Six Dimensional Trajectory Solver for Automomous Proximity Operations

by

Ronald J. Simmons

B.S. Aero Eng., United States Air Force Academy, (1988)
B.S. Astro Eng., United States Air Force Academy, (1988)

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

MASTER OF SCIENCE
in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1990

Signature of Author ______________________________
Department of Aeronautics and Astronautics
May 1990

Certified by ______________________________
Professor Walter M. Hollister, Thesis Advisor
Department of Aeronautics and Astronautics

Approved by ______________________________
Edward V. Bergmann, Technical Supervisor
Charles Stark Draper Laboratory, Inc.

Accepted by ______________________________
Professor Harold Y. Wachman, Chairman
Department Graduate Committee

Six Dimensional Trajectory Solver for Autonomous Proximity Operations

by

Ronald J. Simmons

Submitted to the Department of Aeronautics and Astronautics
on May 11, 1990 in partial fulfillment of the requirements
for the degree of Master of Science.

Abstract

Many future space missions will require that proximity operations be accomplished reliably, precisely and efficiently without a pilot in the loop. This thesis examines one element of an autonomous proximity operations controller, the trajectory planner. The trajectory planner uses a modified gradient search to find a locally optimal trajectory from the initial state to the target state that does not violate any of the mission constraints.

The mission constraints are defined as the maximum time of flight for the operation, the maximum closing velocity allowed, and the obstacles in close proximity to the chase and target craft. The obstacles in this space are not assumed to be stationary; therefore, the planner must be able to develop a solution that, although not guaranteed globally optimal, meets all mission constraints in real time. This will enable the autonomous controller to avoid obstacles moving rapidly with respect to it and to correct for failed actuators.

The Clohessy-Wiltshire equations for relative position and quaternions for relative attitude are used to define a state space relationship between the initial state and the final state as a function of time. The trajectory solver then uses these equations to find the minimum fuel solution to the problem of maneuvering to a target position and attitude while evading moving obstacles. Example results and simulations are included for various initial conditions and maneuvering constraints.

The trajectory planner algorithm can find a trajectory from any initial state to any final state which satisfies the input constraints and uses minimum fuel.

Thesis Supervisor:	Professor Walter M. Hollister
Title:	Professor of Aeronautics and Astronautics, M.I.T.
Technical Supervisor:	Edward V. Bergmann
Title:	Section Chief, Flight Systems Division Charles Stark Draper Laboratory, Inc.

Acknowledgements

I wish to thank my technical supervisor Ed Bergmann for his invaluable assistance in the development of the Trajectory Planner. He gave me latitude to explore new aspects of the problem and then pulled me back to reality when I went too far. I extend thanks to my thesis advisor Prof. Walter Hollister for explaining the motivation behind and usefulness of the C-W equations, without which the Planner would not be possible. Thanks also to Capt Craig Niiya for providing insight into many of the algorithm problems encountered along the way. And thanks to Bruce Persson and Charles Cooke for your infinite patience in linking the Trajectory Planner to the OEX autopilot. To all the other members of the Charles Stark Draper Laboratory who provided technical assistance and support during the last two years; I am grateful for your help.

I would also like to thank the Charles Stark Draper Laboratory for sponsoring this research and for providing a graduate fellowship.

I wish also to thank my friends Bill, Bob, Carol, Carole (both of them), Dave, Dino, Duncan, George, Irene, Jeff, Jesse, Kellie, Kelly, Lance, Moshe, Om, Paul, the Spring Break Crew (Fred, Pete, Ralph), Steve, Susan, T. Montana, Todd, Tom, Trish, Walter, and all the other Draper Meats. Thanks, you made even the bad times good. Thank you Audrey for just being across the hall.

Most importantly, I wish to thank my family for their love, support, and encouragement through the years; I dedicate this thesis to you. (And to my mom who told me "if I don't read it, nobody will" ... good luck.)

This report was prepared at The Charles Stark Draper Laboratory, Inc. under IR&D funding. However, publication of this report does not constitute approval by The Charles Stark Draper Laboratory or the Massachusetts Institute of Technology of the findings or conclusions contained herein. It is published solely for the exchange and stimulation of ideas.

Ronald J. Simmons, 2Lt USAF

Contents

List of Figures

List of Tables

Introduction

Autonomous Docking

Current proximity operations are highly constrained by both vehicle specifications and operating restrictions. Consequently, these operations need to be extensively orchestrated on the ground prior to execution. In spite of this exhaustive planning, the execution often deviates from the planned trajectory due to unmodelled dynamics or failed actuators. As a result, the active craft is forced to return to one of the prechosen standoff points while the operation is replanned on the ground.

Future proximity operations are expected to be considerably more complex in both planning and execution. Docking with the US space station may require maneuvering around other vehicles, solar arrays or appendages of the station itself. Other vehicles could be simultaneously maneuvering near the space station, creating moving obstacles that cannot be accounted for prior to execution. Another highly constrained mission is the Mars sample return mission, in which two unmanned spacecraft will rendezvous in orbit about Mars. This rendezvous suffers from the problem that no man can be in the loop; not only is the mission unmanned, but the communications time between Earth and Mars is at best 4 minutes each way, making ground control impossible.

To accomplish these missions and to improve the reliability of proximity operations for rescue missions, an autonomous system for proximity operations must be developed. This system must be reliable, precise, and fuel efficient.

Current and Proposed Solutions

Knowing that future missions will require spacecraft to exercise a greater level of autonomy in executing proximity operations, a great deal of research has been conducted to this end. Unfortunately, most of the research has been conducted on developing the hardware required for the imaging systems and the docking platforms. Little effort has gone into the development of an intelligent planner. The navigation and guidance portion of the autonomous system is often a simple, fixed trajectory between two known parking orbits. These methods have a standard solution for any contingency that may occur, which is break off the maneuver, return to a prechosen standoff point, and try again. Although these methods may have been acceptable in the past, it is unlikely that they could accomplish a mission in a more constrained maneuver space.

Many people advocate creating a simple algorithm which calculates a series of small closing trajectories along the velocity vector of the target craft. This technique of hopping along the velocity vector has been in use since the early days of Apollo because each of the intermediate burn positions is on the target velocity vector and is therefore stable with respect to the target state. This method suffers from the same limitations that plagued the astronauts who flew earlier missions; the simplistic algorithm does not account well for failed actuators and the maneuver is prone to extensive replanning if a failure does occur. Furthermore, it is by no means fuel optimal and therefore requires larger fuel reserves on the chase craft.

Others suggest that all currently planned unmanned vehicles should be remotely piloted. This may be a valid solution for many missions such as orbital resupply or "space tug" missions which use the Orbital Maneuvering Vehicle, OMV. While these mission might be accomplished without an intelligent planner, the pilot could still benefit from its use. A planner could aid the pilot by suggesting trajectories and associated burn histories as well as warning of impending collisions and suggesting evasive action. Such an aid could significantly increase the probability of success in a highly constrained maneuver space and could make real time replanning possible in the event of failures.

Although progress toward autonomous planners has been slow, their usefulness is difficult to deny. This thesis suggests an autonomous system that eliminates the need for man in the planning or execution loop. The autonomous controller proceeds by defining the maneuver, assessing the maneuver constraints, planning the trajectory to the target state, and monitoring the execution. As stated earlier, the autonomous controller could also be used as a pilot aid on manned missions.

Elements of Planner

The intelligent autonomous controller will consist of three distinct sections: the maneuver manager, the trajectory planner, and the execution manager. All three controller sections will be in communication with each other. (see Figure 1.1)

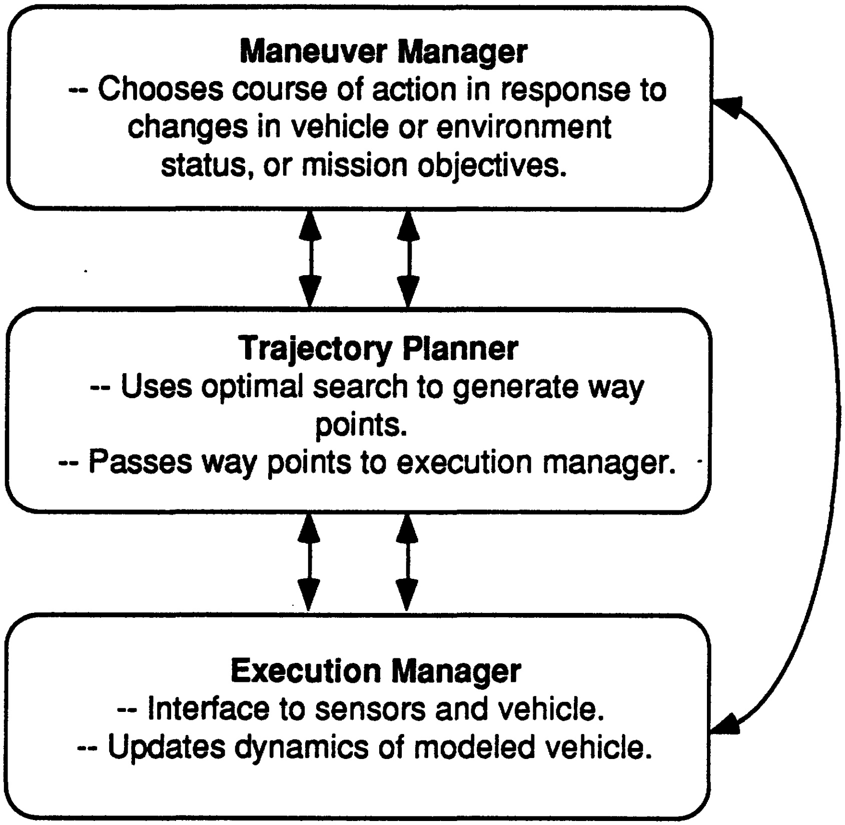

Maneuver Manager
-- Chooses course of action in response to changes in vehicle or environment status, or mission objectives.
Trajectory Planner
-- Uses optimal search to generate way points.
-- Passes way points to execution manager.
Execution Manager
-- Interface to sensors and vehicle.
-- Updates dynamics of modeled vehicle.

Figure 1.1: *Intelligent Autonomous Controller*

The maneuver manager is an intelligent decision maker, or a human pilot, which chooses a course of action based on mission requirements and information provided by the trajectory planner, execution manager and spacecraft sensors. Since the maneuver manager needs to assimilate information and make mission critical decisions such as continuing or aborting the proximity operation, unmanned missions will need to pay special attention to this element. The manager has access to information about mission objectives, vehicle limitations, and the maneuvering environment. If the maneuvering environment changes during the execution of proximity operations, the manager will be made aware of this and will have the option of replanning the maneuver to compensate for the new constraints or terminating the maneuver.

The trajectory planner creates a series of waypoints, intermediate states, between the current state and the target state for the vehicle to follow. This series of waypoints defines the optimal path between the initial and target states subject to maneuvering constraints from the maneuver manager and the execution manager. Although the search algorithm used in the planner is applicable to any spacecraft, the cost function, fuel use, will be vehicle specific. A trajectory planning time of a few minutes is usually insignificant in light of a maneuver duration of several hundreds of minutes; however, conditions such as a jet failure when the vehicle is close to the target state may dictate the need for real time planning. The planner should therefore have the ability to find a solution in real time that, although not optimal, does not violate any of the constraints and allows the vehicle to complete the maneuver.

The final element in the autonomous controller is the execution manager. It is the interface between the planner and the vehicle which implements the plan. Before the maneuver is initiated, the execution manager determines the maneuver environment and the vehicle condition. This information is then sent to both the maneuver manager and the trajectory planner. Vehicle malfunctions, such as inoperative thrusters, are relayed to the trajectory planner as constraints in the maneuver environment; states requiring these thrusters simply cannot be achieved. During the maneuver it uses a model of the vehicle's actuators to determine the firings necessary to achieve the way points and ultimately the goal state. If the maneuver environment changes or a failure occurs during the maneuver, this information is made available for replanning. One of the most important roles of the execution manager is that it allows the controller to correct for dynamics that may have not been properly modelled in the trajectory planner. If these unmodelled dynamics or incorrect implementation of desired rate changes cause the vehicle to stray from the planned path, the execution manager will sense this and can correct the vehicle path.

This thesis will discuss the development of one of these three sections, the trajectory planner. Since only the cost function is vehicle specific, the discussion of the search algorithm will be appropriate for the trajectory planner of any craft. Example trajectories are included for various initial conditions, obstacle sets, and terminal conditions. These trajectories, however, are executed on the Space Shuttle OEX autopilot [2] and are therefore specific to the Space Shuttle. The other two elements of this controller are topics of continuing research at the Draper Laboratory.

Equations of Motion

Reference Frames

The target reference coordinate system is centered on the desired position of the satellite, target position. The **x** axis is measured along the radius vector from the center of mass of the attracting body to the satellite, with the positive direction away from the attracting body. The **y** direction is measured along the velocity vector of the target ,with the positive direction in the direction of positive velocity. For this coordinate system to be orthogonal the orbit must be circular; this makes the velocity vector normal to the radius vector for all time. The final axis **z**, is normal to the **x**, **y** plane completing the right handed coordinate system. (see Figure 2.1) This reference frame rotates in time about its **z** axis at the orbital rate of its center so that the **x** axis is always aligned with the radius vector.

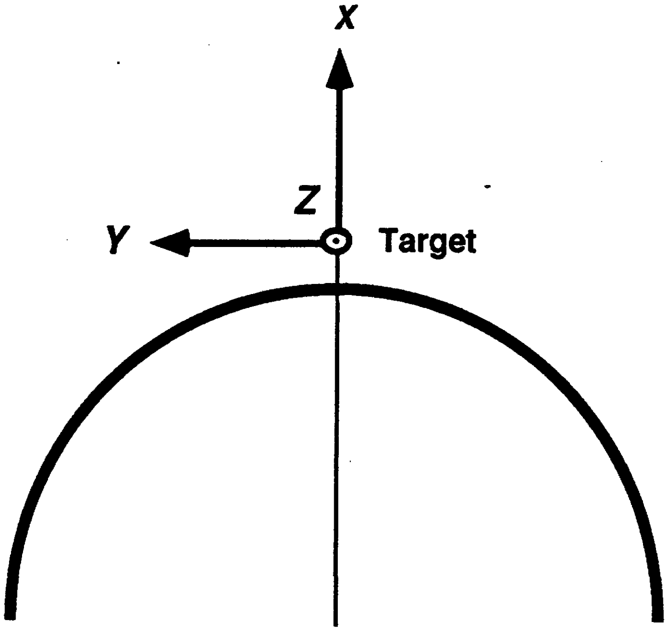


Figure 2.1: *Target Reference Frame*

The body reference coordinate system is centered on the center of mass of the chase vehicle. For the Shuttle example presented in this paper, the axes are standard aircraft axes. The positive **x** axis is located from the center of mass through the nose of the craft. The positive **y** axis is located from the center of mass through the right wing of the craft. The positive **z** axis rounds out the right handed coordinate system by pointing through the "bottom" of the craft. (see Figure 2.2) Since both the body and target reference frames,LVLH, are rotating in time, current position and orientation of the body is referenced to the current LVLH reference frame.

