NASA-CR-192087

145601
P. 34
(1+2)

Final Technical Report

National Aeronautics and Space Administration

Grant No. NAG – 1 – 944

# ROBUST NEIGHBORING OPTIMAL GUIDANCE FOR THE ADVANCED LAUNCH SYSTEM

by

David G. Hull

Guidance and Control Group

Department of Aerospace Engineering and Engineering Mechanics

The University of Texas at Austin

January, 1993

(NASA-CR-192087) ROBUST NEIGHBORING OPTIMAL GUIDANCE FOR THE ADVANCED LAUNCH SYSTEM (Texas Univ.) 34 p

N93-22009
--THRU--
N93-22011
Unclas

G3/15 0145601

# Robust Neighboring Optimal Guidance for the Advanced Launch System

David G. Hull*

In recent years, optimization has become an engineering tool through the availability of numerous successful nonlinear programming codes. Optimal control problems are converted into parameter optimization (nonlinear programming) problems by assuming the control to be piecewise linear, making the unknowns the nodes or junction points of the linear control segments. Once the optimal piecewise linear control (suboptimal) control is known, a guidance law for operating near the suboptimal path is the neighboring optimal piecewise linear control (neighboring suboptimal control). Research conducted under this grant has been directed toward the investigation of neighboring suboptimal control as a guidance scheme for an advanced launch system. The list of references is a list of papers presented at technical meetings; these papers are included at the end of the report.

The first step is to obtain the optimal piecewise linear control for the advanced launch system, upon which the neighboring optimal piecewise linear control is based. These results have been obtained by using a nonlinear programming code based on recursive quadratic programming with numerical partial derivatives and are reported in Ref. 1. In an effort to improve the results obtained by nonlinear programming, the suboptimal control problem is solved by the shooting method which requires analytical derivatives (Ref. 2). By guessing the control history, the shooting method is completely desensitized to the guesses of the initial

* M. J. Thompson Regents Professor in ASE/EM.

Lagrange multipliers. Ref. 2 shows that the optimal piecewise linear control can be a good approximation of the actual optimal control.

Once the suboptimal control has been obtained, the next step is to develop the neighboring suboptimal control for guidance about the suboptimal path. Since this is a completely new research area, a simpler launch model, the lunar launch problem, has been used to determine the feasibility of this guidance strategy.

First results are reported in Ref. 3. Given a perturbation in the state from the state corresponding to the suboptimal control, the control parameter perturbations are obtained by minimizing the increase in the performance index (second variation) subject to the constraint that the final conditions are satisfied. This process leads to a set of gains which multiply the state perturbations to get the control parameter perturbations. At this point, time is the running variable, and while the results are satisfactory, they give some indication that the method has not yet been applied properly.

Refs. 4 and 5 are further steps to clarify the fundamental issues of this guidance approach. In Ref. 5, the problem is reduced to a "fixed final time" problem by using the horizontal component of velocity as the variable of integration. These results indicate that neighboring suboptimal control is now being formulated and applied correctly.

Continuing work is associated with using the time as the running variable since this is the way most guidance systems operate.

Finally, a list of master's degrees awarded during this research effort is given in Table 1.

## References


1. Shaver, D. A., and D. G. Hull, "Advanced Launch System Trajectory Optimization Using Suboptimal Control", Proceedings of the AIAA Guidance, Navigation, and Control Conference, Portland, Oregon, August, 1990.

2. Hull, D. G, and J. J. Sheen, "A Shooting Approach to Suboptimal Control, Proceedings of the American Control Conference, Boston, Massachusetts, June, 1991.

3. Hull, D. G., and C. E. Helfrich, "Neighboring Extremal Guidance for Systems with Piecewise Linear Control", Presented at the AIAA Third Aerospace Planes Conference, Orlando, Florida, December, 1991.

4. Hull, D. G., "Neighboring Suboptimal Control", Proceedings of the American Control Conference, Chicago, Illinois, June, 1992.

5. Hull, D. G., and M. J. Nowak, "Neighboring Suboptimal Control for Vehicle Guidance", Proceedings of the Third Annual AAS/AIAA Space Flight Mechanics Conference, Pasadena, California, February, 1993.

**Table 1**

MASTER OF SCIENCE DEGREES AWARDED

1. Douglas A. Shaver, "Advanced Launch System Trajectory Optimization Using Suboptimal Control, December, 1989.

2. John R. Bain, Jr., "Suboptimal Guidance for the Advanced Launch System", May 1990.

3. Clifford E. Hilfrich, "Neighboring Extremals for Systems with Piecewise Linear Control", August, 1991.

4. Michael J. Nowak, "Neighboring Extremal Guidance for Systems with Piecewise Linear Control Using Multiple Optimization", August, 1992.

5. William A. Libby, "Neighboring Suboptimal Control with Time as the Running Variable", May 1993.

AIAA GNC Conference
Portland, Oregon Aug. 1990

# Advanced Launch System Trajectory Optimization Using Suboptimal Control



Douglas A. Shaver* and David G. Hull†
Department of Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin
Austin, Texas 78712



## Abstract

The maximum-final-mass trajectory of a proposed configuration of the Advanced Launch System is presented. A model for the two-stage rocket is given; the optimal control problem is formulated as a parameter optimization problem; and the optimal trajectory is computed using a nonlinear programming code called VF02AD. Numerical results are presented for the controls (angle of attack and velocity roll angle) and the states. After the initial rotation, the angle of attack goes to a positive value to keep the trajectory as high as possible, returns to near zero to pass through the transonic regime and satisfy the dynamic pressure constraint, returns to a psotive value to keep the trajectory high and to take advantage of minimum drag at positive angle of attack due to aerodynamic shading of the booster, and then rolls off to negative values to satisfy the constraints. Because the engines cannot be throttled, the maximum dynamic pressure occurs at a single point; there is no maximum dynamic pressure subarc.

To test approximations for obtaining analytical solutions for guidance, two additional optimal trajectories are computed: one using untrimmed aerodynamics and one using no atmospheric effects except for the dynamic pressure constraint. It is concluded that untrimmed aerodynamics has a negligible effect on the optimal trajectory and that approximate optimal controls should be able to be obtained by treating atmospheric effects as perturbations.

## List of Symbols

$C_L$ lift coefficient
$C_D$ drag coefficient
$C_m$ pitching moment coefficient
$D$ aerodynamic drag (lb)
$g$ local gravitational acceleration (ft/sec$^2$)
$h$ altitude (ft)
$i$ orbit inclination (rad)
$I_{sp}$ vacuum specific impulse (sec)
$J$ performance index
$l$ aerodynamic reference length (ft)
$l_t$ distance from exit plane to vehicle cg (ft)
$L$ aerodynamic lift (lb)
$m$ vehicle mass (slugs)
$M_A$ aerodynamic pitching moment (ft lb)
$M$ Mach number
$P$ penalty function
$p$ atmospheric pressure (lb/ft$^2$)
$q$ dynamic pressure (lb/ft$^2$)
$S_b$ aerodynamic reference area (ft$^2$)
$T$ thrust (lb)
$T_{vac}$ vacuum thrust (lb)
$t_s$ staging time (sec)
$V$ velocity (ft/sec)

$\alpha$ angle of attack (rad)
$\gamma$ flight path angle (rad)
$\delta$ thrust gimbal angle (rad)
$\theta$ pitch angle (rad)
$\lambda$ longitude (rad)
$\mu$ velocity roll angle (rad)
$\rho$ atmospheric density (slug/ft$^3$)
$\tau$ latitude (rad)
$\tau$ normalized time
$\omega$ rotational velocity of earth (rad/sec)
$\psi$ heading angle (rad)

Subscripts

$b$ body axes
$cg$ center of gravity
$e$ exit
$f$ final
$i$ inertial
$o$ initial
$s$ sea-level
$w$ wind axes

## I. Introduction

A program is under way to develope an unmanned, all-weather, launch system for placing medium to large payloads (~ 120,000 lb) into low-earth orbit. A prospective design for this Advanced Launch System (ALS) is shown in Fig. 1 to be composed of a core vehicle and a booster. Both the booster and the core are ignited at launch, and staging occurs when all the booster pro-

*Graduate Research Assistant, Member AIAA.
†M.J. Thompson Regents Professor, Associate Fellow, AIAA.

pellant is consumed. Payload mass can be increased by adding another booster.

Part of the design process is to iterate the vehicle design and trajectory design until a reasonable combination is achieved. This paper is concerned solely with the optimal trajectory design of the proposed configuration. The objective is to find the trajectory which maximizes the final mass (since the engines burn throughout the trajectory, this is also a minimum final time problem). Any remaining propellant can be considered for conversion to payload or a decrease in launch weight. The physical model is that of flight over a rotating, spherical earth with an exponential atmosphere. Launched vertically from the surface of the earth, the payload is to be placed into perigee of an 80nm by 150nm transfer orbit. Because of structural considerations, there is a limit on the amount of dynamic pressure the vehicle can withstand.

This study has had several goals: (a) to determine the maximum-final-mass trajectory of the proposed ALS, (b) to generate initial Lagrange multipliers for a shooting code to investigate neighboring extremal guidance, and (c) to determine if atmospheric effects (pressure thrust and aerodynamics) can be considered as a perturbation to vacuum thrust and gravity for guidance law development. While only (a) and (c) are reported here, (b) requires the use of an exponential atmosphere. Hence, the dynamic pressure limit based on a standard atmosphere has been lowered to have the same effect in an exponential atmosphere.

In Section 2, a model is presented for the proposed ALS configuration. Then, the optimal control problem is formulated in Section 3 and converted into a parameter optimization problem in Section 4. This is done for relative ease in obtaining an optimal trajectory. Numerical results are presented in Section 5 in the form of optimal controls, states, and dynamic pressure. Also contained in Section 5 are two additional optimal trajectories based on untrimmed aerodynamics and neglected atmospheric effects. Finally, conclusions are presented in Section 6.

## II. Physical Model

In this section, a physical model for the Advanced Launch System (ALS) is defined. It includes the equations of motion for flight over a rotating, spherical earth with an exponential atmosphere and the mass, propulsion, and aerodynamic properties of the vehicle.

### Equations of Motion

Since sideslip causes drag, the vehicle is assumed to fly at zero sideslip angle, so that only the angle of attack gives the orientation of the vehicle relative to the free stream. The direction of the lift vector is then controlled through the bank angle or, more specifically, through the velocity roll angle.

A three-degree-of-freedom model for vehicle motion can be obtained from a six-degree-of-freedom model by one of two aerodynamic approximations: untrimmed aerodynamics or trimmed aerodynamics. For a rocket, untrimmed aerodynamics is equivalent to setting the thrust gimbal angle to zero and ignoring the aerodynamic pitching moment. On the other hand, with trimmed aerodynamics, it is assumed that the pitch rate is zero (pitching moment equals zero) so that the gimbal angle can be determined as a function of the angle of attack.

