AD-A258 191

MENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1992	THESIS/DISSERTATION

4. TITLE AND SUBTITLE

Neighboring Extremal Guidance for Systems with Piecewise Linear Control Using Multiple Optimization

5. FUNDING NUMBERS

6. AUTHOR(S)

Michael James Nowak, Major

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFIT Student Attending: University of Texas

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/CI/CIA-92-102

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFIT/CI
Wright-Patterson AFB OH 45433-6583

DTIC
ELECTE
DEC 1 0 1992
S D
C

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release IAW 190-1
Distributed Unlimited
ERNEST A. HAYGOOD, Captain, USAF
Executive Officer

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

14. SUBJECT TERMS			15. NUMBER OF PAGES
			36
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Standard Form 298 (Rev 2-89)

NEIGHBORING EXTREMAL GUIDANCE FOR SYSTEMS WITH PIECEWISE LINEAR CONTROL USING MULTIPLE OPTIMIZATION

by

MICHAEL JAMES NOWAK, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

August, 1992

NEIGHBORING EXTREMAL GUIDANCE FOR SYSTEMS WITH PIECEWISE LINEAR CONTROL USING MULTIPLE OPTIMIZATION

APPROVED:

Supervisor: _David G. Hull_
Dr. David G. Hull

R. Bishop
Dr. Robert H. Bishop

Acknowledgments

Abstract

NEIGHBORING EXTREMAL GUIDANCE FOR SYSTEMS WITH PIECEWISE LINEAR CONTROL USING MULTIPLE OPTIMIZATION

by

MICHAEL JAMES NOWAK, B.S.

SUPERVISING PROFESSOR: Dr. David G. Hull

A neighboring extremal guidance law for systems with piecewise linear control using multiple optimization along the suboptimal trajectory is developed. The guidance law computes control gains that relate the difference in the perturbed and nominal trajectory with a control update. Previous research in neighboring extremal control has concentrated on calculating the control gains using a single optimization from the initial conditions to the final constraint manifold. The purpose of this study is to develop a more optimal method of computing the control gains at each node through multiple optimization.

iv

Multiple optimization reoptimizes the trajectory from each control node on the suboptimal trajectory. The suboptimal control vectors for each subtrajectory are used to compute a more optimal set of control gains. An example optimization problem, the lunar launch problem, is used to compare the performance of both the single optimization and multiple optimization control laws. Overall, the multiple optimization gains returned results comparable to those achieved with single optimization. For perturbations of ±5% in either gravity or thrust acceleration, endpoint conditions for multiple optimization gains never exceeded more than 50 feet in position or 4.3 ft/sec in vertical velocity from the desired boundary conditions. Direct comparisons of performance for specific model perturbations result in mixed conclusions. For thrust perturbations, single optimization gains deleiver smaller errors in meeting end point conditions, while for gravity perturbations, multiple gains give better results.

Table of Contents

List of Tables

List of Figures

List of Symbols

English Symbols

a	parameter vector for suboptimal control problem
G	augmented performance index
$\mathcal{J}$	performance index of neighboring extremal problem
J	performance index
K	gain matrix
$f(\ )$	state differential equation vector
g	acceleration due to gravity ($\frac{ft}{sec^2}$)
$g(\ )$	paramaterized state differential equation vector
t	time (*sec*)
u	horizontal component of velocity ($\frac{ft}{sec}$)
$\bar{u}$	normalized horizontal velocity
v	vertical component of velocity ($\frac{ft}{sec}$)
x	state vector
y	altitude (*ft*)

Greek Symbols

α	thrust acceleration ($\frac{ft}{sec^2}$)
δ	variation
θ	angle between thrust vector and horizontal
ν	vector of Lagrange multipliers

τ	normalized time
Φ	state transition matrix
ϕ	performance index in suboptimal control problem
ψ	equality final constraint vector in suboptimal control problem
Ψ	transition matrix

Subscripts and Superscripts

$(\ )_0$	value at time zero
$(\ )_{0_s}$	specified value at time zero
$(\ )_a$	derivative with respect to the parameters
$(\ )_f$	final value
$(\ )_p$	pertaining to arbitrary time or $\bar{u}$
$(\ )_s$	specified value
$(\ )_x$	derivative with respect to the states
$(\ )_{x_f}$	derivative with respect to state at final point
$(\ \dot{)}$	derivative with respect to t
$(\ )'$	derivative with respect to τ or $\bar{u}$
$(\ )^{-1}$	inverse
$(\ )^T$	transpose

Chapter 1

Introduction

An inherent problem in optimal and suboptimal control is that the optimal trajectory is not usually realizable. Unpredicted model errors and/or perturbations in the initial conditions can result in deviations from the optimal trajectory. Recomputing the revised optimal control from the point of perturbation to the final constraint manifold is not always possible due to computer processing limitations. As a result, the precomputed optimal control does not produce the desired trajectory.

In light of these deviations, one approach to optimal control is to provide guidance in the neighborhood of the optimal trajectory. Assuming model errors and perturbations are small enough, the neighboring extremal guidance should keep the perturbed trajectory in the neighborhood of the extremal, or optimal, path.

The purpose of this research is to develop a multiple optimization neighboring extremal guidance law for systems using piecewise linear control and apply it to a representative problem. The guidance law provides updates to the suboptimal control vector based on errors in the perturbed versus the nominal trajectory and precomputed neighboring extremal gains. The neighboring extremal control gains are found by finding the change in control required to minimize the change in the performance index subject to the trajectory reach-

ing the final constraints.

Two methods of computing these gains are presented and discussed. In the first method the gains are computed along the optimal trajectory at node points corresponding to the piecewise linear control nodes. The second method reoptimizes from each node along the optimal trajectory to the final constraint manifold. The neighboring extremal gains are then computed by integrating back from the final constraint manifold along the reoptimized trajectory to each node.

To test each approach, a simple example, the lunar launch problem, is used. The lunar launch problem involves placing a vehicle in a low moon orbit, in minimum time, subject to constraints on the final conditions. Perturbations in vehicle acceleration and lunar gravity are used to test the ability of the control law to correct for "model errors" and reach the desired final conditions.

Chapter 2

Suboptimal Control Problem

The fixed final time problem is to find the control, u(t), that minimizes the performance index

$$J = \phi(x_f), \tag{2.1}$$

with system dynamics

$$\dot{x} = f(t, x, u), \tag{2.2}$$

and initial conditions

$$t_0 = t_{0_s} , \quad x_0 = x_{0_s}, \tag{2.3}$$

subject to boundary constraints

$$\psi(x_f) = 0. \tag{2.4}$$

The solution to this system of equations produces the optimal control history, $u(t)$. However, often the system of equations cannot be solved in a closed form or may be too complicated to solve numerically. By parametrizing the control, a nonlinear programming code is employed to solve for the suboptimal control.

