AAS/AIAA Space Flight Conf.
Pasadena, California, 1993

N/S
53-65 145604
N93-22011
p. 11

NEIGHBORING SUBOPTIMAL CONTROL FOR VEHICLE GUIDANCE

David G. Hull* and Michael J. Nowak†

Department of Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin, Austin, TX 78712

The neighboring optimal feedback control law is developed for systems with a piecewise linear control for the case where the optimal control is obtained by nonlinear programming techniques. To develop the control perturbation for a given deviation from the nominal path, the second variation is minimized subject to the constraint that the final conditions be satisfied (neighboring suboptimal control). This process leads to a feedback relationship between the control perturbation and the measured deviation from the nominal state. Neighboring suboptimal control is applied to the lunar launch problem. Two approaches, single optimization and multiple optimization, for calculating the gains are used, and the gains are tested in a guidance simulation with a mismatch in the acceleration of gravity. Both approaches give acceptable results, but multiple optimization keeps the perturbed path closer to the nominal path.

INTRODUCTION

In order to develop the neighboring optimal guidance law for a dynamical system, it is first necessary to obtain the optimal control. Currently, most trajectory optimization (see Ref. 1, for example) is accomplished by restricting the class of control functions to some subclass, say piecewise linear functions (suboptimal control). Then, the control parameters are the nodes of a piecewise linear function, and the suboptimal control is found by applying nonlinear programming methods. The subject of this paper is neighboring optimal control for systems with piecewise linear controls, or neighboring suboptimal control, and its application to vehicle guidance.

In Refs. 2 and 3, the neighboring suboptimal control problem is formulated as a free final time problem and applied to the lunar launch problem. This formulation requires an iteration at each sample point to find the normalized time. In this paper, neighboring suboptimal control is formulated as a fixed final time problem and applied

*M. J. Thompson Regents Professor
†Graduate Student, Major, USAF

to the lunar launch problem. While this problem is a minimum time problem, it can be converted to a "fixed final time" problem by using the horizontal component of velocity, whose final value is fixed, as the variable of integration.

Two approaches for computing the control gains are presented. In the single optimization approach, the nominal suboptimal control is viewed as a sequence of reduced-node suboptimal controls to the final constraint manifold. Hence, the quality of the suboptimal control diminishes along the flight path. In the multiple optimization approach, a new full-node suboptimal control is computed from each node of the nominal suboptimal trajectory to the final constraint manifold. Hence, the quality of the suboptimal control along the flight path is maintained.

After the suboptimal control problem and the neighboring suboptimal control problem are summarized, the lunar launch problem is defined. Then, the single optimization and multiple optimization approaches are used to compute the gains which are, in turn, tested in a simulation with a mismatch in the acceleration of gravity. Finally, some conclusions are reached about the use of these two approaches.

SUBOPTIMAL CONTROL PROBLEM

The fixed final time optimal control problem being considered here is to find the control history $u(\tau)$ which minimizes the performance index

$$J = \phi(x_f) \tag{1}$$

subject to the state differential equations

$$\frac{dx}{d\tau} = f(\tau, x, u), \tag{2}$$

the prescribed initial conditions

$$\tau_0 = \tau_{0_s}, \quad x_0 = x_{0_s}, \tag{3}$$

and the prescribed final conditions

$$\tau_f = 1, \qquad \psi(x_f) = 0. \tag{4}$$

Here, the time has been normalized by the final time, that is, $\tau = t/t_f$. This optimal control problem is converted into a suboptimal control problem (parameter optimization problem) by assuming that controls are piecewise linear, meaning that the unknowns become the nodes of the linear control segments.