In view of the above comments, the three-degree-of-freedom equations of motion relative to the earth are given by (Ref. 3)

$$
\begin{aligned}
\dot{\lambda} &= \frac{V\cos\gamma\cos\psi}{r\cos\tau} \\
\dot{\tau} &= \frac{V\cos\gamma\sin\psi}{r} \\
\dot{h} &= V\sin\gamma \\
\dot{V} &= \frac{1}{m}(T\cos(\alpha+\delta) - D - mg\sin\gamma) \\
&\quad + r\omega^2\cos\tau(\cos\tau\sin\gamma - \sin\tau\cos\gamma\sin\psi) \\
\dot{\gamma} &= \frac{1}{mV}[(T\sin(\alpha+\delta) + L)\cos\mu - mg\cos\gamma] \\
&\quad + \frac{V\cos\gamma}{r} + 2\omega\cos\tau\cos\psi + \frac{r\omega^2}{V}\cos\tau(\cos\tau\cos\gamma \\
&\quad + \sin\tau\sin\gamma\sin\psi) \\
\dot{\psi} &= -\frac{1}{mV\cos\gamma}(T\sin(\alpha+\delta) + L)\sin\mu \\
&\quad - \frac{V}{r}\tan\tau\cos\gamma\cos\psi + 2\omega(\cos\tau\tan\gamma\sin\psi - \sin\tau) \\
&\quad - \frac{r\omega^2}{V\cos\gamma}\cos\tau\sin\tau\cos\psi \\
\dot{m} &= -\frac{1}{I_{sp}g_s}T_{vac}
\end{aligned}
\tag{1}
$$

In these equations, $\lambda$ is the longitude, $\tau$ is the latitude, $h$ is the altitude above mean sea level, $V$ is the velocity, $\gamma$ is the flight path angle, $\psi$ is the heading angle, $m$ is the mass, $r = r_s + h$ is the distance from the center of the earth to the vehicle center of gravity, $\omega$ is the angular velocity of the eath, $D$ is the drag, $L$ is the lift, $T$ is the thrust, $I_{sp}$ is the specific impulse, $\delta$ is the gimbal angle of the thrust vector, $\alpha$ is the angle of attack, and $\mu$ is the velocity roll angle. With regard to signs, a positive roll angle generates a negative heading toward the south.

For trimmed aerodynamics, the pitching moment, which is the sum of the aerodynamic pitching moment and the thrust pitching moment, is set equal to zero, and the resulting expression solved for the thrust gimbal angle. With reference to Fig. 1 and by assuming that $\delta$ is small, this process leads to

$$\delta = -\frac{M_A}{Tl_t} \tag{2}$$

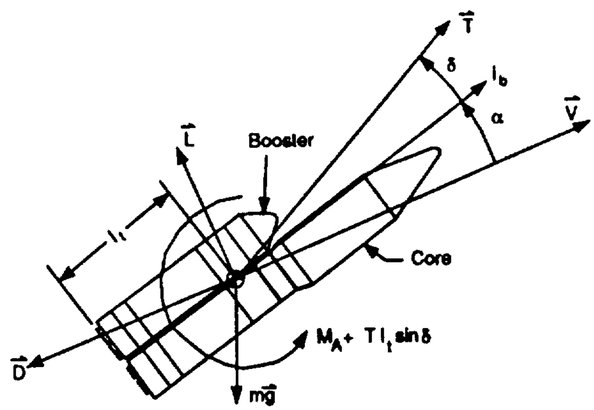


Figure 1: Force and Moment Nomenclature

where $M_A$ is the aerodynamic pitching moment ant $l_t$ is the distance from the center of gravity to the exit plane of the engines. Because $\delta$ is dependent on the aerodynamic pitching moment and the moment is dependent on the pitching moment coefficient, it results that $\delta$ is linear in $\alpha$ with the coefficients varying with time. Aerodynamics is discussed in further detail later in this section.

Eqs. (1) have two singularities: $V = 0$ in the $\dot{\gamma}$ and the $\dot{\psi}$ equations and $\gamma = \frac{\pi}{2}$ in the $\dot{\psi}$ equation. To remove the $V$ singularity and to clear the launch tower, the vehicle is flown vertically for 3 sec with the angle of attack and the bank angle being chosen so that $\dot{\gamma} = 0$ and $\dot{\psi} = 0$. To remove the $\gamma$ singularity, the vehicle is pitched over at constant heading ($\psi = 0$) for 1.0 sec at a constant negative pitch rate $\dot{\theta}$ whose optimal value is determined. Since $\theta = \gamma + \alpha$, the angle of attack during pitch-over is given by

$$\alpha = \frac{\pi}{2} - \gamma + \dot{\theta}(t-3) \ . \tag{3}$$

Finally, the bank angle is chosen to make $\dot{\psi} = 0$. With a flat earth model, $\mu = 0$.

### Earth

The earth is taken to be a rotating, spherical body whose surface is described by the mean sea-level radius $r_s$ and whose gravitational acceleration varies with altitude according to the inverse-square law

$$g = g_s\left(\frac{r_s}{r}\right)^2 \tag{4}$$

where $g_s r_s^2$ represents the earth's gravitational parameter. Sea-level gravitational acceleration $g_s$, $r_s$, and the rotational velocity of earth $\omega$ are known constants given as

$$r_s = 2.09256725E{+}7 \text{ ft} \ , \ g_s = 32.174 \ \frac{\text{ft}}{\text{sec}^2}$$

$$\omega = 7.2921158E{-}5 \ \frac{\text{rad}}{\text{sec}} \ . \tag{5}$$

### Atmosphere

The atmosphere is represented by the exponential functions

$$\frac{\rho}{\rho_s} = exp\left(\frac{-h}{\lambda_1}\right) \ , \ \frac{p}{p_s} = exp\left(\frac{-h}{\lambda_2}\right) \tag{6}$$

where the scale-height constants are given by

$$\lambda_1 = 23{,}800 \text{ ft} \ , \ \lambda_2 = 23{,}200 \text{ ft} \tag{7}$$

and the sea-level values of the density and pressure are

$$\rho_s = .002377 \ \frac{\text{slugs}}{\text{ft}^3} \ , \ p_s = 2{,}116.24 \ \frac{\text{lb}}{\text{ft}^2} \ . \tag{8}$$

Finally, the speed of sound is given by

$$a = \sqrt{\gamma \frac{p}{\rho}} \tag{9}$$

where $\gamma = 1.4$ is the ratio of specific heats of air.

### Mass Characteristics

The ALS configuration consists of a core vehicle as depicted in Fig. 1. The take-off mass of the ALS consists of the inert vehicle mass, the propellant mass, payload mass, payload margin mass, and the payload fairing mass (Table 1).

Table 1: Mass Characteristics

| Vehicle | Vehicle Component | Take-off Mass (slugs) |
|---|---|---|
| Core | Inert Mass | 5,474.29 |
| | Propellant | 45,974.38 |
| | Payload | 3,729.71 |
| | Payload Margin | 372.97 |
| | Payload Fairing | 1,215.89 |
| | Total Core | 56,767.26 |
| Booster | Inert Mass | 6,740.85 |
| | Propellant | 45,066.82 |
| | Total Booster | 51,807.67 |
| Core + Booster | Total Take-off Mass | 108,574.93 |

The center of gravity is located relative to a coordinate system whose origin is at the tip of the core vehicle, whose $x$ axis is down the longitudinal axis, and whose $y$ axis is toward the booster. For the first stage, the vehicle center of gravity is assumed to have coordinates

$$x_{cg} = 165.45 \text{ ft} \ , \ y_{cg} = 10.36 - .0388t \text{ ft} \tag{10}$$

so that $l_t$ is constant and has the value

$$l_t = l - x_{cg} = 110.81 \text{ ft} \tag{11}$$

where $l$ = 276.26 ft is the length of the core vehicle. Actually, $x_{cg}$ varies slightly but this variation has been neglected. For the second stage, untrimmed aerodynamics is used so the cg position is not needed.

### Propulsion

The ALS is powered by ten liquid hydrogen/liquid oxygen low cost rocket engines (LCE): seven power the booster and three power the core. All engines are ignited at launch; staging occurs when the booster fuel is depleted; and the core engines burn until insertion.

Propulsion characteristics of interest are the thrust $T$, vacuum thrust $T_{vac}$, and the specific impulse $I_{sp}$(see Eqs. 1). If the exit pressure is conservatively approximated as $p_e = 0$, the thrust of a single engine is modeled as

$$T' = T_{vac}' - pA_e' \tag{12}$$

where the prime denotes one engine, $p$ is the atmospheric pressure at the altitude of the rocket, and $A_e'$ is the exit area. Date relevant to one LCE are as follows:

$$\begin{aligned} T_{vac}' &= 580,110.0 \text{ lb} \\ A_e' &= 40.381 \text{ ft}^2 \\ I_{sp}' &= 430.0 \text{ sec} \ . \end{aligned} \tag{13}$$

For the complete vehicle,

$$T = kT' \ , \ I_{sp} = I_{sp}' \ , \ T_{vac} = kT_{vac}' \tag{14}$$

where $k$ = 10 before staging and $k$ = 3 after staging. Specific impulse is like specific propellant consumption (weight flow rate of propellant per pound of thrust); hence, it has the same value regardless of the number of engines operating.

### Aerodynamics

The drag, lift, and pitching moment are related to their respective coefficients by the standard equations

$$D = qS_bC_D \ , \ L = qS_bC_L \ , M_A = qS_blC_m \tag{15}$$

where $q = \frac{1}{2}\rho V^2$ is the dynamic pressure, $S_b$ = 1413.71 ft$^2$ is the cross-sectional area of the combined vehicle (booster + core), and $l$ is the length of the core. While the aerodynamic coefficients are needed at and about the center of gravity (cg), the aerodynamic data has been provided at and about the launch cg. Although the drag and lift transfer directly, the moment changes with cg position. Therefore, the aerodynamic data at the cg must be related to the launch cg.

The aerodynamic data are preliminary estimates associated with the development of the six-degreee-of-freedom simulation presented in Ref. 4. These data are provided in tabular form (Tables 2 through 6) consistent with the functional relations

$$C_D = C_D(M,\alpha) \ , C_L = C_L(M,\alpha) \ ,$$

$$\overline{C}_m = \overline{C}_m(M,\alpha) \tag{16}$$

where $M$ denotes the Mach number and the bar indicates that the moment is about a fixed point (launch cg). About the actual center of gravity, the moment is given by

$$C_m = \overline{C}_m - C_D\frac{y_{cg} - 10.36}{l} \tag{17}$$

since $x_{cg}$ is assumed not to change.

While the aerodynamic data could have been used in tabular form with linear interpolation to read the tables, the approach taken is to assume polynomials in $\alpha$ with Mach-number-dependent coefficients. For the first stage, the coefficients are written as

$$\begin{aligned} C_D &= C_{D_0}(M) + C_{D_{\alpha^2}}(M)\alpha^2 + C_{D_{\alpha^3}}(M)\alpha^3 \\ C_L &= C_{L_\alpha}(M)\alpha \\ \overline{C}_m &= \overline{C}_{m_0}(M) + \overline{C}_{m_\alpha}(M)\alpha \end{aligned} \tag{18}$$

where the Mach-number-dependent terms have been obtained from cubic-spline curve fits of the tabular data. After staging, the flow regime is hypersonic and the aerodynamic force coefficients are modeled as

$$\begin{aligned} C_D &= C_{D_0} + C_{D_\alpha}\alpha + C_{D_{\alpha^2}}\alpha^2 \\ C_L &= C_{L_\alpha}\alpha + C_{L_{\alpha^2}}\alpha^2 \end{aligned} \tag{19}$$

where the coefficients of $\alpha$ are constants. Also, pitching moments are assumed to be negligible after staging, that is, untrimmed aerodynamics are used ($\delta = 0$).

A peculiarity of the aerodynamics of the combined vehicle at supersonic and hypersonic speeds is that the drag coefficient has a minimum at a positive angle of attack. This is caused by the aerodynamic shading of the booster by the flow field of the core.