Consider a suboptimal control that is of the class of piecewise linear functions. For this study only a single control is considered; however, the results

can be generalized to include multiple control systems. Hence, the control is expressed as a parameter vector

$$a^T = [a_{11}, a_{12}, \cdots, a_{1r}], \tag{2.5}$$

where r is the number of control points or nodes along the trajectory. By normalizing the variable of integration so the integration limits are 0 to 1, the control nodes become fixed as fractional representations. If a new variable of integration, τ, is defined as

$$\tau = \frac{t}{t_f}, \tag{2.6}$$

the system dynamics become

$$x' = \frac{dx}{d\tau} = t_f f(t_f \tau, x, a) = g(\tau, t_f, x, a), \tag{2.7}$$

where

$$\tau_0 = 0 , \quad \tau_f = 1. \tag{2.8}$$

Since the states are a function of the parametrized control, the fixed final time, suboptimal control problem is to find the piecewise continuous control, a, that minimizes the performance index

$$J = \phi(x_f(a)), \tag{2.9}$$

with boundary constraints

$$\psi(x_f(a)) = 0, \tag{2.10}$$

subject to system dynamics

$$x' = \frac{dx}{d\tau} = g(\tau, t_f, x, a), \tag{2.11}$$

where

$$\tau_0 = 0 , \quad \tau_f = 1. \qquad (2.12)$$

The suboptimal control vector, a, is obtained using the nonlinear programming code VF02AD. Equipped with the suboptimal control and trajectory, the neighboring extremal guidance law can be developed.

Chapter 3

Neighboring Extremal Control

The suboptimal control solution provides a control history and a nominal trajectory that minimizes the performance index. As long as the vehicle remains on the nominal trajectory, the control law guides the vehicle to the desired final conditions in the minimum time. Realistically, however, the nominal trajectory is rarely achieved. Perturbations in initial conditions or model errors produce deviations from this trajectory. The goal of neighboring extremal control is to develop a guidance law for the perturbed trajectory in the neighborhood of the nominal trajectory. The guidance law provides control corrections that ultimately reduce the change in the performance index of the perturbed trajectory from that of the nominal trajectory.

3.1 Neighboring Control Derivation

The basis of developing a neighboring extremal control law is to find the change in control required to minimize the increase in the performance index from the point of the perturbation to the final constraint manifold. The derivation of this control law was originally developed by Hull and Helfrich [1]. A summary of the basic derivation is included here for completeness.

Since $x_f = x_f(a)$ from Eq. (2.11), the augmented performance index

for the suboptimal control problem is expressed as

$$J' = \phi(x_f) + \nu^T \psi(x_f) \equiv G(a, \nu), \qquad (3.1)$$

where ϕ is the performance index, ψ is the vector of final constraints and ν is the vector of Lagrange multipliers. The change in the performance index for a perturbation in the control vector, a, to second order is

$$\Delta J' = G_a \delta a + \frac{1}{2!} \delta a^T G_{aa}^T \delta a \qquad (3.2)$$

Since the first variation vanishes on the extremal path, the change in the performance index is approximated as

$$\Delta J' = \frac{1}{2!} \delta a^T G_{aa}^T \delta a. \qquad (3.3)$$

In Eq. (3.3), the δa's are not independent since the final conditions must be satisfied. Hence, it is necessary to determine the relationship between δa and δx_0 such that $\delta \psi = \psi_{x_f} \delta x_f = 0$. Eq. (2.11) relates the state and the control through a set of differential equations. Taking the variation of this equation results in

$$\delta x' = g_x \delta x + g_a \delta a, \qquad (3.4)$$

with boundary conditions

$$\tau_0 = 0, \quad \tau_f = 1, \qquad (3.5)$$

$$\delta x_0 = \delta x_{0_s}, \qquad (3.6)$$

$$\psi_{x_f} \delta x_f = 0. \qquad (3.7)$$

A solution of the form

$$\delta x = \Phi \delta x_f + \Psi \delta a \qquad (3.8)$$

is assumed, with boundary conditions

$$\Phi_f = I, \tag{3.9}$$

$$\Psi_f = 0, \tag{3.10}$$

and substituted into Eq. (3.4). This results in

$$(\Phi' - g_x\Phi)\delta x_f + (\Psi' - g_x\Psi - g_a)\delta a = 0. \tag{3.11}$$

To satisfy Eq. (3.11) the coefficients of δx_f and δa are chosen to equal zero. This produces the differential equations governing Φ and Ψ:

$$\Phi' = g_x\Phi \tag{3.12}$$

and

$$\Psi' = g_x\Psi + g_a \tag{3.13}$$

with final conditions

$$\Phi_f = I, \tag{3.14}$$

$$\Psi_f = 0. \tag{3.15}$$

These differential equations are integrated backward from τ_f to find Φ and Ψ at any node. Multiplying Eq. (3.8) by Φ^{-1}, solving for δx_f, and substituting into Eq. (3.7) gives the desired relation:

$$\psi_{x_f}\Phi_0^{-1}\Psi_0\delta a - \psi_{x_f}\Phi_0^{-1}\delta x_0 = 0. \tag{3.16}$$

This boundary constraint is combined with the performance index in Eq. (3.3) to form a new augmented performance index

$$\mathcal{J}' = \frac{1}{2!}\delta a^T G_{aa}\delta a + \delta\nu^T(\psi_{x_f}\Phi_0^{-1}\Psi_0\delta a - \psi_{x_f}\Phi_0^{-1}\delta x_0). \tag{3.17}$$

Minimizing $\mathcal{J}'$ minimizes the change in the performance index due to a perturbation in the control, subject to the boundary constraint that relates the control and the state vector. The matrix G_{aa} is not necessarily positive definite since δa has both dependent and independent variations. However, the boundary constraint is used to eliminate the dependent variations so the resulting matrix is composed of only independent variations, and the performance index is in quadratic form.

The first variation of this performance index is

$$\delta \mathcal{J}' = (\delta a^T G_{aa} + \delta \nu^T \psi_{x_f} \Phi_0^{-1} \Psi_0) \delta(\delta a). \tag{3.18}$$

To satisfy the requirement that the first variation vanish, the coefficient of $\delta(\delta a)$ must be zero. Therefore,

$$\delta a^T G_{aa} + \delta \nu^T \psi_{x_f} \Phi_0^{-1} \Psi_0 = 0. \tag{3.19}$$

Solving for δa,

$$\delta a = -G_{aa}^{-1} \Psi_0^T \Phi_0^{-T} \psi_{x_f} \delta \nu \tag{3.20}$$

substituting into the constraint, Eq. (3.16), and solving for $\delta \nu$ gives

$$\delta \nu = -(\psi_{x_f} \Phi_0^{-1} \Psi_0 G_{AA}^{-1} \Psi_0^T \Phi_0^{-T} \psi_{x_f}^T)^{-1} \psi_{x_f} \Phi_0^{-1} \delta x_0 . \tag{3.21}$$

Hence, Eq. (3.20) leads to

$$\delta a = K_0 \delta x_0, \tag{3.22}$$

where

$$K_0 = G_{aa}^{-1} \Psi_0^T \Phi_0^{-T} \psi_{x_f}^T (\psi_{x_f} \Phi_0^{-1} \Psi_0 G_{aa}^{-1} \Psi_0^T \Phi_0^{-T} \psi_{x_f}^T)^{-1} \psi_{x_f} \Phi_0^{-1} \tag{3.23}$$

is the gain matrix at τ_0. Eq. (3.22) relates the change in the control required to minimize the change in the performance index for a perturbation in the state at x_0 and still satisfy the final conditions.