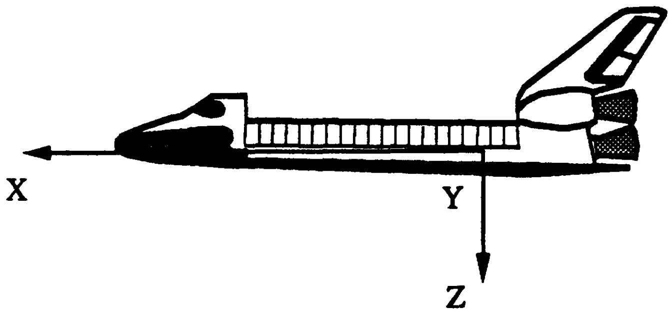


Figure 2.2: *Body Reference Frame*

Derivation of Hills Equations [5]

From Newton's second law, the acceleration of the chasing craft can be expressed as the sum of the forces acting on the craft per unit mass,

$$\ddot{\mathbf{R}} = \frac{1}{M}\sum \textit{forces on the chase craft} \qquad (2.1)$$
$$\text{where:} \quad \mathbf{R} = X\hat{i} + Y\hat{j} + Z\hat{k}$$

assuming that the velocities are non relativistic. **R** is expressed here in inertial coordinates of the central force body (capital X, Y, Z will be used for inertial coordinates while lower case x, y, z will be used for LVLH coordinates). Furthermore, it is assumed that the external forces are small compared to the central force; this keeps the orbit nearly circular. Equation 2.1 can be rewritten in the form,

$$\ddot{\mathbf{R}} = \mathbf{F} - \frac{GM}{R^3}\mathbf{R} \qquad (2.2)$$
$$\text{where:} \quad \mathbf{F} = \mathbf{F_X} + \mathbf{F_Y} + \mathbf{F_Z}$$

where **F** is any non gravitational force, including control inputs, per unit mass acting on the chase craft. The second term in Equation 2.2 is the gravitational acceleration on the chase craft where M is the mass of the central body and G is the universal gravitational constant. This equation can be further separated into three components,

$$\ddot{\mathbf{R}} = \left(\mathbf{F_X} - \frac{GM}{R^3} X\right)\hat{\mathbf{i}} + \left(\mathbf{F_Y} - \frac{GM}{R^3} Y\right)\hat{\mathbf{j}} + \left(\mathbf{F_Z} - \frac{GM}{R^3} Z\right)\hat{\mathbf{k}} \qquad \mathbf{(2.3)}$$

Another expression for $\ddot{\mathbf{R}}$ may be obtained by differentiating the **R** vector twice,

$$\ddot{\mathbf{R}} \;=\; \ddot{\rho}\,\mathbf{u}_R + \left(2\mathbf{n} \times \dot{\rho}\,\mathbf{u}_R\right) + \langle \dot{\mathbf{n}} \times \mathbf{R} \rangle + \mathbf{n} \times \langle \mathbf{n} \times \mathbf{R} \rangle \qquad \mathbf{(2.4)}$$

Expanding and collecting components gives,

$$\ddot{\mathbf{R}} \;=\; \left(\ddot{X} - 2n\dot{Y} - n^2X\right)\hat{\mathbf{i}} + \left(\ddot{Y} + 2n\dot{X} - n^2Y\right)\hat{\mathbf{j}} + \ddot{Z}\,\hat{\mathbf{k}} \qquad \mathbf{(2.5)}$$

where **n** is the constant orbital rate, $\mathbf{u}_R$ is a unit vector in the **R** direction, and **R** $= \rho\, \mathbf{u}_R$. Equating the coefficients of **i, j,** and **k** in Equations 2.3 and 2.5 yields,

$$\ddot{X} \;=\; \left(n^2 - \frac{GM}{R^3}\right) X + 2n\dot{Y} + \mathbf{F_X} \qquad \mathbf{(2.6)}$$

$$\ddot{Y} \;=\; \left(n^2 - \frac{GM}{R^3}\right) Y - 2n\dot{X} + \mathbf{F_Y} \qquad \mathbf{(2.7)}$$

$$\ddot{Z} \;=\; -\frac{GM}{R^3} Z + \mathbf{F_Z} \qquad \mathbf{(2.8)}$$

To make the derivation more straight forward, the central body inertial coordinate system is initially aligned with the ascending node along the $\hat{\mathbf{i}}$ axis. Then, the origin of the target centered coordinate frame during node passage is,

$$(X, Y, Z)|_{\textit{target origin}} = (\rho, 0, 0) \qquad \textbf{(2.9)}$$

where ρ is the radius of the target from the center of the central body. The chase craft position is therefore,

$$(X, Y, Z)|_{\textit{chase craft}} = (\rho{+}x, y, z) \qquad \textbf{(2.10)}$$

Since the orbit is nearly circular and the orbit perturbations are assumed small, the magnitude of **R** is approximately constant and can be expressed as,

$$R = [\ (x{+}\rho)^2 + y^2 + z^2\ ]^{1/2} \qquad \textbf{(2.11)}$$

and then R^3 can be written as,

$$R^3 = [\ (x{+}\rho)^2 + y^2 + z^2\ ]^{3/2} \qquad \textbf{(2.12)}$$

If the orbit is assumed to be low Earth, ρ will be on the order of 6600 Kilometers while x, y, z are on the order of 500 meters for close proximity operations. It is therefore safe to say that $\rho >> x, y, z$. The following approximations follow,

$$\begin{aligned} \frac{GM}{R^3}\,y &= \frac{GM}{[\ (x+\rho)^2 + y^2 + z^2\ ]^{3/2}}\,y \\ &\approx -\frac{GM}{\rho^3}\,y \end{aligned} \qquad \textbf{(2.13)}$$

$$\begin{aligned} \frac{GM}{R^3}\,(x+\rho) &= \frac{GM}{[\ (x+\rho)^2 + y^2 + z^2\ ]^{3/2}}\,(x+\rho) \\ &\approx -\frac{GM}{(x+\rho)^3}\,(x{+}\rho) \end{aligned}$$

$$\approx -\frac{GM}{\rho^3\left(1+\frac{x}{\rho}\right)^3}(x+\rho) \tag{2.14}$$

using the binomial expansion theorem to further reduce Equation 2.14 yields,

$$-\frac{GM}{\rho^3\left(1+\frac{x}{\rho}\right)^3}(x+\rho) = \frac{GM\,(x+\rho)}{\rho^3\frac{1}{\rho}\,(x+\rho)\left(1+\frac{x}{\rho}\right)^2}$$

$$= \frac{GM\,\rho}{\rho^3}\left(1+\frac{x}{\rho}\right)^{-2}$$

$$\approx \frac{GM\,\rho}{\rho^3}\left(1-\frac{2x}{\rho}\right)$$

$$\approx -\frac{GM}{\rho^3}(\rho-2x) \tag{2.15}$$

Kepler's Third Law can be used to relate the central body mass, orbital radius and orbital rate,

$$n^2 = \frac{\mu}{a^3}$$

$$= \frac{G\,(M+m)}{a^3}$$

$$\approx \frac{G\,M}{\rho^3} \tag{2.16}$$

Where m is the mass of the smaller second body, orbiting craft, and a is the semi major axis of the orbit. This is a good approximation since M>>m and a is approximately equal to ρ for a nearly circular orbit. Equations 2.5 through 2.6 can be rewritten using the approximations in 2.13 through 2.16,

$$\ddot{X} = \left(n^2 - \frac{GM}{R^3}\right)X + 2n\dot{Y} + \mathbf{F_X}$$

$$\begin{aligned} \ddot{x} &= n^2 (x + \rho) - \frac{GM}{R^3} (x + \rho) + 2n\dot{y} + \mathbf{F}_x \\ &\approx n^2 (x + \rho) - \frac{GM}{\rho^3} (\rho - 2x) + 2n\dot{y} + \mathbf{F}_x \\ &\approx n^2 (x + \rho) - n^2 (\rho - 2x) + 2n\dot{y} + \mathbf{F}_x \\ &\approx 2n\dot{y} + 3n^2 x + \mathbf{F}_x \end{aligned} \tag{2.17}$$

$$\begin{aligned} \ddot{Y} &= \left(n^2 - \frac{GM}{R^3}\right) Y - 2n\dot{X} + \mathbf{F}_Y \\ \ddot{y} &= \left(n^2 - \frac{GM}{R^3}\right) y - 2n\dot{x} + \mathbf{F}_y \\ &\approx \left(n^2 - n^2\right) y - 2n\dot{x} + \mathbf{F}_y \\ &\approx - 2n\dot{x} + \mathbf{F}_y \end{aligned} \tag{2.18}$$

$$\begin{aligned} \ddot{Z} &= - \frac{GM}{R^3} Z + \mathbf{F}_Z \\ \ddot{z} &= - \frac{GM}{R^3} z + \mathbf{F}_z \\ &\approx -n^2 z + \mathbf{F}_z \end{aligned} \tag{2.19}$$

The three equations 2.17 through 2.19 are approximations of motion with respect to the target centered reference frame. These three equations will be defined as the linearized dynamics model. They are,

$$\ddot{x} = 2n\dot{y} + 3n^2 x + \mathbf{F}_x \tag{2.20}$$

$$\ddot{y} = - 2n\dot{x} + \mathbf{F}_y \tag{2.21}$$

$$\ddot{z} = -n^2 z + \mathbf{F}_z \tag{2.22}$$

which are commonly termed the Hill's Equations for forced motion. George William Hill, after whom these equations are named, first published this solution to the restricted three body problem in 1878 in the first issue of the *American Journal of Mathematics* [4].

Summarizing the assumptions made in the derivation above:

(1) The target is in a near circular orbit. This makes the coordinate system orthogonal and makes the orbital rate constant.

(2) **F** is small compared to the central force. This keeps the orbit perturbations small and the orbit nearly circular.

(3) The displacements x, y and z are small compared to the distance ρ. Even for low Earth orbits, ρ is on the order of 6×10^6 meters while x and y will be no more than 1×10^3 meters in this study; ρ is at least three orders of magnitude greater than x and y for these proximity operations.

(4) The impulses applied to the satellites normal to the orbital plane are small. This keeps the magnitude of z much smaller than ρ.

(5) The central body is assumed to be spherical; J_2 and higher order perturbations are ignored. If desired, these effects could be accounted for in the external forces terms.

(6) The mass of the second orbiting craft is much smaller than the central force mass.

Since none of the assumptions are unreasonable for the proximity operations scenario, the Hill's Equations are valid for this application. The errors in this method are caused by neglecting x, y and z in the approximations and are therefore of order $[x^2 + y^2 + z^2]^{1/2} / \rho$. Since x, y and z are three orders of magnitude smaller than ρ, the errors will be insignificant.

Force Free Solution

If no force is applied to the chase craft, the Hill's equations reduce to

$$\ddot{x} - 2n\dot{y} - 3n^2 x = 0 \qquad (2.23)$$

$$\ddot{y} + 2n\dot{x} = 0 \qquad (2.24)$$

$$\ddot{z} + n^2 z = 0 \qquad (2.25)$$

which are commonly referred to as the Clohessy-Wiltshire equations. Although the equations are attributed to W. Clohessy and R. Wiltshire for their paper in the September 1960 issue of the *Journal of Aerospace Science*, another author developed the solution concurrently [3]. Mr. A. Wheelon published his own version of the force free solutions in the chapter "Midcourse and Terminal Guidance" in the book *Space Technology* published in 1959 [9].

The y in Equation 2.23 can be eliminated by integrating Equation 2.24 and substituting the value for y. The resultant equation is,

$$\ddot{x} + n^2 x = 2n\left(\dot{y}_o + 2nx_o\right) \tag{2.26}$$

where (x_o, y_o, z_o) are the initial conditions of (x, y, z); ie. values prior to application of an external force.

The unforced positions and velocities at time t (indicated by a subscript 1) in terms of the initial position and velocities (indicated by a subscript 0) are presented below in state space representation.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \dot{x}_1 \\ \dot{y}_1 \\ \dot{z}_1 \end{bmatrix} = \phi \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix}$$

$$\phi = \begin{bmatrix} 4 - 3\cos(nt) & 0 & 0 & 1/n \sin(nt) & 2/n\,(1-\cos(nt)) & 0 \\ -6(nt - \sin(nt)) & 1 & 0 & -2/n\,(1-\cos(nt)) & 4/n \sin(nt) - 3t & 0 \\ 0 & 0 & \cos(nt) & 0 & 0 & \sin(nt)/n \\ 3n \sin(nt) & 0 & 0 & \cos(nt) & 2\sin(nt) & 0 \\ -6n(1-\cos(nt)) & 0 & 0 & -2\sin(nt) & 4\cos(nt) - 3 & 0 \\ 0 & 0 & -n\sin(nt) & 0 & 0 & \cos(nt) \end{bmatrix} \tag{2.27}$$

This state space relationship implies that any target state can be commanded at a time t by picking the correct initial velocity. The correct velocity is determined by pre-multiplying the target state by the inverse of the state transition matrix, assuming that an inverse exists. To find the conditions under which this matrix has no inverse, its determinant is calculated and set equal to zero:

$$\sin(nt)\,[\,8 - 8\cos(nt) - 3nt\sin(nt)\,] = 0 \tag{2.28}$$

which has two zeros. The periodic zeros of the characteristic equation are created by the sin(nt) which premultiplies the bracketed quantitiy. These zeros occur when,

$$nt = N\pi \tag{2.29}$$

where N is an integer. Or equivalently,

$$t = N/2\ P \tag{2.30}$$

where P is the orbital period of the craft. Another set of aperiodic zeros are created by the bracketed quantity, the first of which occurs at nt = 8.84 r. At these values of nt , the state space relationship reduces to,

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \dot{x}_1 \\ \dot{y}_1 \\ \dot{z}_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -6nt & 1 & 0 & 0 & -3t & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \tag{2.31}$$

This transition matrix implies that only y_1 can be controlled. x_1 and z_1 can never be controlled for transfers of these times since the satellite must have the same magnitude of radius at integer multiples of the period (for a two impulse transfer).

The force free solution provides an adequate model for the translational dynamics of proximity operations. Since the burn times are small in relation to the total time of the maneuvers, the burns can be assumed to be impulsive and the change in velocity vectors instantaneous. The transfer is therefore characterized by a series of coasting trajectories defined by the force free solution.

Attitude Dynamics

The current orientation of the chase craft is represented by the current quaternion from the LVLH frame to the Body Frame. This can be computed quickly and efficiently if the angular rates in the Body Frame are assumed constant between impulses. This is a good assumption because the effects of atmospheric drag, gravity gradients, and Euler coupling on the chase craft, which slowly change the body rates, will be small for transfer times investigated. This quaternion is made up of two parts; a vector about which the current reference frame must be rotated, and the angle of rotation about that vector. Therefore, if the LVLH reference frame is rotated about the vector defined in the quaternion through the proper angle, the body reference frame will be found. (see Figure 2.3)

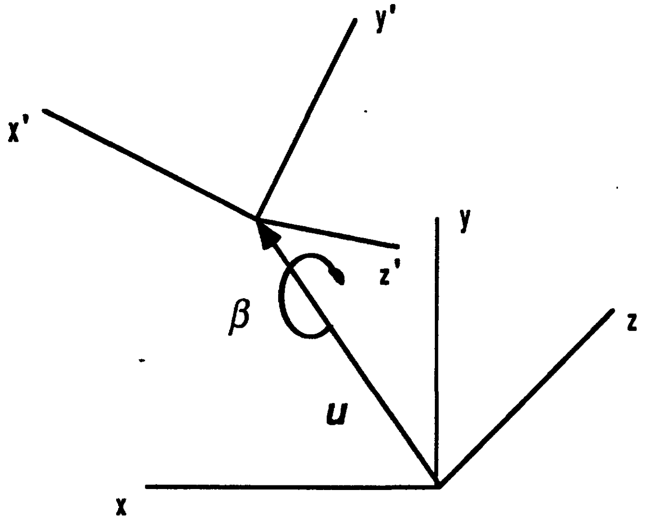


Figure 2.3: *Quaternion Rotation*

Where **u** is the vector of rotation and β is the angle of rotation. The quaternion chosen is an ordered quadruple with a norm of one,

$$\mathbf{q} = \begin{bmatrix} \cos(\beta/2) \\ -u_1 \sin(\beta/2) \\ -u_2 \sin(\beta/2) \\ -u_3 \sin(\beta/2) \end{bmatrix} \qquad \textbf{(2.32)}$$

As stated earlier, both the LVLH reference frame and the body reference frame are rotating with time. Therefore, finding the current orientation of the chase craft is a four step process.

The first step is to find the orientation of the current body frame to the initial body frame. This step takes advantage of the two assumptions previously made, that the angular rates are constant between impulses and that all rate changes are assumed impulsive. If the angular rates are constant , the quaternion angle and vector between any two body frames is given by ,

$$\beta = |\omega|\, t \qquad \mathbf{u} = \begin{bmatrix} \omega_1 / |\omega| \\ \omega_2 / |\omega| \\ \omega_3 / |\omega| \end{bmatrix} \tag{2.33}$$

where ω is the constant body rate during this period. The first step is satisfied by substituting this angle and vector set into equation 2.32 defining the quaternion between the two body frames.

The second step is to find the orientation of the current body frame to the initial LVLH frame. This is accomplished by multiplying the quaternion from LVLH $_0$ to body $_0$ by the quaternion from body $_0$ to body $_1$ (subscipt 0 indicates time zero, subscript 1 indicates current time).

$$\mathbf{q}_{\mathrm{LVLH}\,0}^{\mathrm{body}\,1} = \mathbf{q}_{\mathrm{body}\,0}^{\mathrm{body}\,1}\ \mathbf{q}_{\mathrm{LVLH}\,0}^{\mathrm{body}\,0} \tag{2.34}$$

For the definition of quaternion multiplication, see reference 6.

The third step is to find the orientation of the current LVLH frame to the initial LVLH frame. Knowing that the LVLH reference frame also rotates at the constant rate n about its second axis, the quaternion angle and vector between any two LVLH frames is given by ,

$$\beta = nt \qquad \mathbf{u} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{2.35}$$

The third step is satisfied by substituting this angle and vector set into equation 2.32 defining the quaternion between the two LVLH frames.

The final step is to find the orientation of the current body frame to the current LVLH frame. This is accomplished by multiplying the quaternion from LVLH $_1$ to LVLH $_0$ by the quaternion from LVLH $_0$ to body $_1$.

$$q_{\mathrm{LVLH}\,1}^{\mathrm{body}\,1} = q_{\mathrm{LVLH}\,0}^{\mathrm{body}\,1}\; q_{\mathrm{LVLH}\,1}^{\mathrm{LVLH}\,0} \tag{2.36}$$

This four part process is a simplistic method of tracking current attitude of the body frame. With the quaternions given here and the force free solution from the previous section, the chase position and attitude can be found at any time with respect to the target state.

Trajectory Planner

The purpose of the trajectory planner is to create a series of intermediate states that the chase vehicle should fly through on its way to the target state. At several of these intermediate states the chase vehicle will be required to execute rate changes so that it can continue along the planned trajectory. This set of rate changes, along with the associated positions and times, defines the impulse profile that the chase craft must follow to reach the target state.