## III. The Optimal Control Problem

Formally the optimal control problem considered here is to find the control history $u(t)$ which minimizes a performance index of the form

$$J = \Phi(x_f) \tag{20}$$

subject to the differential constraints

$$\dot{x} = f(x,u) \ , \tag{21}$$

the prescribed initial conditions

$$t_o = t_{o_s} \ , \ x_o = x_{o_s} \ , \tag{22}$$

the prescribed final conditions

$$\Psi(x_f) = 0 \ , \tag{23}$$

and a state-variable inequality constraint

$$S(x) \leq 0 \ . \tag{24}$$

Each of these quantities is discussed below.

### State Variables and Control Varables

The state variables are $x^T = [\lambda\ \tau\ h\ V\ \gamma\ \psi\ m]$ while the control variables are $u^T = [\alpha\ \ \mu]$.

### Performance Index

It is desired to maximize the final mass. Hence, the performance index is taken to be

$$\Phi = -\frac{m_f}{m_{ref}} \tag{25}$$

where the minus sign is included because the performance index is actually minimized and where $m_{ref}$ is the sum of the payload mass, the payload margin mass, and the payload fairing mass. A performance index of $\Phi = -1.0$ means that the reference mass is inserted into orbit with no extra fuel.

### Differential Constraints

The differential constraints are the equations of motion (Eqs. 1) completely expressed in terms of the state variables and the control variables.

### Prescribed Initial Conditions

For the trajectory design problem, the initial conditions are taken to be

$$\begin{gathered} t_o = 0\,\text{sec}\ \ ,\lambda_o = -80.54\,\text{deg}\ \ ,\tau_o = 28.5\,\text{deg} \\ h_o = 0\,\text{ft}\ \ ,V_o = 0\,\frac{\text{ft}}{\text{sec}}\ \ ,\gamma_o = 90.0\,\text{deg} \\ \psi_o = 0.0\,\text{deg}\ \ ,m_o = 108,574.93\,\text{slugs} \end{gathered} \tag{26}$$

During the vertical rise segment, the heading angle is undefined, so the initial condition on $\psi$ is actually the heading angle during the pitch-over segment.

### Prescribed Final Conditions

The Advanced Launch System is being designed to place a nominal payload at perigee of an 80nm by 150nm transfer orbit of 28.5 deg inclination. As a consequence, the equality constraint residuals are

$$\begin{gathered} \Psi_1 = h_f - 486,080\,\text{ft}\ ,\ \Psi_2 = V_{i_f} - 25,776.9\,\frac{\text{ft}}{\text{sec}} \\ \Psi_3 = \gamma_f\ ,\ \Psi_4 = \cos i_f - \cos(28.5\,\text{deg}) \end{gathered} \tag{27}$$

where the inertial velocity and the inclination are related to the relative states as follows (Ref. 5 and 6)

$$V_i = \left[V^2 + 2Vr\omega\cos\gamma\cos\psi\cos\tau + (r\omega\cos\tau)^2\right]^{\frac{1}{2}} \tag{28}$$

$$\cos i = \frac{\cos\tau(V\cos\gamma\cos\psi + r\omega\cos\tau)}{\left[V^2\cos^2\gamma + 2Vr\omega\cos\gamma\cos\psi\cos\tau + (r\omega\cos\tau)^2\right]^{\frac{1}{2}}}\ .$$

### State-Variable Inequality Constraint

Based on structural considerations, the ALS must not exceed a maximum dynamic pressure of $q$ = 650 lb/ft$^2$. Therefore, the state-variable inequality constraint residue $S$ is

$$S = \frac{1}{2}\rho V^2 - 650\ \text{lb/ft}^2\ \ . \tag{29}$$

Actually, in a standard atmosphere, the limit is $q_{max}$ = 850 lb/ft$^2$. The value of 650 lb/ft$^2$ is chosen because the value of $\rho$ is approximately 20% smaller in the exponential atmosphere than the standard atmosphere around the maximum dynamic pressure portion of the trajectory.

## IV. The Suboptimal Control Problem

The optimal control problem is converted to a parameter optimization problem (suboptimal control problem) as follow: (a) the time is normalized by introducing the transformation $\tau = \frac{t}{t_f}$; (b) the control $u(t)$ is replaced by a set of nodal points which is linearly interpolated, and (c) the state-variable inequality constraint is converted to a point constraint by using a penalty function.

Because of the time transformation, the boundary values of $\tau$ are given by

$$\begin{gathered} \tau_o = 0\ ,\ \tau_p = \frac{3}{t_f}\ ,\ \tau_1 = \frac{4}{t_f}\ , \\ \tau_s = \frac{153.54}{t_f}\ ,\ \tau_f = 1 \end{gathered} \tag{30}$$

where $t_p$ = 3 sec is the time at the beginning of pitch-over and $t_1$ = 4 sec is the time when three-dimensional flight begins. Staging occurs when all of the booster propellant is consumed; hence, $t_s$ = 153.54 sec.

Figure 2 shows the arrangement of nodal points in each stage. Nine nodes are used for the control during the first stage, and five for the control during the second stage. Even though the duration of the first stage is shorter than that of the second, there is more activity in $\alpha$ during the first stage, making more nodes desireable. The nodes are equally spaced in each stage so that the node times are

$$\begin{gathered} \tau_i = \tau_1 + \frac{\tau_s - \tau_1}{8}(i-1)\ ,\ i = 1 \rightarrow 9 \\ \tau_i = \tau_s + \frac{1-\tau_s}{4}(i-10)\ ,\ i = 10 \rightarrow 14\ \ . \end{gathered} \tag{31}$$

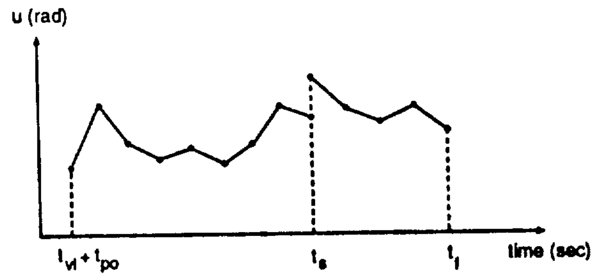


Figure 2: Example Control History

Note that there are two control nodes at the stage time. This has been done in order to find the true suboptimal control.

The dynamic pressure constraint is converted to a parameter inequality constraint by introducing the penalty function

$$P = -\int_{t_o}^{t} min^2[(1 - \frac{q}{q_{max}}), 0]dt \geq 0 \qquad (32)$$

which accumulates value when $q > q_{max}$. The constraint becomes

$$P_f \geq 0 \ . \qquad (33)$$

To compute $P_f$, the penalty function is differentiated to form

$$\dot{P} = -min^2[(1 - \frac{q}{q_{max}}), 0] \qquad (34)$$

where

$$P_o = 0 \ . \qquad (35)$$

In all, the nonlinear programming problem involves 30 parameters, that is, the parameter vector is given by

$$X = \left[\dot{\theta} \ , \ \alpha_1 \ , \ldots \ , \ \alpha_{14} \ , \ \mu_1 \ , \ldots \ , \ \mu_{14} \ , \ t_f\right] \qquad (36)$$

where $\dot{\theta}$ is the pitch rate during the pitch-over, $\alpha_k$ , $\mu_k$ are the angle of attack and the bank angle nodes, and $t_f$ is the final time.

If values of the parameters (36) are known, the differential equations (1) and (34) can be integrated through the mission to determine the states and $P$ at the final time. Then, the performance index (25), the orbital insertion equality constraint residuals (27), and the dynamic pressure inequality constraint (29) can be computed. It follows that the performance index and the constraints are functions of the parameters (36) such that the nonlinear programming problem can be expressed as follows:
Find the set of parameters $X$ which minimizes the performance index

$$J = -\frac{m_f(X)}{m_{ref}} \qquad (37)$$

subject to the equality constraints

$$C_1 = \frac{h_f(X)}{h_{f_s}} - 1 = 0$$

$$C_2 = \frac{V_{i_f}(X)}{V_{i_{f_s}}} - 1 = 0$$
$$C_3 = \gamma_f(X) = 0 \qquad (38)$$
$$C_4 = \frac{cos i_f(X)}{cos i_{f_s}} - 1 = 0$$

and the inequality constraint

$$C_5 = P_f(X) \geq 0 \ . \qquad (39)$$

Derivatives required by the nonlinear programming algorithm are computed by central differences.

## V. Numerical Results

The optimal trajectory has been computed using a nonlinear programming code known as VF02AD which is based on quadratic programming. Optimal control histories are presented in Fig. 3, while the resulting states are shown in Figs. 4 through 7. The magnitude of the performance index is 103.94% where 100% = 171,120 lb. This means that an additional 6,742 lb of payload can be placed in orbit with this vehicle by using the optimal trajectory. The vehicle is inserted into orbit at $t_f$ = 363.8 sec and the optimal value of the pitch rate during the 1.0 sec pitch-over is -.02005 rad/sec.

Shown in Fig. 8 is the dynamic pressure. It is seen that the maximum dynamic pressure occurs at a single point and not along a $q_{max}$ subarc. This is due to the no-throttling design of the vehicle and the fact that the aerodynamic forces needed to fly along $q = q_{max}$ cannot be achieved. Optimal trajectories with lower values of $q_{max}$ have been calculated, and the results are the same.

It is difficult to completely determine the meanings of the optimal control histories because performance-index minimization and constraint satisfaction are going on all through the trajectory. For angle of attack, it is seen from Fig. 3 that the vehicle initially goes to positive $\alpha$ to achieve altitude and decrease $q$. Then, the dip in $\alpha$ from $t$ = 40 to 60 sec allows the vehicle to pass through the transonic regime efficiently (Mach 1 occurs at $t \approx 50$ sec) and to satisfy the dynamic pressure inequality constraint ($q_{max}$ occurs at $t \approx 70$ sec). Next, the vehicle returns to positive $\alpha$ to get low drag and to decrease the magnitude of $\dot{\gamma}$. Staging occurs around Mach 8 and the roll off in $\alpha$ from positive to negative values during the second stage helps pull the trajectory down to meet the final conditions. For the velocity roll angle, the nonzero values at the beginning of the trajectory seem to be caused by the rotational effects of earth where the vehicle wants to fly at constant latitude throughout most of the first stage. Changes in $\mu$ near the end of the trajectory help cause constraint satisfaction, particularly in the orbit inclination.

Additional optimal trajectories have been computed with the intent of determining what kinds of approximations can be made in order to obtain approximate analytical solutions for guidance purposes. First, the effect of using untrimmed aerodynamics ($\delta = 0$) rather than trimmed aerodynamics is shown in Fig. 9 and 10 to change only slightly the optimal controls and to cause a relative change in the performance index of 0.2% (376.5 lb). Hence, untrimmed aerodynamics is a reasonable approximation. Second, the question of whether or not atmospheric effects can be considered a perturbation is considered. This means that the pressure term in the thrust and the aerodynamics are neglected; however, the dynamic pressure constraint is maintained because it is a structural constraint. The optimal controls for this case are shown in Fig. 11 and 12 and lead to a relative increase in the performance index of 16% (27,379 lb). Trajectory profiles for the atmosphere and no-atmosphere cases are shown in Fig. 13. The optimal control which results from the no-atmosphere case is reasonably close to that of the atmosphere case and has the same general trend. This seems to indicate that atmospheric effects can be treated as a perturbation.

