JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 19, 565-598 (1967)

The Successive Sweep Method and Dynamic Programming

STEPHEN R. MCREYNOLDS*

Jet Propulsion Laboratory,

Pasadena, California 91103

Submitted by Richard Bellman

INTRODUCTION

The "successive sweep method" by McReynolds and Bryson [1] is an iterative algorithm for obtaining solutions to optimal control problems. A similar algorithm was developed independently by Mitter [2]. Both algorithms extend Merriam's second order relaxation method [3], [4] to optimal control problems with side contraints. Similar techniques for the solution of two-point boundary value problems are employed by Bellman and Kalaba [5] [6] in "invariant imbedding" and "quasilinearization."

The successive sweep method is related to the Newton-Raphson-Kantorovich techniques for solving two point boundary value problems [7]-[16]. Although the terminology is not uniform, all these techniques may be viewed as successive linearization or, equivalently, as successive maximization of a second order expansion of the performance index.

The successive sweep method, while sharing certain characteristics of these other methods, possesses a unique combination of features. One important feature is that the successive approximation is determined about an arbitrary point in the control space. This is a common and logical approach because the control space is the lowest dimensional space that may be used to specify an optimal solution. Various techniques involving successive approximation in control space were suggested by Bellman [17] and Kalaba [18]. Iteration in control space was also a fundamental part of the "gradient method" due to Bryson [19] and Kelley [20]. In an effort to obtain a technique with more rapid convergence, "Newton-Raphson" iterative procedures in control space were suggested by Merriam [4] and Kelley *et al.* [11]. These "Newton-Raphson"

* Senior Research Engineer, Jet Propulsion Laboratory, Pasadena, California. This work originated as part of the author's doctorial thesis at Harvard University under the guidance of Professor A. E. Bryson Jr., and supported by a grant from the Office of Naval Research. The present version represents one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS7-100, sponsored by the National Aeronautics and Space Administration.

techniques are different than those suggested by Kantorovich [7], Hestenes [8], and Stein [9] in that emphasis is placed on considering the control function as the fundamental unknown rather than either the state functions and/or adjoint functions.

For the methods involving iteration in control space, the state and adjoint functions are constructed to satisfy the proper differential equations. However, the optimality conditions that characterize the optimal control policy, together with terminal boundary conditions, are relaxed.

In the classical Newton-Raphson-Kantorovich method, as suggested by Stein [9] and more recently by Kopp and McGill [5], Moyer and Pinkham [6], Van Dine *et al.* [12], the state and adjoint vectors are picked arbitrarily but satisfy boundary conditions at both ends, while the control functions are determined from the optimality conditions. Both methods have certain advantages and disadvantages. However, iteration in policy space is favored here for several reasons. First, the control vector has a lower dimension than the state space. Secondly, for the purposes of obtaining good initial guess, policy space is usually more convenient to deal with than adjoint space.

Since iteration is performed in function space, the successive sweep method is distinct from "neighboring external methods" suggested by Kelly [11], Breakwell *et al.* [10], Jazwinski [21]. In these algorithms successive approximations were made on the initial values of the adjoint vector. The disadvantages with neighboring extremal methods are that for most problems a reasonably good guess of the initial value of the adjoint variables is very difficult.

The successive sweep method freely employs Lagrange Multipliers when needed. These are considered as additional control parameters that must be adjusted in order that the corresponding constraint be satisfied. The use of penalty functions is dispensed with. In this regard the successive sweep method is an extension of Merriam's algorithm [4].

The successive sweep method, and the related methods of Merriam [4] and Bellman and Kalaba [5], distinguish themselves from more standard Newton-Raphson method in that the "sweep method" (Gelfand and Fomin [22]) is applied to the linearized equations. The terminal boundary condition is "swept" back to the initial time, and the problem is converted to an initial value problem.

The forward sweep method takes on the special form of a linear corrective feedback law. The sweep method is particularly advantageous for damped systems in that it avoids the difficulty of integrating unstable equations.

In other places [1]-[3], the successive sweep method has been derived using the classical approaches employed in the calculus of variations. In [1] it is developed as solving a linearized two-point boundary value problem. In [2], [3] it is formulated as successively maximizing a second order variational expansion of the performance index.

In this paper, a different approach is taken by showing how dynamic programming may be used to furnish an elegent derivation and interpretation of the successive sweep method.

For the purposes of application to numerical problems a discrete algorithm is developed. Our development, although independently arrived at, closely resembles that of Mayne [23]. A similar algorithm, although slightly different but equivalent, appears in the author's thesis.

The Problem

The optimal control problem that shall be considered in this paper will be the Bolza problem. It consists of finding certain unknown control functions $u(t) = (u_1(t), \ldots, u_m(t))$ and control parameters $a = (a_1, \ldots, a_p)$ that optimize the performance of some physical system. The performance of the system is measured by a performance index J which assumes the general form

$$J = \phi(x(t_N), a, t_N) + \int_{t_0}^{t_N} L(x(t), u(t), a, t)\, dt. \tag{1}$$

The vector $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))$ denotes the state of the physical system. The control functions and the control parameters affect the state of the system through a set of system equations

$$\dot{x} = f(x, u, a, t). \tag{2}$$

For the purposes of this paper the time interval (t_0, t_N) will be assumed fixed. The initial state will be assumed given by the equations

$$x(t_0) = x_0 \text{ specified.} \tag{3}$$

An optimal control problem is usually complicated by the presence of side contraints. Of the various possible types of constraints, we shall consider the following kinds:

$$\Psi_0 = \Psi(x(t_N), a, t_N) + \int_{t_0}^{t_N} M(x(t), u(t), a, t)\, dt \tag{4}$$

$$C(x, u, a, t) \leqslant 0. \tag{5}$$

A set of functions and parameters $(x(t), u(t), a)$ that satisfies Eqs. (2)-(5) over $[t_0, t_N]$ is referred to as an "admissible solution." The problem in optimal control is to find the admissible solution for which the performance index attains its maximum (or minimum).

The Dynamic Programming Characterization of the Optimal Field

The first step in obtaining the dynamic theory is to imbed the original problem in a field of similar problems. Define

$$J(t^*) = \phi(x(t_N), a, t_N) + \int_{t^*}^{t_N} L(x(t), u(t), a, t)\, dt \tag{6}$$

$$\Psi_0(t^*) = \Psi(x(t), a, t_N) + \int_{t^*}^{t_N} M(x(t), u(t), a, t)\, dt. \tag{7}$$

Let us replace boundary condition (3) with the general boundary condition

$$x(t^*) = x_0^*, \tag{8}$$

where x_0^* is variable.

The field of problems can now be defined as follows: Find $u(t)$ over $(t^* \leqslant t \leqslant t_n)$ that maximizes $J(t^*)$ given by Eq. (6) subject to the constraints given by Eq. (2), (4), (7), and (8). The original problem is recovered in part by replacing t^*, $\Psi_0(t^*)$ and x_0^* by the specified values t_0, Ψ_0, x_0.

The solution to the field of problems is expressed in terms of an optimal control law $u_{\text{opt}}(x_0^*, t^*, a, \Psi_0(t^*))$, a function that expresses the optimal value of the control at time t^* explicitly as a function of the parameters x_0^*, t^*, a, and $\Psi_0(t^*)$. This function, together with the system equations (Eq. (2)) may be used to generate the field of optimal control functions and the optimal trajectories. In particular, the solution to the original problem may be generated. The optimal control law naturally defines the optimal return function $S_{\text{opt}}(x_0^*, t^*, a, \Psi_0(t^*))$, the function which assigns to x_0^*, t^*, a, and $\Psi_0(t^*)$ the optimal value of the return. This may be written as

$$S_{\text{opt}}(x_0^*, t^*, a, \Psi_0(t^*)) = \phi(x_{\text{opt}}(t_N), a) + \int_{t^*}^{t_N} L(x_{\text{opt}}(t), u_{\text{opt}}(t), a)\, dt, \tag{9}$$

where it is understood that $x_{\text{opt}}(t)$, and $u_{\text{op}t}(t)$ are the optimal trajectory and control function that corresponds to $(x_0^*, t^*, a, \Psi_0(t^*))$.

Dynamic Programming characterizes the optimal return function $S_{\text{opt}}(x_0, t_0, a, \Psi_0(t_0))$ and the optimal control law $u_{\text{opt}}(x, t, a, \Psi_0)$ in terms of a boundary value problem, which consists of a first order partial differential equation for $S_{\text{opt}}(x_0, t_0, a, \Psi(t))$, with a boundary condition at the terminal time, and a principle of optimality which characterizes the optimal control law. The derivation of the dynamic programming equations is contained in several places (in [17], for example). However, since the arguments involved are crucial for the development in the next section, a short, nonrigorous derivation will be included here.

Let us assume that $u_{\text{opt}}(x, t, a, \Psi_0)$ and $S_{\text{opt}}(x, t, a, \Psi_0)$ are known for a range of values of x, a and Ψ_0 over the interval $[t + \Delta t, t_N]$. Now let x, a, Ψ_0 be values within the range for which the optimal return function is known. We now consider the problem with the initial time t and with the parameter x, a, Ψ_0. In order to obtain the solution to this problem, it is only necessary to find the optimal choice of u over the interval $[t, t + \Delta t]$. Let $u(t)$ be an arbitrary continuous function over the interval $(t, t + \Delta t)$. The value of the return obtained from $u(t)$ followed by the optimal control law is given by

$$S(x, t, a, \Psi_0; u(t)) = S_{\text{opt}}(x + \Delta x, t + \Delta t, a, \Psi_0 + \Delta\Psi) + \int_t^{t+\Delta t} L(x(t), u(t), a, t)\, dt, \tag{10}$$

where

$$\Delta x = \int_t^{t+\Delta t} f(x(t), u(t), a, t)\, dt \tag{11}$$

$$\Delta\Psi = -\int_t^{t+\Delta t} M(x(t), u(t), a, t)\, dt. \tag{12}$$

By choosing Δt small the right-hand side of (10) will exist. If Δt is small enough $u(t)$ may be approximated to first order by a constant u and (10), (11), and (12) may be approximated to first order in Δt by

$$S(x, t, a, \Psi_0; u) = S_{\text{opt}}(x + \Delta x, t + \Delta t, a, \Psi_0 + \Delta\Psi) + L(x, u, a, t)\, \Delta t \tag{13}$$

$$\Delta x = f(x, u, a, t)\, \Delta t \tag{14}$$

$$\Delta\Psi = -M(x, u, a, t)\, \Delta t. \tag{15}$$

We now expand the right-hand side equation (13) around x, $t + \Delta t$, Ψ_0. Using (14) and (15), one obtains

$$\begin{aligned} &S(x, t, a, \Psi_0; u) \\ &= S_{\text{opt}}(x, t + \Delta t, a, \Psi_0) + \Big[\frac{\partial S_{\text{opt}}}{\partial x}(x, t + \Delta t, a, \Psi_0) f(x, u, a, t) \\ &- \frac{\partial S_{\text{opt}}}{\partial \Psi_0}(x, t + \Delta t, a, \Psi_0)\, M(x, u, t, a) + L(x, u, a, t)\Big] \Delta t + 0(\Delta t^2). \end{aligned} \tag{16}$$