If a denotes the unknown parameter vector which for one control is written as $a^T = [u_1, u_2, \ldots, u_r]$, the differential equation (2) and its boundary conditions can be rewritten as

$$\frac{dx}{d\tau} = g(\tau, x, a) \tag{5}$$

$$\tau_0 = \tau_{0_s}, \quad x_0 = x_{0_s}, \quad \tau_f = 1. \tag{6}$$

Given a, these equations can be integrated to obtain $x_f = x_f(a)$ so that $J = \phi[x_f(a)] = F(a)$ and $\psi[x_f(a)] = C(a)$ Then, the suboptimal control problem is to find the parameter vector a which minimizes the performance index $J = F(a)$ subject to the constraint $C(a) = 0$.

To solve the suboptimal control problem analytically, the augmented performance index $J' = F(a) + \nu^T C(a) \triangleq G(a, \nu)$ is formed. The first variation conditions are $G_a = 0$ and $C = 0$ which determine a and ν. The second variation becomes $\delta^2 J' = \delta a^T G_{aa} \delta a > 0$ where $C_a \delta a = 0$. δa can be divided into dependent and independent parts; the dependent parts can be eliminated; and the second variation condition becomes the positive definiteness of a matrix.

At this point, it is assumed that the suboptimal control problem is solved by using a nonlinear programming code (see Ref. 1, for example), and the next step is to find the neighboring suboptimal control.

NEIGHBORING SUBOPTIMAL CONTROL

The solution of the suboptimal control problem gives nominal control and state histories to be followed by the vehicle. However, because of modeling errors, the vehicle deviates from the nominal state. Hence, it is desired to find the neighboring suboptimal control perturbation which enables the vehicle to operate in the neighborhood of the nominal trajectory. The general philosophy is to find the control perturbation which minimizes the increase in the performance index while satisfying the prescribed final conditions.

Since the first variation vanishes along the suboptimal path, the increase in the performance index is the second variation

$$\Delta J = \frac{1}{2} \delta a^T G_{aa} \delta a \tag{7}$$

where the second derivative matrix G_{aa} can be computed numerically. The elements of δa are not independent but are constrained by the need to satisfy the final conditions

$$\delta\psi = \psi_{x_f} \delta x_f = 0. \tag{8}$$

The variation of the state equation (5) leads to the differential equation

$$\frac{d}{d\tau} \delta x = g_x \, \delta x + g_a \, \delta a \tag{9}$$

which must be solved subject to the boundary conditions

$$\begin{aligned} \tau_0 &= \tau_{0_s}, \qquad & \delta x_0 &= \delta x_{0_s} \\ \tau_f &= 1, \qquad & \psi_{x_f} \delta x_f &= 0. \end{aligned} \tag{10}$$

Next, the solution of Eq. (9) is assumed to have the transition matrix form

$$\delta x = \Phi \delta x_f + \Psi \delta a \tag{11}$$

where

$$\Phi_f = I, \qquad \Psi_f = 0 \tag{12}$$

to guarantee that $\delta x_f = \delta x_f$. Then, substituting Eq. (11) into Eq. (9) and equating like coefficients leads to the differential equations

$$\begin{aligned} \Psi' &= g_x \Psi + g_a \\ \Phi' &= g_x \Phi \end{aligned} \tag{13}$$

which must be solved subject to the boundary conditions (12). Once Φ and Ψ have been obtained, Eq. (11) can be used.

To satisfy the final condition (10), Eq. (11) is evaluated at τ_0 and rewritten as

$$\delta x_f = \Phi_0^{-1} \delta x_0 - \Phi_0^{-1} \Psi_0 \delta a \tag{14}$$

Then, Eq. (10) leads to

$$\psi_{x_f} \Phi_0^{-1} \Psi_0 \delta a - \psi_{x_f} \Phi_0^{-1} \delta x_0 = 0 \tag{15}$$

which is the constraint on the control node perturbation, δa, imposed by the final condition.