## VI. Discussion and Conclusions

The maximum-final-mass trajectory has been computed for a two-stage rocket representing the Advanced Launch System and operating over a rotating, spherical earth with an exponential atmosphere. The problem is converted into a parameter optimization problem by replacing the control histories by node points and using straight-line interpolation to form functions. Then, a nonlinear programming code known as VF02AD is used to perform the optimization. Optimal trajectories have been calculated for three cases: (a) trimmed aerodynamics, (b) untrimmed aerodynamics, and (c) no atmosphere. With the assumption of trimmed aerodynamics, the aerodynamic model is as accurate as possible for a three-degree-of-freedom analysis. The optimal trajectory is characterized by positive angles of attack over most of the path with a prominant decrease during passage through maximum dynamic pressure. The maximum dynamic pressure occurs at a single point rather than over a subarc because the engines cannot be throttled.

To obtain analytical solutions for guidance purposes, approximations must be introduced. The effect of replacing trimmed aerodynamics by untrimmed aerodynamics has been examined, and it is concluded that untrimmed aerodynamics gives good results.

Next, the effect of neglecting atmospheric effects (pressure thrust and aerodynamics) has been investigated. With the exception of the transonic and maximum dynamic pressure portion of the trajectory, it is clear that atmospheric effects can be considered as perturbations to the trajectory generated by vacuum thrust and gravity. During the passage through the transonic and maximum dynamic pressure part of the trajectory, there is a difference of 14 deg between the atmosphere and no-atmosphere solutions. Since this region constitutes less than fifteen percent of the whole trajectory, treating atmospheric effects as perturbations could yield satisfactory results.


## Acknowledgement

This research was sponsered in part by NASA Langley Research Center Grant #NAG-1945 in association with Dr. Daniel D. Moerder.


## References


1. Chenard, S., "Advanced Launch System Progress," *Interavia Space Markets*, March-April 1989, pp. 6-10.
2. DeMeis, R., "Sweetening the Orbital Bottom Line," *Aerospace America*, August 1988, pp. 26-30.
3. Miele, A., *Theory of Flight Paths*, Addison-Wesley, Reading, Mass., 1962.
4. Pamadi, B.N., and Dutton, K., "An Aerodynamic Model for the Advanced Launch System ALS-L Vehicle," to appear as a NASA TN and to be presented at the AIAA Applied Aerodynamics Conference, Portland, Oregon, August 1990.
5. Bate, R.R., Mueller, D.D., and White, J.E., *Fundamentals of Astrodynamics*, Dover Publications, New York, 1971.
6. Hull, D.G., and Speyer, J.L., "Optimal Reentry and Plane-Change Trajectories," *The Journal of the Astronautical Sciences*, Vol. 30, No. 2, April-June 1962, pp. 117-130.