The optimal u must maximize the right hand-side of Eq. (16). As $\Delta t \to 0$, the $0(\Delta t^2)$ term may be ignored, and the optimal control $u_{\text{opt}}(x, t, a, \Psi_0)$ is given by

$$\begin{aligned} u_{\text{opt}}(x, t, a, \Psi_0) = \arg\max_u \Big\{&\frac{\partial S_{\text{opt}}}{\partial x}(x, t, a, \Psi_0) f(x, u, a, t) \\ &- \frac{\partial S_{\text{opt}}}{\partial \Psi_0}(x, t, a, \Psi_0)\, M(x, u, a, t) + L(x, u, a, t)\Big\}. \end{aligned} \tag{17}$$

If the optimal choice of u is employed on the right-hand side of Eq. (16), the left-hand side becomes $S_{\text{opt}}(x, t, a, \Psi_0)$. Expanding the right side of Eq. (16) around (x, t, a, Ψ_0), one obtains, after cancelling:

$$0 = \Big[\frac{\partial S_{\text{opt}}}{\partial t}(x,t,a,\Psi_0) + \frac{\partial S_{\text{opt}}}{\partial x}(x, t, a, \Psi_0) f(x, u_{\text{opt}}, a, t) - \frac{\partial S_{\text{opt}}}{\partial \Psi_0} M(x, u_{\text{opt}}, a, t) + L(x, u_{\text{opt}}, a, t)\Big] \Delta t + O(\Delta t^2). \tag{18}$$

Since Eq. (18) must hold over a range of Δt, including arbitrarily small Δt, the quantity in the brackets may be set to zero, resulting in the partial differential equation:

$$0 = \frac{\partial S_{\text{opt}}}{\partial t}(x, t, a, \Psi_0) + \frac{\partial S_{\text{opt}}}{\partial x}(x, t, a, \Psi_0) f(x, u_{\text{opt}}, a, t) - \frac{\partial S_{\text{opt}}}{\partial \Psi_0} M(x, u_{\text{opt}}, a, t) + L(x, u_{\text{opt}}, a, t). \tag{19}$$

The boundary conditions to this partial differential equation are given by

$$S_{\text{opt}}(x, t_N, a) = \Psi(x, t_N, a) \tag{20}$$

$$\Psi_0(t_N) = \Psi(x_N, t_N, a). \tag{21}$$

If the parameters $[a]$ are to be chosen to optimize the return function, they must be chosen at the initial time according to the rule:

$$a_{\text{opt}}(x_0, t_0, \Psi_0) = \arg \operatorname*{Max}_a S_{\text{opt}}(x_0, t_0, a, \Psi_0). \tag{22}$$

The dynamic programming equations are given by Eqs. (17), (19), (20)-(22). The Eq. (17) is the optimality principle by which the optimal control function is determined; note that it is a function only of x, a, t, Ψ_0 and the first partials of the optimal return function evaluated at x, a, t. Equation (19) is a partial differential equation for the return function, which can be solved "backward" in time in conjunction with Eq. (17) and the boundary conditions given by Eqs. (20) and (21).

Side Constraints—A Legendre Transformation

The formulation of dynamic programming theory introduced in the last section must be modified if side constraints of the form given by Eq. (7) are present with $\Psi(x(t_N), a, t_N)$ functionally dependent on $x(t_N)$. The reason is this: in order to evalute u_{opt} at the final time, as given by Eq. (17), it is necessary to compute S_x and S_{Ψ_0}, but these partial derivatives are not well defined at the final time, because from Eq. (21) it is seen that at the final time Ψ_0 is functionally dependent on x.

In order to avoid this difficulty a set of variables (Lagrange multipliers) ν_i $(i = 1, p)$ are introduced; they are defined by

$$\nu^T = -\frac{\partial S_{\text{opt}}}{\partial \Psi_0}(x, t, \Psi_0). \tag{23}$$[1]

The ν_i $(i = 1, p)$ are constant along an optimal trajectory. To show this, differentiate (19) with respect to Ψ_0, which yields

$$0 = \frac{\partial^2 S_{\text{opt}}}{\partial t\, \partial \Psi_0} + \frac{\partial^2 S}{\partial x\, \partial \Psi_0} f - \frac{\partial^2 S}{\partial \Psi_0{}^2} M(x, u, a, t). \tag{24}$$

Partials of u with respect Ψ_0 do not appear because Eq. (19) is extremized with respect u. The above expression can be interpreted as

$$\frac{d}{dt}\frac{\partial S}{\partial \Psi} = \frac{d}{dt}(-\nu) = 0,$$

where $(d/dt)(\ )$ is the total derivative of a quantity along a trajectory. Employing a Legendre-transformation ([22], p. 71) a new performance index can be defined by

$$J^* = \phi(x_1, a, t_N) + \nu^T(\Psi(x_N, a, t_N) - \Psi_0) + \int_{t_0}^{t_N} (L(x, u, a, t) + \nu^T M(x, u, a, t))\, dt. \tag{25}$$[1]

The solution of the old problem can be obtained by extremizing J^* with respect to u, a, and ν. Thus the technique of handling side constraints may be considered as a special case of the technique employed for control constants.

The Successive Sweep Correction: Control Constants and Initial Conditions

If control functions $u(t)$ are absent (or have been chosen optimally) the problem becomes that of finding the optimal choice of the parameters $[a]$

[1] The superscript $[T]$ denotes the transpose of a matrix and vector.

and $[\nu]$. Let us combine both sets of constants into $[a]$ so that without loss of generality we may consider the problem of finding $[a]$ to extremize

$$J = \phi(x_N, a, t_N) + \int_{t_0}^{t_N} L(x, a, t)\, dt, \tag{26}$$

subject to the constraints

$$\dot{x}_k = f^k(x, a, t). \tag{27}$$

$x(t_0)$ may be given or to be chosen optimally. The return function $V(x, a, t)$ satisfies

$$V(x, a, t_N) = \phi(x, a, t_N) \tag{28}$$

$$V_t + V_{x_k} f^k(x, a, t) + L(x, a, t) = 0. \tag{29}$$ [1]

The optimal choice of a satisfies

$$a_{\text{opt}}(x_0, t_0) = \arg \operatorname{Max} V(x_0, a, t_0). \tag{30}$$

The Successive Sweep Method for obtaining a_{opt} consists of generating a sequence of nominal solutions $a^{(0)}, a^{(1)}, \ldots, a^{(k)}, \ldots$ that converges to a_{opt}. The correction da to $a^{(k)}$ is obtained by maximizing the quadratic expansion

$$V(x_0, a^{(k)} + da, t_0) \cong V(x_0, a^{(k)}, t_0) + V_{a_i}(x, a^{(k)}, t_0)\, da_i$$
$$+ \tfrac{1}{2}\, da_i V_{a_i a_j}(x_0, a^{(k)}, t_0)\, da_j. \tag{31}$$

Setting the partial of this expansion with respect to da yields

$$0 = Va_i + V_{a_i a_j}\, da_j. \tag{32}$$

Solving for da_i yields

$$da_i = -(V^{-1}_{a_i a_j})\, V_{a_j} \big|_{a^{(k)}, x_0, t_0}. \tag{33}$$

The next iterate $a^{(k+1)}$ is obtained by adding this correction to $a^{(k)}$. Repeating this process, convergence to the maximum is obtained (see Fig. 1).

---

[1] In this and following section subscripts x, u, a, t are used to denote partial derivatives e.g. $V_{x_i} = \partial V / \partial_{x_i}$. Subscripts i, j, k, etc. denote vector components. For these subscripts, repetition implies summation.

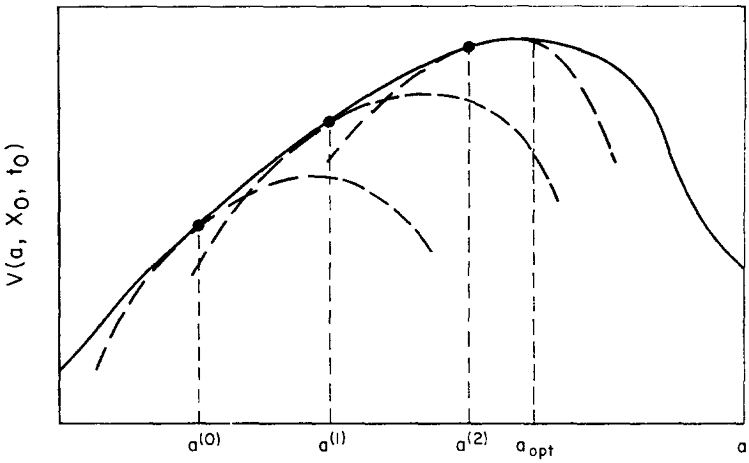

FIG. 1. The successive sweep method.

