (193, 194) merit some special comment however. The function of these equations is to produce a good estimate of Q_n at time t_o given that a good estimate of T_{go} (or T) had been established at some earlier time. Suppose that a good estimate of T has been obtained and that at some subsequent instant t_o the guidance loop is to be closed again and a new T_{go} and $\alpha(t)$ is to be computed. A reasonable estimate of time-to-go should be obtainable from Eq. (193). Equation (196) can be solved for Q_n so that given P, τ_o and the reasonable estimate of $T_{go,n}$, a reasonable estimate of Q_n - Eq. (194) - can be obtained. After the first entry to the program and the first accurate estimate of T, Eqs. (193, 194) function very effectively upon each subsequent entry to the program to obtain current good estimates of $T_{go,n}$ and Q_n. Consequently, the criterion for accuracy, Eq. (213), is normally satisfied as soon as it is encountered after a program entry. Thus, on any entry to the program subsequent to the first, the time-to-go prediction requires only a single pass through Eqs. (199 to 215). [A second pass using the refined time-to-go prediction is then made through the radial coordinate control block to obtain a very accurate $\alpha(t)$.] When the program is entered for the first time, there is no previous estimate of T available. In this event Q_n is set to 1 and Eq. (194, 194) are bypassed. The variable j, which is 0 to 1, is set to zero on entering the program and set to 1 upon satisfaction of the criterion of precision, Eq. (213). The variable j is tested after each pass through the Radial Coordinate Control subroutine. When j is found equal to 1, the program branches to the computations which convert $\alpha(t)$ to appropriate spacecraft commands. The sample "program" follows.

$j = 0$	(188)
Do state vector update and observe clock: $t = t_o$	(189)
Do τ_o, v_e, and a_{To} estimation	(190)
$P = \exp[-(v_{\theta D} - v_{\theta o})/v_e]$	(191)
If this is 1st entry to program, $Q_n = 1$, go to (196)	(192)
$T_{go,n} = T - t_o$	(193)
$Q_n = (1 - T_{go,n}/\tau_o)/P$	(194)
Go to (197)	(195)
$T_{go,n} = \tau_o(1 - PQ_n)$	(196)
Do Radial Coordinate Control Subroutine	(197)
If $j = 1$, $T = t_o + T_{go,n}$, go to angle transformation subroutine	(198)
$\dot{v}_{\theta o}$ = Eq. (175)	(199)
$\dot{g}_{eff}(t_o)$ = Eq. (186)	(200)
$\ddot{r}_o$ = Eq. (177)	(201)
$\dot{p}_1(t_o)$ = Eq. (179)	(202)
$\dot{p}_2(t_o)$ = Eq. (180)	(203)
$r_o^{(3)}$ = Eq. (178)	(204)
$\dot{\alpha}_o$ = Eq.(184)	(205)
$\ddot{v}_{\theta o}$ = Eq. (176)	(206)
$v_{\theta o}^{(3)}$ = Eq. (175)	(207)
$\dot{H}(t_o) = 1$	(208)
$\dot{H}(t_o)$ = Eq. (168)	(209)
$\ddot{H}(t_o)$ = Eq. (169)	(210)
$H^{(3)}(t_o)$ = Eq. (170)	(211)
$H_{F,n}$ = Eq. (162) (Note that $H(T_n) = H_{F,n}$)	(212)

If $|P - H_{F,n}| < \epsilon_o$, $j = 1$ (213)

$Q_{n+1} = P\, Q_n / H_{F,n}$ (214)

$Q_n \leftarrow Q_{n+1}$ (215)

Go to (196) (216)

Note that the bulk of this program - Eqs. (199 to 212) - is concerned with predicting the terminal value of H(t). Under many circumstances the rather high accuracy of these equations is inappropriate and the computations can be considerably simplified and shortened.

Guidance Law for Injection Onto a Specified Conic Trajectory

Because of the great utility of the concept of conic trajectories for spaceflight guidance and navigation, this section deals with guidance of a rocket-powered spacecraft onto a specified conic. It is assumed that it is the objective of guidance to achieve a conic of specified specific energy, E, and specific angular momentum, h; but that the location of the line of apsides is not explicitly specified. These specifications do not normally require the explicit control of the burnout altitude of the vehicle. The specification of E and h is equivalent to specifying the semi-major axis and eccentricity of the achieved orbit; this fixes the size and shape of the orbit. By specifying E and h, the periapsis altitude is fixed, assuring (except for instrumentation errors) the minimum altitude of the spacecraft's orbit.

The guidance law for achieving specified final values of v_θ and $\dot{r}$ (as well as r, when required) has already been constructed. (It is assumed that the rocket's thrust is constant). It is necessary simply to add a few equations for determining $v_{\theta D}$ and $\dot{r}_D$. The equations for $v_\theta(T)$ and $\dot{r}(T)$ are derived from the well-known expressions for the specific energy and specific angular momentum of a particle in a central force field.

$$E = v^2/2 - \mu/r \tag{217}$$

$$h = r\, v_\theta \tag{218}$$

Adapting these equations to the present purpose yields

$$E_D = v_{\theta D}^2/2 + \dot{r}_D^2/2 - \mu/r_{BO} \tag{219}$$

$$h_D = r_{BO}\, v_{\theta D} \tag{220}$$

Note that

$$\underline{v}_D = \underline{v}_D(r_{BO}) \tag{221}$$

which is a simplification of the more general case

$$\underline{v}_D = \underline{v}_D(r_{BO}, T) \tag{222}$$

Solving Eqs. (219, 220) for $v_{\theta D}$ and $\dot{r}_D$ yields

$$v_{\theta D} = h_D/r_{BO} \tag{223}$$

$$\dot{r}_D = \sqrt{2 E_D - (h_D/r_{BO})^2 + 2\mu/r_{BO}} \tag{224}$$

Note that the positive square root has been chosen for $\dot{r}_D$ in Eq. (224). This choice implicitly assumes that the spacecraft ascends into orbit. The negative square root should be used if the spacecraft descends into orbit.

Consideration must be given to the possibility of the argument of the square root in Eq. (224) being negative. The zeroes of the expression under the radical are the periapsis and apoapsis radii. The radial rate is zero at these radii because the fall of the body toward the central field source is arrested at the periapsis radius and the body begins to fall outward. The body falls outward until it reaches its apoapsis radius and then begins to fall inward again. These two turning points of the trajectory are

$$r_{peri} = -(\mu - \sqrt{\mu^2 + 2 E_D h_D^2})\,/\,2 E_D \tag{225}$$

$$r_{apo} = -(\mu + \sqrt{\mu^2 + 2 E_D h_D^2})\,/\,2 E_D \tag{226}$$

(Note that neither Eq. (225) nor Eq. (226) is valid for a parabola ($E_D = 0$).

$$\left.\begin{aligned} r_{peri} &= \mu\, h_D^2/2 \\ r_{apo} &= \infty \end{aligned}\right\}\ \text{parabola}$$