The last step is to minimize ΔJ as given by Eq. (7) with respect to δa subject to the constraint (15). Standard parameter optimization methods lead to

$$\delta a = K_0 \delta x_0 \tag{16}$$

where the gain K_0 is given by

$$K_0 = G_{aa}^{-1} \Psi_0^T \Phi_0^{-T} \psi_{x_f}^T (\psi_{x_f} \Phi_0^{-1} \Psi_0 G_{aa}^{-1} \Psi_0^T \Phi_0^{-1} \psi_{x_f}^T)^{-1} \psi_{x_f} \Phi_0^{-1}. \tag{17}$$

The computation of the gains can be checked by observing that $K_0 = \partial a^{opt} / \partial x_0$ and using numerical differentiation. Given a suboptimal control and state history, a perturbation in the state is introduced at some node, and the suboptimal control from that perturbed state to the final constraint manifold is computed. The gains are computed as $K_0(i,j) = \Delta a(i) / \Delta x_0(j)$ where Δa is the change in the suboptimal control caused by the change in the state.

The application of neighboring suboptimal control as a guidance law is discussed in terms of the lunar launch problem which is defined in the next section.

LUNAR LAUNCH PROBLEM

The lunar launch problem is to insert a payload in circular lunar orbit over a flat moon using a rocket with constant thrust acceleration. While this is a free final time problem, it can be converted to a "fixed final time" problem by choosing the horizontal component of velocity as the variable of integration. With the variable of integration normalized as $\bar{u} = (u - u_0)/(u_f - u_0)$, the optimal control problem is stated as follows: Find the thrust inclination history $\theta(\bar{u})$ which minimizes the performance index

$$J = t_f \tag{18}$$

subject to the equations of motion

$$\frac{dt}{d\bar{u}} = \frac{(u_f - u_0)}{\alpha \cos\theta} \tag{19}$$

$$\frac{dy}{d\bar{u}} = \frac{(u_f - u_0)v}{\alpha \cos\theta} \tag{20}$$

$$\frac{dv}{d\bar{u}} = \frac{(u_f - u_0)(\alpha \sin\theta - g)}{\alpha \cos\theta} \tag{21}$$

and the boundary conditions

$$\bar{u}_0 = 0,\ t_0 = 0,\ y_0 = 0,\ v_0 = 0\ , \tag{22}$$

$$\bar{u}_f = 1,\ y_f = 50{,}000\ ft,\ v_f = 0\ ft/sec. \tag{23}$$

In these equations, $\alpha = 20.8$ ft/sec^2 is the thrust acceleration, $g = 5.32$ ft/sec^2 is the acceleration of gravity, $u_f = 5444$ ft/sec is the satellite speed, and $u_0 = 0$ ft/sec.

For a piecewise linear control involving nine nodes, the nonlinear programming code VF02AD gives the following suboptimal control in degrees:

$$\begin{array}{lll} \theta_1 = 26.01 & \theta_2 = 23.31 & \theta_3 = 20.51 \\ \theta_4 = 17.65 & \theta_5 = 14.86 & \theta_6 = 11.90 \\ \theta_7 = 8.98 & \theta_8 = 6.01 & \theta_9 = 3.03 \end{array} \tag{24}$$

Two approaches for applying neighboring suboptimal control are discussed: the single optimization approach and the multiple optimization approach. Here, $u_0 = 0$ for the single optimization approach or a node value for the multiple optimization approach. In Ref. 4, neighboring suboptimal control results are presented for the cases where there is a thrust acceleration or a gravity modeling error. Only the gravity case is discussed here because it has the largest errors.

SINGLE OPTIMIZATION APPROACH

In this approach, the suboptimal control from node 1 to node 9 is considered to be a sequence of reduced-node suboptimal controls. In other words, the suboptimal control from node 1 to node 9 is a nine-node suboptimal control. From node 2 to node 9, it is an eight-node suboptimal control; from node 3 to node 9, it is a seven-node suboptimal control; and so on. At node 8, there are only two nodes available, but these are enough to satisfy the boundary conditions (no optimization). Next, the 9×3 gain matrix, K_0 in Eq. (17), is computed backward to each node and saved. The gains associated with the state t are all zero because there is no condition imposed on t_f. Hence, the gain matrix, reduces to a 9×2 matrix, and the states are now $\delta x_0^T = [\delta y_0 \;\; \delta v_0]$.