In order to obtain this correction one may calculate the derivatives of $V(x, a, t)$ by the technique of characteristics employed by Dreyfus [25]. Since Eq. (29) and Eq. (28) must fold for arbitrary a, the equations obtained by taking first and second partials of the equations with respect to a must also hold, yielding

$$V_{a_i} = \phi_{a_i}(x, a, t_N) \tag{34}$$

$$V_{a_i a_j} = \phi_{a_i a_j}(x, a, t_N) \tag{35}$$

$$V_{ta_i} + V_{x_k a_i} f^k + V_{x_k} f^k_{a_i} + L_{a_i} = 0 \tag{36}$$

$$V_{ta_i a_j} + V_{x_k a_i a_j} f^k + V_{x_k a_i} f^k_{a_j} + V_{x_k a_j} f^k_{a_i} + V_{x_k} f^k_{a_i a_j} + L_{a_i a_j} = 0. \tag{37}$$

In terms of the operator

$$(\dot{\ }) = \frac{\partial}{\partial t}(\ ) + \frac{\partial}{\partial x_i}(\ )f,$$

which expresses the total derivative along a trajectory, Eqs. (36) and (37) become

$$\dot{V}_{a_i} = - V_{x_k} f^k_{a_i} - L_{a_i} \tag{38}$$

$$\dot{V}_{a_i a_j} = - V_{x_k a_i} f^k_{a_j} - V_{x_k a_i} f^k_{a_j} - V_{x_k} f^k_{a_i a_j} - L_{a_i a_j}\,. \tag{39}$$

These are characteristic equations for V_{a_i} and $V_{a_i a_j}$. Together with boundary conditions (34) and (35), V_{a_i} and $V_{a_i a_j}$ may be computed backward along a trajectory. In order to integrate these characteristic equations, it will be necessary to also compute V_{x_i} and $V_{a_i x_j}$, which in turn requires $V_{x_i x_j}$. Characteristic equations for these quantities and conditions can be obtained by taking the first and second partials of Eqs. (28) and (29).

$$V_{x_i}(x, a, t_N) = \phi_{x_i}(x, a, t_N) \tag{40}$$

$$V_{x_i a_j}(x, a, t_N) = \phi_{x_i a_j}(x, a, t_N) \tag{41}$$

$$V_{x_i x_j}(x, a, t_N) = \phi_{x_i x_j}(x, a, t_N) \tag{42}$$

$$\dot{V}_{x_i} = - V_{x_k} F^k_{x_i} - L_{x_i} \tag{43}$$

$$\dot{V}_{x_i a_j} = - V_{x_k a_j} f^k_{x_i} - V_{x_k x_i} f^k_{a_j} - V_{x_k} f^k_{x_i a_j} - L_{x_i a_j} \tag{44}$$

$$\dot{V}_{x_i x_j} = - V_{x_i x_k} f^k_{x_j} - V_{x_j x_k} f^k_{x_i} - V_{x_k} f^k_{x_i x_j} - L_{x_i x_j}. \tag{45}$$

If the initial conditions are not fully specified, it will be necessary to iterate on those unknown components of the initial state vector. Not much more effort is required because the first and second partials of the return function with respect to x are known. If $x(t_0)$ is completely unspecified, then $dx(t_0)$ and da are determined simultaneously by maximizing the quadratic expansion.

$$\begin{aligned} &V(x_0^{(k)} + dx, a^{(k)} + da) \\ &= V(x_0^{(k)}, a^{(k)}) + V_{x_i}\, dx_i + V_{a_i}\, da_i + \tfrac{1}{2} V_{x_i x_j}\, dx_i\, dx_j + V_{x_i a_j}\, dx_i\, da_j \\ &\quad + \tfrac{1}{2} V_{a_i a_j}\, da_i\, da_j; \end{aligned} \tag{46}$$

this results in correction given by

$$\begin{bmatrix} dx \\ da \end{bmatrix} = - \begin{bmatrix} V_{xx} V_{xa} \\ V_{ax} V_{aa} \end{bmatrix}^{-1} \begin{bmatrix} V_x^T \\ V_a^T \end{bmatrix}. \tag{47}$$

The Successive Sweep Correction to the Nominal Control $u(t)$

To obtain corrections to the control it is necessary to compute the functional gradient and curvature. The concept of functional gradient arises in the gradient method of Bryson [19] and Kelly [20]. The concept functional gradient has been formulated in terms of dynamic programming by

Dreyfus [26]. In this section the concept of functional curvature is similarly introduced.

For now let us consider a performance index simply of the form

$$J = \phi(x_N, t_N) + \int_{t_0}^{t_N} L(x, u, t)\, dt, \tag{48}$$

and system equations are given by:

$$\dot{x} = f(x, u, t). \tag{49}$$

To arrive at the concept of functional gradient and curvature, let us consider a small change in u over the interval $(t, t + \Delta t)$. Δt is chosen small enough a continuous control function may be considered approximately constant over this interval, which we designate as u. Now the return function

$$V(x, t; u) = V(x, t; u, u(t): t \in [t + \Delta t, t_N])$$

satisfies

$$V(x, t; u) = V(x + \Delta x, t + \Delta t; u(t) : t \in [t + \Delta t, t_N]) + \int_{t}^{t+\Delta t} L(x; u, t)\, dt, \tag{50}$$

where

$$\Delta x = \int_{t}^{t+\Delta t} f(x, u, t)\, dt. \tag{51}$$

To compute the effect of a change in u, Eq. (51) may be used to eliminate Δx from Eq. (50), and the right-hand side of Eq. (51) may be expanded in terms of δu.

$$\begin{aligned} \delta V(x, t; \delta u) &= V(x, t; u + \delta u) - V(x, t; u) \\ &= \Delta t\{[V_{x_k} f^k_{u_j} + L_{u_j}]\, \delta u_j \\ &\qquad + \tfrac{1}{2}\, [V_{x_k} f^k_{u_i u_j} + L_{u_i u_j}]\, \delta u_i\, \delta u_j + 0(\delta u^3, \Delta t)\}. \end{aligned} \tag{52}$$

In the above expression, all the partials of V are evaluated at (x, t^*). In the expansion in Eq. (52) appears the functional gradient $V_{x_k} f^k_{u_i} + L_{u_i}$ and the functional curvature $V_{x_k} f^k_{u_i u_j} + L_{u_i u_j}$. These may be recognized as familiar quantities appearing in the theory of optimal control. $V_{x_k} f^k_{u_i} + L_{u_i} = 0$ and $V_{x_k} f^k_{u_i u_j} + L_{u_i u_j} \leqslant 0$ are necessary conditions for an optimal solution.

The first naieve formulation the successive sweep correction is to chose δu to maximize the approximation in Eq. (52) by ignoring the $0(\delta u^3, \Delta t^2)$ term, resulting in

$$\delta u_i = -\, (V_{x_k} f^k_{u_i u_j} + L_{u_i u_j})^{-1}\, (V_{x_k} f^k_{u_j} + L_{u_j}). \tag{53}$$

This correction suggest an iterative scheme by which the optimal control law may be obtained. Let $u^{(k)}(x, t)$ be the kth optimal control law, then $u^{(k+1)}(x, t)$ may be obtained by

$$u_i^{(k+1)}(x, t) = u_i^{(k)}(x, t) - (V_{x_n} f^n_{u_i u_j} + L_{u_i u_j})^{-1} (V_{x_m} f^m_{u_j} + L_{u_j}). \quad (54)$$

$V(x, t)$ is the return function that corresponds the control law $u^{(k+1)}$. The $u^{(k+1)}(x, t)$ must be computed backward in time simultaneously with the solution of the partial differential equation for $V(x, t)$. This process is pictured in Figs. 2-4 for the case $L \equiv 0$. In Fig. 2 the improved control law

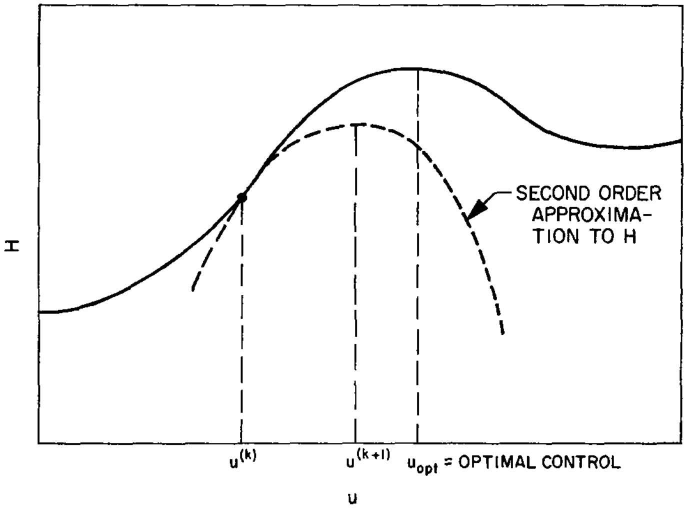

FIG. 2. The successive sweep correction to the control.

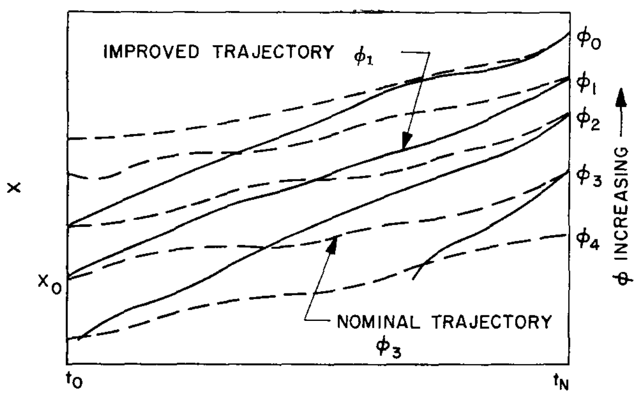

FIG. 3. Field of improved trajectories compared to the old; $J = \phi(x(N))$.

$u^{(k+1)}$ is pictured geometrically as maximizing an quadratic approximation of $V_{x_k} f^k$ (the "Hamiltonian"). In Fig. 3 the field of improved trajectories are pictured superimposed on the field of old trajectories. Because $L = 0$, the return functions are constant along the corresponding field of trajectories. The correction in the control at time t may be interpreted as yielding the direction in which the return is increasing most rapidly from among all the possible admissible directions, according to a second order perturbation in u; this is pictured in Fig. 4.

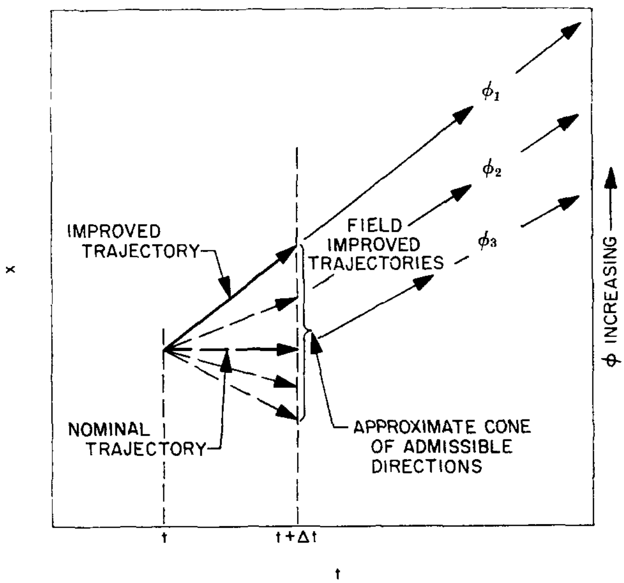

FIG. 4. The backward sweep.