Equation (225) is valid for a hyperbola; but Eq. (226) is meaningless). An obvious requirement is that

$$r_{peri} \le r_{BO} \le r_{apo} \tag{227}$$

If r_{BO} violates Eq. (227), the radicand in Eq. (224) is negative and $\dot{r}_D$ is imaginary! If the unconstrained burnout radius is greater than r_{apo} or less than r_{peri}, it is impossible, in the unconstrained radius mode, to insert the spacecraft onto the specified conic. The policy is to switch to the constrained radius mode whenever predicted r_{BO} violates Eq. (227). If r_{BO} exceeds r_{apo} it is reasonable to require that

$$\left.\begin{aligned} r_D &= r_{apo} \\ \dot{r}_D &= 0 \\ v_{\theta D} &= h_D/r_{apo} \end{aligned}\right\} \quad \text{If } r_{BO} > r_{apo} \tag{228}$$

On the other hand

$$\left.\begin{aligned} r_D &= r_{peri} \\ \dot{r}_D &= 0 \\ v_{\theta D} &= h_D/r_{peri} \end{aligned}\right\} \quad \text{If } r_{BO} < r_{peri} \tag{229}$$

The policy, in summary, is to insert onto the conic at an unspecified radius (if this is possible); if the unconstrained burnout radius violates Eq. (227), the policy is to control the burnout radius and enter the trajectory at the apsidal distance closer to the unconstrained burnout radius, r_{BO}. This policy, which does not control r(T) unless necessary, tends to result in a minimum ΔV expenditure.

Figure 11 is a block diagram of the guidance law for achieving a specified conic trajectory (of unspecified orientation of the apsidal line.) Initially, r_{BO} is not known, of course. To start the procedure, r_o is used for r_{BO}, i.e., the present radius is used for the predicted burnout radius. Letting r_{BO} tentatively equal r_o allows the calculation of $v_{\theta D}$ and r_D from Eqs. (223 and 224). Using these values for the desired horizontal and radial rates, a solution $\alpha(t)$ and T_{go} are computed, and r_{BO} is predicted. This r_{BO} is used to recompute $v_{\theta D}$ and r_D and the process is repeated. After a few consecutive cycles in this manner, an accurate value of r_{BO} is obtained. (The procedure of guessing that $\underline{r}_{BO}$ is equal to $\underline{r}_o$ in order to start a iterative loop for obtaining $\underline{r}_{BO}$ accurately, is very important. It is used for other, more complex boundary-valued problems discussed later.) If r_o is presently greater than the apoapsis radius or less than the periapsis radius, the nearest apsidal radius could be used for the first estimate of r_{BO}.

Guidance Law for Insertion into a Circular Orbit of Unspecified Altitude

The E Guidance method can be used to insert a rocket-propelled vehicle into a circular orbit of specified altitude. There are occasions when operational requirements and considerations make it appropriate to guide into a circular orbit of "natural" altitude. For example, the "natural" altitude circular orbit generally requires less fuel expenditure than the constrained-altitude circular orbit. The method of E Guidance can guide into a circular orbit of unconstrained altitude. This is quite evident in the previous section where guidance onto a general conic was discussed; it is brought up here for specific discussion because of the great utility of circular orbits and the simplifications which result when the objective of guidance is to achieve "any circular orbit".

The components of the burnout velocity vector are very simple

$$\dot{r}_D = 0 \tag{230}$$

$$v_{\theta D} = \sqrt{\mu/r_{BO}} \tag{231}$$

The radial coordinate control equations are operated in the unconstrained radius mode, of course. To begin the computations r_o is used for r_{BO} and few cycles of iteration establish an accurate value of r_{BO}. Figure 12 is a block diagram of the system.

Guidance Law for Injection onto an Intercept Trajectory

An extremely important guidance problem is injection onto a trajectory which passes through a specified point in space. This is the intercept problem. If the objective is to intercept a moving body such as the moon, a point on the surface of the terrestrial globe, or an orbiting spacecraft, then the intercepting vehicle must pass through the given point at a specified time. ("Moving", here, means moving in the inertial coordinate system in which the problem is being solved.) The guidance scheme must synchonize the arrival of the intercepting vehicle at the given point with the arrival of the target body at this point. It is assumed that the trajectory of the target body is known and that some point of intercept and time of intercept has been chosen. This section of the paper presents a guidance scheme for injecting the intercepting vehicle onto a trajectory that coasts through the given point at the appointed time.

The solution to this problem has many applications in space guidance and navigation. For example, if the problem is to inject a spacecraft onto a trajectory which freefalls from the earth to the moon, the guidance law presented here is applicable. The guidance law is also applicable to ascent powered guidance whose objective is the establishment of a coasting trajectory which intercepts another spacecraft with which rendezvous is desired.

Figure 13 illustrates the intercept-trajectory geometry and time-scale, and introduces some of the terminology which is used to discuss this problem. The outside orbit (which is drawn closed) is the trajectory of the target body. The inside trajectory is composed of two parts: a portion of the powered arc, shown as a solid bold line; and a coasting arc, shown as a dotted line terminating at point C. Point C in the intercept point, where target body and intercepting vehicle meet. It is assumed that point C and the time at which the target body will arrive at C are known. The objective of powered-flight guidance is to acquire the velocity vector required to coast through C at the right instant. When this velocity vector, denoted by $\underline{v}_R$, is achieved, the thrust is terminated and the freefall from B to C begins. Therefore, the powered maneuver, a portion of which is shown as the bold line between A and B, is conducted to achieve $\underline{v}_R$. The vector $\underline{v}_R$ is sometimes called the Lambert velocity vector, after one of the first men who investigated the properties of trajectories which coast from a given point B to another given point C in a specified duration of freefall, T_{ff}. The Lambert velocity vector $\underline{v}_R$ is a function of B, C, and T_{ff}.

$$\underline{v}_R = \underline{v}_R (B, C, T_{ff}) \tag{232}$$

or, using the vectors defined in Fig. 13

$$\underline{v}_R = \underline{v}_R (\underline{r}_{BO}, \underline{r}_{INT}, T_{ff}) \tag{233}$$

Vector $\underline{r}_{BO}$ is the position vector of the spacecraft at the instant of thrust termination ("burnout"). The space required to derive an expression for $\underline{v}_R$ is not available here. Several mathematical procedures for obtaining $\underline{v}_R$ are given in reference 1.

Time is a very important parameter in the intercept problem. The arrival of the spacecraft at the intercept - point position vector $\underline{r}_{INT}$ must be synchronized with the arrival of the target body there. The time of engine ignition for the powered maneuver may be very critical because the fuel required to establish $\underline{v}_R$ is usually very sensitive to the timing of the maneuver. There is a best time of engine ignition for the maneuver. A delay of engine ignition can be compensated by the guidance system; but a fuel penalty is incurred because the best or "nominal" time of engine ignition minimizes the fuel required. It is convenient to locate the origin of the time coordinate axis at the instant of nominal engine ignition time. Figure 13 illustrates this time coordinate axis. On this axis T_{INT} is the time at which the spacecraft must arrive at $\underline{r}_{INT}$; i.e., T_{INT} is the time-of-intercept. T_{BO} is the time-of-burnout; and t is any instant during the powered flight. From Figure 13, the following equalities can be written

$$T_{ff} = T_{INT} - T_{BO} \tag{234}$$

But

$$T_{BO} = t + T_{go} \tag{235}$$

Consequently

$$T_{ff} = T_{INT} - (t + T_{go}) \tag{236}$$

The quantity $(t + T_{go})$ is the sum of engine ignition delay (if any) and the time of powered flight; but this quantity is computed as the sum of the time elapsed since the nominal time of engine ignition and the remaining time till burnout. Substituting Eq. (236) into Eq. (233) yields

$$\underline{v}_R = \underline{v}_R \left[\underline{r}_{BO}, \underline{r}_{INT}, T_{INT} - (t + T_{go})\right] \tag{237}$$