3.2 Single Trajectory Optimization

For continuous systems the neighboring extremal guidance law can be applied throughout the trajectory. By the optimality condition [2], the trajectory from each point along the optimal path to the final constraint manifold can be thought of as the optimal trajectory from that point. As a result, the gain equation for the continuous optimal control problem is valid throughout the trajectory.

For the suboptimal problem, however, the optimality condition does not apply. If the suboptimal trajectory has r control nodes, the suboptimal trajectory from node 1 is an r-node optimal trajectory. From the second node though, the suboptimal trajectory is an $(r-1)$-node suboptimal trajectory. Furthermore, in order to satisfy a suboptimal control problem with m boundary constraints, at least m control nodes must be available. For a parametrized control with r control nodes, m nodes are necessary to solve the boundary conditions while the remaining $r-m$ nodes ($m \leq r$) provide the optimization. As a result, neighboring extremal gains can only be computed along the suboptimal trajectory at the first $r-m+1$ node points. The remaining $m-1$ node points do not have enough control nodes to satisfy the boundary constraints.

The single trajectory optimization method computes the neighboring extremal gains by evaluating the gain equation (Eq. 3.23) with the values of Φ and Ψ along the suboptimal trajectory. The differential equations, Φ' and Ψ', are integrated back from the final constraint manifold to each node on the suboptimal trajectory using the suboptimal control vector. Gain matrices are computed at each node point up to node $r-m+1$. Since the calculation of

K_p is based on the nominal trajectory, the gains can be precomputed at node points and stored.

Although the gain matrix at each node is an $r \times n$ matrix, where r is the number of nodes and n are the number of states, only gains associated with the current node are retained. Each gain matrix relates the change in control along the suboptimal trajectory to an error in the state vector at a node point along the trajectory. Obviously, control changes at nodes that have been "passed" do not affect the current control and are eliminated. Similarly, gains related to nodes "in front" of the current node are also eliminated. As a result, the matrix reduces to a $1 \times n$ vector.

Since each node along the trajectory corresponds to a particular τ, the gain vector at any τ_p is found by linearly interpolating adjacent gain vectors. As the last gain vector is at node $r - m + 1$, gain vectors beyond this node are linearly extrapolated based on the last two gain vectors.

3.3   Multiple Trajectory Optimization

The focus of this research has turned to developing another method of computing the control gains along the suboptimal trajectory. This approach uses reoptimized r-node control vectors from each node along the suboptimal trajectory to compute the gain matrices. Fig. 3.1 depicts a five-node example of a trajectory and subtrajectory. To prevent confusion, the trajectory from any node to the final constraint manifold, except for the first node, is called a suboptimal subtrajectory. The suboptimal subtrajectory from the first node to the final constraint manifold is also the suboptimal trajectory.

Each subtrajectory control vector is slightly different from the suboptimal trajectory. As expected the suboptimal trajectory from node 1 is the same as in the single optimization method. However, from the second node, *multiple optimization provides an r-node subtrajectory* to the final constraint manifold, while the single optimization method only has $r - 1$ nodes from the second node to the final constraint manifold.

In the multiple optimization approach, the differential equations for Φ and Ψ are integrated back to each node using the optimized control vector from that node. Only the gain matrices at nodes along the suboptimal trajectory are stored. In the case of node 1, the suboptimal control vector is used to integrate Φ and Ψ back from the final constraint manifold to node 1. For a subsequent node, the optimal control vector along the subtrajectory is used to integrate Φ and Ψ back to the node. This process is repeated for the first $r - m + 1$ nodes along the suboptimal trajectory.

The gain matrix at each node is reduced to a $1 \times n$ vector by eliminat-
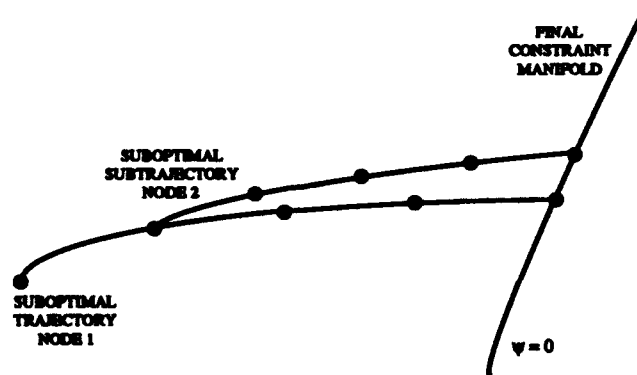
Figure 3.1: Optimal trajectory/subtrajectory

ing gains that do not affect the control along the suboptimal trajectory. Since the neighboring extremal path is based on the original suboptimal trajectory, only those gains that affect the control along the suboptimal trajectory need to be retained. Only the first row of the gain matrix at each node relates to the control along the trajectory.

The hypothesis behind optimizing from each node point is that the resulting gains are based on the most optimal r node trajectory from any node along the suboptimal trajectory. In the single optimization method the gain matrices are based on the original suboptimal trajectory from node 1, and the control vector updates are also along this trajectory. Since the multiple optimization approach bases its gains on the optimal r node subtrajectory from each node point, the gain vector should produce a control update that is more aligned with the most optimum path to the final constraint manifold but still in the neighborhood of the extremal.

Chapter 4

Lunar Launch Problem

4.1 Optimal Control Problem

The test problem for this research involves injecting a vehicle into lunar orbit in minimal time (refer to Fig. 4.1). The nominal solution to this free final time problem produces a control history (thrust angle versus time) that minimizes the performance index

$$J = t_f. \tag{4.1}$$

The differential equations governing the trajectory are

$$\dot{y} \;=\; v \tag{4.2}$$

$$\dot{u} \;=\; \alpha \cos\theta \tag{4.3}$$

$$\dot{v} \;=\; \alpha \sin\theta - g \;, \tag{4.4}$$

where α is the vehicle acceleration and g is the acceleration of gravity. The differential equations for the downrange component are not included since they are not constrained. The initial conditions are given by

$$t_0 = 0, \; y_0 = 0, \; u_0 = 0, \; v_0 = 0 \;, \tag{4.5}$$

and the final conditions by

$$y_f \;=\; 50,000 \; ft \tag{4.6}$$

14
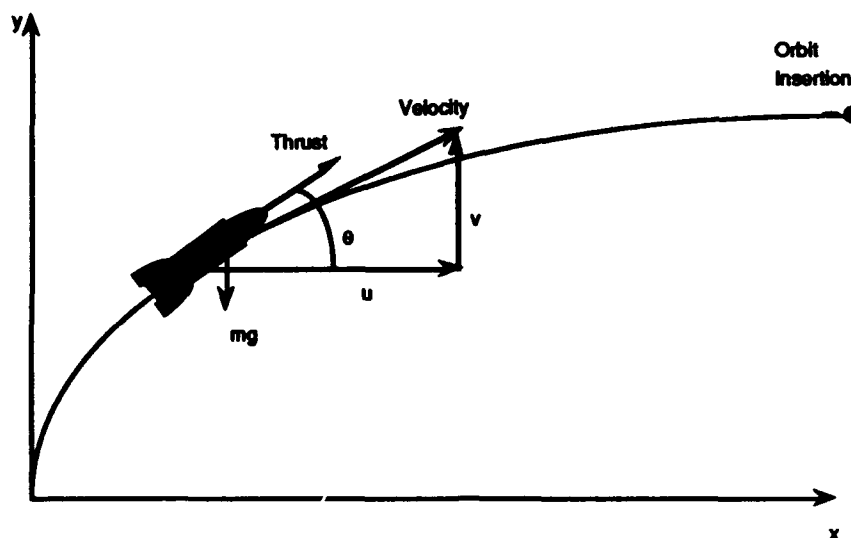
Figure 4.1: The lunar launch problem

$$u_f = 5,444 \quad ft/sec \tag{4.7}$$

$$v_f = 0 \quad ft/sec. \tag{4.8}$$

These final conditions are the desired final states to achieve lunar orbit. The quantities u and v denote the horizontal and vertical components of the velocity, respectively. The final boundary constraints are combined into a vector, ψ, where