If the state perturbation occurs at node 8, only δa_8 is of interest for a sample and hold system. Hence, only the gains $K_0(8,1)$ and $K_0(8,2)$ are needed. Similarly, if the state perturbation occurs at node 7, only $K_0(7,1)$ and $K_0(7,2)$ are needed to compute δa_7, and so on. For a state perturbation between nodes, the gains are obtained by linearly interpolating the gains at adjacent nodes. To have gains over the last or 8th interval, the gains at nodes 7 and 8 are linearly extrapolated. In conclusion, only the gains $K_0(i,1)$ and $K_0(i,2)$ where $i = 1,\ldots,9$ need to be stored in the flight computer.

This approach to neighboring extremal control is tested by introducing a $\mp 5\%$ error in the acceleration of gravity. In other words, the true value of g is taken to be $\mp 5\%$ different than the value being used in the computation of the gains. Gains are computed and stored at every node or at every $0.125\bar{u}$ for 9 nodes (Table 1). The sample points are assumed to occur at every integration step of the simulation. Here, 64 integration steps are used so that a sample point occurs every $0.015625\bar{u}$. The nominal states are obtained by numerical integration of the equations of motion subject to the suboptimal control (24). The true states are obtained by integrating the equations of motion with the true acceleration of gravity subject to the neighboring suboptimal control. At each sample point, the true states and nominal states are differenced and the differences multiplied by the gains to obtain the control perturbation. The control perturbation is assumed constant over the sample period, but it is added to the piecewise-linear nominal control. Hence, the applied control varies linearly over the sample period.

The deviations between the true states and the desired values at the final point are presented in Table 2 along with the values which would have been obtained had the nominal control (24) been applied open loop. On a relative basis, the improvement is substantial. However, a statement about the absolute quality of the closed-loop results cannot be made without some performance criteria, say for example, that the vehicle has only so much ΔV to meet the desired final conditions precisely.

Time histories of the deviations are shown in Fig. 1. Throughout the trajectory, the deviations are small, but they do not go to zero at the end. There are two possible reasons for this: (a) the quality of the suboptimal trajectory as the vehicle moves along its path and (b) the size of the last interval over which the gains are

Table 1

9-NODE SINGLE OPTIMIZATION GAINS

Node	y Gain	v Gain
1	-0.369E-5	-0.673E-3
2	-0.289E-5	-0.462E-3
3	-0.385E-5	-0.521E-3
4	-0.573E-5	-0.640E-3
5	-0.940E-5	-0.831E-3
6	-0.179E-4	-0.118E-2
7	-0.461E-4	-0.201E-2
8	-0.267E-3	-0.581E-2
9	-0.488E-3	-0.961E-2

obtained by extrapolation.

Both of these concerns can be addressed by increasing the number of nodes. Hence, the computations have been repeated for 17 nodes. The final point deviations are presented in Table 2 and show considerable improvement relative to those of 9 nodes. However, the deviation histories do not change appreciably relative to Fig. 1.

Table 2

DEVIATION FROM DESIRED FINAL CONDITIONS

% Change in g	State	Open Loop	Closed Loop 9 Node Single Opt.	Closed Loop 17 Node Single Opt.	Closed Loop 9 Node Mult. Opt.
-5.0	y_f	9891.024	65.178	20.959	48.137
	v_f	72.540	-2.705	-1.977	-1.566
+5.0	y_f	-9891.023	-63.989	-19.917	-47.891
	v_f	-72.540	2.616	1.832	1.542