The number of search methods which could be utilized to calculate this impulse profile is almost endless; however, some are better suited to the task than others. Furthermore, their is no obvious choice of independent variables to use in the search algorithm. For this reason, two different algorithms were investigated. The first was an application of simulated annealing which used three independent variables: number of impulses, position of each impulse, and time of each impulse. Although this method did create reasonable solutions, the computation time required was exceptionally high. The second algorithm, a modified gradient search, creates more efficient trajectories with greatly reduced computation time. The modified gradient search algorithm is outlined in this section.

Cost Function

The cost function used in this planner is the total fuel use required to complete the maneuver,

$$\mathbf{Cost} = \sum_{i=1}^{n} \text{fuel use}_{\text{impulse } i} \qquad \mathbf{(3.1)}$$

where n is the total number of impulsive velocity changes commanded by the trajectory planner. The fuel use required for a given maneuver is a function of specific impulse, jet location, thrust direction, jet coupling, and minimum impulse of the jets; therefore, this cost function is highly vehicle specific.

To make the algorithm as general as possible, the impulses are represented as changes in velocity and attitude rates. The output of the algorithm is merely a schedule of combined rate changes at given locations and times along the trajectory. Since the planner commands rate changes and not jet on times, the algorithm can be adapted to any vehicle by simply changing the jet select algorithm. For testing purposes, a linear programming jet selection for the Space Shuttle [1] has been chosen to compute fuel use.

As most vehicles performing proximity operations will have range and range rate information to the target state, feedback control of the maneuver execution is possible. The feedback control is accomplished by defining a region around the reference trajectory beyond which the chase craft cannot stray; the limits of this region are called deadbands. If the chase vehicle reaches one of these deadbands, the autopilot will correct the vehicle state with additional jet firings. As the Shuttle jets have a minimum firing interval of eighty milliseconds and therefore cannot exactly nullify the undesired rates, the deadbanding firings are biased to drive the state toward the center of the deadband region. Since the vehicle will continue to oscillate between the limits defined by the deadbands, these additional firings are termed limit cycling effects.

The trajectory planner should have the capability to account for these additional firings since they may account for more than fifty percent of the fuel use for a long transfer. The cost function used when considering feedback control of the execution is

$$\textbf{Cost} = \sum_{i=1}^{n} (\text{fuel use}_{\text{impulse } i}) + (\Gamma * \text{tof}) \qquad \textbf{(3.2)}$$

where Γ is the limit cycling constant and tof is the maneuver time. The limit cycling constant is an average fuel use as a function of time due to the added jet firings. It describes the magnitude and frequency of the firings required to keep the average deviation from the planned trajectory approximately zero. Since it is a function of the fuel use considerations noted above as well as vehicle orientation and magnitude of the position and attitude deadbands, it is an empirically determined constant. As the correction for limit cycling effects is only of first order, a slight error in fuel use predictions is expected. Adding this correction reduces the maneuver time of flight so that limit cycling effects no longer dominate total fuel use.

Search Space

The search method used in this algorithm is a modified gradient search, incorporating heuristics to trim the search space when necessary. The algorithm searches over a set of independent variables simultaneously to minimize the cost function. The search begins by finding a two impulse solution to the problem, via equation 2.27, which meets all of the constraints defined by the maneuver environment. This solution is then made more efficient by adding impulses and then determining the best position, magnitude, direction, and time of these impulses. The gradient search moves in the direction of the maximum negative gradient of the cost function and terminates when it encounters a positive gradient in all directions.

Independent Variables

To optimize the cost function, the algorithm searches over five independent variables: time between each impulse, number of impulses, position of the impulses, time of the first impulse, and the attitude at each impulse. An infinite number of possible trajectories to the target state exist for any given value of each independent variable, as long as it is not one of the singular times of flight defined by Equation 2.28. Therefore, the search space and computation time grows rapidly as the number of independent variables is increased. For this reason, the search space has been limited slightly.

The cost function is most sensitive to changes in the time of flight. For this reason, this variable's search space is not constrained. The search space on time runs from instantaneous transfer, infinite cost, to the maximum time of flight allowed for the maneuver . Any combination of second, third, and fourth impulse times that adds up to this total transfer time is allowable.

The number of impulses has a large effect on the cost function, but only through four impulses for reasonable obstacle sets and terminal conditions. This has been empirically determined by investigating many different types of obstacle sets and closing velocity constraints. For closing velocities greater than 0.01 ft/s, four impulse solutions are at least as efficient as solutions incorporating more impulses. In fact, the only time that the algorithm returned optimal trajectories requiring more than four impulses was when the chase craft had to maneuver around a complex obstacle that encompassed the target state. Based on these considerations, the number of impulses allowed in the search space is limited to four.

It is more efficient at times to delay the time of the first impulse and allow the vehicle to coast around obstacles. However, the algorithm is only asked to search over this variable for two impulse solutions. This is because the three and four impulse solutions can implicitly search over this variable by changing the position and time of the second impulse so that the initial impulse is zero. The range of values for the first delay time is the same as that for the time of flight with the stipulation that the total transfer time must be less than the maximum transfer time allowed.

The intermediate impulse positions can be optimized for both three and four impulse solutions. These positions are optimization by a gradient search and therefore have no imposed limits except that goal state and the initial state cannot be chosen. This search space, however, has certain physical limits because many positions are impossible to achieve given the control authority of the vehicle and the limitations on time of flight.

To take advantage of the coupling of the chase craft's jets, the vehicle attitude at each of the intermediate impulse positions can be varied for three and four impulse solutions. By varying the chase craft's attitude at the impulse positions, the angular rates during each coast period is implicitly varied. The attitude search space runs from zero to three hundred sixty degrees in roll, pitch, and yaw for the second and third impulse positions. Since the initial and final states are predetermined, the attitudes at these positions cannot be varied.

Dependent Variables

Two dependent variables are tracked as the independent variables are optimized; these are total fuel required (the cost function) and closing velocity near the target state. The reason for calculating the total fuel required is obvious; it is the quantity being minimized by the algorithm. The closing velocity is calculated so that the two craft do not close at a rate faster than that specified by the maneuver manager. The closing velocity is particularly important when the final state is a dock rather than station keeping.

Constraints

The maneuver space is constrained by three quantities: maximum time of flight, closing velocity, and physical obstacles in close proximity to the chase craft and target state. The maximum time of flight is an upper limit on the total maneuver time: no limitation is placed on how this time is distributed between the impulses. The closing velocity is an instantaneous limitation on the velocity at the terminal point. Obstacles limit the maneuver space because no two vehicles are allowed to occupy the same location in space at the same time. If any of the constraints are violated by a trajectory, this trajectory is unacceptable and it is abandoned by the trajectory planner.

Algorithm Operation

The algorithm requires a great deal of information about the maneuver space before it can generate a trajectory to the target state. Since this thesis deals solely with trajectory planning and not vehicle sensors, all information on relative positions and rates is considered perfect. The information required by the planner includes:

- Initial position and velocity of the chase craft with respect to the target state.
- Altitude of the orbit above Earth (spherical Earth is assumed).
- Maximum time allowed for the maneuver.
- Maximum terminal closing velocity with respect to the target state.
- Radius of a sphere, centered at the center of mass, required to enclose the chase vehicle.
- Number of degrees of freedom in the search. Three corresponding to translation only and six to translation and rotation.
- Cost function to be used; eg. whether to account for limit cycling.
- If the six degree of freedom search is selected,
    - Initial roll, pitch, and yaw with respect to the initial LVLH frame.
    - Final roll, pitch, and yaw with respect to the final LVLH frame.
    - Initial body rates with respect to the chase craft body frame.

With this information, the algorithm is able to proceed with the search.

The search begins by finding a two impulse solution to the target state using eq. 2.27, if one exists, subject to all the above constraints. This solution serves as a baseline solution which meets every constraint except for obstacles in the maneuver space. The search can continue without obstacles in the search space, however, if they are to be considered they must be input at this point. The necessary input for an obstacle set is,

-- Initial position and velocity of each obstacle with respect to the target state.
-- Status of each obstacle, moving or fixed with respect to the target.
-- Radius of a sphere, centered at the center of mass, required to enclose each obstacle.

It is important to note that all moving obstacles are assumed to be coasting after time zero. The radius required to enclose the chase vehicle and each obstacle is necessary so that the planner can simplify the propagation of each in time. Furthermore, a collision between the chase vehicle and an obstacle is indicated when the distance between their centers of mass is less than the sum of the sphere radii.

The baseline case is then altered to seek the optimal two impulse solution around the obstacles if one exists. This is done by varying the time of flight from the reference time found earlier until the obstacles are avoided. While not necessarily optimal, this solution is guaranteed to meet all of the input constraints. Since this solution can be calculated almost immediately, this is a very valuable solution if replanning is necessary due to impending collision. As two impulse solutions often have trouble meeting closing velocity constraints, it is recommended that this constraint not be considered until after the collision is avoided and more time is available for replanning.

The two impulse solution may be made even more nearly optimal by varying the time of the first impulse. Delaying the first impulse can improve efficiency of the solution if the initial velocity is in a favorable direction. However, if the initial velocity vector tends to move the chase craft away from the target state, a delayed burn is detrimental and the resultant cost will be greater. At this point in the search, the best two impulse solution has been found and the algorithm proceeds to add intermediate impulses.

As the number of impulses increases, the search space increases drastically. To increase the search speed, the user may vary both the time and spatial step sizes. Each coast period between impulses is incremented by the time step while the impulse positions are incremented by the spatial step. As with all gradient searches, the search time required is greatly dependent on the accuracy of the initial guess. Additional impulse positions are therefore considered at several logical positions along the reference trajectory. The best of these is chosen as the initial guess. Intermediate impulse positions are then optimized by searching each of the adjacent spatial nodes and selecting the node with the minimum cost. The search terminates when each of the adjacent nodes has a higher cost than the current node. Notice that this search may get caught in a local minimum; however, this can be avoided by cleverly choosing the spatial step size.

At this point the algorithm has calculated the most optimal two, three, and four impulse solutions to the input set. If the three degree of freedom search was selected, the search terminates at this point. Attitude angles and rates are assumed to be zero for the entire maneuver and the best of the calculated trajectories is output to the user.

If the six degree of freedom search was selected, the best current solution is sent to the attitude post optimizer. In each of the sections of the search outlined above, a constant attitude rate was assumed during the entire maneuver. If the best solution is a three or four impulse solution, however, the solution can be further improved by varying the attitude rates at each of the impulses. This portion of the search is a simple search over the entire range of possible attitude angles where both the search space and step size are reduced between each iteration. This portion of the search is not intended to produce the best solution possible by a six degree of freedom search; rather, it simply creates a more efficient solution than the solution given it by the three degree of freedom search. With this more efficient solution, insight can be gained into how varying the attitude at each impulse can reduce the total cost.

Mathematical Description

In the three previous sections the basic algorithm was described. In this section a simple functional overview is presented on how the algorithm was practically implemented in computer code. This is done so that the reader can gain some insight into the strengths and weaknesses of the algorithm without delving into the mathematics in excruciating detail.

To be able to design any proximity operation, one must be able to readily calculate four pieces of information. The first two deal with how to get to a given location in space at a specified time. To accomplish such a maneuver with impulses applied only at the initial and final positions, one must be able to determine the initial velocity and attitude rates that allow the vehicle to coast to the desired position at the correct time . The third piece of information is the fuel cost associated with any specified rate change. The final quantity that must be calculable is the state of the chase vehicle, position and velocity, at any given time between impulses. The equations used in calculating these quantities are outlined in the chapter Equations of Motion.

From equation 2.27, the calculation of the required initial velocity can be represented as an operation of the form,

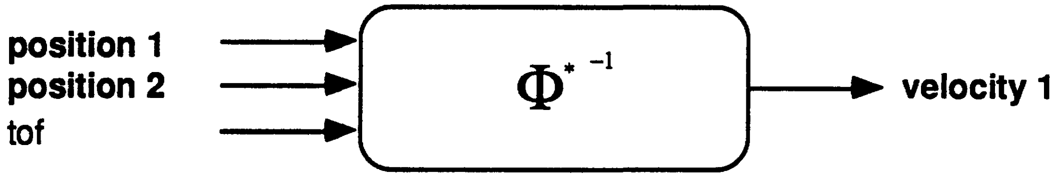


Figure 3.1: *Required Initial Velocity Calculation -- operation VEL*

Where the matrix Φ^* is used to represent the first three rows of the Φ matrix evaluated at time tof, eg. time when position two is achieved. Only the first three rows of the Φ matrix are necessary because a final position is desired, not a final position and velocity. Since Φ^* is a three by six matrix, this operation is slightly more complicated than a simple inversion. The operation actually solves for v_0 in equation 3.3 below.

$$[x_1] = [\mathbf{A}\ \mathbf{B}] \begin{bmatrix} x_0 \\ v_0 \end{bmatrix} \qquad (3.3)$$

Where **A** and **B** are the three by three partitions of the Φ^* matrix, x is the position vector, and v is the velocity vector. The solution to this equation is

$$v_o = \mathbf{B}^{-1}\,[x_1 - \mathbf{A}\,x_0] \qquad (3.4)$$

This calculation will be referred to as operation VEL.

A similar operation can be performed to determine the attitude rates necessary to reach a final attitude at a specified time using equation 2.32. This operation can be represented as,

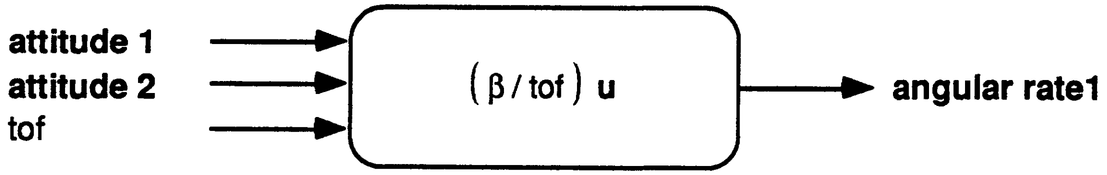


Figure 3.2: *Required Initial Body Rates Calculation -- operation RATE*

Where **u** is the vector of rotation and β is the angle of rotation required to get from attitude 1 to attitude 2 in the time tof. Since the two attitudes are input in quaternions with respect to the rotating LVLH reference frame, the calculation requires the three quaternion multiplications shown below.

$$q^{body_2}_{body_1} = q^{body_2}_{LVLH_2}\, q^{LVLH_2}_{LVLH_1}\, q^{LVLH_1}_{body_1} \tag{3.5}$$

The values of **u** and β are then read from the resulting quadruple as outlined in equations 2.32 and 2.33. This calculation will be referred to as operation RATE.

Now that the required velocity and angular rates can be calculated for a transfer between any two points in space with a specified transfer time, it is possible to determine the cost associated with this transfer. This is done with the cost functions in equations 3.1 and 3.2, where the fuel use for each impulse is obtained from the jet select algorithm. This operation is represented as,

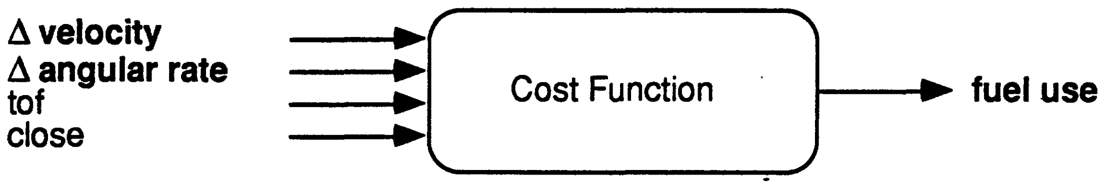


Figure 3.3: *Fuel Cost Calculation -- operation FUEL*

Where tof is the time until the next burn, close is a flag for considering limit cycling effects, and Δ velocity and angular rates are the requested rate changes. The rate changes are found by simply subtracting the rates immediately before the impulse from the requested rates. This calculation is referred to as operation FUEL.

When the vehicle translates between impulses, the state at any time can be determined by simply propagating the linear equations of motion forward in time. The translational portion of the state vector is calculated with equation 2.27 while the attitude portion of the state vector is calculated with equations 2.32 thru 2.36. This operation can be represented as,

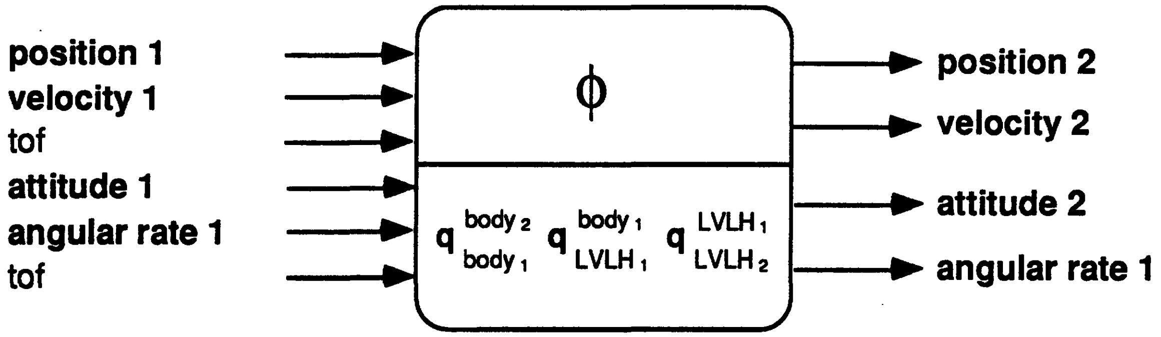


Figure 3.4: *Current State Calculation -- operation STATE*

Where tof refers to the time since the last impulse. Since gravity gradient and atmospheric drag effects are not considered, the angular rates are constant with time. This calculation is referred to as operation STATE. In all of the diagrams that follow, the state vector will refer to position, velocity, attitude, and angular rates. The rate vector will refer to both velocity and angular rates.