$$\psi = \begin{bmatrix} \frac{y_f}{50000} - 1 \\ \frac{u_f}{5444} - 1 \\ \frac{v_f}{5444} \end{bmatrix} = 0. \tag{4.9}$$

For simplicity several assumptions are made. First, the thrust vector and all motion is in the x–y plane. Second, it is assumed that the moon is flat over the course of the trajectory so the lunar gravity vector, $\bar{g}$, remains constant with $|\bar{g}| = 5.32$ ft/sec^2. Finally, the ratio of vehicle mass and thrust is assumed constant throughout the trajectory so thrust acceleration, α, is constant. Based on previous research into the lunar launch problem [1], a

nominal value of $\alpha = 20.8$ ft/sec^2 is assumed.

In its current form as a free final time problem, the upper integration limit of the equations of motion is not fixed. By selecting a different independent variable, the integration limit can be fixed, and the problem handled more conveniently.

For the lunar launch problem, the control angle, θ, stays between 0 and 90 degrees during the ascent. As a result, u is monotonic and it is used as the variable of integration.

With u as the variable of integration, several benefits are realized. Since u_f is the integration limit, the vehicle is certain to meet the required horizontal velocity to achieve orbit. In addition, incorporating u into the limits of integration reduces the number of boundary constraints since u_f will, by definition, be satisfied.

The state vector is rewritten as

$$x^T = \begin{bmatrix} t & y & v \end{bmatrix}. \tag{4.10}$$

To normalize by u_f the variable $\bar{u}$ is introduced where

$$\bar{u} = \frac{u - u_0}{u_f - u_0}. \tag{4.11}$$

For the lunar launch problem, $u_f = 5444$ ft/sec and u_0 is the initial horizontal velocity of the vehicle. For simulations and optimization of the nominal trajectory from $\bar{u} = 0$, u_0 equals zero. For the single trajectory optimization $u_0 = 0$ during gain calculations, while in the multiple optimization scheme, u_0 equals the horizontal velocity at each control node.

The normalized differential equations are

$$x' = \frac{dx}{d\bar{u}} = \frac{dx}{dt}\frac{dt}{du}\frac{du}{d\bar{u}} = g(x, \bar{u}, a). \tag{4.12}$$

Therefore, the equations of motion are rewritten as

$$t' = \frac{(u_f - u_0)}{\alpha \cos\theta} \tag{4.13}$$

$$y' = v\frac{(u_f - u_0)}{\alpha \cos\theta} \tag{4.14}$$

$$v' = \frac{(\alpha \sin\theta - g)(u_f - u_0)}{\alpha \cos\theta}, \tag{4.15}$$

where

$$x_0^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \tag{4.16}$$

$$y_f = 50000, \tag{4.17}$$

$$v_f = 0 \tag{4.18}$$

and the limits of integration are

$$\bar{u}_0 = 0 \ , \quad \bar{u}_f = 1. \tag{4.19}$$

The suboptimal control vector is computed using the nonlinear programming code VF02AD. VF02AD is an extension of a variable metric method for unconstrained optimization applied to the constrained case. It searches for an optimal solution until reaching a preset convergence criteria. Convergence occurs when the change in the performance index between iterations is less than the preset criteria. For all optimizations a convergence criteria of 10^{-9} is used.

For the single trajectory optimization, only the suboptimal control from $u_0 = 0$ to $u_f = 5444$ ft/sec is computed. The suboptimal control consists of nine nodes with an interval of 680.5 ft/sec between node points. Nine nodes are selected for convenience, resulting in control nodes every $0.125\bar{u}$.

For the multiple optimization method, the nine-node suboptimal trajectory from $u = 0$ is computed and used as a basis for the nominal, unperturbed control. Then the nine-node suboptimal subtrajectory from each node on the suboptimal trajectory to the final constraint manifold is calculated. The control vector for each subtrajectory gives the suboptimal nine-node control from $u_0 = 680.5(N - 1)$ ft/sec to $u_f = 5444$ ft/sec, where N is a node on the suboptimal trajectory. This results in eight suboptimal control vectors for the multiple optimization approach.

4.2   Computation of the Neighboring Extremal Gains

To compute the neighboring extremal gains, expressions for G_{aa}, ψ_{x_f}, Ψ, and Φ are found.

The derivative, G_{aa}, is calculated numerically using central differences. G_{aa} represents the second derivative effect of a change in the parametrized control vector on the performance index, G. Each element of a is perturbed and the effect on the performance index is computed.

ψ_{x_f} is the partial derivative of the constraints with respect to the state vector. ψ_{x_f} is a constant matrix with

$$\psi_{x_f} = \begin{bmatrix} 0 & \frac{1}{50000} & 0 \\ 0 & 0 & \frac{1}{5444} \end{bmatrix} . \tag{4.20}$$

The differential equations for Φ and Ψ are integrated backwards from $\bar{u}_f$. The differential equation governing Φ is

$$\frac{d\Phi}{d\bar{u}} = \Phi' = g_x \Phi, \tag{4.21}$$

where

$$g_x = \frac{\partial g}{\partial x} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{u_f - u_0}{\alpha \cos\theta} \\ 0 & 0 & 0 \end{bmatrix} \tag{4.22}$$

and

$$\Phi_f = I \tag{4.23}$$

The matrix differential equation for Ψ is

$$\Psi' = g_x \Psi + g_a \tag{4.24}$$

where

$$\Psi_f = 0. \tag{4.25}$$

In this problem, Ψ is a 3×9 matrix and g_a is the partial derivative of the performance index with respect to the parameter vector a. Since $a^T = [\theta_1, \cdots, \theta_r]$, g_a can be written as

$$\frac{(u_f - u_0)\sin\theta}{\alpha \cos^2\theta} \begin{bmatrix} \frac{\partial\theta}{\partial\theta_1} & \frac{\partial\theta}{\partial\theta_2} & \cdots & \frac{\partial\theta}{\partial\theta_9} \\ v\frac{\partial\theta}{\partial\theta_1} & v\frac{\partial\theta}{\partial\theta_2} & \cdots & v\frac{\partial\theta}{\partial\theta_9} \\ \frac{(\alpha - g\sin\theta)}{\sin\theta}\frac{\partial\theta}{\partial\theta_1} & \frac{(\alpha - g\sin\theta)}{\sin\theta}\frac{\partial\theta}{\partial\theta_2} & \cdots & \frac{(\alpha - g\sin\theta)}{\sin\theta}\frac{\partial\theta}{\partial\theta_9} \end{bmatrix}. \tag{4.26}$$