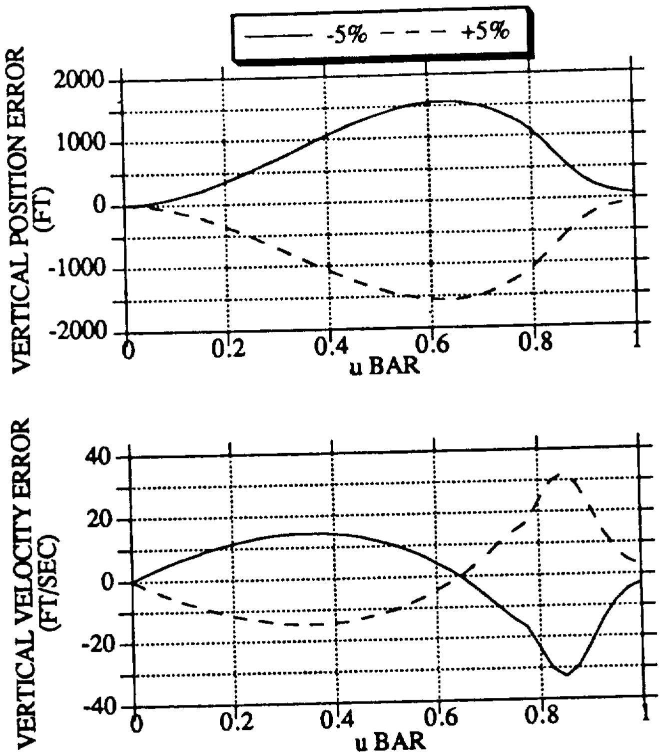

Figure 1: 9-Node Single Optimization Deviation Histories

MULTIPLE OPTIMIZATION APPROACH

In an attempt to improve just the quality of the neighboring suboptimal control, a 9-node suboptimal control to the final constraint manifold is computed from each node of the nominal trajectory (Fig. 2), and the gains are computed for each subtrajectory by Eq. (17). These gains are presented in Table 3 and are seen to be larger than those of the single optimization approach and uniformly increasing toward the final point. The use of these gains in the simulation with a $\mp 5\%$ mismatch in the acceleration of gravity leads to the final results of Table 2. These closed-loop results are somewhat better than those of the single optimization results for 9 nodes.

The time histories of the deviations are shown in Fig. 3. Overall these deviations are smaller than those of single optimization. Again, the fact that the deviations do not go to zero can probably be attributed to the extrapolation of the gains at nodes 7 and 8 over the last interval.

DISCUSSION AND CONCLUSIONS

Two approaches for computing the gains for the neighboring suboptimal control guidance law have been tested in a simulation of a lunar launch vehicle: the single optimization approach and the multiple optimization approach. In both approaches,

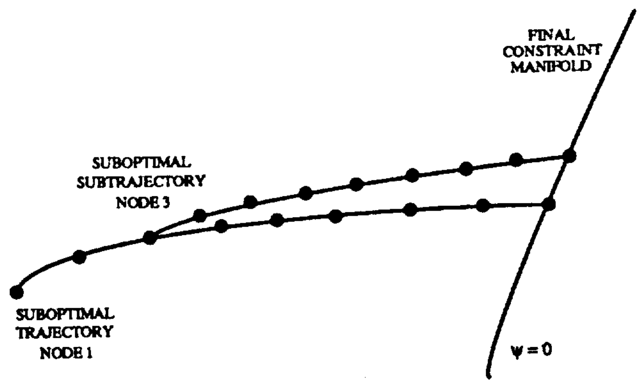

Figure 2: Multiple Optimization Approach

Table 3

9-NODE MULTIPLE OPTIMIZATION GAINS

Node	y Gain	v Gain
1	-0.369E-5	-0.673E-3
2	-0.494E-5	-0.780E-3
3	-0.688E-5	-0.921E-3
4	-0.101E-4	-0.112E-2
5	-0.161E-4	-0.141E-2
6	-0.290E-4	-0.190E-2
7	-0.661E-4	-0.288E-2
8	-0.267E-3	-0.581E-2
9	-0.468E-3	-0.874E-2