The intercept position vector and time, $\underline{r}_{INT}$ and T_{INT}, are known; the current time, t, is observed on the spacecraft clock; and time-to-go, T_{go}, is predicted by the fixed-thrust rocket guidance law. If $\underline{r}_{BO}$ could be predicted, $\underline{v}_R$ could be computed from Eq. (237). The magnitude of the burnout position vector $\underline{r}_{BO}$ is predicted by the fixed-thrust rocket guidance law in the unconstrained radius mode; and $|\underline{r}_{BO}|$ is constrained to be equal to r_D in the constrained radius mode. Consequently, only θ_{go}, the remaining central angle to be traveled during powered flight, must be predicted in addition in order to obtain $\underline{r}_{BO}$. A block diagram of the guidance system for attaining $\underline{v}_R$ is shown in Fig. 14. Since $\underline{r}_{BO}$ is not initially known, the current position vector $\underline{r}_o$ is used for $\underline{r}_{BO}$ on the first entry to the program, and several cycles through the computation loop produces an accurate estimate of the burnout position vector. The block diagram is otherwise self-explanatory.

The contents of the block labeled "Burnout Angle Predictor" are now described. The prediction of θ_{BO} is based on a truncated Taylor's series expansion of $\theta(t)$ about $t = t_o$.

$$\theta(t) = \theta(t_o) + \dot{\theta}(t_o)(t - t_o) + \ddot{\theta}(t_o)(t - t_o)^2/2 + \ldots \tag{238}$$

Evaluating Eq. (238) for t = T yields

$$\theta_{BO} = \theta(T) = \theta(t_o) + \dot{\theta}(t_o) T_{go} + \ddot{\theta}(t_o) T_{go}^2/2 + \ldots \tag{239}$$

Because

$$\dot{\theta} = v_\theta / r \tag{240}$$

the derivatives of $\theta(t)$ are expressible in terms of the derivatives of v_θ and r. These derivatives are given below, evaluated at $t = t_o$.

$$\dot{\theta}_o = v_{\theta o}/r_o \tag{241}$$

$$\ddot{\theta}_o = \dot{v}_{\theta o}/r_o - \dot{\theta}_0 (\dot{r}_o/r_o) \tag{242}$$

$$\theta_o^{(3)} = \ddot{v}_{\theta o}/r_o - 2\ddot{\theta}_o(\dot{r}_o/r_o) - \dot{\theta}_o(\ddot{r}_o/r_o) \tag{243}$$

The expansion could easily be extended to higher powers in T_{go} for additional accuracy, if desired. Using the derivatives in Eq. (239) yields.

$$\theta_{BO} \approx \theta_o + \dot{\theta}_o T_{go} + \ddot{\theta}_o T_{go}^2/2 + \theta_o^{(3)} T_{go}^3/6 \tag{244}$$

The Plane Control Guidance Law

For many missions it is necessary to control the plane of the trajectory achieved by the powered-flight maneuver. The E Guidance formulation is a natural and convenient way to determine the allocation of thrust acceleration required for the plane control.

Figures 15 and 16 illustrate the plane control boundary-valued problem. Figure 15 shows the desired orbital plane passing through the center of the gravitational body around which the powered maneuver is performed. A coordinate axis, Y, which is normal to the desired plane, is used to measure the deviation of the spacecraft from the plane. (The X and Z axes lie in the desired plane.) If $\underline{y}_1$ is a unit vector along the Y axis, the deviation of the spacecraft from the plane and its velocity normal to the plane are simply

$$y = \underline{r} \cdot \underline{y}_1 \qquad (245)$$

$$\dot{y} = \underline{v} \cdot \underline{y}_1 \qquad (246)$$

By the terminal time T the vehicle's displacement from the plane and component of velocity normal to the plane must be zero. Hence

$$y_D = 0 \qquad (247)$$

$$\dot{y}_D = 0 \qquad (248)$$

The differential equation for y is

$$\ddot{y} = a_{Ty} + \underline{g} \cdot \underline{y}_1 \qquad (249)$$

where a_{Ty} is the component of thrust acceleration along the Y axis and $\underline{g} \cdot \underline{y}_1$ is the component of gravitational acceleration along the Y axis. Equations (48-50) are easily adapted to this problem.

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} -\dot{y}_o \\ -(y_o + \dot{y}_o T_{go}) \end{bmatrix} \qquad (250)$$

$$\ddot{y} = c_1 \, p_1(t) + c_2 \, p_2(t) \qquad (251)$$

Therefore, a solution allocation of thrust acceleration is

$$a_{Ty} = c_1 \, p_1(t) + c_2 \, p_2(t) - \underline{g} \cdot \underline{y}_1 \qquad (252)$$

A very efficient choice of functions for $p_1(t)$ and $p_2(t)$ is given in Appendix A.

The very same E matrix can be used for the plane-control problem and the radial coordinate control problem.

Note that the allocation of thrust acceleration along the Y axis will, in general, increase the "acceleration lost" and increase the time-to-go required to achieve the desired horizontal speed. If the plane change required is large, the effect of this "acceleration lost" must be included in the time-to-go predictor.

Controlling the Spacecraft's Final Attitude

There are sometimes operational requirements for controlling the spacecraft's final attitude orientation. One example is a lunar landing: the spacecraft must be in a vertical or erect orientation as it touches down upon the lunar surface. An interesting example of how final attitude control can be of benefit when minimizing fuel expenditure is escape from a planet along a specified asymptote or to a specified distant intercept point. The variational calculus proves * that the thrust acceleration vector and the velocity vector should be aligned at burnout for optimum fuel performance. Since the instantaneous rate of increase of energy of the spacecraft is given by

$$dE/dt = \underline{a}_T \cdot \underline{V} \qquad (253)$$

it is intuitively plausible that at the terminus of the powered maneuver, when both V and a_T are largest, the greatest benefit will be derived from collinearity of the thrust acceleration and velocity vectors.

This section of the paper illustrates how the final thrust acceleration vector is controlled for the throttleable engine guidance law. The principle of controlling the final attitude of the vehicle is clearly illustrated and can be extended by the reader to the fixed-thrust rocket guidance law.

Consider the control of one axis of the three axis problem. The X axis differential equation is

$$\ddot{x} = a_{Tx} + g_x \qquad (254)$$

It is desired to control the terminal values of $x(t)$, $\dot{x}(t)$, and $a_{Tx}(t)$.

$$\dot{x}(T) = \dot{x}_D \qquad (255)$$

$$x(T) = x_D \qquad (256)$$

$$a_{Tx}(T) = a_{Tx,D} \qquad (257)$$

There are three equations of constraint which $\ddot{x}(t)$ must satisfy.