The control, θ, is linearly interpolated between adjacent nodes, so that

$$\theta(\bar{u}) = \theta_{i-1} + \frac{\theta_i - \theta_{i-1}}{\bar{u}_i - \bar{u}_{i-1}}(\bar{u} - \bar{u}_{i-1}), \quad \text{for } \bar{u}_{i-1} \le \bar{u} \le \bar{u}_i, \tag{4.27}$$

and

$$\theta(\bar{u}) = \theta_i + \frac{\theta_{i+1} - \theta_i}{\bar{u}_{i+1} - \bar{u}_i}(\bar{u} - \bar{u}_i), \quad \text{for } \bar{u}_i \le \bar{u} \le \bar{u}_{i+1}. \tag{4.28}$$

The partial derivatives with respect to θ_i, respectively, are

$$\frac{\partial \theta}{\partial \theta_i} = \frac{\bar{u} - \bar{u}_{i-1}}{\bar{u}_i - \bar{u}_{i-1}}, \qquad (4.29)$$

and

$$\frac{\partial \theta}{\partial \theta_i} = 1 - \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+1} - \bar{u}_i}. \qquad (4.30)$$

The partial derivative terms determine the impact of a change in the control on the two adjacent control intervals. However, Eqs. (4.29) and (4.30) are both valid only for interior node points. For boundary nodes, these equations break down since at either boundary, $\bar{u}_{i-1}$ or $\bar{u}_{i+1}$ does not exist. Therefore, over the first interval only Eq. (4.30) is valid while over the last interval only Eq. (4.29) is used. Elsewhere, the partial derivative is zero.

With the differential equations for Φ and Ψ, and the general gain equation (Eq. 3.23), the gain matrices for the single and multiple optimization trajectory are determined. Because the ψ_{x_f} matrix has zeroes in the first column, the gain matrix, K, also has zeroes in the first column. These zero gains are not stored in either the single or multiple trajectory optimization.

Intuitively, the zero value of these gains makes sense. The gains in this column relate deviations in t to control changes. As the final value of t is not constrained, its variation from the optimal trajectory does not influence the control. Conversely, a deviation from the nominal trajectory for y and/or v will have an impact on the desired final conditions so a control change is necessary.

For both methods eight gain matrices are computed from $\bar{u} = 0$ up to, and including, $\bar{u} = 0.875$. At $\bar{u} = 0.875$ there are two nodes and two boundary

constraints. At the last node ($\bar{u} = 1$) there are not enough nodes to satisfy the boundary conditions. Also, since the last node is at the final constraint manifold, it would take an infinite control to correct for perturbations.

4.3   Simulation

To test both gain calculation methods, simulations of the lunar launch problem are made. Each simulation uses a fourth-order Runge-Kutta integrator to simultaneously integrate the differential equations of motion for a perturbed and unperturbed trajectory. The simulation integrates the equations of motion using 40 steps for a step size of $\Delta\bar{u} = 0.025$. The perturbed trajectory has either lunar gravity or thrust acceleration perturbed $\pm 5\%$ from their nominal values. The perturbed and unperturbed trajectories are sampled at each integration step to determine the error vector, δx. Simulations are done closed-loop using a sample and hold technique for the control update. On each integration step the nominal control vector is linearly interpolated while the control update is held constant. To provide a basis for comparison, another simulation using open-loop control is also made. The open-loop method uses the optimal control in the presence of the perturbation with no correction to the nominal control vector.

Chapter 5

Results

5.1 Single Trajectory Optimization and Simulation

For the single trajectory optimization the nonlinear programming code VF02AD is used to calculate the nine node suboptimal control starting at $\bar{u} = 0$ up to, and including, $\bar{u} = 1$. VF02AD returns the following results:

$$
\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \end{bmatrix} = \begin{bmatrix} 0.4539471496 \\ 0.4068478813 \\ 0.3580472572 \\ 0.3079740487 \\ 0.2592985250 \\ 0.2077640929 \\ 0.1567470292 \\ 0.1048252859 \\ 0.0528867033 \end{bmatrix} \text{rad.} = \begin{bmatrix} 26.00925579 \\ 23.31066650 \\ 20.51459670 \\ 17.64561319 \\ 14.85671112 \\ 11.90400566 \\ 8.98094322 \\ 6.00604647 \\ 3.03018489 \end{bmatrix} \text{deg.} \qquad (5.1)
$$

With this control history, the vehicle reaches the following final conditions at $u_f = 5444$ ft/sec on the unperturbed, nominal trajectory:

$$
t_f = 272.7061 \quad \text{sec} \qquad (5.2)
$$

$$
y_f = 50000.0005 \quad \text{ft} \qquad (5.3)
$$

$$
v_f = 1.3987 \times 10^{-6} \quad \text{ft/sec} . \qquad (5.4)
$$

The gain program uses the suboptimal control history and the gain equation to compute the optimal control gains. The gain vectors to three significant figures are shown in Table 5.1.

NODE	$\bar{u}$	NODE GAINS	
		y	v
1	0.000	-0.369E-5	-0.673E-3
2	0.125	-0.289E-5	-0.462E-3
3	0.250	-0.385E-5	-0.521E-3
4	0.375	-0.573E-5	-0.640E-2
5	0.500	-0.940E-5	-0.831E-2
6	0.625	-0.179E-4	-0.118E-2
7	0.750	-0.461E-4	-0.201E-2
8	0.875	-0.267E-3	-0.581E-2

Table 5.1: Single trajectory optimal control gains for nodes 1 through 8

To check the accuracy of the gains, a numerical derivative method is used. The original gain matrix at each node relates the error in the state (difference between the perturbed and nominal trajectory) with the change in the control at each node along the trajectory, required to keep the perturbed trajectory in the neighborhood of the extremal. This relationship can be expressed as

$$\delta a = K \delta x. \tag{5.5}$$

As a result, K can be approximated as

$$K \approx \frac{\partial a}{\partial x}. \tag{5.6}$$

Table 5.2 shows a comparison of the gain matrix at node 1 (before eliminating extraneous gains) computed with the gain equation and numerical partials.

Most of the differences in the values of the numerical partials can be attributed to accuracy and round off errors in VF02AD. To compute the derivatives, VF02AD calculates the optimal control based on a perturbation in the state vector. The state pertrubation is 10^{-4} while the convergence accuracy

GAIN MATRIX FOR $\bar{u} = 0$			
GAIN EQUATION		NUMERICAL PARTIALS	
-0.369E-5	-0.673E-3	-0.353E-5	-0.676E-3
-0.233E-5	-0.559E-3	-0.283E-5	-0.558E-3
-0.192E-5	-0.439E-3	-0.193E-5	-0.439E-3
-0.980E-6	-0.313E-3	-0.999E-6	-0.313E-3
-0.265E-8	-0.183E-3	-0.117E-6	-0.185E-3
0.100E-5	-0.480E-4	0.110E-5	-0.446E-4
0.203E-5	0.898E-4	0.200E-5	0.904E-4
0.306E-5	0.230E-3	0.252E-5	0.228E-3
0.410E-5	0.370E-3	0.511E-5	0.369E-3

Table 5.2: Numerical partial approximation of control gains for node 1

for VF02AD is 10^{-9}. This produces significant figures on the order of 10^{-5}. Therefore, for extremely small gains (i.e. -.265E-8) the numerical partials do not agree favorably with the calculated result.