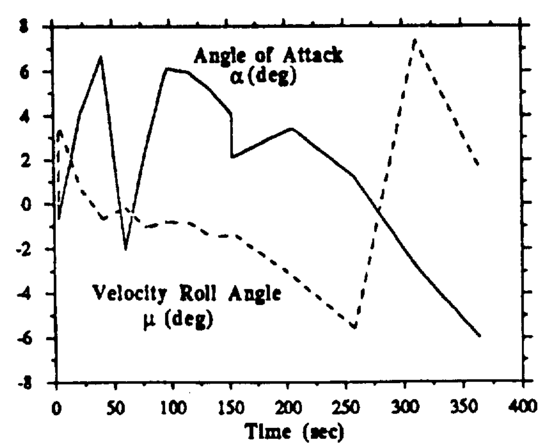


Figure 3: Trimmed Aerodynamic Control Histories

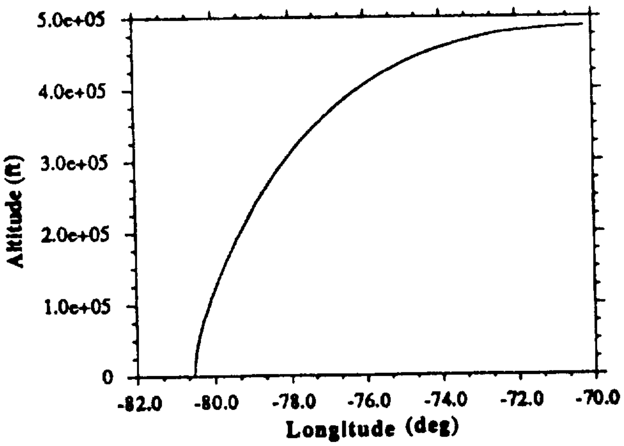


Figure 4: Trajectory Profile; Trimmed Flight

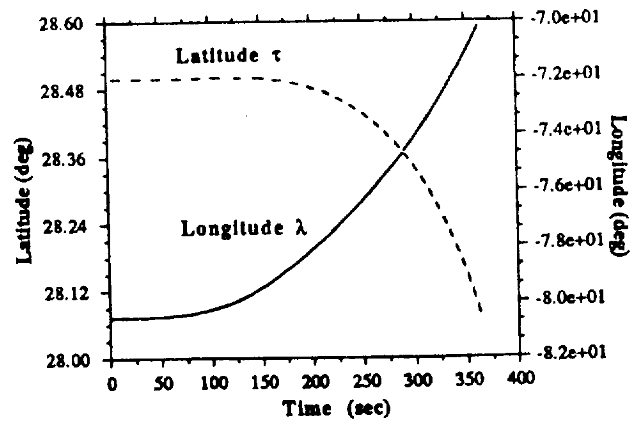


Figure 7: Latitude and Longitude vs. Time; Trimmed Flight

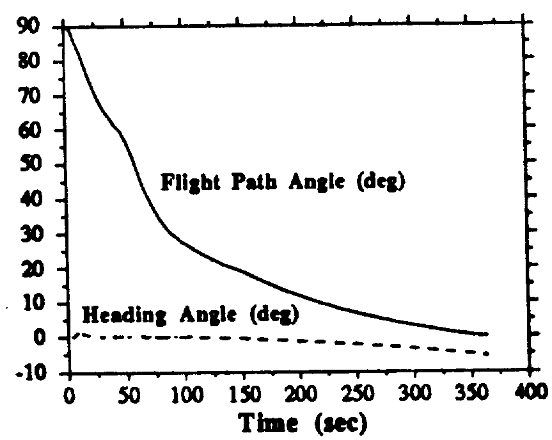


Figure 5: Flight Path and Heading Angle vs. Time; Trimmed Flight

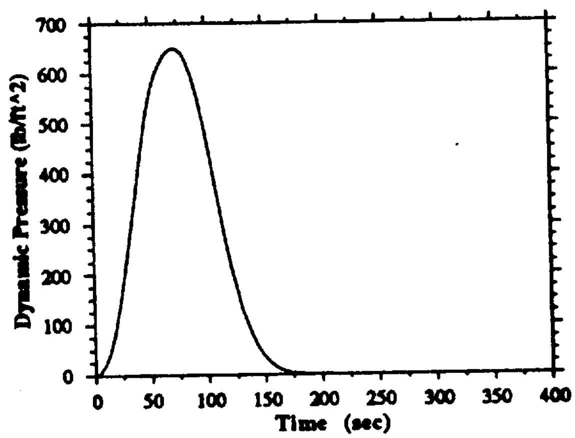


Figure 8: Dynamic Pressure vs. Time; Trimmed Flight

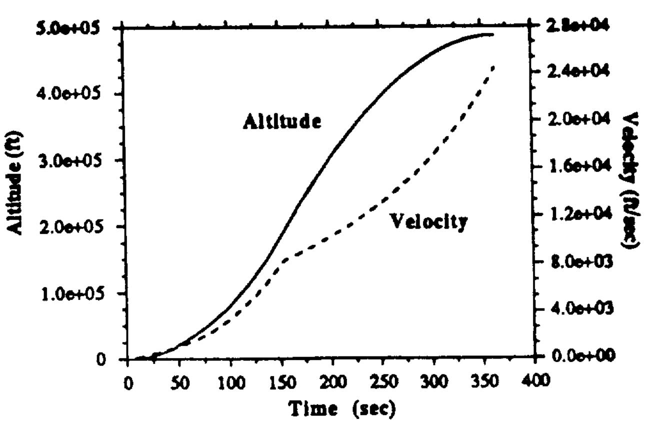


Figure 6: Altitude and Velocity vs. Time; Trimmed Flight

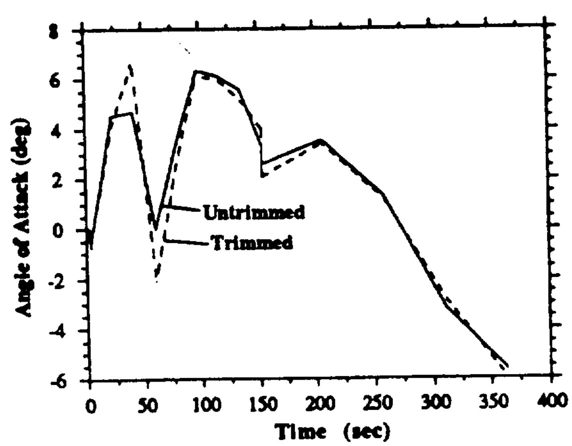


Figure 9: Angle of Attack vs. Time

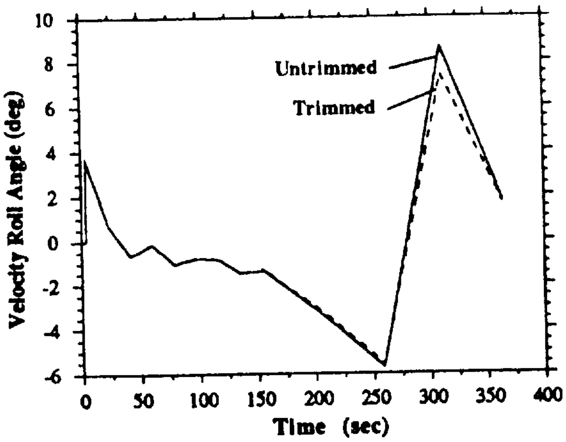


Figure 10: Velocity Roll Angle vs. Time

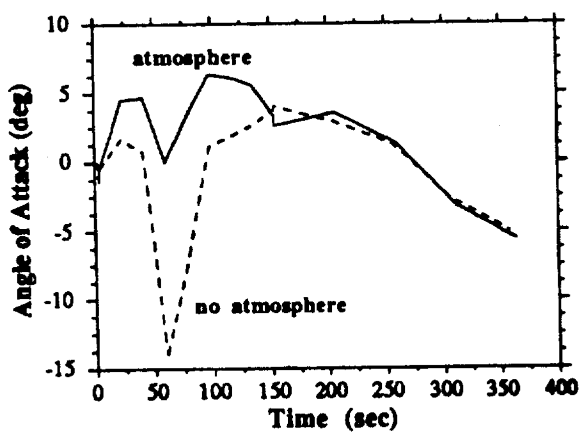


Figure 11: Angle of Attack vs. Time

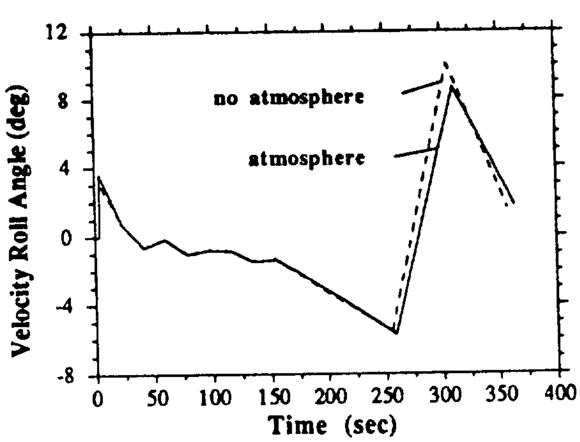


Figure 12: Velocity Roll Angle vs. Time

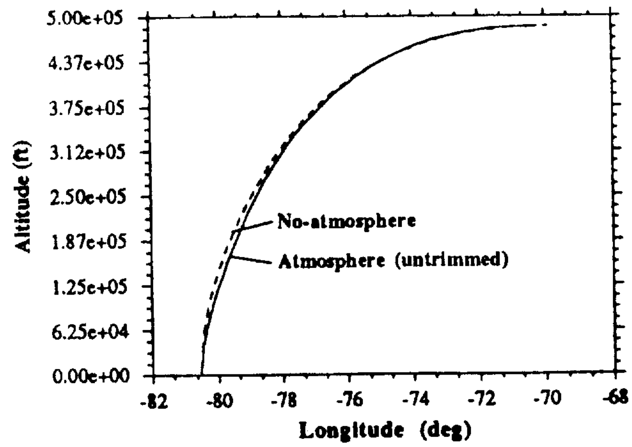


Figure 13: Trajectory Profiles; Atmosphere and No-atmosphere

**Table 2. Lift Coefficient (core + booster)**

Subsonic Data

| M | Angle of Attack (deg) ±0.0 | ±2.0 | ±4.0 | ±6.0 | ±8.0 | ±10.0 |
|---|---|---|---|---|---|---|
| 0.0 | 0.0 | 0.08876 | 0.1775 | 0.2663 | 0.355 | 0.4438 |
| 0.2 | 0.0 | 0.08876 | 0.1775 | 0.2663 | 0.355 | 0.4438 |
| 0.4 | 0.0 | 0.08876 | 0.1775 | 0.2663 | 0.355 | 0.4438 |
| 0.6 | 0.0 | 0.08876 | 0.1775 | 0.2663 | 0.355 | 0.4438 |
| 0.8 | 0.0 | 0.08876 | 0.1775 | 0.2663 | 0.355 | 0.4438 |
| 1.0 | 0.0 | 0.08720 | 0.1744 | 0.2616 | 0.3488 | 0.4360 |

Supersonic/Hypersonic Data

| M | Angle of Attack (deg) 0.0 | 2.0 | 4.0 | 6.0 | 8.0 | 10.0 |
|---|---|---|---|---|---|---|
| 1.2 | 0.0 | 0.0862 | 0.1724 | 0.2586 | 0.3448 | 0.431 |
| 1.5 | 0.0 | 0.086 | 0.171 | 0.260 | 0.351 | 0.431 |
| 2.0 | 0.0 | 0.090 | 0.175 | 0.262 | 0.354 | 0.435 |
| 2.5 | 0.0 | 0.098 | 0.181 | 0.268 | 0.370 | 0.460 |
| 3.0 | 0.0 | 0.100 | 0.192 | 0.278 | 0.385 | 0.490 |
| 3.5 | 0.0 | 0.102 | 0.200 | 0.290 | 0.401 | 0.510 |
| 4.0 | 0.0 | 0.104 | 0.202 | 0.291 | 0.405 | 0.510 |
| 5.0 | 0.0 | 0.104 | 0.206 | 0.298 | 0.410 | 0.509 |
| 6.0 | 0.0 | 0.103 | 0.203 | 0.300 | 0.408 | 0.508 |
| 7.0 | 0.0 | 0.100 | 0.195 | 0.298 | 0.400 | 0.502 |
| 8.0 | 0.0 | 0.095 | 0.185 | 0.290 | 0.395 | 0.500 |

| M | Angle of Attack (deg) -2.0 | -4.0 | -6.0 | -8.0 | -10.0 |
|---|---|---|---|---|---|
| 1.2 | -0.084 | -0.170 | -0.260 | -0.350 | -0.431 |
| 1.5 | -0.086 | -0.171 | -0.260 | -0.351 | -0.431 |
| 2.0 | -0.090 | -0.175 | -0.262 | -0.354 | -0.435 |
| 2.5 | -0.098 | -0.181 | -0.268 | -0.370 | -0.460 |
| 3.0 | -0.100 | -0.192 | -0.278 | -0.385 | -0.490 |
| 3.5 | -0.120 | -0.200 | -0.290 | -0.401 | -0.510 |
| 4.0 | -0.120 | -0.215 | -0.310 | -0.420 | -0.520 |
| 5.0 | -0.120 | -0.225 | -0.327 | -0.442 | -0.542 |
| 6.0 | -0.125 | -0.225 | -0.334 | -0.451 | -0.567 |
| 7.0 | -0.115 | -0.222 | -0.332 | -0.452 | -0.580 |
| 8.0 | -0.110 | -0.218 | -0.325 | -0.450 | -0.565 |

## Table 3. Drag Coefficient (core + booster)

Subsonic Data

| M | Angle of Attack (deg) ±0.0 | ±2.0 | ±4.0 | ±6.0 | ±8.0 | ±10.0 |
|---|---|---|---|---|---|---|
| 0.0 | 0.1870 | 0.1904 | 0.2024 | 0.2254 | 0.262 | 0.314 |
| 0.2 | 0.1872 | 0.1906 | 0.2026 | 0.2256 | 0.2622 | 0.3142 |
| 0.4 | 0.2062 | 0.2096 | 0.2216 | 0.2446 | 0.2812 | 0.3340 |
| 0.6 | 0.2599 | 0.2633 | 0.2753 | 0.2983 | 0.3349 | 0.3877 |
| 0.8 | 0.3480 | 0.3514 | 0.3634 | 0.3864 | 0.4230 | 0.4758 |
| 1.0 | 0.7800 | 0.7834 | 0.7954 | 0.8184 | 0.8550 | 0.9078 |

Supersonic/Hypersonic Data

| M | Angle of Attack (deg) 0.0 | 2.0 | 4.0 | 6.0 | 8.0 | 10.0 |
|---|---|---|---|---|---|---|
| 1.2 | 0.800 | 0.805 | 0.815 | 0.838 | 0.875 | 0.928 |
| 1.5 | 0.740 | 0.703 | 0.645 | 0.640 | 0.635 | 0.635 |
| 2.0 | 0.672 | 0.656 | 0.555 | 0.525 | 0.525 | 0.525 |
| 2.5 | 0.648 | 0.628 | 0.512 | 0.468 | 0.465 | 0.455 |
| 3.0 | 0.637 | 0.608 | 0.486 | 0.448 | 0.431 | 0.418 |
| 3.5 | 0.630 | 0.596 | 0.470 | 0.425 | 0.406 | 0.392 |
| 4.0 | 0.628 | 0.587 | 0.460 | 0.410 | 0.385 | 0.368 |
| 5.0 | 0.620 | 0.572 | 0.448 | 0.392 | 0.355 | 0.352 |
| 6.0 | 0.617 | 0.570 | 0.446 | 0.382 | 0.348 | 0.348 |
| 7.0 | 0.615 | 0.567 | 0.445 | 0.378 | 0.340 | 0.340 |
| 8.0 | 0.615 | 0.565 | 0.445 | 0.372 | 0.340 | 0.338 |

| M | Angle of Attack (deg) -2.0 | -4.0 | -6.0 | -8.0 | -10.0 |
|---|---|---|---|---|---|
| 1.2 | 0.803 | 0.815 | 0.838 | 0.875 | 0.928 |
| 1.5 | 0.745 | 0.750 | 0.773 | 0.800 | 0.871 |
| 2.0 | 0.690 | 0.708 | 0.731 | 0.768 | 0.822 |
| 2.5 | 0.665 | 0.680 | 0.706 | 0.745 | 0.790 |
| 3.0 | 0.648 | 0.651 | 0.688 | 0.730 | 0.771 |
| 3.5 | 0.640 | 0.650 | 0.675 | 0.716 | 0.757 |
| 4.0 | 0.631 | 0.641 | 0.665 | 0.706 | 0.745 |
| 5.0 | 0.625 | 0.635 | 0.651 | 0.692 | 0.731 |
| 6.0 | 0.610 | 0.625 | 0.646 | 0.686 | 0.727 |
| 7.0 | 0.610 | 0.620 | 0.640 | 0.685 | 0.730 |
| 8.0 | 0.610 | 0.620 | 0.640 | 0.684 | 0.725 |

## Table 4. Pitching Moment Coefficient (core + booster)

| M | Angle of Attack (deg) 0.0 | 2.0 | 4.0 | 6.0 | 8.0 | 10.0 |
|---|---|---|---|---|---|---|
| 0.0 | 0.0035 | 0.0271 | 0.0508 | 0.0744 | 0.0981 | 0.1217 |
| 0.2 | 0.0035 | 0.0271 | 0.0508 | 0.0744 | 0.0981 | 0.1217 |
| 0.4 | 0.0040 | 0.0276 | 0.0513 | 0.0745 | 0.0986 | 0.1222 |
| 0.6 | 0.0052 | 0.0288 | 0.0538 | 0.0757 | 0.0998 | 0.1234 |
| 0.8 | 0.0072 | 0.0308 | 0.0558 | 0.0777 | 0.1018 | 0.1254 |
| 1.0 | 0.020 | 0.046 | 0.072 | 0.098 | 0.124 | 0.150 |
| 1.2 | 0.033 | 0.062 | 0.093 | 0.123 | 0.153 | 0.183 |
| 1.5 | 0.038 | 0.066 | 0.095 | 0.124 | 0.153 | 0.182 |
| 2.0 | 0.033 | 0.059 | 0.085 | 0.111 | 0.134 | 0.162 |
| 2.5 | 0.030 | 0.056 | 0.077 | 0.097 | 0.120 | 0.135 |
| 3.0 | 0.029 | 0.052 | 0.071 | 0.087 | 0.103 | 0.116 |
| 3.5 | 0.028 | 0.049 | 0.066 | 0.080 | 0.094 | 0.099 |
| 4.0 | 0.027 | 0.047 | 0.061 | 0.076 | 0.0895 | 0.099 |
| 5.0 | 0.026 | 0.045 | 0.057 | 0.068 | 0.085 | 0.098 |
| 6.0 | 0.026 | 0.042 | 0.054 | 0.068 | 0.082 | 0.096 |
| 7.0 | 0.0255 | 0.042 | 0.053 | 0.068 | 0.082 | 0.096 |
| 8.0 | 0.0255 | 0.042 | 0.052 | 0.068 | 0.082 | 0.097 |

| M | Angle of Attack (deg) -2.0 | -4.0 | -6.0 | -8.0 | -10.0 |
|---|---|---|---|---|---|
| 0.0 | -0.0201 | -0.044 | -0.067 | -0.091 | -0.115 |
| 0.2 | -0.020 | -0.044 | -0.067 | -0.091 | -0.115 |
| 0.4 | -0.019 | -0.043 | -0.067 | -0.091 | -0.115 |
| 0.6 | -0.018 | -0.042 | -0.066 | -0.089 | -0.113 |
| 0.8 | -0.016 | -0.040 | -0.064 | -0.087 | -0.111 |
| 1.0 | -0.004 | -0.027 | -0.051 | -0.075 | -0.098 |
| 1.2 | 0.003 | -0.029 | -0.058 | -0.089 | -0.119 |
| 1.5 | 0.009 | -0.019 | -0.048 | -0.077 | -0.106 |
| 2.0 | 0.009 | -0.0155 | -0.045 | -0.071 | -0.097 |
| 2.5 | 0.007 | -0.016 | -0.043 | -0.067 | -0.092 |
| 3.0 | 0.005 | -0.018 | -0.041 | -0.063 | -0.089 |
| 3.5 | 0.004 | -0.018 | -0.040 | -0.062 | -0.086 |
| 4.0 | 0.004 | -0.019 | -0.040 | -0.062 | -0.085 |
| 5.0 | 0.005 | -0.018 | -0.038 | -0.058 | -0.082 |
| 6.0 | 0.008 | -0.017 | -0.028 | -0.058 | -0.078 |
| 7.0 | 0.008 | -0.017 | -0.028 | -0.058 | -0.076 |
| 8.0 | 0.008 | -0.017 | -0.028 | -0.058 | -0.075 |