* Reference 3.

$$\dot{x}_D - \dot{x}_o = \int_{t_o}^{T} \ddot{x}(t)\,dt \tag{258}$$

$$x_D - x_o - \dot{x}_o T_{go} = \int_{t_o}^{T} \left[\int_{t_o}^{t} \ddot{x}(s)\,ds \right] dt \tag{259}$$

$$\ddot{x}(T) = a_{Tx,D} + g_x(T) \tag{260}$$

If a central force gravitational field is assumed

$$g_x(T) = -\mu x_D / r_D^3 \tag{261}$$

Since there are three equations of constraint for $\ddot{x}(t)$, a three degree-of-freedom $\ddot{x}(t)$ is appropriate

$$\ddot{x}(t) = c_1 + c_2(T-t) + c_3(T-t)^2 \tag{262}$$

where c_1, c_2, and c_3 must be chosen to satisfy Eqs. (258, 259, and 260). Substituting the three degree-of-freedom definition of $\ddot{x}(t)$ into Eqs. (258, 259, and 260) leads to three simultaneous linear algebraic equations in the unknowns c_1, c_2, and c_3

$$\dot{x}_D - \dot{x}_o = c_1 T_{go} + c_2 (T_{go}^2/2) + c_3 (T_{go}^3/3) \tag{263}$$

$$x_D - x_o - \dot{x}_o T_{go} = c_1 (T_{go}^2/2) + c_2 (T_{go}^3/3) + c_3 (T_{go}^4/4) \tag{264}$$

$$a_{Tx,D} + g_x(T) = c_1 \tag{265}$$

The solution for the c_i,s is

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 18/T_{go}^2 & -24/T_{go}^3 & -6/T_{go} \\ -24/T_{go}^3 & 36/T_{go}^4 & 6/T_{go}^2 \end{bmatrix} \begin{bmatrix} \dot{x}_D - \dot{x}_o \\ x_D - x_o - \dot{x}_o T_{go} \\ a_{Tx,D} + g_x(T) \end{bmatrix} \tag{266}$$

and the solution allocation of thrust acceleration along the X axis is

$$a_{Tx}(t) = c_1 + c_2(T-t) + c_3(T-t)^2 - g_x(t). \tag{267}$$

Similar equations, using the same E matrix, can be written for the y and z allocation of thrust acceleration.

Note that these equations allow complete specification of the terminal thrust-acceleration vector.

Appendix A

Analytic Determination of a_i, s in $p_1(t)$ and $p_2(t)$

In the section of this paper which is entitled "Optimization of the Guidance Law", $p_1(t)$ and $p_2(t)$ were defined as

$$p_1(t) = a_o + a_1(T-t) + a_2(T-t)^2 + \cdots + a_n(T-t)^n \tag{A1}$$

$$p_2(t) = p_1(t)(T-t) \tag{A2}$$

The a_i, s are to be chosen to optimize the fuel expended. Any choice of a_i, s is admissible and leads to a solution to the boundary-valued problem, providing that at least one of the a_i, s is non-zero. This appendix describes a choice of a_i, s which has produced remarkably efficient results, often leading to more economical fuel expenditures than a steepest descent numerical optimization program.

The determination of the a_i, s developed here was originally motivated by knowledge of the calculus of variations solution to the plane-control steering problem. The differential equation of motion normal to the desired plane is

$$\ddot{y} = a_T \sin \alpha_Y + \underline{g} \cdot \underline{y}_1 \tag{A3}$$

(See the section entitled "The Plane-Control Guidance Law" and Fig. 16.) Because $\underline{g} \cdot \underline{y}_1$ is very small, Eq. (A3) is approximately

$$\ddot{y} \approx a_T \sin \alpha_Y \tag{A4}$$

and the boundary conditions are

$$y_o = y(t_o) \tag{A5}$$

$$\dot{y}_o = \dot{y}(t_o) \tag{A6}$$

$$y(T) = y_D = 0 \tag{A7}$$

$$\dot{y}(T) = \dot{y}_D = 0 \tag{A8}$$

The terminal time T is fixed. (T is determined to satisfy the horizontal speed constraint.) The variational calculus solution is

$$\tan \alpha_Y(t) = A + Bt \tag{A9}$$

where A and B must be chosed to satisfy Eqs. (A5 - A8). For moderate α_Y, since $\sin\alpha_Y$ and $\tan \alpha_Y$ are nearly equal for small α_Y, the calculus of variations prescription can be very closely approximated by

$$\sin \alpha_Y(t) = C + Dt - \underline{g}\cdot\underline{y}_1/a_T \tag{A10}$$

(Remember that $\underline{g}\cdot\underline{y}_1$ is small; therefore $\underline{g}\cdot\underline{y}_1/a_T$ is even smaller.) The motivation for defining $\alpha_Y(t)$ as in Eq. (A10) rather than as in Eq. (A9) is that the determination of the C and D which satisfy Eqs. (A5 - A8) is very simple and direct whereas the determination of the A and B which satisfy Eqs. (A5 - A8) is very involved and lengthy. To see how the evaluation of C and D proceeds, substitute $\sin \alpha_Y(t)$ as defined in Eq. (A10) into Eq. (A3).

$$\ddot{y} = Ca_T + Da_T t \tag{A11}$$

Thus the optimum, or nearly optimum, $\ddot{y}(t)$, is a linear combination of $a_T(t)$ and $a_T(t)\,t$. In the notation introduced previously,

$$\ddot{y} = c_1 p_1(t) + c_2 p_2(t) \tag{A12}$$

is equivalent to Eq. (A11) if

$$p_1(t) = a_T(t) \tag{A13}$$

$$p_2(t) = (T-t)\,a_T(t) \tag{A14}$$

That is, Eqs. (A12 - A14), also define $\ddot{y}(t)$ as a linear combination of $a_T(t)$ and $a_T(t)\,t$.

Now the a_i, s can be evaluated. Consider the Taylor's Series expansion of $a_T(t)$ about $t = T$.

$$a_T(t) = a_T(T) + \dot{a}_T(T)(t-T) + \ddot{a}_T(T)(t-T)^2/2 + \cdots \tag{A15}$$

Since

$$p_1(t) = a_o + a_1(T-t) + a_2(T-t)^2 + \cdots \tag{A16}$$

Therefore

$$a_o = a_T(T) = V_e/(\tau - T) \tag{A17}$$

$$a_1 = -\dot{a}_T(T) = -a_o^2/V_e \tag{A18}$$

$$a_2 = \ddot{a}_T(T)\ 2 = a_o^3/V_e^2 \tag{A19}$$

$$a_3 = -a_T^{(3)}(T)\ 6 = -a_o^4/V_e^3 \tag{A20}$$

etc.

It is not necessary to carry the expansion very far because the terms decrease rapidly in magnitude. Note that the solution to the boundary-valued problem remains exact; the prescription of the variational calculus is not followed exactly; but the minima in these problems are very broad, in general, and a negligible amount of fuel is sacrificed.