Tables 5.3 and 5.4 show the simulation results using the single optimization gains. Each table lists the error in the final values for y and v from the desired final conditions. To provide an additional basis for comparison, the open-loop results have also been included.

Simulations using single optimization gains show a marked improvement in the final conditions over the open-loop results. Positional errors due to acceleration perturbations are better handled than errors in gravity, however. Velocity errors at u_f are approximately the same regardless of whether the perturbation is due to gravity or thrust accelerations.

Figs. 5.1 and 5.2 depict the history of the position and velocity error as a function of the variable of integration, $\bar{u}$. When viewing the graphs it

α $\left(\frac{ft}{sec^2}\right)$	% Change in α	State	Deviation from Optimal	
			Closed Loop	Open Loop
19.760	-5.0	y	10.530	-8328.003
		v	-2.821	-76.358
21.84	+5.0	y	-17.592	6590.498
		v	+0.811	2.987

Table 5.3: Single optimization method for ±5% modeling error in thrust acceleration
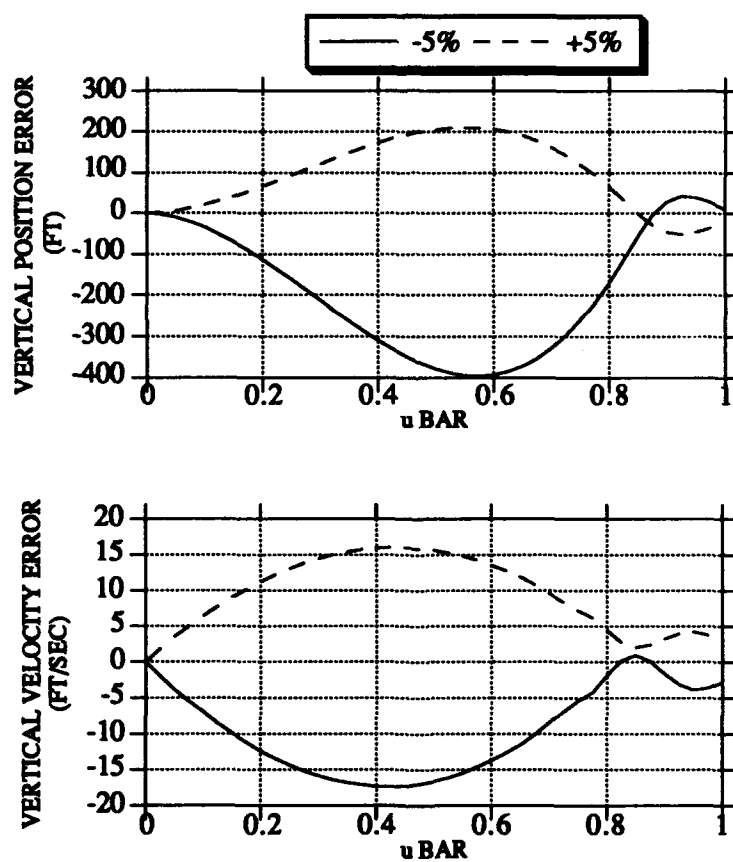
Figure 5.1: Single optimization, acceleration perturbation

g $\left(\frac{ft}{sec^2}\right)$	% Change in g	State	Deviation from Optimal	
			Closed Loop	Open Loop
5.054	-5.0	y	65.178	9891.024
		v	-2.705	72.540
5.586	+5.0	y	-63.989	-9891.023
		v	2.616	-72.540

Table 5.4: Single optimization method for ±5% modeling error in lunar gravity

Figure 5.2: Single optimization, gravity perturbation

must be remembered that the objective of the guidance law is to operate in the neighborhood of the extremal path. As a result the position and velocity errors will not normally converge to zero until later in the trajectory.
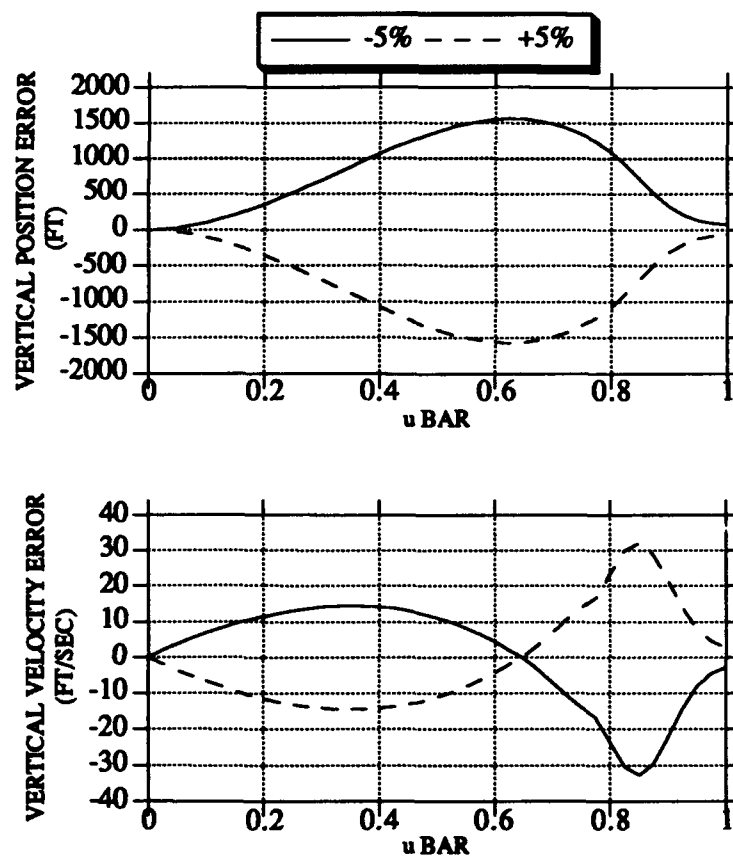
5.2 Multiple Trajectory Optimization and Simulation

The multiple trajectory optimization method also uses a nine-node optimal trajectory as the basis for the nominal control. A nine node optimization is then computed from each node on the suboptimal trajectory to the final constraint manifold. Table 5.5 shows the resulting optimal control vectors from each node. As expected, the optimal control from node 1 is the same as the single trajectory optimization.

Table 5.6 lists the optimal control gains at each node. Compared with the single optimization gains, the multiple optimization are up to twice as large as gains produced under the single optimization method. Gains at either ends of the trajectory are approximately equal.

To verify the accuracy of the gains, a numerical partial approximation is made similar to that discussed in the previous section. Table 5.7 shows that the numerical partials agree extremely well the gains computed using the gain equation.

Tables 5.8 and 5.9 and Figs. 5.3 and 5.4 present the results of the simulation for the multiple optimization case.