## Table 5. Lift Coefficient (core vehicle)

| M | Angle of Attack (deg) ±0.0 | ±2.0 | ±4.0 | ±6.0 | ±8.0 | ±10.0 |
|---|---|---|---|---|---|---|
| 8.0 | 0.2062 | 0.2089 | 0.2206 | 0.2417 | 0.2733 | 0.3160 |
| 10.0 | 0.2180 | 0.2201 | 0.2313 | 0.2523 | 0.2835 | 0.3262 |
| 12.0 | 0.2353 | 0.2374 | 0.2486 | 0.2703 | 0.3024 | 0.3459 |

## Table 6. Drag Coefficient (core vehicle)

| M | Angle of Attack (deg) ±0.0 | ±2.0 | ±4.0 | ±6.0 | ±8.0 | ±10.0 |
|---|---|---|---|---|---|---|
| 8.0 | 0.2062 | 0.2089 | 0.2206 | 0.2417 | 0.2733 | 0.3160 |
| 10.0 | 0.2180 | 0.2201 | 0.2313 | 0.2523 | 0.2835 | 0.3262 |
| 12.0 | 0.2353 | 0.2374 | 0.2486 | 0.2703 | 0.3024 | 0.3459 |

PRE 1. ANN
92A29302

# A SHOOTING APPROACH TO SUBOPTIMAL CONTROL

David G. Hull[1] and Jyh-Jong Sheen[2]

Department of Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin

## Abstract

The shooting method is used to solve the suboptimal control problem where the control history is assumed to be piecewise linear. Suboptimal solutions can be obtained without difficulty and can by increasing the number of nodes lead to accurate approximate controls and good starting multipliers for the regular shooting method. Optimal planar launch trajectories are presented for the Advanced Launch System.

## 1. Introduction

The original motivation for using the shooting method to solve the suboptimal control problem (piecewise linear control) has been to calculate an accurate suboptimal control and ultimately to find the corresponding neighboring extremal feedback control rule. Since aerospace minima are usually quite flat, an approximate optimal control can deliver most of the optimal performance. Then, the ability to compute the suboptimal control and the neighboring extremal without difficulty would be useful.

In this paper, the shooting method is developed for the suboptimal control problem and used to optimize the Advanced Launch System trajectory. The usual sensitivity of the solution process to the initial guesses disappears completely, and solutions are obtained without difficulty. Of course, only an approximate optimal control is achieved, but if it is not good enought, its accuracy can be improved by increasing the number of control nodes.

## 2. Suboptimal Control Problem

The standard optimal control problem is to find the control $u(t)$ which minimizes the scalar performance index

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

subject to the system dynamics

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

and the prescribed boundary conditions

$$t_0 = 0, \quad x_0 = x_{0_s}, \quad \psi(x_f, t_f) = 0. \tag{3}$$

The dimensions of $x$, $u$, and $\psi$ are $n \times 1$, $r \times 1$, and $p \times 1$, respectively. This problem is made into a suboptimal control problem by normalizing the final time through the transformation $\tau = t/t_f$ and by restricting the class of functions to which the optimal control can belong. Here, the restricted class is that of piecewise linear functions The end points $u_1, \ldots, u_m$ of the straight line segments are called nodes.

Formally, this fixed-final-time suboptimal control problem is to find the parameters $u_1, \ldots, u_m$, $t_f$ which minimize the performance index

$$J = \phi(x_f, t_f) + \int_{\tau_0}^{\tau_f} \bar{L}(\tau, x, u_1, \ldots, u_m, t_f)d\tau \tag{4}$$

subject to the dynamics

$$x' = \bar{f}(\tau, x, u_1, \ldots, u_m, t_f), \tag{5}$$

the prescribed boundary conditions

$$\tau_0 = 0, \quad x_0 = x_{0_s}, \quad \tau_f = 1, \quad \psi(x_f, t_f) = 0. \tag{6}$$

In these equations, the prime denotes a derivative with respect to $\tau$, and

$$\begin{aligned} \bar{L}(\tau, x, u_1, \ldots, u_m t_f) &= t_f L(t_f \tau, x, u) \\ \bar{f}(\tau, x, u_1, \ldots, u_m, t_f) &= t_f f(t_f \tau, x, u) \end{aligned} \tag{7}$$

where

$$u(\tau) = u_k + \frac{u_{k+1} - u_k}{\tau_{k+1} - \tau_k}(\tau - \tau_k), \quad \tau_k \le \tau \le \tau_{k+1} \tag{8}$$

and the node times $\tau_k$ are fixed.

By the usual arguments of the calculus of variations, the equations defining the suboptimal solution are given by

$$\begin{aligned} & x' = \bar{f} \\ & \lambda' = -\bar{H}_x^T && \bar{H} = \bar{L} + \lambda^T \bar{L} \\ & \int_{\tau_0}^{\tau_f} \bar{H}_{u_k} d\tau = 0 && k = 1, \ldots, m \\ & \int_{\tau_0}^{\tau_f} \bar{H}_{t_f} d\tau = -G_{t_f} && G = \phi + \nu^T \psi \end{aligned} \tag{9}$$

and

$$\tau_0 = 0, \quad x_0 = x_{0_s}, \quad \tau_f = 1, \quad \psi = 0, \quad \lambda_f = G_{x_f}^T. \tag{10}$$

## 3. Shooting Method

To put the suboptimal control problem in a form suitable for applying the shooting method, new states $v_k(\tau)$ and $w(\tau)$ are introduced to eliminate the integrals in Eq. (9). The optimality conditions become

$$\begin{aligned} & x' = \bar{f}, \\ & \lambda' = -\bar{H}_x^T \\ & v_k' = \bar{H}_{u_k} \quad k = 1, \ldots, m \\ & w' = \bar{H}_{t_f} \end{aligned} \tag{11}$$

and

$$\begin{aligned} & \tau_0 = 0, && x_0 = x_{0_s}, && v_{k_0} = 0, && w_0 = 0 \\ & \tau_f = 1, && \psi = 0, && \lambda_f = G_{x_f}^T, && \\ & && v_{k_f} = 0, && w_f = -G_{t_f}. && \end{aligned} \tag{12}$$

[1] M.J. Thompson Regents Professor
[2] Graduate Research Assistant

If a new state vector $z(\tau)$ is defined as

$$z^T = [x^T\ \lambda^T\ v_1\ \dots\ v_m\ w] \tag{13}$$

and a parameter vector is introduced as

$$a^T = [u_1\ \dots\ u_m\ t_f]\,, \tag{14}$$

the differential equations (11) can be rewritten in the form

$$z' = F(\tau, z, a) \tag{15}$$

Of the initial states, there are $1 + n + m + 1$ conditions; only $\lambda_0$ is unknown. At the final time, there are in Eqs. (12) $1 + p + n + m + 1$ final conditions. Of these, $p$ equations are solved for the $p$ Lagrange multipliers $\nu$ which are in turn eliminated from the remaining conditions to form

$$h(z_f, a) = 0 \tag{16}$$

whose dimension is $(n + m + 1) \times 1$.

The derivation of the equations for the shooting method is straightforward and leads to the following algorithm:

1. Guess $\lambda_0$ and $a$
2. Integrate from $\tau_0 = 0$ to $\tau_f = 1$

$$\begin{aligned} z' &= F & z_0 &\text{ known} \\ \Phi_2' &= F_z \Phi_2 & \Phi_{2_0} &= [0\ I\ 0\ 0]^T \\ \Psi' &= F_z \Psi + F_a & \Psi_0 &= 0 \end{aligned} \tag{17}$$

3. Calculate $\|h\|$.
4. Calculate $\delta\lambda_0$ and $\delta a$ by solving

$$[h_{z_f}\Phi_{2_f} \vdots\ h_{z_f}\Psi_f] \begin{bmatrix} \delta\lambda_0 \\ \delta a \end{bmatrix} = -\alpha h \tag{18}$$

and using a norm reduction scheme to determine $\alpha$.

5. Check for convergence ($\|h\| < \epsilon$). If not, go to 2.

The advantage of this method is that there is absolutely no influence of $\lambda_0$ on $x$. On the other hand, the sensitivity of the shooting method to $\lambda_0$ is replaced by having to accept an approximate solution. However, by using a reasonable number of nodes, it should be possible to obtain $\lambda_0$'s for which the exact shooting method can be converged.

## 4. Optimal Planar Trajectory for the ALS

The Advanced Launch System is a two-stage rocket consisting of a core with a side-mounted booster. Staging occurs at the fixed time of burnout of the booster. Ref. 1 contains a description of the physical model.

In the optimization problem, the performance index is the final mass; the state equations are the equations of motion for flight in a great circle plane over a nonrotating spherical earth where the control is the angle of attack; the initial conditions are all specified; and final conditions are imposed on altitude, velocity, and flight path angle.

Converged results are presented in Table 1 and Fig. 1 for three first and second stage node arrangements. Starting multipliers for the 3-2 case are given in Table 1, and the control nodes are taken to be $\alpha$ = -.5, 10., 6., 4., -4. deg and $t_f$ = 300. sec. Convergence required 17 iterations and 241 sec of CPU time on a CDC Cyber computer. Also presented are the converged values obtained from the standard shooting method. Note that the optimal results are approached as the number of nodes is increased.

Other than having to derive the multiplier equations, no difficulties have been encountered during these calculations.

## 5. Conclusions

The shooting approach to suboptimal control is an effective way to obtain approximate optimal trajectories and to obtain starting Lagrange multipliers for the regular shooting method.

## Reference


1. Shaver, D. A., and Hull, D. G., "Advanced Launch System Trajectory Optimization Using Suboptimal Control," AIAA Guidance, Navigation, and Control Conference, Portland, Oregon, August, 1990.


## Acknowledgement


This research was supported by NASA Langley Grant NAG-1944 monitored by Dr. Daniel D. Moerder.