With these four basic operations, it is straightforward to show how the algorithm has been implemented. The following algorithm flow charts use these primitive operations to show how the various trajectories are created and checked against the input constraints. The first of these algorithm flow charts, figure 3.5, is for the two impulse solution with the first impulse occurring at time zero.

This trajectory is created by calling the four primitive operations above with several different times of flight. The minimum cost solution that meets the closing velocity constraint is then returned. If none of the two burn solutions meet the closing velocity constraint, the time of flight of the minimum fuel use solution between the two states is returned as the reference time of flight and no further two burn solutions are attempted. Notice that the rates are driven to zero relative to the LVLH reference frame by the final impulse. This is done so that the chase craft will track the origin of the LVLH frame in both position and attitude after the maneuver is completed.

The process of calculating the fuel use between two states in a specified period of time has been grouped into a macro operation called SEGMENT. This macro operation defines what is referred to as a coasting segment. The initial impulse generates the rates needed to coast to the desired position, and the final impulse generates the final desired rates. By combining these

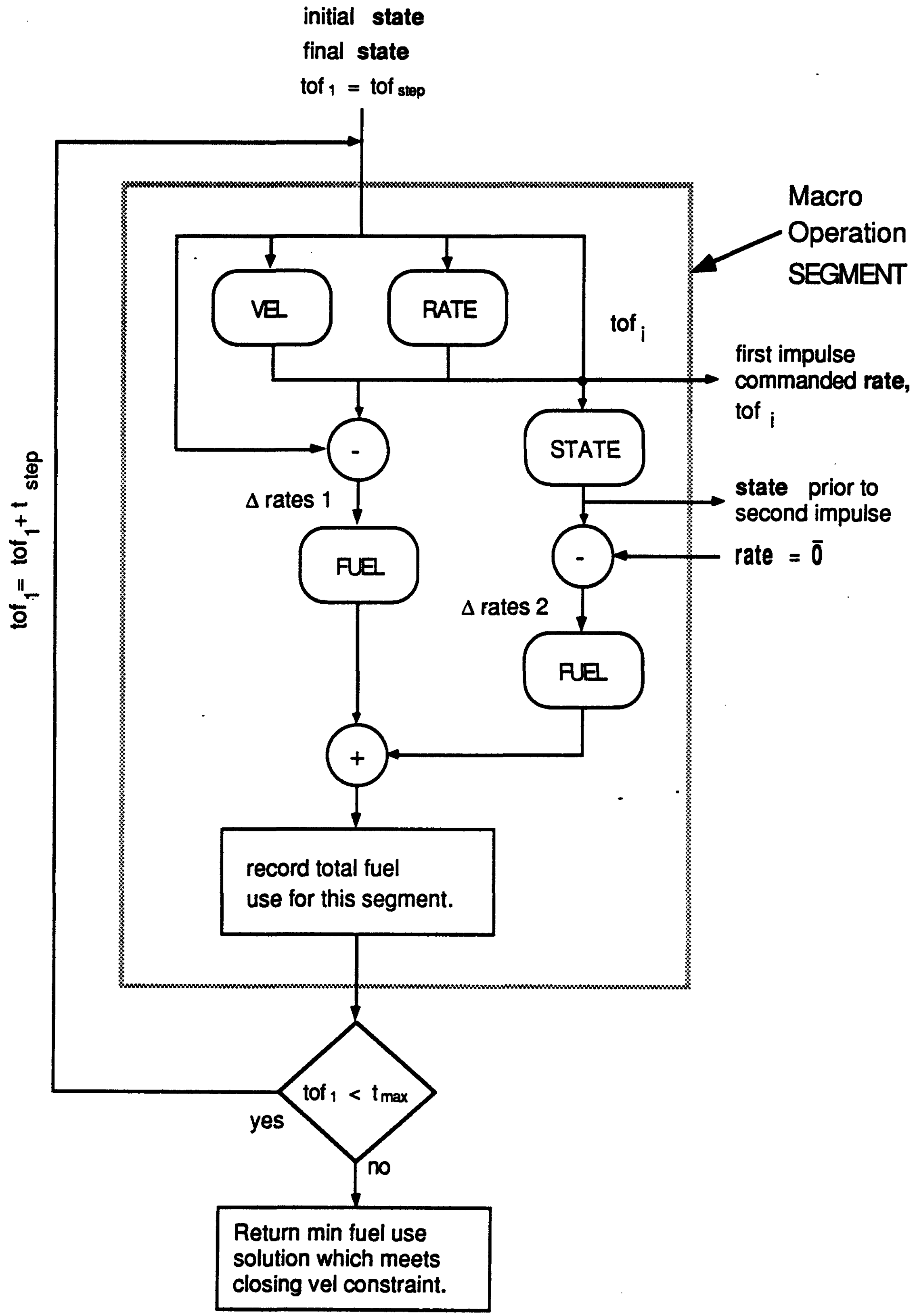


Figure 3.5: *Two Impulse Solution -- Macro Operation SEGMENT*

segments a trajectory with multiple burns can be achieved. The inputs required for each segment are simply the initial and final state vectors and the current time of flight, tof, of each segment. For multiple segment solutions, the tof of each segment is varied simultaneously. In the two impulse solution presented in Figure 3.5, only tof_1 is defined and the time step is hard wired to one minute.

After obstacles are input, the algorithm searches for a two impulse solution around the obstacle set. This is done as before except that the initial value of tof_1 is the reference time of flight. Since the obstacles can be avoided by either increasing or decreasing the time of flight, the logic depicted in Figure 3.5 is called first with a negative one minute time step and then with a positive one minute time step. The best solution which meets all of the constraints is returned.

The next type of solution created by the algorithm is the two impulse solution with a delayed first impulse. Here, the first impulse occurs at integer multiples of the time step. The chase craft state is propagated forward by the delay time and used as the initial condition input to SEGMENT, see Figure 3.6. Here two flight segments exist, one for the coast before the first impulse and one for the coast period between the two impulses. The sum of these times of flight never exceeds the maximum time of flight. Since the first impulse must be delayed by at least one time step, the solution returned by this routine may be less fuel efficient than the two impulse solution found earlier.

As stated earlier, the multiple segment trajectories are found by combining more than one of the segment operations. Since multiple times of flight exist for these solutions, the logic must be created to vary each of these segment times of flight simultaneously. Figures 3.7 and 3.8 illustrate the procedures for two and three segment solutions. These flow charts show only the time of flight variation; the gradient search on the impulse positions will be illustrated later in this section.

In Figure 3.7 a three impulse solution is created with two calls to operation SEGMENT. In the first call of SEGMENT, the current state is fed back as the desired state, thus eliminating one of the two commanded impulses. This is necessary to create a trajectory with an odd number of impulses because operation SEGMENT was created to command two impulses. The rates after the first impulse are then propagated forward in time to the position of

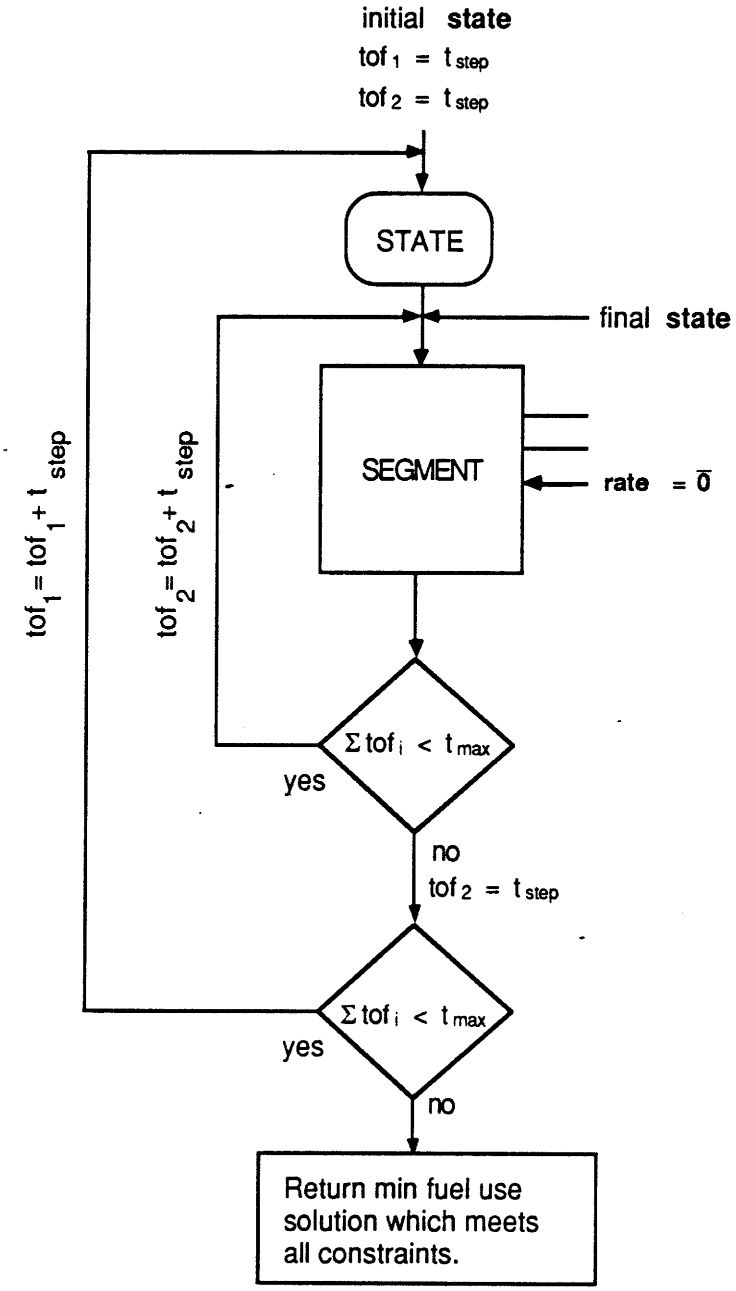


Figure 3.6: *Two Impulse Solution , delayed first impulse*

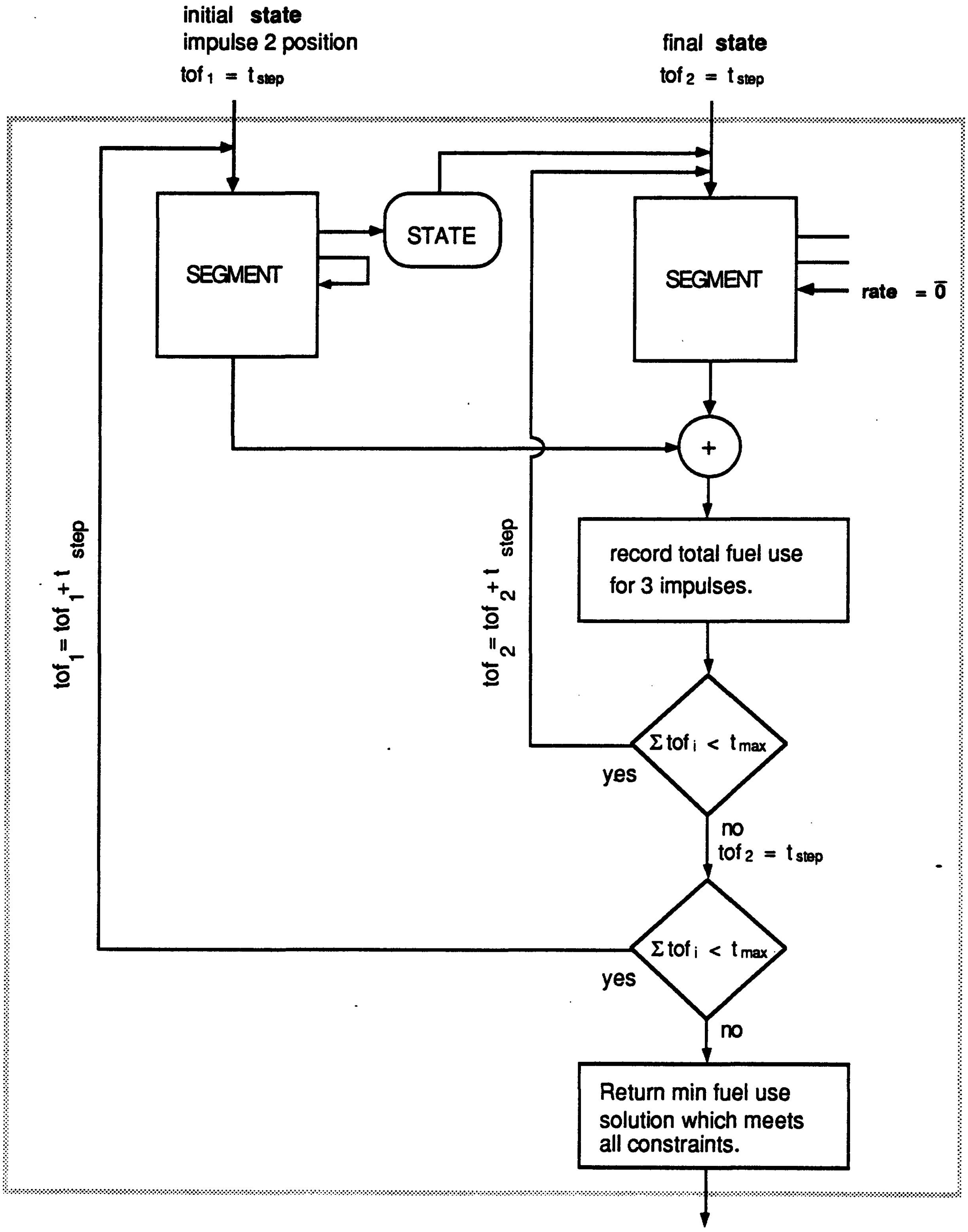


Figure 3.7: *Macro Operation Two Segments*

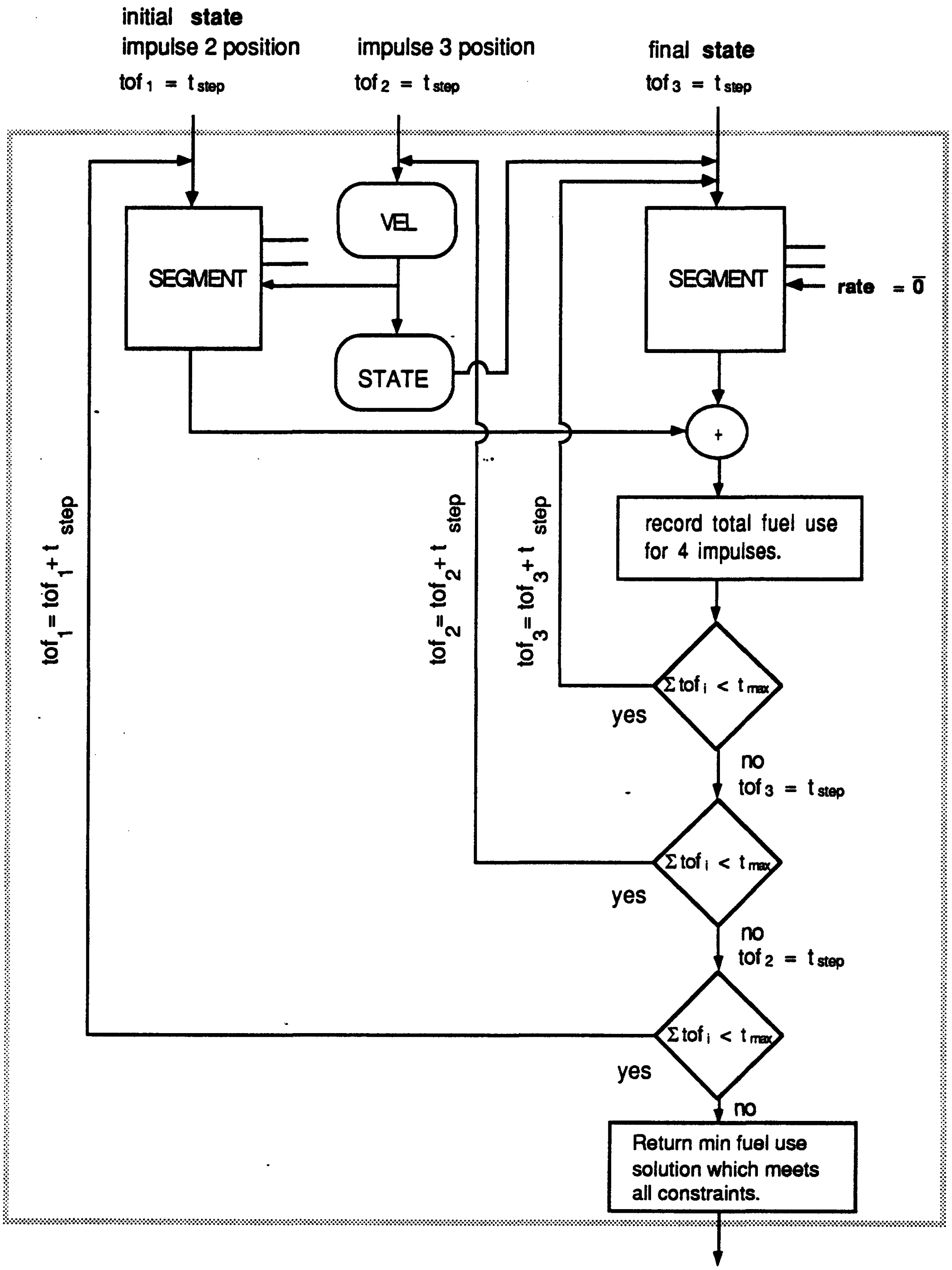


Figure 3.8: *Macro Operation Three Segments*

the second impulse by the operation STATE. A second call to SEGMENT creates the last two impulses that define this trajectory. This process is combined into a macro operation called Two Segments.