The Successive Sweep procedure that has been described so far, although esthetically pleasing, is not practical because it involves much work, even more than required for a dynamic programming solution.

In order to obtain a practical means of obtaining solutions to a problem, a method of obtaining an approximation to the control law given in Eq. (53) is necessary. Let us restrict the nominal control law $u^{(k)}(x, t)$ to be merely a function of time $u^{(k)}(t)$. We seek to approximate the improved control law in the neighborhood of a nominal trajectory $x^{(k)}(t)$. Returning to Eq.(50), let us expand $V(x, t; u)$ around $x^{(k)}$, $u^{(k)}$ to obtain

$$\delta V(x^{(k)} + dx, t, \delta u) = \Delta t[Z_0 + Z_{u_i}\,\delta u_i + Z_{x_i}\,\delta x_i + \tfrac{1}{2} Z_{u_i u_j}\,\delta u_i\,\delta u_j$$

$$+ Z_{u_i x_j}\,\delta x_j\,\delta u + \tfrac{1}{2} Z_{x_i x_j}\,\delta x_i\,\delta x_j + 0(\delta x^3, \delta u^3)] + 0(\Delta t^2), \qquad (55)$$

where

$$Z_{u_i} = V_{x_j} f^j_{u_i} + L_{u_i}$$

$$Z_{x_i} = V_{x_j} f^j_{x_i} + L_{x_i}$$

$$Z_{u_i u_j} = V_{x_k} f^k_{u_i u_j} + L_{u_i u_j}$$

$$Z_{u_i x_j} = V_{x_j x_k} f^k_{u_i} + L_{u_i x_j} + V_{x_k} f^k_{u_i x_j}$$

$$Z_{x_i x_j} = V_{x_i x_k} f^k_{x_j} + V_{x_j x_k} f^k_{x_i} + V_{x_k} f^k_{x_i x_j} + L_{x_i x_j} . \tag{56}$$

All the terms in these and succeeding expressions are evaluated at $x^{(k)}$, $u^{(k)}$, $a^{(k)}$. The Successive Sweep improvement is obtained by chosing δu to maximize the expression on the right-hand side of Eq. (55), neglecting third-order quantities δx and δu and second-order quantities in Δt.

$$\delta u_i = - Z^{-1}_{u_i u_j}(Z_{u_j} + Z_{u_j x_k} \delta x_k); \tag{56}$$

this is the desired approximation to equation and may be appropriately termed as a "corrective feedback law." If parameters are present, the corrective feedback law takes on the form

$$\delta u_i = - Z^{-1}_{u_i u_j}(Z_{u_j} + Z_{u_j x_k} \delta x_k + Z_{u_j a_k} \delta a_k), \tag{57}$$

where

$$Z_{u_i a_j} = V_{x_k a_j} f^k_{u_i} + V_{x_k} f^k_{u_i a_j} + L_{u_i a_j} . \tag{58}$$

In order to evaluate the expressions in Eq. (57), it is necessary to obtain the first and second partial derivatives of V with respect a and x. To do so, the method of characteristics may be employed. To obtain the characteristic equations in this case, we shall consider a quadratic expansion of the partial differential equations for the return function.

The boundary value problem defining the return function in the case where both control parameters and control functions are present is:

$$V(x, a, t_N) = \phi(x, a, t_N) \tag{59}$$

$$V_t(x, a, t) + V_{x_i}(x, a, t) f^i(x, a, u, t) + L(x, a, u, t) = 0. \tag{60}$$

A quadratic expansion of Eq. (60) around a reference solution yields

$$0 = (\dot{V}_0 + L) + (\dot{V}_{x_i} + Z_{x_i})\,\delta x_i + (\dot{V}_{a_i} + Z_{a_i})\,\delta a_i + Z_{u_i}\,\delta u_i$$
$$+ \tfrac{1}{2}\,(\dot{V}_{x_i x_j} + Z_{x_i x_j})\,\delta x_i\,\delta x_j + Z_{x_i u_j}\,\delta x_i\,\delta u_j$$
$$+ (\dot{V}_{x_i a_j} + Z_{x_i a_j})\,\delta x_i\,\delta a_j + \tfrac{1}{2}\,Z_{u_i u_j}\,\delta u_i\,\delta u_j + Z_{u_i a_j}\,\delta u_i\,\delta a_j$$
$$+ \tfrac{1}{2}\,(\dot{V}_{a_j a_j} + Z_{a_i a_i})\,\delta a_i\,\delta a_j\,, \tag{61}$$

where $(\cdot)$ denotes the characteristic derivative $(\partial/\partial t)(\ ) + (\partial/\partial x)(\ )$ along the nominal trajectory, and

$$Z_{a_i} = V_{x_j} f^{\,j}_{a_i} + L_{a_i}$$

$$Z_{a_i a_j} = V_{x_k} f^{\,k}_{a_i a_j} + V_{x_k a_i}\, f^{k}_{a_j} + V_{x_k a_j}\, f^{\,k}_{a_i} + L_{a_i a_j}\,. \tag{62}$$

Note that the corrective feedback law given by Eq. (57) maximizes the right-hand side of Eq. (61). Substituting Eq. (57) into Eq. (61) to eliminate δu, the following identity is obtained.

$$0 = (\dot{V}_0 + L - Z_{u_i} Z^{-1}_{u_i u_j} Z_{u_j}) + (\dot{V}_{x_i} + Z_{x_i} - Z_{u_j} Z^{-1}_{u_j u_k} Z_{u_k x_i})\,\delta x_i$$
$$+ (\dot{V}_{a_i} + Z_{a_i} - Z_{u_j} Z^{-1}_{u_j u_k} Z_{u_k a_i})\,\delta a_i$$
$$+ \tfrac{1}{2}\,(\dot{V}_{x_i x_j} + Z_{x_i x_j} - Z_{u_n x_i} Z^{-1}_{u_n u_k} Z_{u_n x_j})\,\delta x_i x_j$$
$$+ (\dot{V}_{x_i a_j} + Z_{x_i a_j} - Z_{x_i u_n} Z^{-1}_{u_n u_k} Z_{u_k a_j})\,\delta x_i\,\delta a_j$$
$$+ \tfrac{1}{2}\,(\dot{V}_{a_i a_j} + Z_{a_i a_j} - Z_{a_i u_m} Z^{-1}_{u_n u_m} Z_{u_m a_j})\,\delta a_i\,\delta a_j\,.$$

Setting to zero the individual terms in the above expression results in characteristic equations for the first and second partials of V.

$$\dot{V}_0 + L - Z_{u_i} Z^{-1}_{u_i u_j} Z_{u_j} = 0, \qquad \dot{V}_{x_i} + Z_{x_i} - Z_{u_j} Z^{-1}_{u_j u_k} Z_{u_k x_i} = 0$$
$$\dot{V}_{a_i} + Z_{a_i} - Z_{u_j} Z^{-1\,k}_{u_j u_k} Z_{u_k a_i} = 0, \qquad \dot{V}_{x_i x_j} + Z_{x_i x_j} - Z_{x_j u_m} Z^{-1}_{u_m u_n} Z_{u_n x_j} = 0$$
$$\dot{V}_{x_i a_j} + Z_{x_i a_j} - Z_{x_i u_k} Z^{-1}_{u_k u_n} Z_{u_n a_j} = 0$$
$$\dot{V}_{a_i a_j} + Z_{a_i a_j} - Z_{u_k a_i} Z^{-1}_{u_k u_n} Z_{u_n a_j} = 0. \tag{63}$$

The boundary conditions for these derivatives are obtained as before, by expanding Eq. (59) around the nominal solution.

$$\begin{aligned}
V_0(t_N) &= \phi(x^{(k)}(t_N), t_N, a)\\
V_{x_i}(t_N) &= \phi_{x_i}(x^{(k)}(t_N), t_N, a^{(k)})\\
V_{a_i}(t_N) &= \phi_{a_i}(x^{(k)}(t_N), t_N, a^{(k)})\\
V_{x_i x_j}(t_N) &= \phi_{x_i x_j}(x^{(k)}(t_N), t_N, a^{(k)})\\
V_{x_i a_j}(t_N) &= \phi_{x_i a_j}(x^{(k)}(t_N), t_N, a^{(k)})\\
V_{a_i a_j}(t_N) &= \phi_{a_i a_j}(x^{(k)}(t_N), t_N, a^{(k)}).
\end{aligned} \tag{64}$$

Corrections in the parameters can be chosen again by Eq. (38). If side constraint of the form given by Eq. (4) are present, then the Legendre-transformation discussed earlier may be used.

If side constraints of the form given in Eq. (5) are present, then standard techniques used in calculus for inequality constraints may be employed. The constraint may be ignored until it is violated. If it is violated, then the equality must be enforced. This can be done by either using the equality to explicitly solve for a control variable, thus reducing the dimension of the unknown control functions. If this cannot be done, then additional multipliers may be introduced. Once the correction in the control parameters has been determined, the corrective feedback law given by Eq. (57) may be used together with the system equations to generate the next nominal solution $u^{(k+1)}(t)$ and $x^{(k+1)}(t)$. Then the process may be repeated until the desired accuracy is obtained.

Summary of the Sweep Method

From the dynamic programming point of view, the successive sweep method consists of a second-order perturbation of the boundary value problem defined by the dynamic programming equations. The perturbation is made around a nominal set of control functions $u^{(k)}(t)$ and parameters $a^{(k)}$ and trajectory $x^{(k)}$.

On the backward part of the sweep method an improved control law $u^{(k+1)}(x, t, a)$ and return function $V(x, t, a)$ are approximated. The improved control law is obtained by maximizing a quadratic approximation to the Hamiltonian.

This leads to a linear (corrective) control law. To obtain the coefficients appearing in the control law, it is necessary to compute the first and second derivatives of the return function, which may be done backwards by employing characteristic equations. At the initial time, δa is chosen to maximize a quadratic expansion $\delta V = V_{a_i}\delta a_i + \frac{1}{2} V_{a_i a_j}\delta a_i a_j$. On the forward sweep, the corrective feedback law is used to generate an improved trajectory and improved control function.

Example

To illustrate the successive sweep method, let us consider the problem of minimizing

$$J = \int_0^1 \frac{\sqrt{1 + u^2}}{\sqrt{1 - x}} \, dt,$$

subject to the constraints

$$x(1) = -.5$$
$$x(0) = 0$$
$$\dot{x} = u.$$

This is a particular example of the brachistochrone problem, which is formulated in detail later on.