On the whole, comparisons of the multiple and single optimization methods produced mixed results. For acceleration perturbations, single optimization gains gave slightly better results than multiple optimization methods. On the other hand, end point conditions for the multiple optimization approach in the face of gravity perturbations were, relatively speaking, much better than single optimization results.

Figs. 5.3 and 5.4 depict the history of the error of y and v from

NODE 1 $\bar{u} = 0$	NODE 2 $\bar{u} = 0.125$	NODE 3 $\bar{u} = 0.250$	NODE 4 $\bar{u} = 0.375$
0.45394715	0.40634061	0.35766840	0.30858725
0.40684788	0.36445945	0.32136083	0.27792256
0.35804726	0.32106646	0.28359996	0.24597137
0.30797405	0.27670197	0.24516890	0.21360186
0.25929852	0.23386572	0.20826123	0.18270982
0.20776409	0.18867334	0.16951886	0.15027212
0.15674703	0.14389432	0.13098104	0.11795366
0.10482529	0.09830906	0.09183870	0.08527328
0.05288670	0.05291161	0.05305770	0.05300320
NODE 5 $\bar{u} = 0.500$	**NODE 6** $\bar{u} = 0.625$	**NODE 7** $\bar{u} = 0.750$	**NODE 8** $\bar{u} = 0.875$
0.25871156	0.20780274	0.15648354	0.10482639
0.23386278	0.18893294	0.14341997	0.09789380
0.20798410	0.16950645	0.13168537	0.09302267
0.18175316	0.14960138	0.11570712	0.08265920
0.15684631	0.13073685	0.10902890	0.08325916
0.13100243	0.11166393	0.08838203	0.06894193
0.10496646	0.09213030	0.08020577	0.06716968
0.07866887	0.07212200	0.06518209	0.05876025
0.05283867	0.05292930	0.05343331	0.05345789

Table 5.5: Optimal control vectors for nodes 1 through 8 (radians)

NODE	$\bar{u}$	NODE GAINS	
		y	v
1	0.000	-0.369E-5	-0.673E-3
2	0.125	-0.494E-5	-0.780E-3
3	0.250	-0.688E-5	-0.921E-3
4	0.375	-0.101E-4	-0.112E-2
5	0.500	-0.161E-4	-0.141E-2
6	0.625	-0.290E-4	-0.190E-2
7	0.750	-0.661E-4	-0.288E-2
8	0.875	-0.267E-3	-0.580E-2

Table 5.6: Multiple optimization control gains

GAIN EQUATION		NUMERICAL PARTIALS	
-0.369E-5	-0.673E-3	-.353E-5	-.675E-3
-0.494E-5	-0.780E-3	-.495E-5	-.781E-3
-0.688E-5	-0.921E-3	-.688E-5	-.922E-3
-0.101E-4	-0.112E-2	-.101E-4	-.112E-2
-0.161E-4	-0.141E-2	-.160E-4	-.141E-2
-0.290E-4	-0.190E-2	-.290E-4	-.190E-2
-0.661E-4	-0.288E-2	-.656E-4	-.286E-2
-0.267E-3	-0.580E-2	-.267E-3	-.577E-2

Table 5.7: Numerical partial approximation of control gains

nominal as a function of u for perturbations due to thrust acceleration and lunar gravity. Both figures show that the multiple optimization gains provide good control of the error and an almost asymptotic approach toward zero error at u_f.

The thrust acceleration perturbation graph (Fig. 5.3) illustrates an apparent inconsistency due to the change of the variable of integration. In the graphs, an error in velocity does not change sign while the position error oscillates once around the zero error axis. The reason for this has to do with normalizing the equations of motion with u_f instead of t_f.

Changing the variable of integration to u means the position vector becomes a function of u instead of t. Therefore, to compute an error vector, the state vector of the perturbed trajectory at some u_p is compared with the state vector of the nominal trajectory at the same u_p. These two points along the trajectory, however, do not occur at the same point in time. As a result, while an element of the perturbed state vector, for example y_p, may be greater than the nominal trajectory at u_p, y_p may still be lower than it should be for

α $\left(\frac{ft}{sec^2}\right)$	% Change in α	State	Deviation from Optimal	
			Closed Loop	Open Loop
19.760	-5.0	y	13.757	-8328.003
		v	-3.630	-76.358
21.84	+5.0	y	-27.812	6590.498
		v	4.300	69.086

Table 5.8: Multiple optimization method for ±5% modeling error in thrust acceleration.
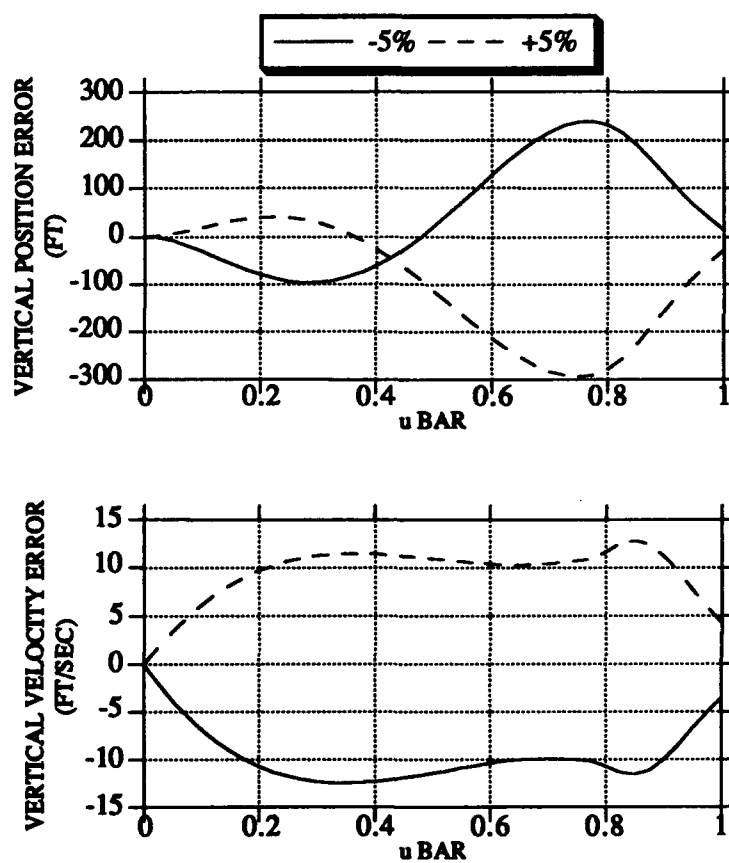

Figure 5.3: Multiple optimization, thrust acceleration perturbation.

g $\left(\frac{ft}{sec^2}\right)$	% Change in g	State	Deviation from Optimal	
			Closed Loop	Open Loop
5.054	-5.0	y	47.558	9891.024
		v	-1.767	72.540
5.586	+5.0	y	-47.295	-9891.023
		v	1.741	-72.540