In Figure 3.8 a four impulse solution is created. Again, only two calls to operation SEGMENT are used. Although three segments exist in the four impulse solution, only two calls to SEGMENT are required. The first and third segments of the trajectory are created by the two calls to SEGMENT. The second segment, however, is simply created by calling operation STATE with the rates required to reach the position of the third impulse. Notice that here three different segment times of flight are varied simultaneously; this explains the increase in run time associated with an increase in the number of impulses. This process is combined into a macro operation call Three Segments.

To complete the three and four impulse solutions, some method must be created for optimizing the positions of the intermediate impulses. The method used in this algorithm is a steepest descent gradient search. An initial guess for the intermediate impulse positions is found by looking at the chase craft position at several different times along the two impulse reference trajectory. Impulses are added at each of the chase craft positions investigated, and the impulse positions associated with the lowest fuel use trajectory are chosen as the starting point of the gradient search.

In Figure 3.9 the three impulse solution is created in this manner. The initial position of the second impulse is chosen as the best of eight positions along the reference trajectory. These positions are found by propagating the state after the first impulse, denoted by initial **state*** , forward in time with the operation STATE. Notice that one of the positions considered is only ten seconds from the final state along the reference trajectory. This position is considered to provide an impulse near the target state so that very small closing velocities can be accommodated. The gradient search then looks a unit step in all directions, a total of twenty seven positions, and selects the impulse position associated with the maximum negative gradient in the cost function; this solution is returned as the optimal three impulse trajectory. The gradient search continues until a positive gradient is found at each of the twenty seven adjacent nodes. Notice that the spatial step size is denoted by 'step' and the time step size by 't_{step}'.

Although Figure 3.10 looks much more complex than the three impulse solution, it is not. The four impulse solution looks much more difficult because

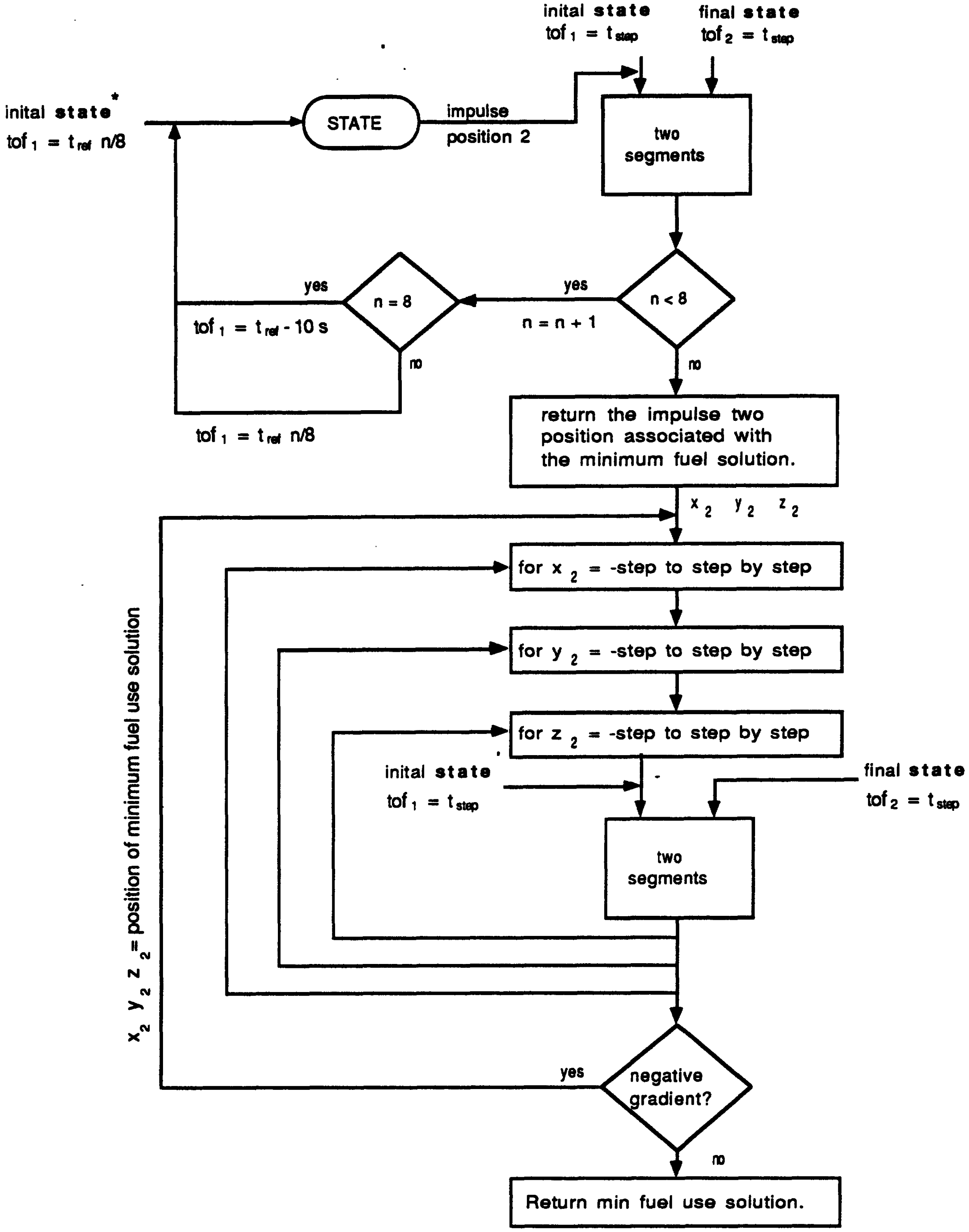


Figure 3.9: *Three Impulse Solution , Gradient Search on Impulse Position 2*

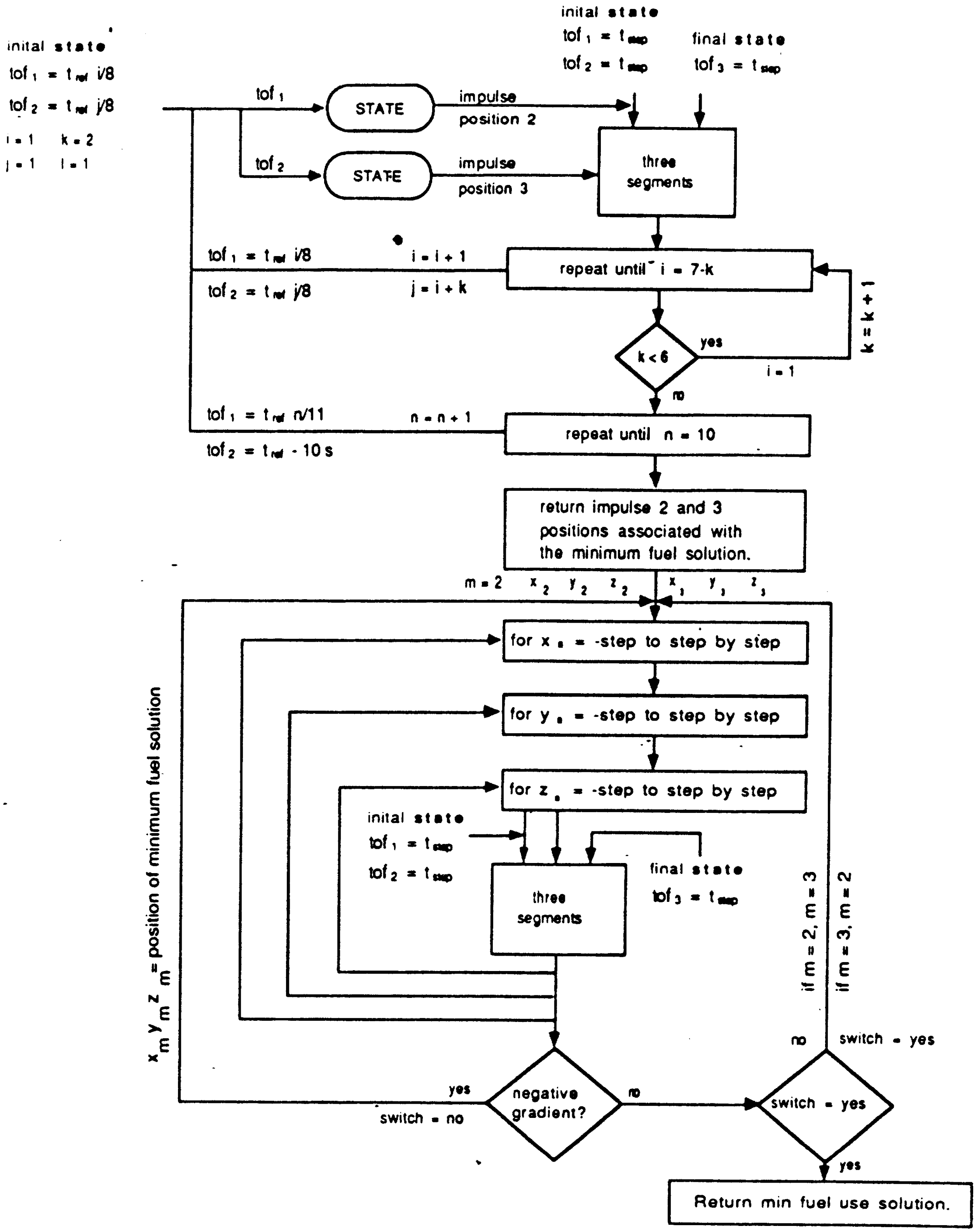
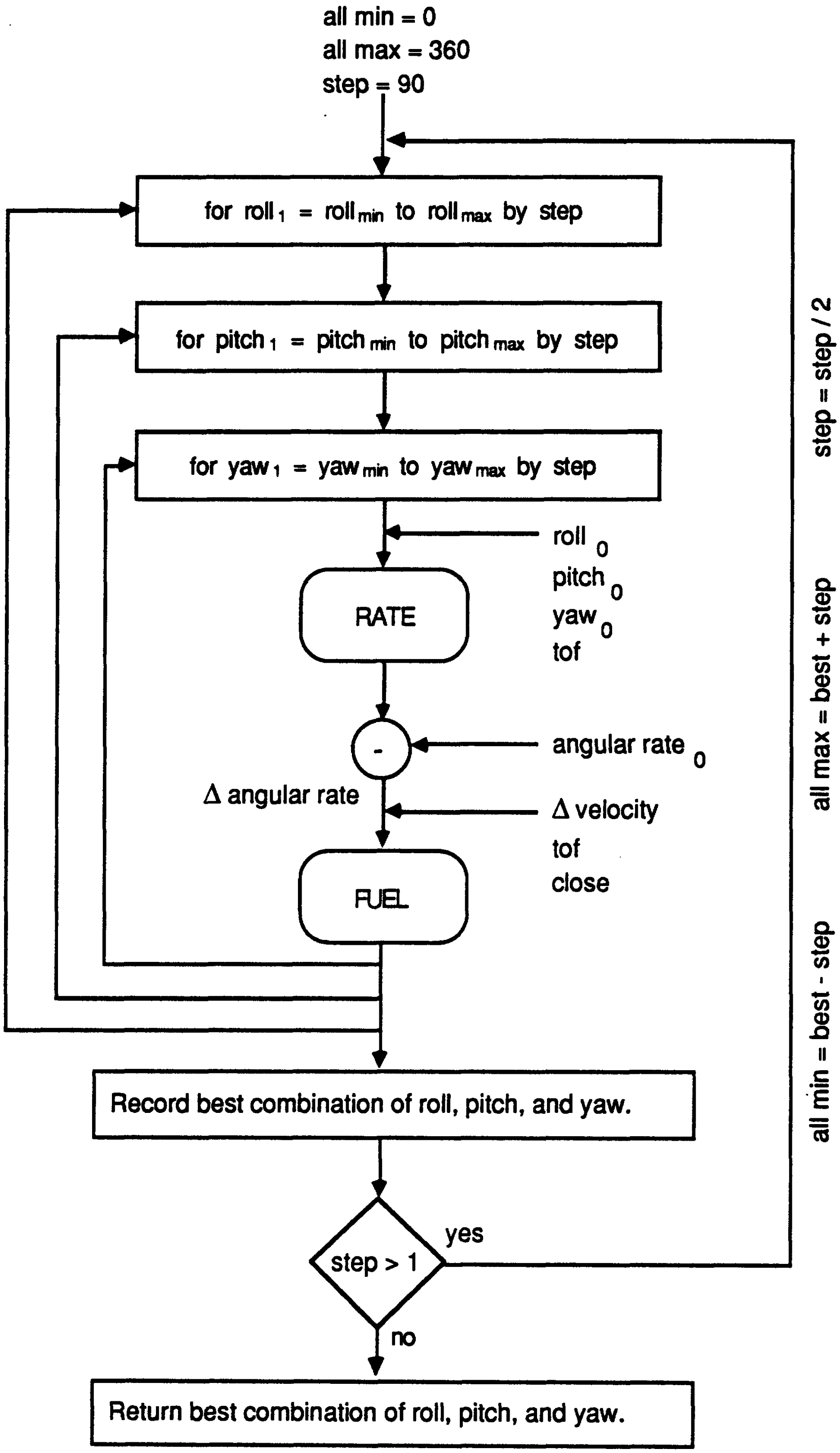


Figure 3.10: *Four Impulse Solution , Gradient Search on Impulse Positions 2 and 3*

a second impulse position must be optimized. Since a call to Three Segments takes considerably longer than a call to Two Segments, more care is taken to start the gradient search with a good set of intermediate impulse positions. Twenty five different sets of impulse positions are investigated here as defined by the three logical loops. Once an initial set of impulse positions is found, the search is started on the second impulse position. The second impulse position is optimized until a positive gradient is encountered at each adjacent node, then the third impulse position is optimized. When the third impulse position encounters a positive gradient at each adjacent node, the second impulse position is again optimized. This process continues until both impulse positions encounter a positive gradient simultaneously; this solution is returned as the optimal four impulse trajectory.

If the best of the above solutions is either a three or four impulse trajectory, the algorithm calls the attitude post optimizer. The post optimizer attempts to decrease the fuel use by taking advantage of the coupling inherent in the jets to produce desired angular rates. Since the attitudes at the initial and terminal states are fixed, only the attitude at the intermediate impulse positions can be varied. The attitude optimization at an impulse position is depicted in Figure 3.11. As the attitude at only one impulse position is optimized in the figure, it is a good description of how the second impulse of a three impulse solution is optimized. To illustrate the optimization of both the second and third impulses of a four impulse solution, the three loops on roll, pitch, and yaw angles would need to be duplicated inside the step size loop.

The inputs to the optimizer are the attitude and angular rates after the previous impulse, the velocity change requested by the current impulse, and the segment time of flight from the last impulse. All of this information is readily available at this point in the search. The references to 'all min' and 'all max' refer to the limits on the roll, pitch, and yaw loops. When the search begins, the limits are all from zero to three hundred sixty degrees with a step size of ninety degrees. As the search progresses, both the range and the step size of the attitude loops are changed. The new range of each of the loops is centered on the best roll, pitch, and yaw currently available. The maximum limit becomes the best current attitude plus the step size; the minimum limit becomes the best current attitude minus the step size. After the new limits are determined, the step size is cut in half. The attitude post optimizer terminates



all min = 0
all max = 360
step = 90
for roll₁ = roll min to roll max by step
for pitch₁ = pitch min to pitch max by step
for yaw₁ = yaw min to yaw max by step
roll 0
pitch 0
yaw 0
tof
RATE
-
angular rate 0
Δ angular rate
Δ velocity
tof
close
FUEL
Record best combination of roll, pitch, and yaw.
step > 1
yes
no
Return best combination of roll, pitch, and yaw.
step = step / 2
all max = best + step
all min = best - step

Figure 3.11: *Attitude Post Optimizer*

when the step size is less than one degree. The best combination of roll, pitch, and yaw angles is returned with the associated trajectory.

Contingency Planning

Although the trajectory planner assumes perfect implementation of the trajectories it creates, the planner can be used for real time contingency planning in the event of failures. Unlike the simpler strategies used previously, the maneuver need no longer be terminated if something unforeseen occurs; instead the danger can be evaluated and the maneuver can either be replanned or terminated depending on the severity of the problem. The decision to terminate or continue can be made by the Maneuver Manager or pilot based on the specific situation .