Appendix B

Implicit Control of Altitude in $\underline{V}_g$ - Nulling Method

While writing the first section of this paper, the author ignorantly slighted successful efforts afoot to give a twist to the $\underline{V}_g$ - nulling method which would permit implicit control of burnout altitude. Mr. Fred H. Martin, Research Assistant, MIT Instrumentation Laboratory, has performed these investigations pursuant to the germ of an idea conceived by Dr. Richard H. Battin and Dr. J. Halcombe Laning, Jr., Deputy Associate Directors, MIT Instrumentation Laboratory. The procedure which is utilized to gain implicit control of altitude is interesting and sheds light on the powered-flight guidance problem.

There are two thrust orientation policies which have particular utility in the $\underline{v}_g$-nulling method. One is to point $\underline{a}_T$ along $\underline{v}_g$. Obviously, this policy should reduce $\underline{v}_g$ since the effect of the policy is to gain speed in the direction of $\underline{v}_g$. The concise mathematical statement of this policy is

$$\underline{a}_T \times \underline{v}_g = \underline{0} \tag{B1}$$

The other policy is to so choose the thrust orientation that $\underline{v}_g$ shrinks without rotating; the mathematical statement of this policy is

$$-\underline{v}_g \times \dot{\underline{v}}_g = \underline{0} \tag{B2}$$

where

$$\dot{\underline{v}}_g = \dot{\underline{v}}_r - (\underline{a}_T + \underline{g}) \tag{B3}$$

because

$$\underline{v}_g = \underline{v}_r - \underline{v} \tag{B4}$$

and

$$\dot{\underline{v}} = \underline{a}_T + \underline{g} \tag{B5}$$

Consequently, Eq. (B2) becomes

$$-\underline{v}_g \times \left[\dot{\underline{v}}_r - (\underline{a}_T + \underline{g})\right] = \underline{0} \tag{B6}$$

Suppose $\underline{a}_T$ is chosen to satisfy a linear combination of (B1) and (B6)

$$a_o(\underline{a}_T \times \underline{v}_g) - a_1 \underline{v}_g \times \left[\dot{\underline{v}}_r - (\underline{a}_T + \underline{g})\right] = \underline{0} \tag{B7}$$

Both the first policy, Eq. (B1), and the second policy, Eq. (B6), tend to shrink $\underline{v}_g$. Therefore, the linear combination of the two policies, Eq. (B7), should shrink $\underline{v}_g$ for an interval of values of a_o and a_1. The original motivation behind combining the two policies was to minimize fuel expenditure by selecting the most effective linear combination of the two policies; i. e., a_o and a_1 were to be varied in pre-flight simulations to minimize the fuel required for a maneuver. Mr. Martin noted that the values of a_o and a_1 affected the final position of the spacecraft as well as the fuel expended in attaining $\underline{v}_r$. The effect on altitude of varying a_o and a_1 was definite but analytically unpredictable. Nevertheless, in pre-flight simulations based on the nominal trajectory, one could so select a_o and a_1 that burnout altitude could be controlled.

The two numbers, a_o and a_1, are, of course, not really independent since Eq. (B7) can be divided through by whichever number is not zero. The linear combination of the two policies can be expressed as

$$c(\underline{a}_T \times \underline{v}_g) - (1 - c)\underline{v}_g \times \left[\dot{\underline{v}}_r - (\underline{a}_T + \underline{g})\right] = \underline{0} \tag{B8}$$

which emphasizes that there is only one degree-of-freedom. If c is 1, the first policy is selected; if c is 0, the second policy is selected; for other values of c both policies play a role. The quantity c can be selected to find the linear combination of the two policies which minimizes fuel expended. (The minimization is with respect to this steering policy and <u>not</u> with respect to all possible steering policies.) Or, the quantity c can be selected to determine a thrust angle policy which results in the vehicle attaining a specified altitude as well as a $\underline{v}_r$. Evidently, one course of action or the other must be chosen. (If c is a function of time, more degrees of freedom are available, however.)

It is illuminating to compare this method of exercising altitude control with the method proposed in this paper. It is suggested that the reader will profit more from the following discussion after reading the main body of this paper.

Suppose that the following boundary-valued problem is to be solved by the method in this paper.

$$\dot{r}(T) = \dot{r}_D \tag{B9}$$

$$v_\theta(T) = v_{\theta D} \tag{B10}$$

where T is the terminal time. Let

$$\ddot{r}(T) = c_1 p_1(t) \tag{B11}$$

where c_1 will be chosen to satisfy Eq. (B9) and $p_1(t)$ is a pre-specified function of time. (T is chosen to satisfy Eq. (B10).) Since

$$\dot{r}(T) - \dot{r}(t_o) = \int_{t_o}^{T} \ddot{r}(t)\,dt \tag{B12}$$

the equation of constraint which c_1 must satisfy is

$$\dot{r}_D - \dot{r}(t_o) = c_1 \int_{t_o}^{T} p_1(t)\,dt \tag{B13}$$

Let $p_1(t)$ be defined as follows

$$p_1(t) = a_o + a_1(T - t) \tag{B14}$$

where a_o and a_1 are arbitrary constants, the purpose of which will be revealed later. Then

$$c_1 = \left[\dot{r}_D - \dot{r}(t_o)\right] / (a_o T_{go} + a_1 T_{go}^2 / 2) \tag{B15}$$

and, because

$$\ddot{r}(t) = -\mu / r^2 + v_\theta^2 / r + a_T \sin\alpha \tag{B16}$$

(α is defined in Fig. 7)

a solution allocation of thrust acceleration along the radius vector is

$$a_T \sin\alpha = -(-\mu / r^2 + v_\theta^2 / r) + c_1 \left[a_o + a_1(T - t)\right] \tag{B17}$$

Equation (B17) satisfies the radial-rate constraint for any T and T_{go}, where

$$T_{go} = T - t_o \tag{B18}$$

For some particular T, Eq. (B10), the horizontal speed constraint, is also satisfied. For this correct value of T, Eqs. (B15 and B17) represent the solution to the problem of attaining the velocity vector which is specified in Eqs. (B9 and B10).

Now, note the following fact: a_o and a_1 can be varied in Eqs. (B15, B17) to minimize fuel expenditure with respect to this steering low--or, they can be varied to control r (T), (which is equivalent to the altitude). Thus, a_o and a_1 for this steering law can perform functions similar to those performed by a_o and a_1 in the $\underline{v}_g$ - nulling steering law, which was expressed in Eq. (B7). Therefore, during pre-flight simulations, a_o and a_1 could be selected (on the basis of the nominal trajectory) to gain implicit control of r (T). But, actually, the method of E Guidance can control the burnout altitude explicitly. In order to attain explicit control of altitude, $\ddot{r}(t)$ must be defined as follows

$$\ddot{r}(t) = c_1 p_1(t) + c_2 p_2(t), \tag{B19}$$

giving $\ddot{r}(t)$ two degrees of freedom. ($p_1(t)$ and $p_2(t)$ are pre-specified.) The two equations of constraint which $\ddot{r}(t)$ must satisfy are

$$\dot{r}_D - \dot{r}(t_o) = \int_{t_o}^{T} \ddot{r}(t)\,dt \tag{B20}$$

$$r_D - r(t_o) - \dot{r}(t_o) T_{go} = \int_{t_o}^{T} \left[\int_{t_o}^{t} \ddot{r}(s)\,ds\right] dt \tag{B21}$$

Equation (B19) can be substituted into Eqs. (B20, B21) to obtain a pair of simultaneous linear algebraic equations in c_1 and c_2; these equations can be readily solved for c_1 and c_2. Then the radial allocation of thrust acceleration which satisfies

$$r(T) = r_D \tag{B22}$$

$$\dot{r}(T) = \dot{r}_D \tag{B23}$$

is

$$a_T \sin\alpha = -(-\mu / r^2 + v_\theta^2 / r) + c_1 p_1(t) + c_2 p_2(t) \tag{B24}$$

Also, note that $p_1(t)$ and $p_2(t)$ can be selected to optimize the Δv performance of the guidance law. Thus, r, $\dot{r}$, and v_θ are explicitly controlled; and optimization is possible by proper selection of $p_1(t)$ and $p_2(t)$. A detailed discussion is given in the main body of the paper.