Table 1: Converged Results

| | node pattern | | | | optimum |
|---|---|---|---|---|---|
| | 3–2 guess | 3–2 | 5–5 | 9–5 | |
| P.I. | | 0.8522 | 0.8529 | 0.8541 | 0.8544 |
| $t_f(sec)$ | 300.00 | 371.72 | 371.69 | 371.64 | 371.63 |
| $\lambda_{h0}$ | 1.0 | -8.862E-6 | -8.925E-6 | -8.914E-6 | -8.939E-6 |
| $\lambda_{v0}$ | 1.0 | -4.148E-4 | -4.150E-4 | -4.129E-4 | -4.125E-4 |
| $\lambda_{\gamma 0}$ | 1.0 | 1.461E-2 | 8.702E-3 | 3.925E-3 | 2.584E-3 |
| $\lambda_{m0}$ | 1.0 | -2.004E-5 | -2.017E-5 | -2.016E-5 | -2.013E-5 |

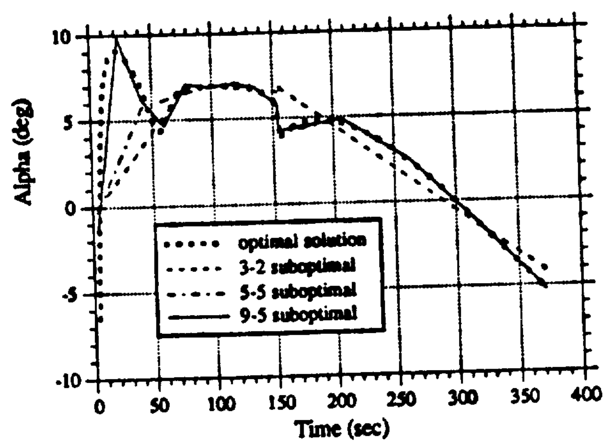


Figure 1: Angle of Attack Histories

ORIGINAL PAGE IS
OF POOR QUALITY

- AIAA Third Aerospace Planes Conference -
Orlando, Fl.--December 1991

# NEIGHBORING EXTREMAL GUIDANCE FOR SYSTEMS WITH A PIECEWISE LINEAR CONTROL



David G. Hull[1] and Clifford E.Helfrich[2]
The University of Texas at Austin
Austin, Texas

## ABSTRACT

The neighboring extremal 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. This process leads to a feedback relationship between the control perturbation and the measured deviation from the nominal state. A simple example, the lunar launch problem, is used to demonstrate the validity of the guidance law. For model errors on the order of 5%, the results indicate that 5% errors occur in the final conditions.

## INTRODUCTION

In order to develop the neighboring optimal guidance law for a dynamical system, it is first necessary to obtain the optimal control, and this can be a formidable task. Currently, most trajectory optimization is accomplished by restricting the class of control functions to some subclass, say piecewise linear functions (suboptimal control). Then, the control variables are parameters (nodes of piecewise linear function), and the suboptimal control is found by applying nonlinear programming methods. Hence, the subject of this paper is the development of the neighboring suboptimal feedback control law, assuming that the suboptimal control law is available.

Given the suboptimal control and a perturbation in the state at some time, the neighboring suboptimal control is found by minimizing the increase is the performance index subject to the constraint that the final conditions must be satisfied. Since the first variation vanishes, minimizing the increase in the performance index is equivalent to minimizing the second variation. The constraint of satisfying the final conditions is obtained through the use of transition matrices to the final point. The above process leads to an analytical expression for the gains of the neighboring suboptimal feedback control law. Because of the simplicity of the control law, the suboptimal control rule can be applied to the vehicles rather than sample and hold. This should allow the sample time to be increased, if errors do not grow too rapidly.

To test this guidance rule, it is applied to a simple trajectory problem with various levels of modeling errors. The results indicate that this guidance approach has merit.

## SUBOPTIMAL CONTROL PROBLEM

The optimal control problem [1] being considered here is to find the control history $u(t)$ which minimizes the performance index

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

subject to the state differential equations

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

the prescribed initial conditions

$$t_0 = t_{0_s}, \qquad x_0 = x_{0_s} \ , \tag{3}$$

and the prescribed final conditions

$$\psi(t_f, x_f) = 0 \ . \tag{4}$$

Here, this problem is converted into a suboptimal control problem [2] by assuming that the controls are piecewise linear, meaning that the unknowns become the junction points (nodes) of the linear control segments and the final time.

If $a$ denotes the unknown parameter vector, that is, $a^T = [t_f, u_{11}, u_{12}, \dots, u_{21}, u_{22}, \dots]$, the suboptimal control problem is stated as follows:

Find the set of parameters $a$ which minimizes the performance index

$$J = F(a) \tag{5}$$

[1] M. J. Thompson Regents Professor, Associate Fellow AIAA
[2] Graduate Research Assistant

subject to the equality constraints

$$C(a) = 0 \, . \tag{6}$$

The differential constraints are an integral part of defining the functions $F$ and $C$ and are written as

$$\begin{array}{l} \dfrac{dx}{d\tau} = g(\tau, x, a) \\ \tau_0 = 0, \quad x_0 = x_{0_s}, \quad \tau_f = 1 \end{array} \tag{7}$$

where $\tau = t/t_f$ and $x_{0_s}$ are the specified values of the initial states.

It is assumed that this problem is solved numerically by using a nonlinear programming code, and the next step is to find the neighboring suboptimal feedback control law.

## 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 modelling errors, the vehicle when using the nominal control 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

$$\Delta J = \tfrac{1}{2} \delta a^T G_{aa} \delta a \tag{8}$$

where $G = \Phi + \nu^T \Psi$ is the augmented performance index and $\nu$ is a constant Lagrange multiplier. Once the suboptimal control has been obtained numerically, the second derivative matrix $G_{aa}$ can be computed numerically. The next step is to find the constraints on $\delta a$ which guarantee satisfaction of the final conditions (4).

The variation of the state equation (7) 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

$$\begin{array}{ll} \tau_0 = \tau_{0_s}, & \delta x_0 = \delta x_{0_s} \\ \tau_f = 1, & \psi_{x_f} \delta x_f + \psi_{\tau_f} \delta t_f = 0 \, . \end{array} \tag{10}$$

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 following differential equations:

$$\begin{array}{rcl} \dot{\Phi} & = & g_x \, \Phi \\ \dot{\Psi} & = & g_x \Psi + g_a \end{array} \tag{13}$$

which must be solved subject to the boundary conditions (12). Once $\Phi$ and $\Psi$ have been obtained, Eq. (11) can be used.

To satisfy the final condition (10), Eq. (11) is rewritten as

$$\delta x_f = \Phi^{-1} \delta x - \Phi^{-1} \Psi \delta a \tag{14}$$

Then, assuming $\Psi_{t_f} = 0$, Eq. (10) leads to

$$\psi_{x_f} \Phi^{-1} \delta x - \psi_{x_f} \Phi^{-1} \Psi \delta a = 0 \, . \tag{15}$$

Applied to $\tau_0$, this equation becomes

$$\psi_{x_f} \Phi_0^{-1} \Psi_0 \delta a - \psi_{x_f} \Phi_0^{-1} \delta x_0 = 0 \tag{16}$$

and is the constraint on the control node perturbation $\delta a$ imposed by the final condition.

The last step is to minimize $\Delta J$ as given by Eq. (8) with respect to $\delta a$ subject to the constraint (16). Standard parameter optimization methods lead to

$$\delta a = K_0 \delta x_0 \tag{17}$$

where the gain $K_0$ is given by

$$\begin{array}{l} K_0 = G_{aa}^{-1} \Psi_0^T \Phi_0^{-T} \psi_{x_f}^T \cdot \\ (\psi_{x_f} \Phi_0^{-1} \Psi_0 G_{aa}^{-1} \Psi_0^T \Phi_0^{-1} \psi_{x_f}^T)^{-1} \psi_{x_f} \Phi_0^{-1} . \end{array} \tag{18}$$

If the sampling is performed continuously, the parameter perturbation becomes

$$\delta a = K \, \delta x \tag{19}$$

where

$$\begin{array}{rcl} K & = & G_{aa}^{-1} \Psi^T \Phi^{-1} \psi_{x_f}^T \cdot \\ & & (\psi_{x_f} \Phi^{-1} \Psi G_{aa}^{-1} \Psi^T \Phi^{-1} \psi_{x_f}^T)^{-1} \psi_{x_f} \Phi^{-1} . \end{array} \tag{20}$$

These gains can be computed at several values of $\tau$ and stored in the onboard computer for interpolation purposes.

Two difficulties occur in the use of Eq. (19) as a guidance law. First, $\Psi$ goes to zero as $\tau$ approaches unity so that the computation of the gains becomes indefinite (zero over zero). This has been handled in the following application by computing the gains at $\tau = .950$ and $\tau = .975$ and extrapolating them to $\tau = 1$. The second problem is determining the value of $\tau$ on the perturbed path since the perturbed final time is unknown. This has been accomplished iteratively by guessing $\delta t_f$, computing $\tau = t/(t_f + \delta t_f)$, computing $\delta a$ and, hence, $\delta t_f$, and repeating the computation until the computed $\delta t_f$ nearly equals the guessed $\delta t_f$.

## EXAMPLE – LUNAR LAUNCH PROBLEM

The lunar launch problem has been selected as a simple example to illustrate the application of this guidance law. The optimal control problem is to find the thrust inclination history $\theta(t)$ which minimizes the time to insertion

$$J = t_f \tag{21}$$

subject to the differential constraints

$$\begin{aligned} \dot{x} &= u \\ \dot{y} &= v \\ \dot{u} &= \alpha \cos\theta \\ \dot{v} &= \alpha \sin\theta - g \ , \end{aligned} \tag{22}$$

the prescribed initial conditions

$$t_0 = x_0 = y_0 = u_0 = v_0 = 0 \ , \tag{23}$$

and the prescribed final conditions

$$y_0 = 50,000 \text{ ft}, \quad u_0 = 5,444 \text{ ft/sec}, \quad v_0 = 0 \text{ ft/sec}. \tag{24}$$

The quantities $\alpha$ and $g$ are the constant thrust acceleration and lunar acceleration of gravity whose nominal values are $\alpha = 20.8$ ft/sec$^2$ and $g = 5.32$ ft/sec$^2$.

Using five nodes for the suboptimal control calculation leads to

$$\begin{aligned} t_f &= 272.7 \text{ sec}, \\ \theta_1 &= 26.09 \text{ deg}, \\ \theta_2 &= 20.68 \text{ deg} \\ \theta_3 &= 15.34 \text{ deg}, \\ \theta_4 &= 9.061 \text{ deg}, \\ \theta_5 &= 3.113 \text{ deg} \end{aligned} \tag{25}$$

To test the guidance law, a 5% error is introduced in $\alpha$ which drives the vehicle away from the nominal. Gains have been computed stored at each .025 in $\tau$. Two implementations have been performed: one is to

| $a$ $\left(\frac{ft}{sec^2}\right)$ | % Change in $a$ | State | % Deviation from Optimal | |
|---|---|---|---|---|
| | | | Sample-Hold | Integrate Control |
| 19.760 | -5.0 | y | 1.15 | 1.15 |
| | | u | 5.96 | 6.35 |
| | | v | 0.88 | 0.81 |
| 21.840 | +5.0 | y | 0.99 | 0.96 |
| | | u | 4.50 | 4.45 |
| | | v | 0.34 | 1.00 |

Table 1: Results for 5% Modeling Error in Thrust

| $g$ $\left(\frac{ft}{sec^2}\right)$ | % Change in $g$ | State | % Deviation from Optimal | |
|---|---|---|---|---|
| | | | Sample-Hold | Integrate Control |
| 5.054 | -5.0 | y | 0.07 | 0.08 |
| | | u | 0.11 | 0.26 |
| | | v | 0.25 | 0.01 |
| 5.586 | +5.0 | y | 0.10 | 0.09 |
| | | u | 0.43 | 0.41 |
| | | v | 0.17 | 0.07 |

Table 2: Results for 5% Modeling Error in Gravity

use sample and hold and the other is to use the actual linear control. Results are shown for a 4 sec sample time in Table 1. Note that a 5% error in $\alpha$ leads to roughly a 5% error in the insertion conditions.