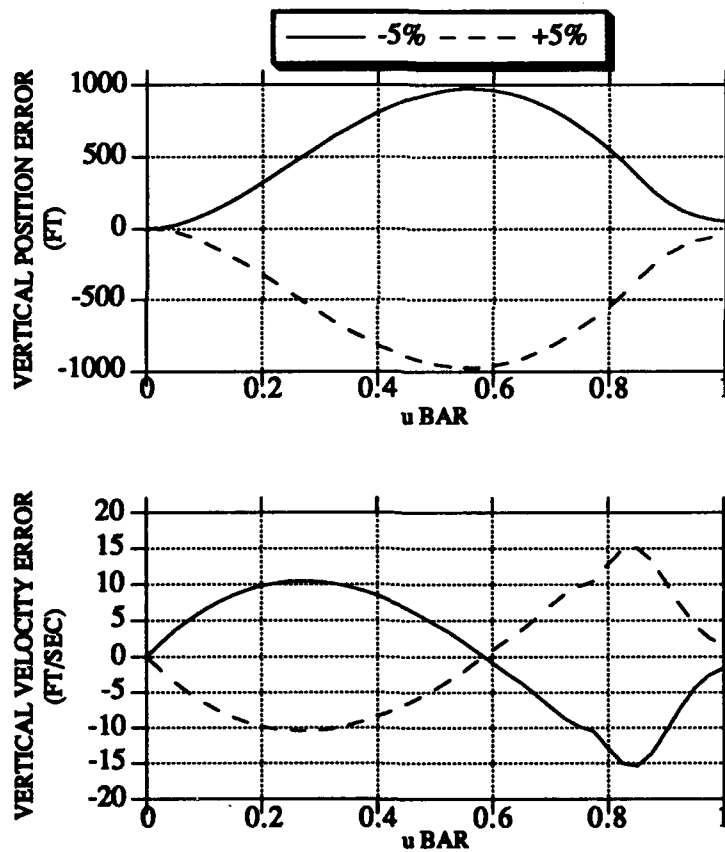Table 5.9: Multiple optimization method for ±5% modeling error in lunar gravity.

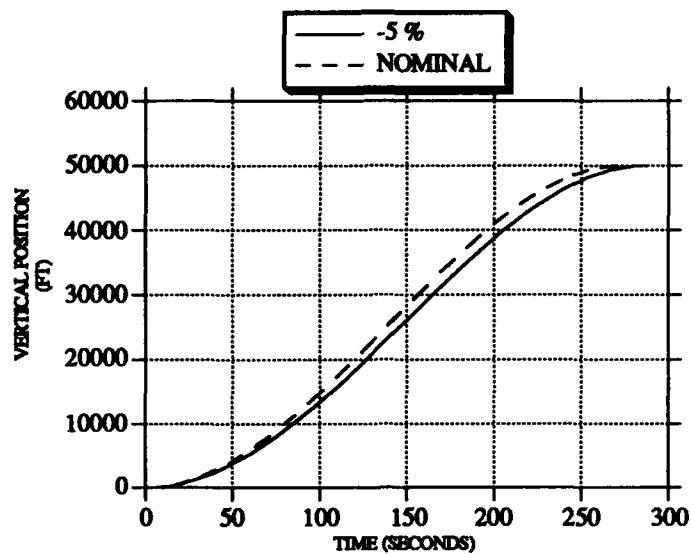Figure 5.4: Multiple optimization, gravity perturbation.

Figure 5.5: Altitude versus time

the same point in time. Fig. 5.5 shows the plot of the y element as a function of time for a -5% error in thrust acceleration. As can be seen, the -5% perturbed trajectory does not exceed the nominal trajectory.

Chapter 6

Conclusions

The purpose of this research is to develop a more optimal method of computing the control gains at each node through multiple optimization. Multiple optimization reoptimizes the trajectory from each control node on the suboptimal trajectory. Each subtrajectory from a node to the final constraint manifold has the same number of control nodes, r, as the original suboptimal trajectory and represents the optimal r-node solution. The suboptimal control vector for each subtrajectory is used to compute the control gains. These gains are more aligned with the most optimal path from a node to the final constraint manifold.

To test the control law, simulations are made using a representative optimization problem, the lunar launch problem. The lunar launch problem involves placing a vehicle into lunar orbit in minimum time subject to constraints on the final states. The simulation integrates the equations of motion for both the perturbed and unperturbed (nominal) trajectory. The value of each state vector at each integration step is sampled and used to determine the error vector. Multiplying the error vector by the control gains gives the control update to be applied to the perturbed trajectory.

Neighboring extremal guidance using gains calculated by multiple optimization methods produce results comparable to the single trajectory opti-

mization method. For perturbations of ±5% in either gravity or thrust acceleration, endpoint conditions using multiple optimization gains never exceed more than 48 feet in position and 4.3 ft/sec in velocity from the desired final conditions. Single optimization gains give nearly the same results with endpoint conditions never varying more than 65 feet and 3 ft/sec from the desired final conditions. Comparisons of each method for a specific model perturbations give mixed results, with multiple optimization delivering smaller errors in the final conditions for perturbations due to gravity and single trajectory optimization performing better for thrust acceleration errors.

These observations are made cognizant that the sample problem in this research does not define an acceptable error for the final conditions. Previous research [1] has used the criterion that the percentage error in the final states should not exceed the percentage error in the perturbation. Unfortunately, for errors in y_f this criterion seems rather liberal and is easily met by both the single and multiple optimization method. Conversely, a 5% error in v_f is zero, a little too restrictive.

One area for further research is to explore the impact of increasing the number of control nodes along the trajectory. Increasing the number of control nodes and, hence, the number of gains decreases the the gap between the point at which the last gain matrix exists, and the final constraint manifold. This approach should improve guidance near the end of the trajectory and subsequently decrease the error at the final constraint manifold.

Bibliography

[1] Hull, D. G. and Helfrich, C. E., "Neighboring Extremal Guidance For Systems With A Piecewise Linear Control." AIAA Third Aerospace Planes Conference, Orlando, Fl. Paper AIAA-91-5067

[2] Bryson, A. E. and Ho, Y. C., *Applied Optimal Control.* Hemisphere Publishing Corp., Washington D. C., 1975.

[3] Atkinson, L. V. et al., *Numerical Methods with FORTRAN 77.* Addison-Wesley Publishing Co., Wokingham, England, 1989

[4] Hull, D. G.,"Optimal Control Theory", unpublished notes, Department of Aerospace Engineering and Engineering Mechanics, University of Texas, Austin, Texas

VITA

Michael James Nowak was born August 2, 1957 in Hampton, Virginia, the son of Donald and Mary Nowak. He graduated from Rome Catholic High School in 1975. Mr. Nowak graduated with a degree in Astronautical Engineering and received his commission as a second lieutenant from the United States Air Force Academy in 1979. Following graduation Mr. Nowak spent five months as a research assistant at the Air Force Academy working on the Global Positioning Satellite. For the next eleven years he was an Air Force pilot and instructor in training and tactical reconnaissance aircraft. He began his masters program at the University of Texas in June 1991.

Permanent address: 409 RIDGEWOOD DR.
                    ROME NEW YORK 13440

This thesis was typeset[1] with LaTeX by the author.

---

[1] LaTeX document preparation system was developed by Leslie Lamport as a special version of Donald Knuth's TeX program for computer typesetting. TeX is a trademark of the American Mathematical Society. The LaTeX macro package for The University of Texas at Austin thesis format was written by Khe-Sing The.