The terminal constraint requires that the Lagrange multiplier ν be introduced, and the performance index becomes

$$J^* = \nu(x(1) + .5) + \int_0^1 \frac{\sqrt{1 + u^2}}{\sqrt{1 - x}} \, dt.$$

The solution to the original problem is obtained by minimizing J^* with respect to u and extremizing J^* with respect to ν.

Let $u^0(t)$, ν^0, $x^0(t)$ designate a nominal solution about which a field of improved solutions shall be approximated. The field of improved solutions has a return function $S(x, v, t)$ and $u(x, v, t)$ which satisfies

$$S(x, \nu, 1) = \nu(x + .5); \qquad S_t + S_x u(x, \nu, t) + \frac{\sqrt{1 + u^2(x, \nu, t)}}{\sqrt{1 - x}} = 0.$$

To construct our approximation, the following quadratic perturbation of the above partial differential equation is obtained,

$$\left(\dot{S}_0 + \frac{a}{b}\right) + \left(\dot{S}_x + \frac{a}{2b^3}\right) \delta x + \left(S_x + \frac{u^0}{ab}\right) \delta u + \dot{S}_\nu \, \delta\nu$$
$$+ \frac{1}{2}\left(\dot{S}_{xx} + \frac{3a}{4b^5}\right) \delta x^2 + \dot{S}_{x\nu} \, dx \, d\nu + S_{x\nu} \, \delta\nu \, \delta u$$
$$+ \left(S_{xx} + \frac{u^0}{2ab^3}\right) \delta u \, \delta x + \frac{1}{2} \dot{S}_{\nu\nu} \, \delta\nu \, \delta\nu + \frac{1}{2} \frac{\delta u^2}{a^3 b} = 0,$$

where

$$a = \sqrt{1 + u^{02}}, \qquad b = \sqrt{1 - x^0}.$$

The Newton-Raphson correction to the control is obtained by maximizing the above expression, which yields

$$\delta u = -a^3 b\left(S_x + \frac{u^0}{2a}\right) - a^3 b S_{x\nu}\, d\nu - a^3 b\left(S_{xx} + \frac{u^0}{2ab^3}\right).$$

Substituting this corrective control law back into the above equation, one obtains the characteristic equations for the first and second derivatives of S. Boundary conditions are obtained by a perturbation of the terminal boundary condition:

$$S_0(t_N) = \nu^0(x^0(t_N) + .5); \qquad \dot{S}_0 + \frac{a}{b} - a^3 b\left(S_x + \frac{u^0}{ab}\right)^2$$

$$S_x(t_N) = \nu^0; \qquad \dot{S}_x + \frac{a}{2b^3} - \left(S_x + \frac{u^0}{ab}\right) a^3 b\left(S_{xx} + \frac{u^0}{2ab^3}\right) = 0$$

$$S_\nu(t_N) = x^0(t_N) + .5; \qquad \dot{S}_\nu - \left(S_x + \frac{u^0}{ab}\right) a^3 b S_{x\nu} = 0$$

$$S_{xx}(t_N) = 0; \qquad \dot{S}_{xx} + \frac{3a}{4b^5} - \left(S_{xx} + \frac{u^0}{2ab^3}\right)^2 a^3 b = 0$$

$$S_{x\nu}(t_N) = 1; \qquad \dot{S}_{x\nu} - \left(S_{xx} + \frac{u^0}{2ab^3}\right) a^3 b S_{x\nu} = 0$$

$$S_{\nu\nu}(t_N) = 0; \qquad \dot{S}_{\nu\nu} - a^3 b S_{x\nu}^2 = 0.$$

The correction $\delta\nu$ is obtained at the initial time by

$$\delta\nu = -S_{\nu\nu} S_\nu \,|_{t_0}\,.$$

A Finite-Difference Algorithm

Before the successive sweep method is applied to obtain numerical solutions, a finite difference analogue will be developed similar to the ones obtained by Henrici [27], Van Dine [12], Dreyfus [28], Mayne [23], Rauch *et al.* [29], and McReynolds [24].

Let us replace the problem defined by Eq. (1) the finite difference approximation

$$J = \phi(x(t_N), a, t_N) + \sum_{i=0}^{n-1} L^i(x(i), u(i), a) \tag{65}$$

$$x(i+1) = F^i(x(i), u(i), a) \tag{66}$$

$x(0) = x_0$ may or may not be specified.

It is understood that the interval (t_0, t_N) is partitioned into the subintervals (t_i, t_{i+1}) and the following notation is introduced.

$$u(i) \equiv u(t_i); \qquad x(i) \equiv x(t_i);$$

$$L^i(x(i), u(i), a) \equiv L(x(i), u(i), a_i, t_i)(t_{i+1} - t_i);$$

$$F^i(x(i), u(i), a) \equiv x(i) + (t_{i-1} - t_i) F(x(i), u(i), a, t_i).$$

The dynamic programming characterization of the optimal return function $S^i_{\text{opt}}(x, a)$ and $u^i_{\text{opt}}(x, a)$ and $a_{\text{opt}}(x_0)$ is

$$S^N(x, a) = \phi(x, a) \tag{67}$$

$$S^i_{\text{opt}}(x, a) = \operatorname*{Max}_u S^{i+1}_{\text{opt}}(F^i(x, u, a), a) + L^i(x, u, a) \tag{68}$$

$$a_{\text{opt}}(x_0) = \arg \operatorname*{Max}_a S^0_{\text{opt}}(x_0, a, t). \tag{69}$$

The discrete successive sweep method consists of approximating an improved return function $S^i(x, a)$ an improved control law $u^i(x, a)$,[2] around a nominal solution $u^{(0)}(t)$, $x^{(0)}(t)$, $a^{(0)}$, from a second-order perturbation of the dynamic programming equations.

Let us assume that this improved return function $S^j(x(j), a)$ and the corrective feedback law $u^j(x(j), a)$ have been approximated for $j = i + 1, \ldots n - 1$. Now consider

$$R^i(x(i), a; u(i)) = S^{i+1}(F^i(x(i), u(i), a); a) + L^i(x(i), u(i), a). \tag{70}$$

$R^i(x(i); a; u(i))$ is the return obtained from $x(i)$, a and an arbitrary $u(i)$, followed by the improved control law. Equation (69) is now expanded quadratically about the reference solution $u^0(i)$, a^0, $x^0(i)$.

$$R^i(x^0(i) + dx(i), a^0 + da, u^0(i) + du(i))$$

$$\begin{aligned} &= R_0(i) + R_{x_j}(i)\, dx_j(i) + R_{u_j}(i)\, du_j(i) + R_{a_j}(i)\, da_j + \tfrac{1}{2} R_{x_j x_k}(i)\, dx_j(i)\, dx_k(i) \\ &\quad + R_{x_j u_k}(i)\, dx_j(i)\, du_k(i) + R_{x_j a_k}(i)\, dx_j(i)\, da_k + \tfrac{1}{2} R_{u_j u_k}(i)\, du_j(i)\, du_k(i) \\ &\quad + R_{u_j a_k}(i)\, du_j(i)\, da_k + \tfrac{1}{2} R_{a_j a_k}(i)\, da_j\, da_k , \end{aligned} \tag{71}$$

[2] Superscripts refer to time. Superscripts denoting the kth iterate are dropped.

where

$$R_0(i) = S^{i+1}(x^0(i+1), a^0) + L^i(x^0(i), u^0(i), a^0)$$

$$R_{x_j}(i) = S^{i+1}_{x_k}(x^0(i+1), a^0)\, F^i_{k,x_j}(x^0(i), u^0(i), a^0) + L^i_{x_j}(x^0(i), u^0(i), a^0)$$

$$R_{u_j}(i) = S^{i+1}_{x_k}(x^0(i+1), a^0)\, F^i_{k,u_j}(x^0(i), u^0(i), a^0) + L^i_{u_j}(x^0(i), u^0(i), a^0)$$

$$R_{a_j}(i) = S^{i+1}_{a_j}(x^0(i+1), a^0) + S^{i+1}_{x_k}(x^0(i+1), a^0)\, F^i_{k,a_j}(x^0(i), u^0(i), a^0)$$
$$+ L^i_{a_j}(x^0(i), u^0(i), a)$$

$$R_{x_jx_k}(i) = S^{i+1}_{x_mx_n} F^i_{m,x_j} F^i_{n,x_k} + S^{i+1}_{x_m} F^i_{m,x_jx_k} + L^i_{x_jx_k}$$

$$R_{x_ju_k}(i) = S^{i+1}_{x_mx_n} F^i_{m,x_j} F^i_{n,u_k} + S^{i+1}_{x_m} F^i_{m,x_ju_k} + L^i_{x_ju_k}$$

$$R_{x_ja_k}(i) = S^{i+1}_{x_mx_n} F^i_{m,x_j} F^i_{n,a_j} + S^{i+1}_{x_ma_k} F^i_{m,x_j} + S^{i+1}_{x_m} F^i_{m,x_ja_k} + L_{x_ja_k}$$

$$R_{u_ju_k}(i) = S^{i+1}_{x_mx_n} F^i_{m,u_j} F^i_{n,u_k} + S^{i+1}_{x_m} F^i_{m,u_ju_k} + L^i_{u_ju_k}$$

$$R_{a_ku_j}(i) = S^{i+1}_{x_mx_n} F^i_{m,u_j} F^i_{n,a_k} + S^{i+1}_{x_ma_k} F^i_{m,u_j} + S^{i+1}_{x_m} F^i_{m,u_ja_k} + L^i_{u_ja_k}$$

$$R_{a_ja_k}(i) = S^{i+}_{x_mx_n} F^i_{m,a_j} F^i_{n,a_k} + S^{i+1}_{x_ma_j} F^i_{m,a_k} + S^{i+1}_{x_ma_k} F^i_{m,a_j} + S^{i+1}_{x_m} F^i_{m,a_ja_k} + L^i_{a_ja_k}. \tag{72}$$

In the above equation the superscripts $i+1$ denotes that the term is evaluated at $x^0(i+1)$, a^0, $u^0(i+1)$ and superscript i denotes that the term is evaluated at $x^0(i)$, a^0, $u^0(i)$.