That the linear control does not do uniformly better than sample and hold is disappointing. It is felt that the sample time could be increased substantially for the linear control relative to sample and hold and still yield good results. At any rate these are preliminary results and further study is warranted.

Similar results have been developed for a 5% error in $g$ and are shown in Table 2. Qualitatively, these results are similar to those in Table 1.

## DISCUSSION AND CONCLUSIONS

The neighboring extremal feedback control law has been developed for systems with a piecewise-linear control whose nominal control and trajectory have been computed using nonlinear programming techniques. Given a perturbation in the state, the neighboring extremal control perturbation is obtained by minimizing the increase in the performance index relative to the nominal value subject to the constraint that the final conditions be satisfied. Numerical results for the lunar launch problem with mismatches in the thrust acceleration and gravity acceleration show that 5% model errors lead to 5% final condition errors. Further study of this guidance law seems warranted.


## ACKNOWLEDGEMENT

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


## REFERENCES


1. Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere Publishing Corporation, New York, 1975.

2. 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, OR, 1990, pp. 892–901.

ACC, Chicago, 1992

PREPRINT
93A 27793

# NEIGHBORING SUBOPTIMAL CONTROL

David G. Hull[1]

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

## Abstract

The neighboring extremal 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. This process leads to a feedback relationship between the control perturbation and the measured deviation from the nominal state.

## 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 variables are parameters (nodes of piecewise linear function), and the suboptimal control is found by applying nonlinear programming methods. Hence, the subject of this paper is the development of the neighboring suboptimal feedback control law, assuming that the suboptimal control law is available.

## Suboptimal Control Problem

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

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

subject to the state differential equations

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

the prescribed initial conditions

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

[1] M. J. Thompson Regents Professor

and the prescribed final conditions

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

Here, the time has been normalized by the final time, that is, $\tau = t/t_f$ where $t_f$ is an unknown parameter. 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 and the final time.

If $a$ denotes the unknown parameter vector, that is, $a^T = [t_f, u_{11}, u_{12}, \dots, u_{21}, u_{22}, \dots]$, the differential equations (2) and its boundary conditions can be rewritten as

$$\frac{dx}{d\tau} = g(\tau, x, a), \quad \tau_0 = \tau_{0_s}, \quad x_0 = x_{0_s}, \quad \tau_f = 1\ . \tag{5}$$

Given $a$, these equations can be integrated to obtain $x_f = x_f(a)$ so that $\phi = \phi[x_f(a), t_f] = F(a)$ and $\psi = \psi[x_f(a), t_f] = C(a)$ 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 $\nu$. The second variation becomes $\delta^2 J' = \delta a^T G_{aa} \delta a > 0$ where $C_a \delta a = 0$. $\delta a$ can be divided into dependent and independent parts, 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 modelling errors, the vehicle when using the nominal control 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

$$\Delta J = \tfrac{1}{2}\, \delta a^T G_{aa} \delta a \tag{6}$$

subject to $C_a \delta a = 0$ which is imposed below. Once the suboptimal control has been obtained, the second derivative matrix $G_{aa}$ can be computed numerically. The next step is to find the constraints on $\delta a$ which guarantee satisfaction of the final conditions (4).

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

$$\frac{d}{d\tau}\delta x = g_x\, \delta x + g_a\, \delta a \tag{7}$$

which must be solved subject to the boundary conditions

$$\begin{array}{ll} \tau_0 = \tau_{0_s}, & \delta x_0 = \delta x_{0_s} \\ \tau_f = 1, & \psi_{x_f} \delta x_f + \psi_{t_f} \delta t_f = 0\ . \end{array} \tag{8}$$

Next, the solution of Eq. (7) is assumed to have the transition matrix form

$$\delta x = \Phi \delta x_f + \Psi \delta a \tag{9}$$

where

$$\Phi_f = I, \qquad \Psi_f = 0 \tag{10}$$

to guarantee that $\delta x_f = \delta x_f$. Then, substituting Eq. (9) into Eq. (7) and equating like coefficients leads to the following differential equations:

$$\begin{array}{rcl} \Phi' & = & g_x \Phi \\ \Psi' & = & g_x \Psi + g_a \end{array} \tag{11}$$

which must be solved subject to the boundary conditions (10). Once $\Phi$ and $\Psi$ have been obtained, Eq. (9) can be used.

To satisfy the final condition (8), Eq. (9) is rewritten as

$$\delta x_f = \Phi^{-1} \delta x - \Phi^{-1} \Psi \delta a \tag{12}$$

Then, for the case where $\psi_{t_f} = 0$, Eq. (8) leads to

$$\psi_{x_f} \Phi^{-1} \delta x - \psi_{x_f} \Phi^{-1} \Psi \delta a = 0\ . \tag{13}$$

Applied to $\tau_0$, this equation becomes

$$\psi_{x_f} \Phi_0^{-1} \Psi_0 \delta a - \psi_{x_f} \Phi_0^{-1} \delta x_0 = 0 \tag{14}$$

and is the constraint on the control node perturbation $\delta a$ imposed by the final condition.

The last step is to minimize $\Delta J$ as given by Eq. (6) with respect to $\delta a$ subject to the constraint (14). Standard parameter optimization methods lead to

$$\delta a = K_0 \delta x_0 \tag{15}$$

where the gain $K_0$ is given by

$$\begin{array}{c} K_0 = G_{aa}^{-1} \Psi_0^T \Phi_0^{-T} \psi_{x_f}^T \cdot \\ (\psi_{x_f} \Phi_0^{-1} \Psi_0 G_{aa}^{-1} \Psi_0^T \Phi_0^{-1} \psi_{x_f}^T)^{-1} \psi_{x_f} \Phi_0^{-1}. \end{array} \tag{16}$$

## Application

In Ref. 2, neighboring suboptimal control has been applied in the same manner as neighboring optimal control, that is, sampling is assumed to occur continuously so that $\tau_0 = \tau$. However, in optimal control, any part of an optimal trajectory to the final constraint manifold is an optimal trajectory, but this is not the case in suboptimal control. In fact, there may not even be enough nodes between the sample point and the final constraint manifold to satisfy the boundary conditions.

Two alternate approaches are being considered. First, additional nodes are placed near the final constraint manifold to make neighboring suboptimal control valid near the end of the trajectory. Second, the suboptimal control is computed from each node to the final constraint manifold, and the gains (16) are computed at each node. These gains are linearly interpolated for the operation of the vehicle. Unfortunately, no results for either case are available at the time of this writing.


## Acknowledgement

This research was 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, OR, 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.

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

N/S
$\overline{52-15}$ 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 $\nu$. The second variation becomes $\delta^2 J' = \delta a^T G_{aa} \delta a > 0$ where $C_a \delta a = 0$. $\delta a$ 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 $\delta a$ 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

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

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 $\Phi$ and $\Psi$ have been obtained, Eq. (11) can be used.

To satisfy the final condition (10), Eq. (11) is evaluated at $\tau_0$ 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, $\delta a$, imposed by the final condition.

The last step is to minimize $\Delta J$ as given by Eq. (7) with respect to $\delta a$ 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 $\Delta a$ 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:

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

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 $9 \times 3$ 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 $9 \times 2$ 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 $\delta a_8$ 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 $\delta a_7$, 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 $\Delta V$ 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 |

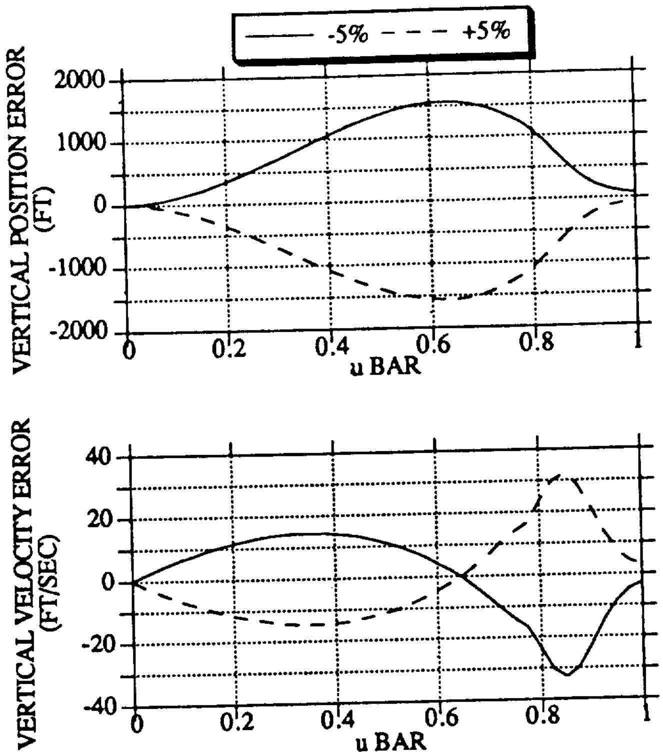


Figure 1: 9-Node Single Optimization Deviation Histories

## 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,

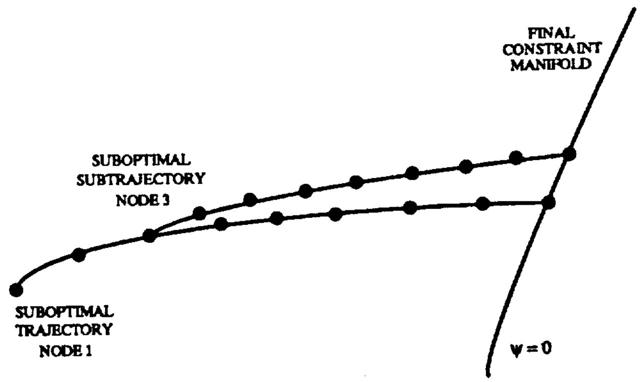


Figure 2: Multiple Optimization Approach

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 |

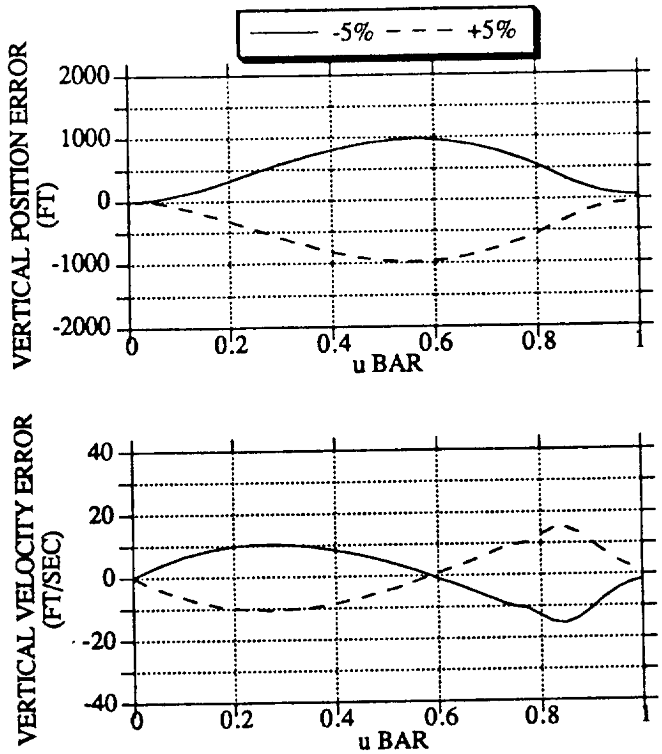


Figure 3: 9-Node Multiple Optimization Deviation Histories

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.