One possible failure is the loss of one or more jets during the maneuver. The effect of this failure on the maneuver is easily evaluated by simply checking whether or not these jets are used in any of the firing sequences. If they are not, the maneuver can continue and the jets will simply be recorded by the execution manager as unusable for future maneuvers. If these jets are used, two options are available to the manager which enable the maneuver to continue. The first option is to find other jet combinations which achieve the same rate request by calling the jet select algorithm with the failed jets disabled. Although this will not be as efficient as the original maneuver, it will likely be more efficient than aborting and reinitiating the maneuver. The second option is to propagate the current state forward, to allow for planning delays, and use this state as the initial condition for a new trajectory planned during that time. This option is more likely to be successful if the attitude angles are optimized because it can utilize attitude changes to favorably align working jets.

Another failure which can occur is a large deviation from the planned trajectory. This can result from jets failing to fire or turn off or improper modelling in the planner. If these deviations are sufficiently small, the autopilot will compensate for them during maneuver execution. If they are too large for the autopilot to correct, a new plan must be developed. Again it should be noted that if the vehicle has strayed so far that a collision is imminent, the replanning can be accomplished more quickly if only two burn solutions are considered.

Limitations

The algorithm outlined above has a few significant limitations. The first is that the gradient search on impulse positions assumes that the cost function has no local minima in which the search can be trapped; this is not necessarily true. Two cases are presented in this section to illustrate this algorithm limitation. Each case involves a maneuver relative to an orbital altitude of 176.7 miles with a 100.0 minute maximum time of flight. The first case has a maximum closing velocity of 10.0 ft/s and no obstacles. For the maneuver specifications in Table 3.1, the second impulse position search terminated at three different locations.

Parameter		Initial State	Final State
position	ft	(100, 200, 300)	(0, 0, 0)
velocity	ft/s	(1.00, 2.00, 3.00)	(0, 0, 0)
ϕ, θ, γ	deg	(180, 0, 0)	(270, 270, 270)
w_1, w_2, w_3	deg/s	(0.00, 0.00, 0.00)	(0.00, 0.00, 0.00)

Table 3.1 *Input for Local Minima Illustration, variable step size*

Where ϕ,θ,γ are the chase craft roll, pitch, and yaw angles (commanded in the order yaw, pitch, roll) with respect to the LVLH frame and w_1,w_2,w_3 are the angular rates of the chase craft body frame with respect to the LVLH frame. Notice that since the final angulare rates are zero, the vehicle will track the rotating LVLH frame at the end of the maneuver.

Three six dimensional searches with no limit cycling effects were run by the trajectory planner on the initial conditions above. In each case the spatial step size used in the gradient search was varied, while the time step was held at a constant twenty minutes. In each of these runs the algorithm returned a three impulse solution as the optimal trajectory; however, the positions of the second impulse were not the same (see table 3.2).

Spatial Step Size (ft)	Second Impulse Position (ft)	Total Fuel Use (lbm)
2	(-6, -2, -21)	154.0
4	(-8, -8, -15)	154.6
20	(-68,-140, -23)	155.1

Table 3.2 *Second Impulse Positions and Resulting Fuel Use*

The three runs in Table 3.2 illustrate a common problem with gradient searches, step size selection. If a step size is too small, the search might not find a global minimum because it is unable to search outside the cost function 'valley' it is currently in (when analyzing gradient techniques, it is often easier to visualize the path to a minimum as a valley in the cost function). If the step size is too large, the search might not even find the 'valley' which contains the global minimum. The cases with step sizes of two and four feet converge on essentially the same solution. It is unclear, however, whether the twenty foot step size case converged to a different local minimum or whether it was unable to follow the 'valley' which defined the minimum found in the other two cases all the way to its lowest point. In spite of the differences in second impulse position, the fuel use differed by less than a percent in all three cases.

The search could better avoid local minimum by incorporating a probablistic approach to impulse position acceptance, eg. simulated annealing. However, the additional heuristics required to implement this approach would greatly increase the run time of the planner. For this reason, no probablistic heuristics were incorporated in this algorithm. Another method of avoiding local minimum would be to change the spatial step size whenever the search finds a local minimum. This would allow the algorithm to identify new 'valleys' in the cost function or better follow the 'valley' it is currently in. Although this is not included in this algorithm, it would be a worthwhile area of future study.

Fortunately, the cost function for this particular problem has an interesting quality that helps the algorithm avoid local minima. If a reasonably low closing velocity constraint is placed on the maneuver, the 'valleys' in the cost function have much steeper 'walls', ie. the cost gradients are much greater. This reduces the risk that the gradient search will get caught in a local minimum (the case presented above had an unrealistically high maximum closing velocity). Since this is a logical constraint for proximity operations, it is not unreasonable to plan a maneuver subject to this constraint. A second case is presented to show this change in the cost function. This case has four spherical obstacles, two moving and two stationary (all obstacle initial conditions are listed in the appendix). The maneuver specifications are listed in Table 3.3.

Parameter		Initial State	Final State
position	ft	(1000, 1000, 1000)	(0, 0, 0)
velocity	ft/s	(1.00, 1.00, 1.00)	(0, 0, 0)
ϕ, θ, γ	deg	(180, 0, 0)	(270, 0, 270)
w_1, w_2, w_3	deg/s	(0.50, 0.50, 0.50)	(0.00, 0.00, 0.00)

Table 3.3 *Input for Local Minima Illustration, variable V_{close}*

Two six dimensional searches with no limit cycling effects were run by the trajectory planner on the initial conditions above, each with different closing velocity constraints. The spatial step size was held constant at one hundred feet, and the time step was held constant at twenty minutes. In each of these runs the algorithm returned a four impulse solution as the optimal trajectory; however, the positions of the second impulse were not the same (see table 3.4).

Maximum V_{close} (ft/s)	Second Impulse Position (ft)	Total Fuel Use (lbm)
1.00	(565, 242, 291)	201.1
0.05	(365, 342, 191)	189.6

Table 3.4 *Second Impulse Positions and Resulting Fuel Use*

Since both of the runs converged on the same third impulse position, only the second impulse positions are discussed here. In both runs the gradient search on the second impulse position begins at the same place. However, in the run where high closing velocity was allowed, the second impulse position gets caught in a local minimum some two hundred and fifty feet from a lower cost minimum. The run with the more constrained closing velocity converged to a lower cost minimum because it had a much better defined 'valley' in the cost function to follow. Since a small closing velocity is typical of most proximity operations, the algorithm is less likely to get caught in a local minimum.

A second limitation is that the gradient search follows the maximum negative gradient in the cost function of acceptable solutions. If an impulse position has a solution to the transfer which requires less fuel but does not meet all of the input constraints, the impulse position is abandoned by the search (see Figure 3.12).

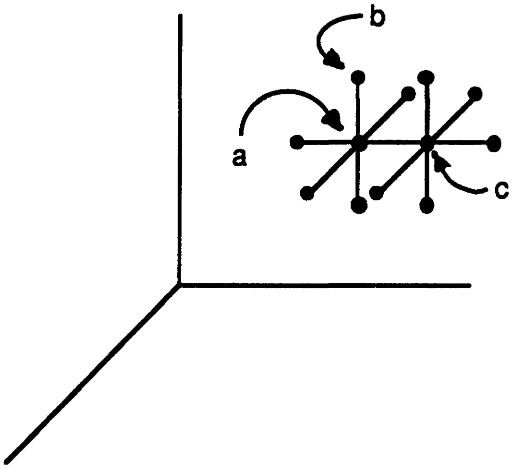


Figure 3.12: *Gradient Search Visualization, impulse position elimination*

In the figure, the gradient search investigates impulse positiōns adjacent to the current impulse position 'a'. The trajectory which requires the minimum fuel uses point 'b' for an intermediate impulse position. However, this trajectory does not meet the closing velocity constraint and it is abandoned in favor of the next best trajectory through point 'c' which does meet the constraint. This means that many impulse positions adjacent to the unacceptable position, point 'b', will not be investigated. This elimination of impulse positions associated with lower fuel use is viewed as a necessary evil to guarantee that the trajectory returned from the algorithm meets the closing velocity constraint.

The last significant limitation is that the computer code which executes the algorithm has not been optimized. Although the coded algorithm is not as efficient as it could be, it does demonstrate that constrained trajectories can be created between a chase state and a target state in a reasonable amount of time. The full algorithm can be run on a one MIPS (millions of operations per second) machine in five to ten minutes, depending on the closing velocity constraint. Further optimization of the code is left to the future.

Evaluation of the Trajectory Planner

Link to Shuttle Autopilot

Before any conclusions can be drawn from maneuvers created by the trajectory planner, the accuracy of the planning algorithm must be determined. This accuracy is determined with the aid of the Space Shuttle simulator at the Draper Laboratory. A given problem solution is evaluated as follows,

- Define a problem to be solved by the trajectory planner.
- Solve the problem with the trajectory planning algorithm and create a waypoint file which contains information about the trajectory at one minute intervals, including
  - position of the chase craft with respect to the target state.
  - velocity of the chase craft with respect to the target state.
  - attitude of the chase craft with respect to the LVLH frame.
  - angular rates of the chase craft with respect to body axes.
  - time.
  - flag indicating a commanded rate change.
- Execute the waypoint file on the Space Shuttle simulator, with the the commanded rate changes implemented impulsively.
- Compare the expected trajectory to the actual trajectory.

Notice that the rate changes are implemented impulsively. This is done to eliminate errors due to velocity changes being implemented in finite time rather than impulsively. As a result, the errors observed in the implementation of the waypoint file are almost entirely modelling errors. The modelling errors dominate for large transfer times, while non-impulsive burn effects dominate for large impulses, hence long firing times.

Twenty different maneuvers were executed successfully in this manner before the algorithm was considered functional; the results of two of these tests are presented here. Although only one optimum solution is output by the trajectory planner for each set of input conditions, four distinct types of solutions are created along the way: the two impulse solution, the two impulse solution with a delayed first impulse, the three impulse solution, and the four impulse solution. The first validation case illustrates the effects of modelling errors on each of these solution types. The second case shows the effects of nonimpulsive burns and how the associated errors can be eliminated by the closed loop autopilot. Both cases involve maneuvers relative to an orbital altitude of 176.7 miles with a 100.0 minute maximum time of flight. The first case has a maximum closing velocity of 1.0 ft/s and three stationary spherical obstacles. The maneuver specifications appear in Table 4.1.

Parameter		Initial State	Final State
position	f t	(100, -200, 300)	(0, 0, 0)
velocity	ft/s	(0.75, -1.25, 1.75)	(0, 0, 0)
ϕ, θ, γ	deg	(45, 135, 270)	(-80, 80, -80)
w_1,w_2,w_3	deg/s	(0.50, 0-.50, 0.50)	(0.00, 0.00, 0.00)

Table 4.1 *Validation Case One Input*

A six dimensional search with no limit cycling effects was run by the trajectory planner on the initial conditions above. When the resulting waypoint files were fed to the Shuttle simulator, the following terminal conditions were reached for each of the solution types (see Table 4.2).

Solution Type	Final Position (ft)	Final Attitude (deg)
2 impulse	(0.1, -0.2, -0.3)	(-80.0, 80.0, -79.9)
delayed, 2 impulse	(0.3, 0.5, -0.3)	(-80.1, 79.9, -80.0)
3 impulse	(0.4, -1.3, 0.1)	(-79.9, 80.1, -80.0)
4 impulse	(1.7, -1.6, 0.1)	(-79.9, 80.1, -80.0)

Table 4.2 *Validation Case One Terminal Conditions*

Notice that all of the solution types very accurately reproduced the desired attitude at the final state. The small errors in final attitude are caused by the truncation of the very small body rates. When these truncated rates are propagated over several thousands of seconds, the result is an attitude angle error on the order of a tenth of a degree.

The final positions, however, exhibit a slightly larger range of errors. The two impulse solutions reach the origin of the LVLH frame within half a foot while the four impulse solution misses the origin by just over two feet; the errors in final position get continuously worse as the number of impulses increases. This is because the intermediate impulses are being made at the correct times but at slightly different positions and attitudes than predicted by the Clohessy-Wiltshire equations. Since the rate change request is correct for the predicted chase craft position but not for the actual chase craft position, the position error is further increased by each impulse. As the magnitudes of both position and attitude errors are small, the linearized equations of motion used by the trajectory planner appear to be an excellent model for proximity operations.

The fuel use predicted by the trajectory planner and actual fuel use required in the simulation for each of the solution types is outlined below (see Table 4.3).

Solution Type	Fuel Use Predicted (lbm)	Fuel Use Actual (lbm)	Percent Error
2 impulse	95.21	95.20	0.01
delayed, 2 impulse	95.61	95.45	0.17
3 impulse	76.87	76.56	0.40
4 impulse	83.53	84:26	0.87

Table 4.3 *Validation Case One Fuel Use*

Again, the small difference between predicted and actual fuel use is caused by the impulse being made at a slightly different attitude than that predicted by the planner. For this reason the error increases with the number of impulses. Even with this source of error, the fuel use prediction is within a percent of the actual fuel use for every case. This is an important result because it implies that the solution obtained by the trajectory planner is based on an accurate fuel use model. If the errors in the model were much larger, it would be difficult to justify that the algorithm actually optimized fuel use because trajectories would be accepted or rejected based on incorrect fuel use predictions.

The second case is presented here to show the accuracy obtained when limit cycling effects are added during the transfer. For this case a maximum closing velocity of 0.25 ft/s is used and four spherical obstacles, two stationary and two moving, are placed in the maneuver space. The maneuver specifications appear in Table 4.1.

Parameter	Initial State	Final State
position ft	(500, 300, 500)	(0, 0, 0)
velocity ft/s	(1.50, 1.00, 0.50)	(0, 0, 0)
ϕ, θ, γ deg	(180, 0, 0)	(33, 90, 33)*
w_1, w_2, w_3 deg/s	(0.50, 0.25,-1.00)	(0.00, 0.00, 0.00)

Table 4.4 *Validation Case Two Input*

The final commanded attitude was actually (0,90,0); however, this is an equivalent attitude to both (33,90,33) and (18,90,18) which were returned by the algorithm.

A six dimensional search with limit cycling effects was run by the trajectory planner on the initial conditions above and a waypoint file was created for the most fuel efficient solution. This case was executed with real jets creating velocity changes nonimpulsively. When the resulting waypoint file was given to the Shuttle simulator, the following terminal conditions were reached (see Table 4.5).

Solution Type	Final Position (ft)	Final Attitude (deg)
3 burn	(-0.1, -0.2, -1.5)	(35.4, 89.3, 33.5)
4 burn	(-0.3, 0.8, -1.0)	(18.4, 88.9, 18.6)

Table 4.5 *Validation Case Two Terminal Conditions*

Only three and four burn solutions are shown in the table since no two burn solution could meet the closing velocity constraint.

Since the jets take a finite time to fire, the craft is both rotating and translating during the firing. As a result, the impulse does not occur entirely at the correct time, position, or attitude. The chase craft, therefore, departs from the planned trajectory during the period of firing. These 'doglegging' effects [1] are kept to a minimum by limiting the burn times to 4.96 seconds. In addition to nonimpulsive burn effects, a second source of error is quantization effects in the jets. Since the digital autopilot on the Shuttle will only change the status of a jet every eighty milliseconds, all rate changes must be created by a firing time equal to an integer multiple of eighty milliseconds. Therefore, the actual rate change created by the jets may not match the requested change.

Although the effect of these two sources of error is to make the vehicle depart from the planned trajectory, the chase craft still arrives at the desired final state. This is due to the closed loop autopilot which applies correction firings when the current state strays too far from the intended state. The cost of this correction is an increase in fuel use over the impulsive, open loop, optimal trajectory. This increased fuel use is a function of the size of the deadband; as the position and angular deadbands decrease, the fuel required to stay within these limits increases. For all of the cases presented in this thesis a constant set of deadbands was used (see the Appendix for these deadbands). The actual and predicted fuel uses are summarized in the table below (see Table 4.6).

Solution Type	Fuel Use Predicted (lbm)	Fuel Use Actual (lbm)	Percent Error
3 burn	210.18	210.46	0.13
4 burn	212.42	220.13	3.63

Table 4.6 *Validation Case Two Fuel Use*

As expected, the fuel use required for the closed loop, finite firing time, execution is much greater than would be predicted by open loop, impulsive, execution. Since the limit cycling effects are modelled as a linear function of time, the planner attempts to minimize the fuel use by decreasing the total time of flight. However, when the time of flight is decreased the magnitude of the forced firings increases. As a result, the times of flight for trajectories planned with limit cycling in the execution typically fall between a quarter and a half of an orbital period; while those planned without limit cycling effects have times of flight between a half and a full period. Notice that the error in the four burn fuel use prediction is much larger than in the open loop cases. This is because the correction for limit cycling effects is only of first order. Some of the fuel use predictions are high while others are low; the average of the prediction errors is approximately zero.

Open Loop Execution

Trajectories which are executed without state feedback and limit cycling are termed open loop; those executed with state feedback and limit cycling are termed closed loop. The next two sections discuss how unforeseen disturbances can affect the open and closed loop execution of a maneuver.

Certain assumptions affect the accuracy of the solution. The critical assumptions are:

-- All rate changes are implemented impulsively.
-- All rate changes are implemented correctly.
-- Orbital changes are caused only by the impulsive rate changes.
-- Gravity gradient torques are ignored.

In the open loop case presented in the previous section the simulator was altered to meet these assumptions. This was necessary to check the accuracy of the model. In this section, however, these assumptions will be checked one by one and the effects on the terminal conditions determined to show how effectively a maneuver manager could perform a maneuver without feedback control.