REFERENCES

1. Battin, R. H., Astronautical Guidance (New York: McGraw-Hill Book Company, Inc., 1964).

2. Cherry, G. W., "A Class of Unified Explicit Methods for Steering Throttleable and Fixed-Thrust Rockets", Progress in Astronautics and Aeronautics, vol. 13, Guidance and Control II, pp. 689-726 (New York: Academic Press, 1964).

3. Halfman, R. L., Dynamics: Systems, Variational Methods, and Relativity (Reading, Mass.: Addison-Wesley Publishing Company, Inc., 1962).

BIBLIOGRAPHY

Battin's book contains the exposition of five methods of finding the conic trajectory which passes from a given initial position vector to a given target position vector in a specified time of flight. The relevant chapter is the third, entitled "Two-Body Orbit Determination". This particular conic boundary-valued problem is referred to on pages 40-42 of this paper, in the section entitled "Guidance Law for Injection onto an Intercept Trajectory"

Battin's book also contains a brief exposition of $\underline{v}_g$-nulling methods. These methods were referred to in the first section of this paper and appendix B to this paper. The relevant pages in Battin's book are pp. 123-126.

Cherry's paper, reference 2, contains the discussion of two subjects which are pertinent to this paper. The first subject is the determination of the best time of engine ignition and best initial time-to-go for a lunar landing utilizing the E Guidance throttleable engine guidance law. The relevant pages for this discussion are pp. 716-719.

The other subject of interest in the reference by Cherry is in Appendix C, p. 722. This appendix shows very simple equations for updating the spacecraft position and velocity vectors with the accelerometer outputs. It has sometimes been said that implicit guidance equations are computationally convenient because they avoid the necessity of computing the gravitational acceleration and integrating the outputs of the accelerometers to obtain position. The equations in the reference for obtaining $\underline{r}$, $\underline{v}$, and $\underline{g}$ show how little computational burden is actually incurred by the requirement to obtain explicit knowledge of these quantities.

Halfman's book, pp. 474-477, shows that to maximize total energy (in escaping from a central-force field) the burnout velocity vector should be aligned with the burnout thrust vector. This result is referred to in the section of this paper which is entitled "Controlling the Spacecraft's Final Attitude".

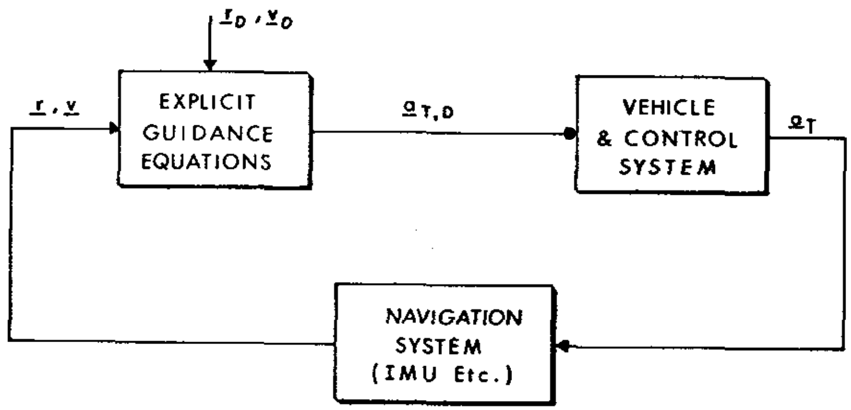
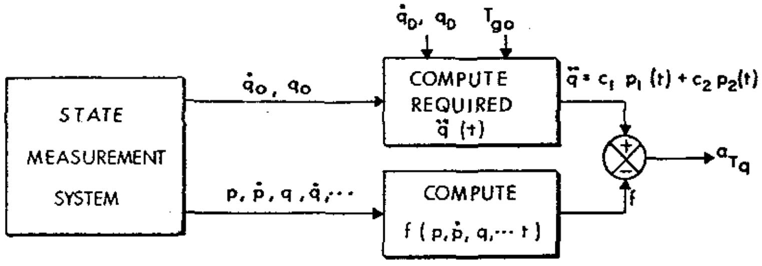


Fig. 1 Explicit steering equations.



$$\ddot{q} = a_{T_q} + f(p, \dot{p}, q, \dot{q}, \cdots, t)$$

Fig. 2 General solution to speed-position boundary-valued problem.

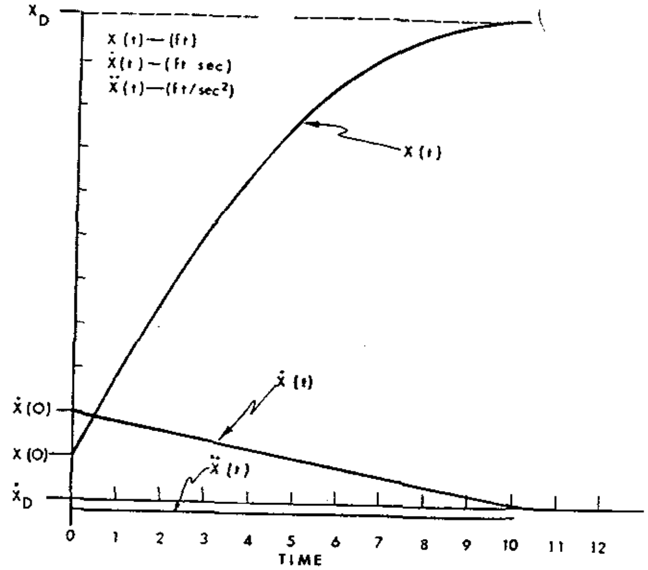


Fig. 3 Controlling final speed and position with a time profile of total acceleration.

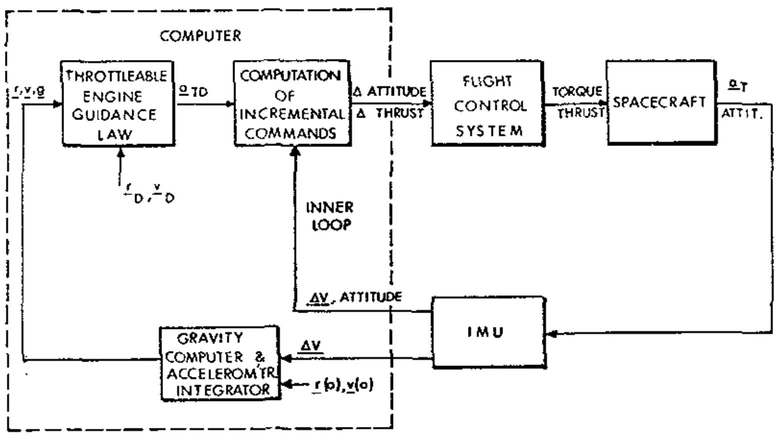


Fig. 4 Throttleable engine guidance law system mechanization.