The improved control law is obtained by chosing $\delta u(i)$ to maximize the right-hand side of Eq. (71). This results in corrective feedback law

$$\delta u_k(i) = -R^{-1}_{u_ku_j}(i)\,(R_{u_j}(i) + R_{u_jx_n}(i)\,\delta x_n + R_{u_ja_n}(i)\,\delta a_n). \tag{73}$$

By substituting this expression to the right-hand side of Eq. (71), the following second-order approximation to the return function S^i is obtained:

$$S^i(x^0(i) + dx(i), a^0 + da) = S_0{}^i + S^i_{x_j}\, dx_j + S^i_{a_j}\, da_j + \tfrac{1}{2} S^i_{x_jx_k}\, dx_j\, dx_k$$
$$+ S_{x_ja_k}\, dx_j\, da_k + \tfrac{1}{2} S^i_{a_ja_k}\, da_j\, da_k, \tag{74}$$

where

$$S_0{}^i = R_0(i) - R_{u_j}(i)\, R^{-1}_{u_j u_k}(i)\, R_{u_k}(i)$$

$$S^i_{x_j} = R_{x_j}(i) - R_{x_j u_k}(i)\, R^{-1}_{u_k u_n}(i)\, R_{u_n}(i)$$

$$S^i_{a_j} = R_{a_j}(i) - R_{a_j u_k}(i)\, R^{-1}_{u_k u_n}(i)\, R_{u_n}(i)$$

$$S^i_{x_j x_k} = R_{x_j x_k}(i) - R_{x_j u_m}(i)\, R^{-1}_{u_m u_n}(i)\, R_{x_k u_n}(i)$$

$$S^i_{x_j a_k} = R_{x_j a_k}(i) - R_{x_j u_m}(i)\, R^{-1}_{u_m u_n}(i)\, R_{a_k u_n}(i)$$

$$S^i_{a_j a_k} = R_{a_j a_k}(i) - R_{a_j u_m}(i)\, R_{u_m u_n}(i)\, R_{a_k u_n}(i). \tag{75}$$

Equation (75) is a backward set of difference equations by which the first and second partials of the improved return function may be computed backwards. Along the reference solution a^0, $u^0(i)$ $(i = 0,\ldots, N-1)$, $x^0(i)$ $(i = 1, N)$. The value of these derivatives at the final time are given by a perturbation of Eq. (66).

$$S_0{}^N = \phi(x^0(N), a^0)$$

$$S^N_{x_i} = \phi_{x_i}(x^0(N), a^0)$$

$$S^N_{a_i} = \phi_{a_i}(x^0(N), a^0,$$

$$S^N_{x_i x_j} = \phi_{x_i x_j}(x^0(N), a\ )$$

$$S^N_{x_i a_j} = \phi_{x_i a_j}(x^0(N), a^0)$$

$$S^N_{a_i a_j} = \phi_{a_i a_j}(x^0(N), a^0). \tag{76}$$

The correction in a is obtained at the initial time by maximizing the quadratic expansion of S^0

$$\delta a_i = - S^0_{a_i a_j}(S^0_{a_j x_n}\, \delta x_n + S^0_{a_j}). \tag{77}$$

The improved control sequence and trajectory may be now determined from the corrective feedback law in conjunction with the system equations.

A Numerical Example: The Brachistochrone Problem

A classic problem in the calculus of variations is the brachistochrone problem, which may be phrased, as follows:

Find the path by which a bead may slide down a frictionless wire between two specified points in minimal time. The initial velocity of the bead will be assumed to be specified by $V_0 \neq 0$, and the gravitational field has a constant g per unit mass. Restricting the bead to a plane, the position of the particle may be described with the rectangular coordinates (y, z), where z is the verticle component and y is the horizontal component of the position vector. (See Fig. 5.) Assuming that along the optimal path, z may be expressed

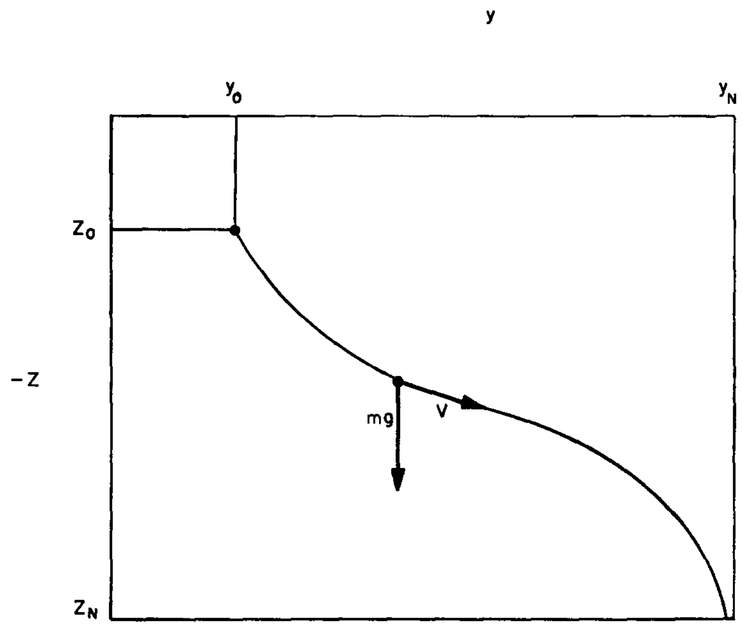

Fig. 5. The brachistochrone problem.

as a piecewise differential function of y, the time of transfer may be expressed by the path integral

$$T = \int_{y_0}^{y_N} \frac{\sqrt{1 + \left(\frac{dz}{dy}\right)^2}}{V(z, y)}\, dy,$$

where y_0 and y_N are the initial and final values of y and $V(x, y)$ is the velocity of the bead.

The conservation of energy implies

$$\frac{V_0^2}{2} + gz_0 = \frac{V^2}{2} + gz.$$

Hence, V may be expressed as terms of z and known parameters

$$V = \sqrt{V_0 + 2g(z_0 - z)}.$$

To simplify the form of the performance index, dimensionless variables are introduced:

$$x = \frac{2g}{V_0^2}(z - z_0), \qquad t = \frac{2g}{V_0^2}(y - y_0), \qquad J = \frac{2gT}{V_0}, \qquad u = \frac{dz}{dy}.$$

The performance index can now become

$$J = \int_0^{t_N} \frac{\sqrt{1 + u^2}}{\sqrt{1 - x}}\, dt.$$

The system equation, relating the "time" derivative of the "state" vector x to "control" u, is simply

$$\frac{dx}{dt} = u.$$

In value of the state at the initial time, $t = 0$, is

$$x(0) = 0.$$

To complete the problem, let the final time t_N and final state x_N be specified by

$$t_N = 1 \qquad x_N = -.5.$$

In order to treat the above constraint, a lagrange multiplier ν is introduced, and the performance index is modified to become

$$J^* = \nu(x_N + .5) + \int_0^t \frac{\sqrt{1 + u^2}}{\sqrt{1 - x}}\, dt.$$

The problem becomes that of minimizing J^* with respects to u, and extremizing J^* with repect ν.

To apply the discrete sweep method to this problem, the performance index may be replaced with the following discrete approximation

$$J^* = \nu(x(N) + .5) + \sum_{i=0}^{N-1} h^i \frac{\sqrt{1 + u(i)^2}}{\sqrt{1 - x(i)}}.$$

The system equation may be approximation by

$$x(i+1) = x(i) + h^i u(i).$$

The boundary conditions for the partial derivatives of the improved return function are:

$$S_x(N) = \nu, \qquad S_\nu(N) = x(N) + .5$$

$$S_{xx}(N) = 0, \qquad S_{x\nu}(N) = 1, \qquad S_{\nu\nu}(N) = 0.$$

These partials may be computed backward using the difference equations given by Eq. (75), with $a \to \nu$ and the R quantities given by

$$R_0^{(i)} = h^i \frac{a(i)}{b(i)}$$

$$R_x^{(i)} = S_x(i+1) + \frac{a(i)\, h^i}{2b^3(i)}$$

$$R_\nu^{(i)} = S_\nu(i+1)$$

$$R_u^{(i)} = h^i \left(S_x(i+1) + \frac{u(i)}{a(i)\, b(i)}\right)$$

$$R_{xx}^{(i)} = S_{xx}(i+1) + \frac{3}{4} h^i \frac{a(i)}{b^5(i)}$$

$$R_{x\nu}(i) = S_{x\nu}(i+1)$$

$$R_{xu}(i) = h^i S_{xx}(i+1) + \frac{3}{4} \frac{a(i)}{b^5(i)}$$

$$R_{\nu\nu}(i) = S_{\nu\nu}(i+1)$$

$$R_{\nu u}(i) = h^i S_{xx}(i+1)$$

$$R_{uu}(i) = h^i \left(h^i S_{xx}(i+1) + \frac{u(i)}{2a(i)\, b^3(i)}\right),$$

where

$$a(i) = \sqrt{1 + u^2(i)}; \qquad b(i) = \sqrt{1 - y(i)}.$$

The corrective feedback law is given by

$$\delta u(i) = -R_{uu}(i)\,(R_{ux}(i)\, dx(i) + R_{u\nu}(i)\, d\nu + R_u(i)).$$

The coefficients appearing in this equation are stored on the backward sweep. $d\nu$ is determined at the initial time from the relation

$$d\nu = -S_{\nu\nu}^{-1}(0)\, S_{\nu}(0).$$

The results of computations are portrayed in Tables 1, 2 and Fig. 6. From Table 1 it can be seen that the fourth and fifth iterates yields good (6 place accuracy) approximations to the optimal solution, which was obtained by an independent calculation with the same program.

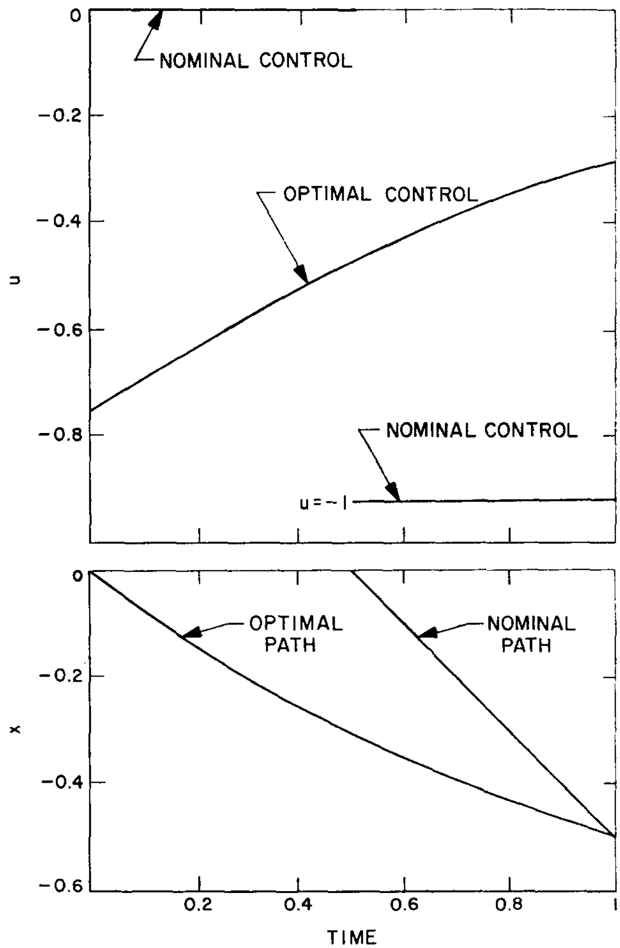

FIG. 6. Results for the brachistochrone.