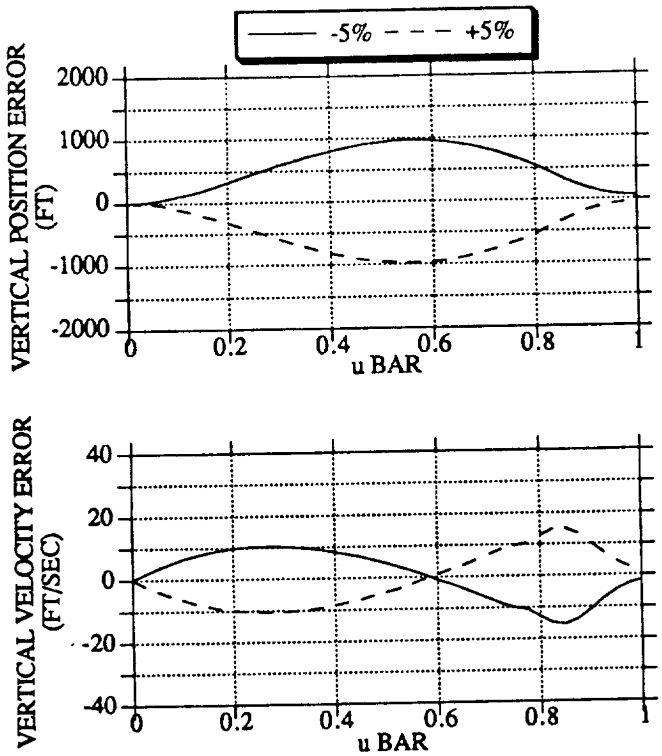

Figure 3: 9-Node Multiple Optimization Deviation Histories

a suboptimal control and trajectory with evenly spaced nodes is used as a base, and the number of gains which must be stored is very small.

For single optimization, that part of the suboptimal trajectory from a generic node to the final constraint manifold is thought of as a reduced-node suboptimal trajectory. Hence, the control becomes less optimal (fewer nodes) toward the end of the trajectory and eventually runs out of nodes for satisfying the boundary conditions. However, the gains generated by this approach produce good results in a guidance simulation. The final point results can be improved by increasing the number of nodes.

The multiple optimization approach is to find a full-node suboptimal control from each node of the nominal path to the final constraint manifold. Gains generated from these subtrajectories are larger than those of the single optimization approach, are uniformly increasing toward the final point, and produce better guidance results, that is, the deviations are smaller along the path.

From these results, it is apparent that the single optimization approach can satisfactorily meet the final conditions. On the other hand, if the perturbed trajectory is to lie close to the nominal trajectory, the quality of the optimization along the path must be improved. Multiple optimization does this, but the amount of computation is considerably more than that of single optimization.

ACKNOWLEDGEMENT

This research has been supported in part by NASA LRC Grant NAG-1944 monitored by Dr. Daniel D. Moerder.

REFERENCES

1. Shaver, D. A., and Hull, D. G., "Advanced Launch System Trajectory Optimization Using Suboptimal Control," Paper No. 90-3413, *Proceedings of the AIAA GNC Conference*, Portland, Oregon, 1990, pp. 892–901.

2. Hull, D. G., and Helfrich, C. E., "Neighboring Extremal Guidance for Systems with a Piecewise Linear Control," AIAA Third International Space Planes Conference, Orlando, Florida, December, 1991.

3. Hull, D.G., "Neighboring Suboptimal Control", American Control Conference, Chicago, Illinois, June, 1992.

4. Nowak, M.J., "Neighboring Extremal Guidance for Systems with Piecewise Linear Control Using Multiple Optimization", M.S. Thesis, University of Texas at Austin, August, 1992.