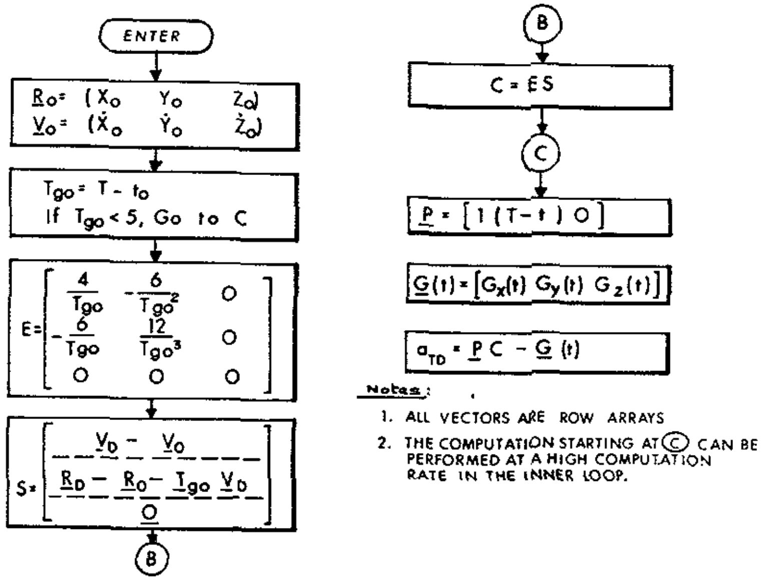


Fig. 5 Flow diagram of throttleable-engine guidance equations.

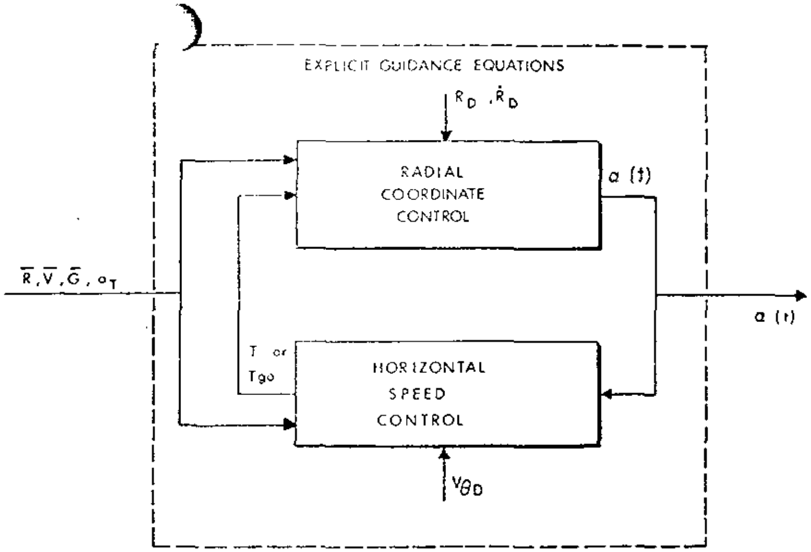


Fig. 6 Simplified block diagram of fixed-thrust rocket guidance law.

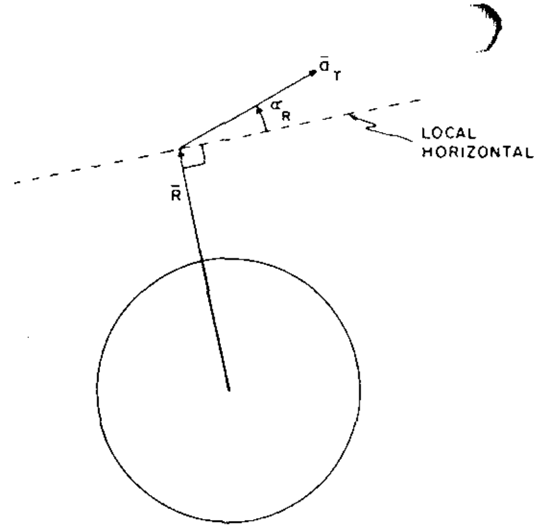


Fig. 7 Definition of thrust angle for fixed-thrust rocket guidance law.

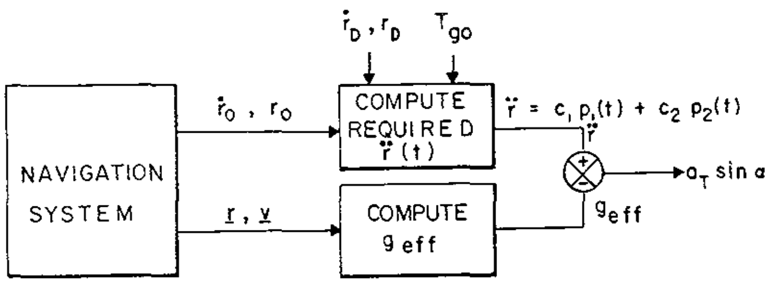


Fig. 8 Block diagram of the radius and radial rate control law.

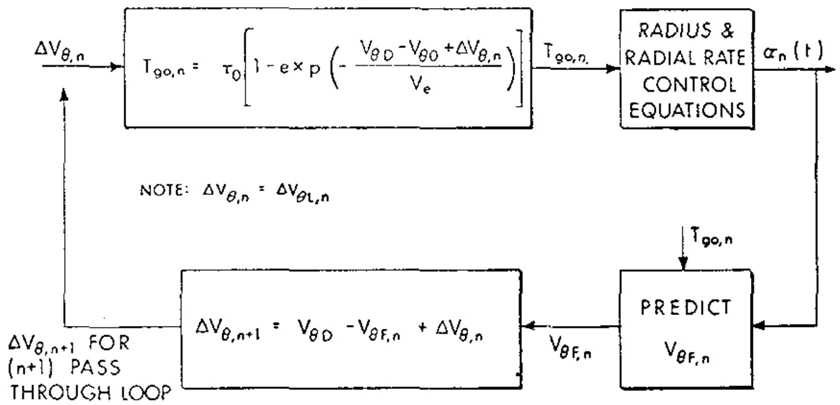


Fig. 9 Strategy of T_{go} prediction loop.