TABLE 1

DATA FROM BRACHISTOCHRONE CALCULATIONS

	T = time of ascent	$u\|_{t=0}$	$x\|_{t=.8}$	$\nu\|$
Nominal	.95222	0	0	0
1st iterate	.891644	—.3089888	—.2781383	.3430585
2nd iterate	.885181	—.8132391	—.3118965	.2257937
3rd iterate	.885396	—.7856565	—.315550	.2229350
4th iterate	.885432	—.7804296	—.3106541	.2229544
5th iterate	.885422	—.7804270	—.3106546	.2229543
optimal	.885422	—.7804245	—.3106546	.2229543

TABLE 2

BRACHISTOCHRONE PROBLEM; BACKWARD SWEEP QUANTITIES

t	u	x	Z_{uu}^{-1}	λ
0	−.780	0	.495	.624
.10	−.707	−.075	.529	.568
.20	−.642	−.142	.561	.515
.30	−.584	−.204	.590	.468
.40	−.531	−.260	.616	.426
.50	−.482	−.311	.640	.387
.60	−.437	−.357	.662	.351
.70	−.395	−.399	.681	.317
.80	−.355	−.436	.699	.285
.90	−.317	−.470	.715	.255
1.0	−.281	−.500	.728	.233

t	P	R	Q	u_x	u_{Ψ_0}
0	.468	.947	−1.62	−1.41	1.16
.10	.399	.968	−1.44	−1.49	1.28
.20	.330	.989	−1.26	−1.57	1.40
.30	.271	1.00	−1.09	−1.70	1.57
.40	.219	1.01	− .92	−1.90	1.80
.50	.174	1.02	− .75	−2.21	2.13
.60	.133	1.02	− .59	−2.68	2.62
.70	.096	1.02	− .44	−3.49	3.48
.80	.062	1.02	− .29	−5.14	5.11
.90	.021	1.01	− .14	−10.1	10.1
1.0	.000	1.	0.00	$-\infty$	$+\infty$

The above results compared with those obtained by gradient procedures, in [21], demonstrate advantages of the successive sweep. In [21], 14 iterations obtained a nominal control $u(i)$ that approximated the optimal control very poorly. No further improvement was observed after more iterations. In comparison with gradient methods, the successive sweep method provides rapid convergence. Theoretically, the number of accurate decimal places should double with each successive iterations. In Table 1 the accurate digits are underlined. In general, the theoretical speed of convergence seems to be substained.

Numerical Example: An Orbit Transfer Problem

Let us consider a problem more closely related to modern technical problems, the orbit transfer problem. This problem has been considered by many authors. (See [11], [16], [20], for example.) For the purpose of this problem, the rocket is constrained to a plane. The rocket position may be characterized by polar coordinates (r, ϕ), where r is the distance from the center of the gravitational field and ϕ is an angle (Fig. 7). The velocity of the rocket may be expressed in terms of its components in the radial direction u and in the tangential direction v. The rocket will be assumed to have constant low thrust T. The only control that is available to the engineer is the direction of the thrust, which we measure by the angle θ that the rocket makes with the

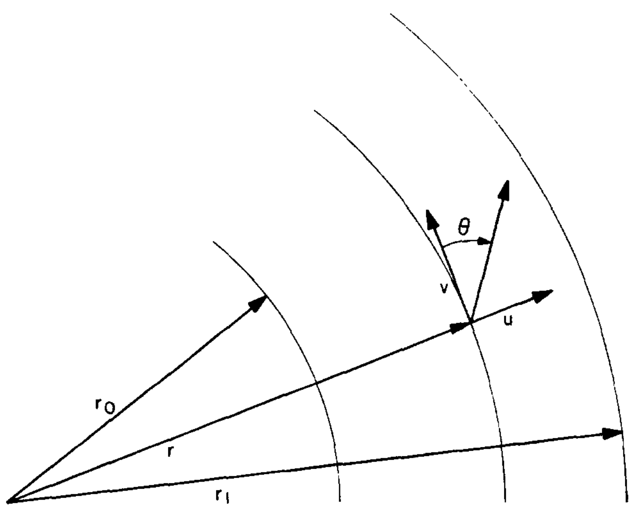

Fig. 7. An orbit transfer problem.

tangent to contours of constant radius. Figure 7 pictorally represents the variables mentioned above. At a particular point in time the dynamics of the rocket are governed by the following rules:

$$\dot{r} = u$$

$$\dot{\phi} = \frac{v}{r}$$

$$\dot{u} = \frac{v^2}{r} - \frac{\mu}{r^2} + \frac{T \sin \theta}{m_0 + \dot{m}(t - t_0)}$$

$$\dot{v} = \frac{uv}{r} + \frac{T \cos \theta}{m_0 + \dot{m}(t - t_0)}.$$

The parameters appearing in the above equations are

$\mu = MG$, M = mass of central body, G = gravitational constant

T = Thrust, m_0 = Initial Time, t_0 = Initial Time

$-\dot{m}$ = Mass Flow/Unit Time.

The problem that shall be considered here is to transfer a spacecraft in a fixed time from a specified circular orbit to another with maximum radius. The following normalized coordinates, shall be introduced

$$x_1 = \frac{r}{r_0}; \qquad x_2 = \frac{u}{\sqrt{\mu/r_0}}; \qquad x_3 = \frac{v}{\sqrt{\mu/r_0}}; \qquad t^1 = t\sqrt{\frac{\mu}{r_0^{\,3}}}; \qquad u = \theta.$$

In the new coordinates, the discrete form of the system equations is

$$x_1(i + 1) = x_1(i) + h^i x_2(i)$$

$$x_2(i + 1) = x_2(i) + h^i \left[\frac{x_3^{\,2}(i)}{x_1(i)} - \frac{1}{x_1^{\,2}(i)} + A(t^i) \sin u(i)\right]$$

$$x_3(i + 1) = x_3(i) + h^i \left[\frac{- x_2(i)\, x_3(i)}{x_1(i)} + A(t^i) \cos u(i)\right].$$

$A(i) = a/(1 - bt_i^{\,1})$. To agree with Moyer and Pinkham [16], $a = 1.405$ and $b = .07487$ shall be chosen. The initial boundary conditions become

$$x_1(0) = 1; \qquad x_2(0) = 0; \qquad x_3(0) = 1.$$

The terminal boundary conditions are

$$\Psi_{0,1} = x_2(N) = 0, \qquad \Psi_{0,2} = x_3(N) - 1\sqrt{x_1(N)} = 0.$$

The final time $t_N = 3.32$ is chosen to agree with the earth—Mars minimal time according to the results obtained by Moyer and Pinkham [31].

The modified performance index becomes

$$J^* = x_1(N) + \nu_1(x_2(N)) + \nu_2\left(x_3(N) - \frac{1}{x_1(N)^{\frac{1}{2}}}\right).$$

The problem becomes that of maximizing J^* with respect to $u(i)$ ($i = 1, N$) and extremizing J^* with respect to ν_1 and ν_2.

Numerical results were obtained using the discrete successive sweep algorithm.

One hundred equal time intervals were employed. Several nominals were used that yielded convergence to the optimal solution. The number of steps required to obtain three place accuracy ranged from 13-18, depending upon the initial nominal. About 2 seconds of IBM computer time was required per iteration.

The converged solution agrees with the results of others. Roughly three place accuracy was obtained. The trajectory is graphed in Fig. 8. The direction of the thrust is indicated at eleven equally space time intervals. The thrust program is graphed in Fig. 9 and is tabulated in Table 3. Note that typical of low-thrust trajectories, the thrust has a positive radial component during the first half of the journey and the rapidly reverses itself to yield a negative radial component on the last half of the trip.

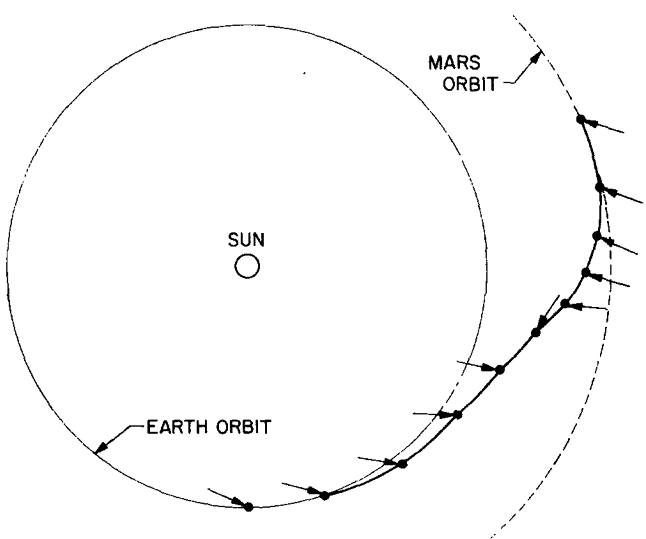

FIG. 8. An optimal orbit transfer.

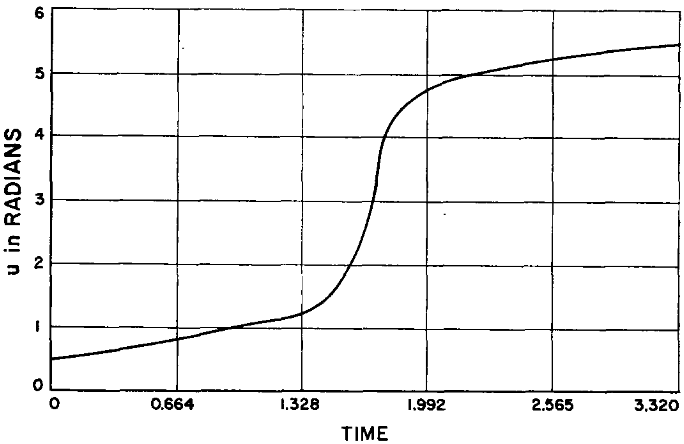

FIG. 9. An optimal thrust angle program.