One set of initial conditions will be used here for all of the cases. As a result, all the terminal conditions and fuel use predictions of the planned trajectory will be the same; only the actual chase vehicle behavior will vary. A six degree of freedom search will be done on the set of initial conditions so that the Shuttle does not fly in a stable gravity gradient orientation during the entire maneuver. This case involves a maneuver relative to an orbital altitude of 176.7 miles with a 100.0 minute maximum time of flight. It has a maximum closing velocity of 0.05 ft/s and four spherical obstacles, two stationary and two moving. The maneuver specifications appear in Table 4.1.

Parameter		Initial State	Final State
position	ft	(1000, 1000, 1000)	(0, 0, 0)
velocity	ft/s	(1.00, 1.00, 1.00)	(0, 0, 0)
ϕ, θ, γ	deg	(180, 0, 0)	(-90, 0, -90)
w_1, w_2, w_3	deg/s	(0.50, 0.50, 0.50)	(0.00, 0.00, 0.00)

Table 4.7 *Input Case Used To Investigate External Acceleration Effects*

This is one of the cases presented in the Limitations section of Chapter 3.

This set of initial conditions was chosen because it stresses many of the algorithm capabilities. The chase craft is placed far from the target state with an initial velocity and angular rate that carries it farther away. Furthermore, the chase craft is constrained to complete the rendezvous in about one orbit without violating the closing velocity constraint. Finally, four obstacles are placed in close proximity to the target. As a result the algorithm chose a four impulse solution with a long transfer time, just over one orbital period. The trajectory chosen is ideal for showing the effect of each assumption listed above on final state since it has a few large rate changes and a long transfer time.

Four cases were executed to show the effect of each of the above assumptions. The first is a reference case executed with perfect jets, eg. jets which provide impulsive and precise rate changes, and no external disturbances. By comparing other cases to this reference, the contribution of each disturbance to the final state error can be determined. The second case uses perfect jets and includes gravity gradient effects. The third case is a perfect jet run with a nonspherical Earth (J_2 is added to the gravitational potential). The final case uses real jets, eg. jets which provide nonimpulsive and imprecise rate changes, and includes no external disturbances. The above cases produced the results in Table 4.8.

Invalid Assumption	Final Position (ft)	Final Attitude (deg)
none	(20, -38, 0)	(-90.0, 0.0, -90.1)
uniform gravity	(20, -38, 0)	(-71.1, -3.4, -165.0)
spherical Earth	(147, -266, -1)	(-91.0, 0.0, 90.1)
perfect jets	(8565, -16428, -84)	(-91.1, -8.7, -73.3)

Table 4.8 *Open Loop Terminal Conditions*

The solution executed with perfect jets and no disturbances missed by a greater distance than would be expected based on previous results. The error is forty three feet, an order of magnitude greater than in any of the previous cases. This miss distance can be explained by the modelling errors noted in the previous section; the modelling errors were simply aggravated by the initial conditions of this case. When the distance between the target state and the chase state increases, modelling errors cause the rate change requests returned from operation VEL to be less precise. This imprecise velocity vector results in position errors over time and the desired final state is not reached. In spite of this error, the final position and attitude provide a reference against which the other solutions can be compared. The reference trajectory is illustrated in Figure 4.1 with the Shuttle drawn at the final state. In this figure, the curve shows the path followed by the Shuttle, the small cylinders represent the obstacle positions at the final time, and the large cylinder represents the desired final position.

Since the acceleration due to gravity is a function of the radius from the attracting body, discrete components of the chase craft will experience different accelerations due to gravity depending on their distance from the central body; this can result in a torque on the body called a gravity gradient torque. The second solution takes gravity gradient effects into account. Since the velocity vector is defined through the center of mass of the orbiting craft, gravity gradient effects cannot affect the velocity vector; angular rates, however, can be altered. Therefore, when the Shuttle Simulator accounted for gravity gradient effects, the final position did not change but the final attitude changed by seventy five degrees in yaw and twenty degrees in roll.

The planner operates under the assumption that the central body is spherical. If the central body is the Earth, this assumption is not quite true. The Earth is actually an oblate spheroid with the largest deviation being a small extra 'band' of mass around the equator. This extra mass makes the acceleration due to gravity different at each position along an orbit inclined to the equator; this is the J_2 term in the gravitational expansion. Fortunately, this disturbance acceleration affects the velocity vector of both the chase craft and the target state similarly; with slight differences caused by variations in the eccentricity and inclination of the chase craft orbit. The addition of J_2 in the Shuttle simulation increased the final position error by two hundred and sixty feet from that achieved by the reference trajectory.

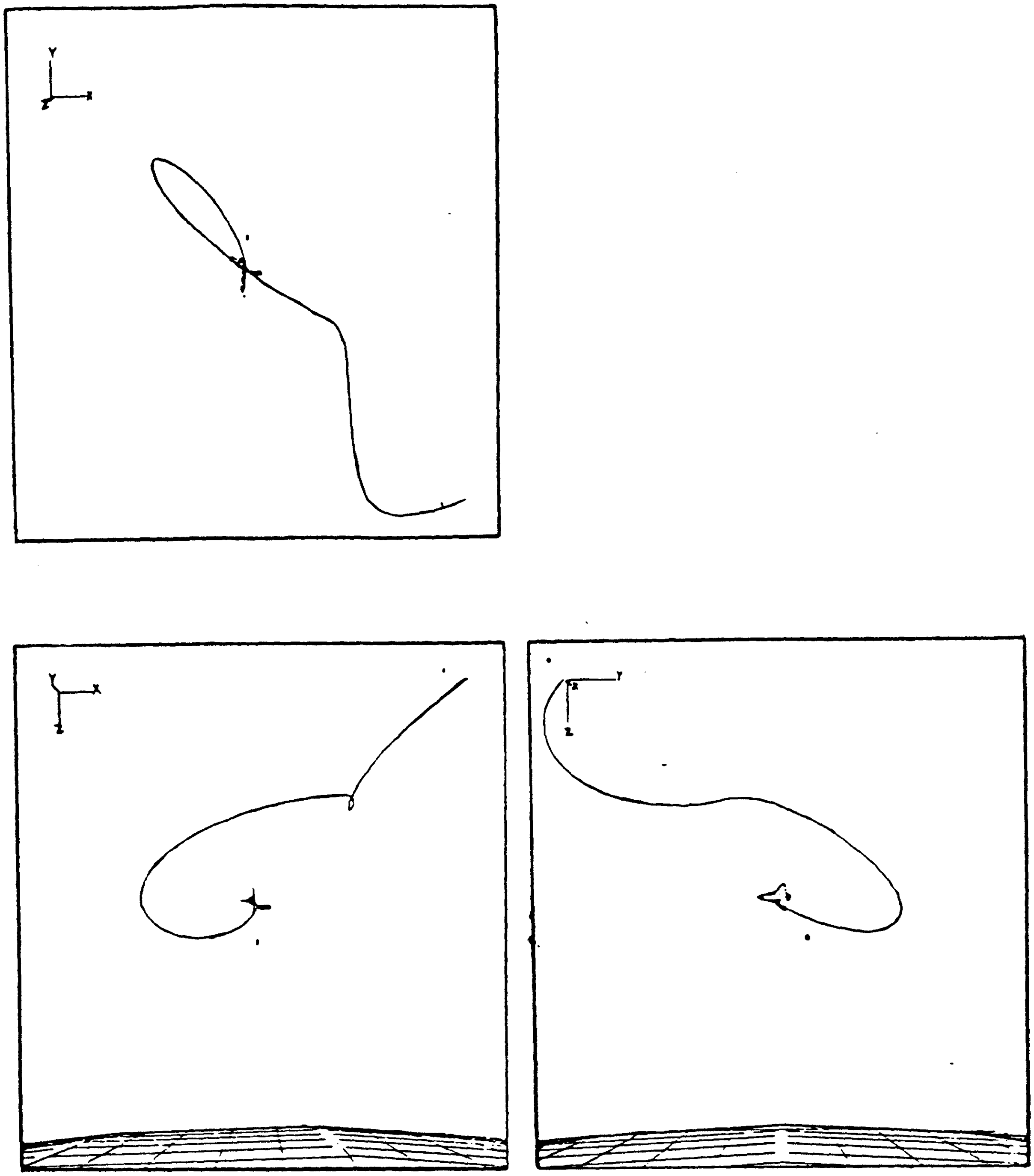


Figure 4.1 *Open Loop Reference Trajectory*

In the final case the maneuver is executed with a model of the Shuttle jets. This means that the rate changes occur over a finite period of time rather than impulsively. As a result, only a small portion of the rate change occurs at the correct position, attitude, and time. Since this violates one of the critical assumptions made by the algorithm, it is unlikely that the chase craft will approach the target state at the desired time. Furthermore, the real jets do not create precisely the desired rate change due to the autopilot quantization effects mentioned earlier. When the imprecise, nonimpulsive rate changes were propagated forward in time, the chase craft reached a position more than three and a half miles from the intended target.

Two of the cases mentioned above discussed unmodelled disturbances which changed the trajectory of the chase craft. Since both J_2 and gravity gradient disturbances are created by the gravity of the central body, their effects will decrease as the orbital radius increases. In the final case, however, the error was created by nonimpulsive and imprecise implementation of the rate changes by the chase craft's jets; this effect is independent of the orbital radius. The actual unmodelled disturbances on a chase craft will include the disturbances mentioned above and others such as atmospheric drag, third body accelerations, and failed actuators.

Since any one of the disturbances mentioned create a sizeable error in the final state, any maneuver planned by the algorithm would need to be executed with state feedback and limit cycling burns. The fuel penalty of doing so must be considered.

Closed Loop Execution

In an effort to keep the chase vehicle from straying too far from the planned trajectory, each of the cases above was executed using state feedback in the maneuver execution. The results are presented in this section to determine if the linearized equations of motion, with jet firings as the only external acceleration, provide an adequate model for the orbital dynamics. If the fuel required to correct for the modelling errors is too large, the errors might better be eliminated by changing the model to account for the external disturbances rather than simply relying on closed loop execution.

The four cases presented here are identical to those above except for the implementation of the rate changes. Only the first of the four cases uses perfect jets to create both the commanded rate changes and the limit cycling impulses.

The other three cases use real jets. Those cases executed with J_2 and gravity gradient effects actually show the fuel required to correct for both real jet and gravity related disturbances. When the waypoint file was given to the Shuttle Simulator with the appropriate disturbances, the following terminal conditions were reached (see Table 4.9).

Invalid Assumption	Final Position (ft)	Final Attitude (deg)
none	(-1.1, 0.6, -0.3)	(-90.1, 0.0, -90.0)
uniform gravity	(-2.1, -1.2, 3.1)	(-89.9, 0.3, -88.2)
spherical Earth	(1.4, 0.4, -0.5)	(-91.4, 0.5, -91.7)
perfect jets	(0.5, 0.0, -0.3)	(-92.0, 0.6, -88.5)

Table 4.9 *Closed Loop Terminal Conditions*

Each of the cases reached the desired final state within the limits set by the attitude and position deadbands. This means that although the model does not account for these external disturbances, the final state can still be achieved through closed loop control of the maneuver execution. The closed loop reference trajectory depicted in Figure 4.2 illustrates the increased accuracy with which the final state is achieved. Although this seems to justify the concept of planning with no disturbance accelerations and correcting for the errors in the execution, the cost associated with adding limit cycling has not been discussed. The fuel cost of adding the limit cycling burns is presented in Table 4.10.

Invalid Assumption	Open Loop Fuel Use (lbm)	Closed Loop Fuel Use (lbm)	Percent Difference
none	190.40	321.89	69.06
uniform gravity	212.27	335.90	58.24
spherical Earth	197.36	342.98	73.78
perfect jets	519.00	342.89	-33.93

Table 4.10 *Fuel Required To Correct For Unmodelled Accelerations*

Here the fuel cost associated with correcting for each of the disturbance accelerations is compared against the open loop maneuver cost. Notice that

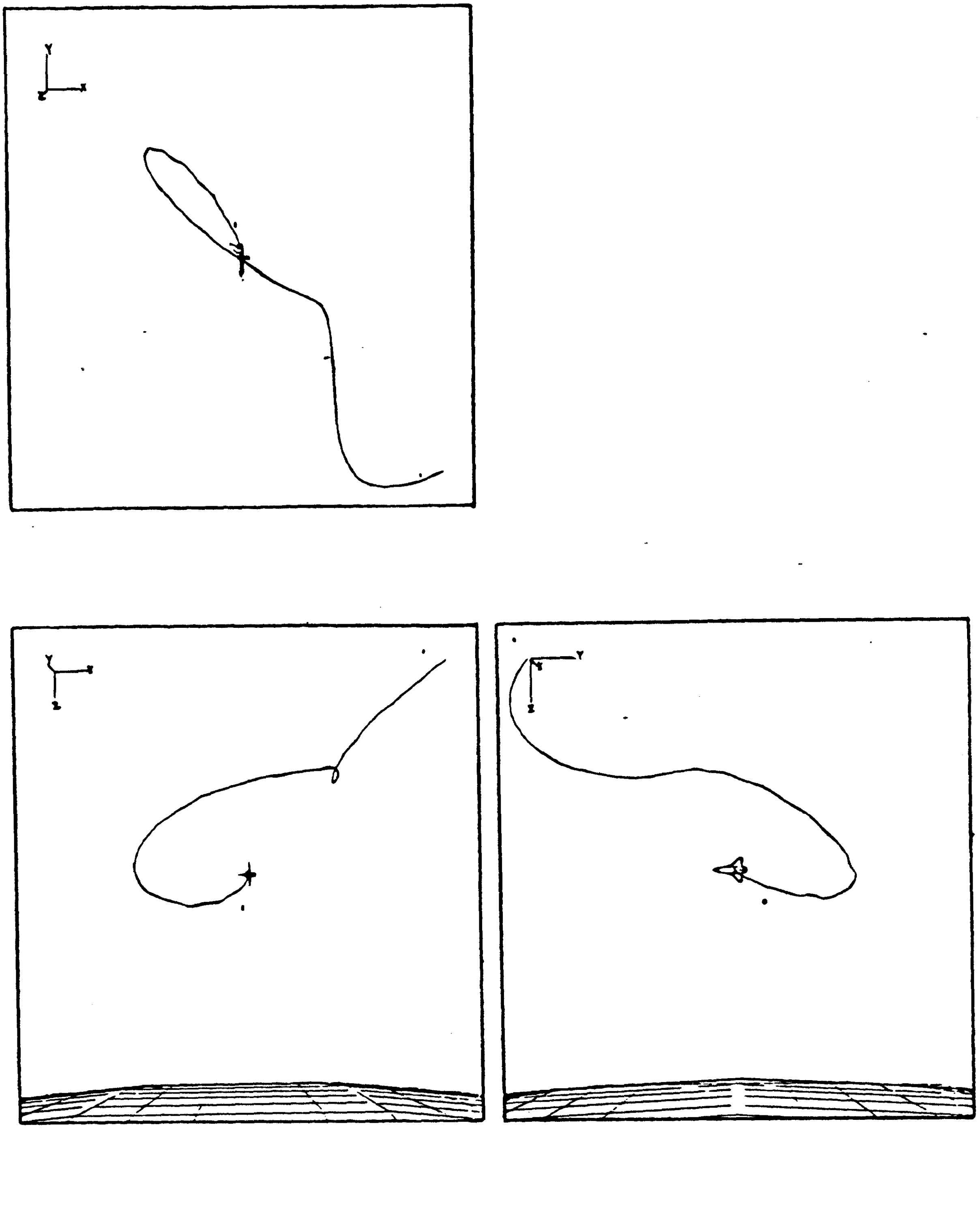


Figure 4.2 *Closed Loop Reference Trajectory*

the closed loop fuel use for each of these cases is essentially the same. This is because the majority of the fuel used in limit cycling is required to generate the rates which carry the chase vehicle from one deadband limit to the other. As a result, the fuel required for limit cycling is accurately modelled as a linear function of time, and not of the disturbances. The first of the cases in the table shows the additional fuel required to implement closed loop control with no external disturbances; it is roughly one hundred thirty pounds of fuel. No more than twenty one pounds of fuel were required beyond this to accommodate the addition of any of the external disturbances.

The fuel increased considerably when the maneuver was executed closed loop, in one case by as much as seventy three percent. However, this is not the minimum fuel, closed loop trajectory between the initial and final states. This trajectory was planned by the algorithm with the cost function that does not consider limit cycling effects. Since the time of flight is very large, so is the closed loop fuel cost. When the maneuver was replanned considering limit cycling effects in the cost function, the following fuel uses were determined (see Figure 4.11) .

Invalid Assumption	Open Loop Fuel Use (lbm)	Closed Loop Fuel Use (lbm)	Percent Difference
none	190.40	264.72	39.03
uniform gravity	212.27	269.82	27.11
spherical Earth	197.36	281.67	42.19
perfect jets	519.00	276.26	-46.77

Table 4.11 *Fuel Required To Correct For Unmodelled Accelerations*

When the planner developed a trajectory including the cost of closed loop control, it decreased the transfer time from ninety five minutes to forty three minutes. Although the forced firings were less efficient with this time of flight, the total fuel use decreased by roughly sixty pounds of fuel in each case due to the decreased limit cycling. Here the increased fuel required for closed loop execution is seventy four pounds in the case without disturbances, or approximately forty percent.