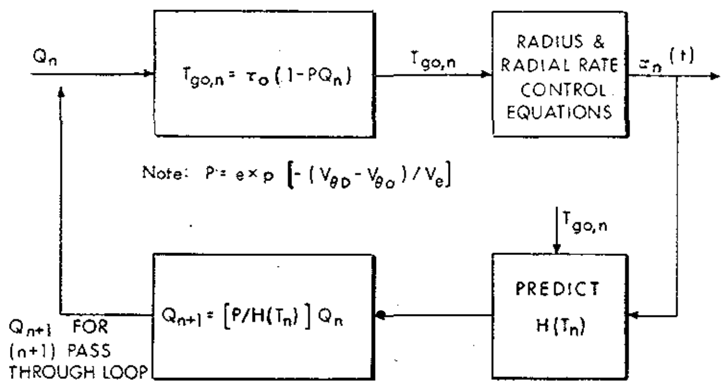


Fig. 10 T_{go} prediction utilizing Q_n and $H(T_n)$

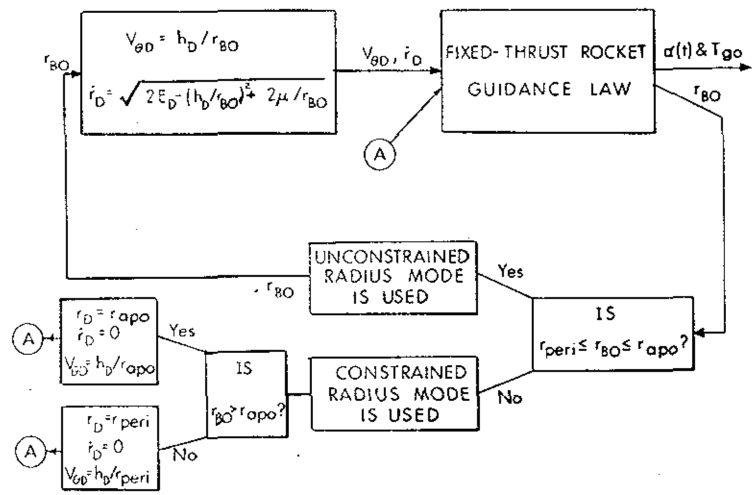


Fig. 11 Guidance law for attaining a specified conic trajectory.

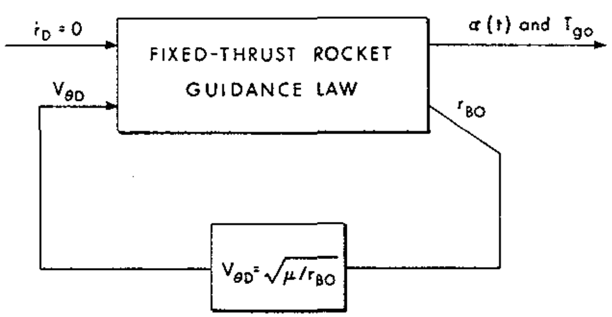


Fig. 12 Guidance law for insertion into a circular orbit of unspecified altitude.

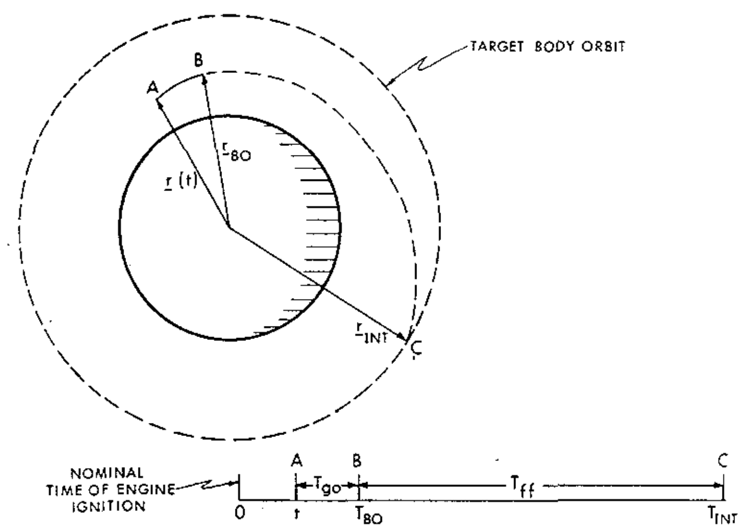


Fig. 13 The intercept problem geometry and time-scale.

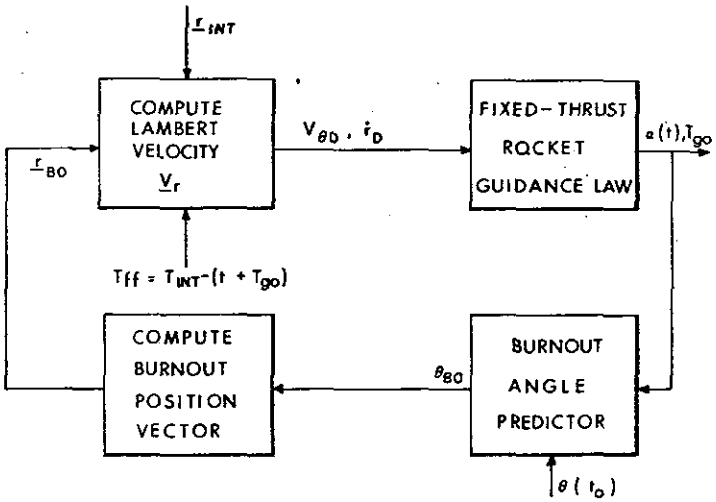


Fig. 14 Guidance law for injection onto an intercept trajectory.

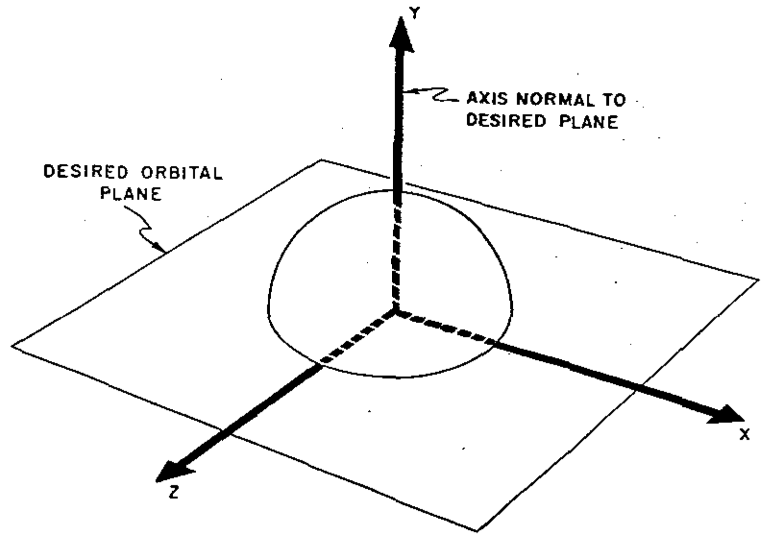


Fig. 15 The specified trajectory plane.

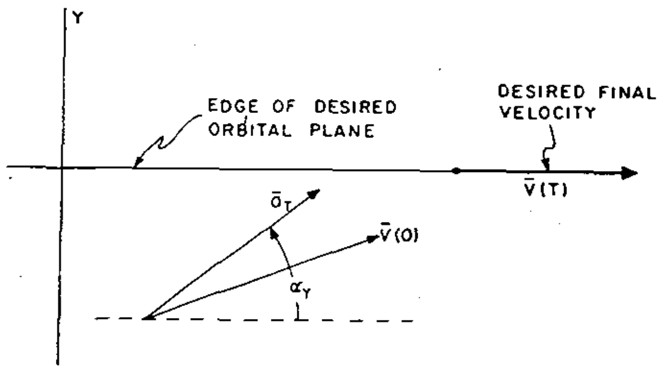


Fig. 16 Edge view of the specified trajectory plane.