TABLE 3

FEEDBACK GAINS ON THE OPTIMAL

t	$u_{x(1)}$	$u_{x(2)}$	$u_{x(3)}$	u_{Ψ_0}	u_{Ψ_1}
0	1.62	.937	1.33	−.537	1.02
.166	1.65	.920	1.40	−.604	1.06
.332	1.70	.935	1.52	−.699	1.16
.498	1.76	.986	1.67	−.828	1.28
.664	1.83	1.08	1.89	−1.00	1.43
.830	1.93	1.24	2.18	−1.29	1.61
.996	2.06	1.49	2.62	−1.61	1.04
1.162	2.30	2.00	2.39	−2.21	2.16
1.328	2.84	2.92	4.89	−3.39	2.83
1.494	4.80	6.75	9.60	−7.39	5.88
1.660	21.88	52.9	44.2	−47.03	54.52
1.826	10.0	90.8	7.30	−9.40	75.00
1.992	−14.0	−76.9	−26.4	64.26	−80.30
2.158	−11.0	−83.4	−20.0	70.00	−86.70
2.324	−5.50	−104.0	−3.64	88.0	−103.1
2.490	−.07	−132.0	15.0	111.0	−123.0
2.656	8.61	−183.0	44.6	162.7	−172.3
2.822	25.0	−280.0	112.0	250.0	−230.0
2.988	75.2	−548.0	292.0	510.6	−495.4
3.154	315.0	-1.42×10^3	$+1.2\times 10^3$	1.8×10^3	-1.8×10^3
3.320	$+\infty$	$-\infty$	$+\infty$	$+\infty$	$-\infty$

TABLE 4

u, $x(1)$, $x(2)$, $x(3)$ ON THE OPTIMAL

t	u	$x(1)$	$x(2)$	$x(3)$
0	.439	1.00	.0	1.00
.166	.500	1.00	.008	1.02
.332	.590	1.00	.030	1.03
.498	.700	1.01	.058	1.04
.664	.804	1.02	.093	1.05
.830	.933	1.04	.133	1.05
.996	1.08	1.06	.176	1.04
1.162	1.24	1.10	.224	1.02
1.328	1.42	1.14	.261	.989
1.494	1.66	1.18	.298	.950
1.660	2.39	1.23	.330	.905
1.826	4.36	1.29	.320	.847
1.992	4.73	1.34	.285	.810
2.158	4.88	1.36	.269	.800
2.324	5.00	1.42	.206	.711
2.490	5.10	1.45	.167	.766
2.656	5.18	1.48	.124	.762
2.822	5.25	1.50	.093	.766
2.988	5.32	1.51	.061	.775
3.154	5.38	1.52	.036	.788
3.32	5.45	1.52	.0	.806

The optimal feedback gains along the optimal are tabulated in Table 4. These are the partial derivatives of the control with respect to the state and constraint levels $\Psi_{0,1}$; $\Psi_{0,2}$. These derivatives are large on the first half of the trajectory, indicating that the optimal control function rapidly changes in this region.

Several difficulties hindered convergence. One of the main difficulties was that the Hamilton $H = S_{x_i} f^i + L$ has both a maximum and minimum with respect to u.

In this case, the Hamiltonian has the form

$$H = c_0(x, t) + c_1(x, t) \sin u + c_2(x, t) \cos u.$$

Since without loss of generality u may be restricted to lie on the interval $[0, 2\pi]$, it can be seen that the Hamiltonian has both a unique maximum and minimum. If the nomial control lies in the wrong region, such as point A in Fig. 10, then the successive sweep method will actually seek out the minimum value rather than the maximum value. Another difficulty will occur if

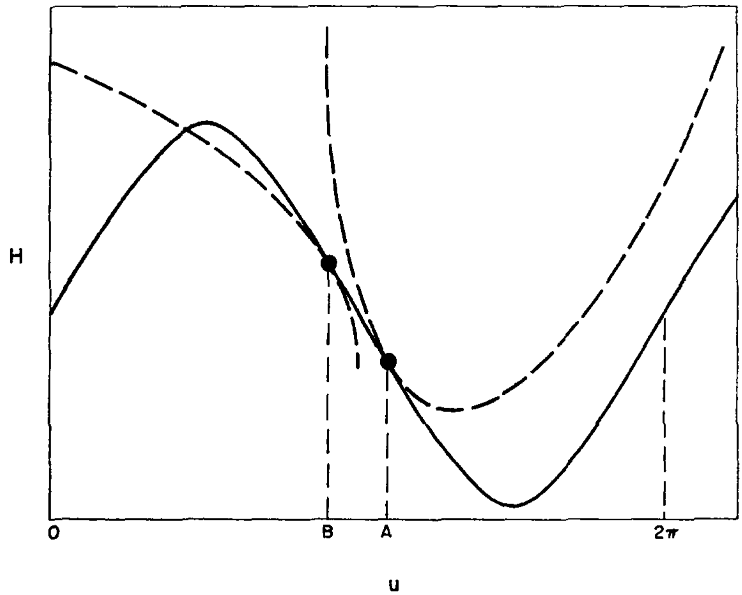

FIG. 10. Hamiltonian for the orbit transfer problem.

the nominal is near the inflextion point, such as point B. In this case the quadratic approximation is poor. To advoid these difficulties the corrective feedback law is replaced by

$$\delta u(i) = - \mid Z_{uu}^{(i)} + A^{(i)} \mid^{-1} [Z_u(i) + Z_{ux}(i)\, \delta x(i) + Z_{uv}(i)\, \partial \nu],$$

where $A(i) > 0$ is chosen to ensure that the correction in the control is not too large. As the optimal solution is approached $A(i)$ can be set to zero, thus ensuring quadratic convergence on the final iterations.

A similar scheme was employed in obtaining corrections to the Lagrange multipliers ν.

REFERENCES

1. S. R. MCREYNOLDS AND A. E. BRYSON. A Successive sweep method for solving optimal programming problems. Joint Automatic Control Conference, Troy, New York, June 1965, Harvard University Craft Lab Report, No. 463, 1965.
2. S. K. MITTER. Successive approximation methods for the solution of optimal control problems. *In* "Automatica" Vol. 3, pp 135-149. Pergamon Press, New York, 1966.
3. C. W. MERRIAM, III. "Optimization Theory and the Design of Feedback Control Systems." McGraw-Hill, New York, New York, 1964.

4. C. W. Merriam. An algorithm for the iterative solution of a class of two-point boundary value problems. *SIAM J. Control, Ser. A.*, **2** (1964), 1-10.
5. R. W. Bellman and R. E. Kalaba. "Quasilinearization and Nonlinear Boundary-Value Problems." American Elsevier, New York, 1965.
6. R. Bellman and R. Kalaba. Dynamic programming, invariant imbedding and quasilinearization: Comparisons and interconnections. *In* "Computing Methods in Optimization Problems" (Balakrishnan and Neustadt, eds.), pp. 135-145. Academic Press, New York, 1964.
7. L. V. Kantorovich. On Newton's method. *Trudy. Mat. Inst. Steklov* **28**, (1949), 104-144.
8. M. R. Hestenes. Numerical methods obtaining solutions of fixed end point problems in the calculus of variations. Rand Report RM-102. August, 1949.
9. J. L. Stein. On methods for obtaining solutions of fixed end point problems in the calculus of variations. *J. Res. Nat. Bur. Standard* **50** (May 1953).
10. J. V. Breakwell, J. L. Speyer and A. E. Bryson. Optimization and control of nonlinear systems using the second variation. *SIAM J. Control, Ser. A.*, **1** (1963), 193-223.
11. H. J. Kelley, R. Kopp, and G. Moyer. A trajectory optimization techniques based upon the theory of the second variation. AIAA Astrodynamics Conference, Yale University, New Haven, Connecticut, August 19-21, 1963.
12. C. D. Van Dine, W. R. Fimple, and J. N. Edelbaum. Application of a finite-difference Newton-Raphson algorithm to problems of low thrust trajectory optimization. *AIAA Paper 65-648* Astrodynamics Specialist Conference, September 1965.
13. W. E. Milne. "Numerical Solutions of Differential Equations." Wiley, New York, 1953.
14. C. T. Leondes and R. A. Niemann. Optimization of aerospace re-entry vehicle trajectories through independent control of lift and drag. *J. Spacecraft Rockets* **3** (1966), 135-149.
15. R. E. Kopp and R. McGill. Several trajectory optimization techniques. Part I. *In* "Computing Method In Optimization Problems" (Balakrishnan and Neustadt, eds.), pp. 65-89. Academic Press, New York, 1964.
16. H. G. Moyer and G. Pinkham. Several trajectory optimization techniques. Part II. *In* "Computing Method in Optimization Problems" (Balakrishnan and Neustadt, eds.), pp. 91-105. Academic Press, New York 1964.
17. R. E. Bellman. "Dynamic Programming." Princeton University Press, Princeton, New Jersey, 1957.
18. R. Kalaba. On nonlinear differential equations, the maximum operation, and monotone convergence. *J. Math. Mech.* **8** (1959), 519-574.
19. A. E. Bryson and W. F. Denham. "A Steepest-Ascent Method for Solving Optimum Programming Problems." *J. Appl. Mech.* 292, pp. 247-257, 1962.
20. J. H. Kelley. Method of gradients. *In* "Optimization Techniques" (G. Leitmann, ed.). Academic Press, New York, 1962.
21. A. Jazwinski. Optimal trajectories and linear control of nonlinear systems. *AIAA J.* **2** (1964), 1371-1379.
22. I. M. Gelfand and S. V. Fomin. "Calculus of Variations." Chap. 6. Prentice Hall, Englewood cliffs, New Jersey, 1963.
23. D. Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *Internat. J. Control* **3** (1966), 85-95.
24. S. R. McReynolds. A successive sweep method for solving optimal programming problems. Ph. D. Thesis, Harvard University, 1965.

25. S. Dreyfus. Dynamic programming and the calculus of variations. *J. Math. Anal. Appl.* 1 (1960), 228-235.
26. S. Dreyfus. The numerical solution of variational problems. *J. Math. Anal. Appl.* 5 (1962), 30-45.
27. P. Henrici. "Discrete Variable Methods in Ordinary Differential Equations." Wiley, New York, 1962.
28. S. E. Dreyfus and J. R. Elliot. "An Optimal Linear Feedback Guidance Scheme." Rand Memorandum, RM-3604-PR, May 1963.
29. H. Rauch, F. Tung, and C. Striebel. On maximum likelihood estimates for linear systems. *AIAA J.* 3 (1965), 1445-1450.