Although each of the disturbances discussed here are predictable and can be included in the model, many other external disturbances such as outgassing or jet failures cannot be predicted. The execution of the maneuver must, therefore, include closed loop control if the maneuver is to have any chance of success. Since the vast majority of the increased fuel use from closing the loop is caused by the limit cycling and not from correcting for the small disturbances, the fuel savings associated with correcting for the predictable disturbances in the model would be marginal. As these maneuvers are proximity operations and a small state error could cause a collision or violation of other constraints, this increased closed loop fuel use is a small price to pay to guarantee that the vehicle will accurately follow the planned trajectory to the target state.

Attitude Optimization

All of the six degree of freedom searches mentioned in this chapter used the attitude post optimizer. The purpose of this section is to give some insight into how much fuel can be saved by varying the angular rates throughout the maneuver in order to take advantage of jet coupling effects. It should be noted once again that the search on attitude is done after the burn positions and times are determined by the algorithm in order to limit the computation time. It is logical to assume that if post optimizing the angular rates can save fuel, optimizing these rates during the gradient search could save even more fuel [8]. This discussion, therefore, gives a conservative estimate of how much fuel can be saved by varying angular rates.

All of the cases discussed in this section were presented earlier in this chapter. The first two cases are those which were used in validating the model, their initial conditions are in Tables 4.1 and 4.4. The last two cases use the initial conditions for the open and closed loop execution discussions, located in Table 4.7. These last two cases differ in that the algorithm used the cost function without limit cycling effects in the first and with them in the second. Since none of these cases were executed on the Shuttle simulator with constant angular rates, ie. before the attitude post optimizer was called, all of the fuel costs discussed here are the costs predicted by the trajectory planner. The fuel costs for each of these cases, both before and after the call to the post optimizer, are presented in Table 4.12.

Case	no burns	Fuel Use, Constant **w** (lbm)	Fuel Use, Variable **w** (lbm)	Percent Difference
Validation 1	3	83.87	76.87	-8.35
	4	87.78	83.53	-4.84
Validation 2	3	223.76	210.18	-6.07
	4	253.05	212.42	-16.06
Open Loop	3	258.69	237.19	-8.31
	4	204.86	189.64	-7.43
Closed Loop	4	300.30	286.15	-4.71

Table 4.12 *Fuel Savings From Attitude Post Optimizer*

Only a four burn solution is presented for the closed loop case since the planner could find no three burn solution to the problem with the cost function which includes limit cycling.

By varying the angular rates at each of the commanded burn positions, the fuel required was reduced in each case; in one case it was reduced by over sixteen percent. Although one might expect greater fuel savings by allowing greater latitude in allowing rate changes, four burn solutions do not necessarily exhibit larger savings from attitude variations. Since fuel savings are observed in every trajectory after the attitude post optimizer is called, even greater fuel savings would be expected if the algorithm were upgraded to allow for angular rate variations during the gradient search on impulse positions. This is an area of continuing research.

Comparison to A* Method

Another algorithm for solving this proximity operation problem was developed simultaneously at the Draper Laboratory. This algorithm made use of a limited tree search known as the A* method [7]. A quick comparison between the A* algorithm and the gradient search algorithm is presented here to show that the two methods, although fundamentally different in their approach, converge to similar fuel use solutions.

There are a few notable differences between the two searches. The first is that the attitudes are fixed during the entire maneuver in the A* algorithm. Fuel expenditures required to keep the vehicle at this attitude are not considered. In the gradient search, an attitude hold can be commanded but the fuel required to do this will be considered. The second difference is that the A* search does not include limit cycling in the cost function. The result of these algorithm differences is that the fuel use returned by the two algorithms will not be the same; they should, however, be close to each other.

Two cases are used to compare the algorithms; one with open loop execution and one with closed loop execution. Two of the input constraints required by the gradient search are not considered by the A* algorithm, maximum time of flight and maximum closing velocity. To keep the input to each algorithm roughly equal, these two values were intentionally kept large in the gradient search. Both cases involve maneuvers relative to an orbital altitude of 176.7 miles with a 100.0 minute maximum time of flight. Both cases also have a maximum closing velocity of 1.0 ft/s and one moving spherical obstacle. The maneuver specifications of the first case appear in Table 4.13.

Parameter		Initial State	Final State
position	ft	(0, 5000, -1000)	(0, 0, 0)
velocity	ft/s	(0.00, 0.00, 0.00)	(0, 0, 0)
ϕ, θ, γ	deg	(0, 0, 0)	(0, 0, 0)
w_1, w_2, w_3	deg/s	(0.00, 0.00, 0.00)	(0.00, 0.00, 0.00)

Table 4.13 *Comparison of A* and Gradient Search Input 1*

The angular rates are zero for all time when the three dimensional search is used. Otherwise, the angular rates are allowed to vary as long as the final state is reached.

The second case discussed here is the closed loop execution case; the maneuver specifications appear in Table 4.14.

Parameter		Initial State	Final State
position	ft	(3000, 0, 0)	(0, 0, 0)
velocity	ft/s	(0.00, -5.22, 0.00)	(0, 0, 0)
ϕ, θ, γ	deg	(0, 0, 0)	(0, 0, 0)
w_1, w_2, w_3	deg/s	(0.00, 0.00, 0.00)	(0.00, 0.00, 0.00)

Table 4.14 *Comparison of A* and Gradient Search Input 2*

Again, the angular rates are zero for all time when the three dimensional search is used. Otherwise, the angular rates are allowed to vary as long as the final state is reached.

A trajectory for both cases was created by each algorithm. However, both three and six degree of freedom searches were executed by the gradient search algorithm while only a three degree of freedom search was executed by the A* algorithm (it cannot vary attitude angles). The resulting fuel uses are summarized in Table 4.15.

Case	DOF	A* Search Fuel Use (lbm)	Gradient Search Fuel Use (lbm)	Difference (lbm)
Open Loop	3	50.9	70.6	19.7
	6	----	60.2	9.3
Closed Loop	3	174.2	193.4	19.2
	6	----	185.1	10.9

Table 4.15 *Fuel Use Comparison of A* and Gradient Methods*

In each case, the fuel cost returned by the gradient search was slightly higher than the cost returned by the A* algorithm. This comes as no surprise when the basic differences between the methods are considered. The A* search does not account for the amount of fuel required to maintain the LVLH aligned attitude. At each of the forced firings, a slight couple is created which excites a change in the angular rates. This angular rate change is a jet coupling effect and is created even if a pure translational rate change is requested. In the A* algorithm, the change in angular rate is ignored and the attitude is set to LVLH alignment for all time. In the gradient search algorithm, however, a second couple in the opposite direction is commanded so that the attitude rates do not change; to create this second couple, jets must be fired. It is the additional jet firings required by the gradient search algorithm that account for the majority of the fuel use differences. The observation that both algorithms use completely different methods to create trajectories, yet converge on similar fuel use solutions, increases confidence in the optimality of each solution.

Conclusions

Capabilities and Efficiency of Algorithm

As the complexity of proximity operations in space continues to increase along with the precision with which they must be performed, the need for an autonomous system to plan and execute these maneuvers becomes more urgent. One of the necessary pieces of any autonomous system is the Trajectory Planner discussed in this thesis.

The equations of motion used in implementing the Trajectory Planner are the Clohessy-Wiltshire equations for translation and quaternions and angular rates for attitudes. They proved to be highly accurate for achieving a desired target state in the absence of external disturbances. The slight errors, on the order of five feet, in the translational equations of motion are a function of both the initial separation of the two vehicles and the maneuver time of flight. As the initial distance and time of flight increase, the error in the final state can increase by an order of magnitude in the absence of closed loop control.

When disturbance accelerations were introduced during maneuver execution, the error in the final state increased by up to three orders of magnitude for executions without limit cycling. In fact, none of the cases implemented without closed loop control and with disturbance accelerations, gravity gradient , J_2, and imperfect jets had an acceptable error in the final state. Therefore, the maneuver can be planned assuming no external accelerations but the execution must include closed loop control. The associated increase in fuel cost is roughly forty percent in each case investigated, regardless of the disturbances. The reduction in fuel use which could be obtained by adding the periodic gravitational disturbances to the model is therefore marginal.

The algorithm includes a post optimizer on the attitude at each of the commanded impulse positions. This is done to take advantage of both jet coupling effects to create desired angular rates and to better align the chase craft in directions in which it has more control authority at each impulse position. By simply commanding the angular rates required to achieve the attitudes returned by the algorithm, an average of eight percent reduction in fuel use was realized.

Since the algorithm described in this thesis uses a steepest descent gradient search, it is prone to getting stuck in local minima as in any other gradient descent method. By careful selection of the time and spatial step sizes as well as the closing velocity constraint, the algorithm can avoid converging on a poor local minimum. To help illustrate this point, cases were presented in which trajectories were created by an A* search trajectory planner and the gradient search trajectory planner. Although the two methods converged on slightly different fuel use solutions, the discrepancy is easily accounted for by a difference in the cost functions used by the algorithms. The similarity in these results lends credibility to the claim that both algorithms create fuel use optimal trajectories.

The OEX autopilot for the Shuttle has been used as an execution manager for the closed loop cases. This means that the software for two of the three autonomous system elements already exist for the Space Shuttle. The software for Maneuver Manager, the third element of the system, is currently being developed at Draper Laboratory.

Future Development

Several areas of future research have been suggested throughout this thesis. However, expanding the algorithm to accommodate any of these would greatly increase the execution time. Therefore, before any of the areas of future development are investigated, it is suggested that the practical aspect of optimizing the algorithm software is considered. By optimizing the algorithm software first, a great deal more research can be accomplished in a given amount of time.

The first area of suggested development is the incorporation of variable step sizes in the gradient search. The addition of variable time and spatial step sizes in the search could help the algorithm to converge to a lower fuel use solution by avoiding poor local minima in the cost function. The logic associated with varying the step size stems from the fact that small step sizes are desired to reach the lowest point in a cost function 'valley', while large step sizes are desired to find other, possibly lower cost, 'valleys.' The process should be to use small step sizes until the gradient is positive in all directions then increase the step sizes to find a new lower cost 'valley'. The search would terminate when the algorithm could no longer find a valley with a negative gradient from the current cost.

The second area of suggested development is the addition of constraints representing jet failures and plume impingement. The jet failure constraint is obvious. If a jet fails during the execution, either on or off, it should not be considered as active by the jet select algorithm. The plume impingement constraint is similar in that jets which will plume the target state should not be considered by the jet select algorithm. Imposing either constraint, therefore, has the same effect on the jet select algorithm; the jets which are disabled cannot be called upon to create a rate change. In adding these constraints, development should concentrate on deciding which jets must be disabled to avoid plume impingement as a function of current attitude angle. Once the determination of which jets to disable has been made, the available jets are passed to the jet select and the rest of the algorithm functions as before. This change, although simple, would greatly increase the usefulness of the algorithm. It should be noted that if this change is made in conjunction with adding attitude optimization during the impulse position gradient search, a great deal of fuel could be saved for the failed jet cases.

The last suggestion for future development promises to create the largest fuel savings; unfortunately, it is also the hardest to implement. This suggestion is that the attitude angles be optimized during the impulse position gradient search. Since the attitude post optimizer was written simply to prove that total fuel use could be reduced by varying the attitude at each impulse position, it is not very efficient. Therefore, a more efficient method of varying the attitude angles than the one illustrated in Figure 3.11 would have to be developed before this change could be practically implemented.

This is by no means an inclusive list of future development topics. Many practical considerations associated with proximity operations have not even been mentioned such as maximum allowable accelerations, which are especially important for maneuvers involving flexible structures, and changes in vehicle systems status which govern the decision to continue or abort the maneuver. These practical considerations lead to the development of the final autonomous system component, the Maneuver Manager.

References

[1] Bergmann, E. and P. Weiler. "Accomodation of Practical Constraints by a Linear Programming Jet Select." AIAA Guidance, Navigation, and Control Conference, 1983.

[2] Bergmann, E. et al. "An Advanced Spacecraft Autopilot Concept." *Journal of Guidance and Control.* Vol 2, No3, May-June 1979.

[3] Clohessy, W. H. and Wiltshire R.S. "Terminal Guadance Systems for Satellite Rendeyvous." *Journal of the Aerospace Science* . September 1980: 653-658,674.

[4] Hill, George W. "Researches in the Lunar Theory." *American Journal of Mathematics.* Vol 1, No. 1, 1878: 5-26.

[5].Leonard, Carolina Lee. *Formationkeeping of Spacecraft Via Differential Drag.* Masters of Science Thesis, M.I.T., July 1986.

[6] McKern, Richard A. *A Study of Transformation Algorithms For Use in a Digital Computer.* Masters of Science Thesis, M.I.T., January 1968.

[7] Niiya, Craig K. *An Application of the A* Search to Trajectory Optimization.* Masters of Science Thesis, M.I.T., June 1990.

[8] Redding, D.C., B. A. Persson, and E.V. Bergmann. "Combined Solution of Spacecraft Rotational and Translational Maneuvers." AIAA Guidance, Navigation and Control Conference, 1986.

[9] Wheelon, Albert D. "Midcourse Terminal Guidance." In *Space Technology.* Ed. Seifert, Howard. New York: John Wiley & Sons, 1959.

Appendix

Additional Information for Test Cases

Each of the cases executed in this program require additional input that was not presented in the main body of this thesis. These include the obstacle positions and velocities as well as the step sizes used in the algorithm execution. The step sizes are summarized in Table A.1,

Case	2 Burn Solution	3 Burn Solution		4 Burn Solution	
Description	Δt (min)	Δt (min)	Δx (ft)	Δt (min)	Δx (ft)
Local Min, Variable Δx	2	20	var	20	var
Local Min, Variable V_{close}	2	20	100	20	100
Validation 1	2	10	20	20	20
Validation 2	2	20	4	20	20
External Accelerations	2	20	100	20	100
A* comparison 1	2	20	100	20	100
A* comparison 2	2	20	50	20	50

Table A.1: *Step Sizes for Sample Cases*

Most of the spatial step sizes were chosen based on experience gained from solving similar maneuvers. For an algorithm which uses constant step sizes, these are good choices for spatial steps. However, the large time step sizes were chosen to minimize computation time. Therefore, lower fuel use solutions could have been determined if the time steps were decreased. To minimize the effect on fuel use of a large time step, the time step is decreased to two minutes for the three burn solutions and four minutes for the four burn solutions to post optimize the solution returned by the gradient search on impulse position.

Each case presented in the main body had a different set of obstacles in the maneuver space. These obstacle sets are summarized in Table A.2.

Case Description	Initial Position (ft)	Initial Velocity (ft/sec)
Local Min, Variable Δx	none	none
Local Min, Variable V_{close}	(1040, 890, 1030)	(0.00, 0.00, 0.00)
	(-100, -20, -100)	(0.00, 0.00, 0.00)
	(400, -200, -500)	(0.05, -0.90, 0.03)
	(-750, 600, 100)	(0.53, 1.60, -0.33)
Validation 1	(150, 380, 480)	(0.00, 0.00, 0.00)
	(-50, -750, -70)	(0.00, 0.00, 0.00)
	(-60, -20, -170)	(0.00, 0.00, 0.00)
Validation 2	(300, 30, 430)	(0.00, 0.00, 0.00)
	(50, -50, 160)	(0.00, 0.00, 0.00)
	(100, -200, 300)	(-0.34, 0.00, -0.14)
	(-50, 100, -50)	(0.22, -0.08, 0.05)
External Accelerations	(1040, 890, 1030)	(0.00, 0.00, 0.00)
	(-100, -20, -100)	(0.00, 0.00, 0.00)
	(400, -200, -500)	(0.05, -0.90, 0.03)
	(-750, 600, 100)	(0.53, 1.60, -0.33)
A* comparison 1	(0, 4000, 500)	(-0.21, 0.48, 0.46)
A* comparison 2	(0, -5000, 500)	(0.07, 0.06, 0.41)

Table A.2: *Initial Obstacle States*

The initial states are given with respect to the rotating LVLH reference frame. All of the obstacles, except those in validation case one, can be encompassed by a sphere of radius ten feet located at their center of mass. In the first validation case, the first obstacle has a radius of twenty feet and the other two have radii of thirty feet. The obstacles with zero initial velocity have their position fixed for all time with respect to the target state. Obstacles with nonzero initial velocity are subject to the same central body force as the chase craft and target state; these obstacles do not fire jets to change their velocities. The initial velocity of each moving obstacle is approximately what is required to rendezvous with the target state.

When executing the trajectories, several parameters were set by the OEX autopilot which affected the closed loop maneuver cost. These parameters included position and attitude deadbands, maximum firing times of the jets allowed for each rate change, and the minimum cycle rate of the jets. Although each of these parameters was mentioned briefly in the main body of the thesis, the specific numbers were often omitted. These parameters were set to be

position dead band	=	5.00	degrees
attitude dead band	=	3.00	degrees
limit cycling constant (derived from deadbands)	=	0.032	lbm$_{fuel}$/sec
maximum on time	=	4.96	seconds
minimum cycle time	=	0.080	seconds

The position and attitude dead bands define limits on the magnitude of the state error. Therefore, these limits define a region around the nominal trajectory in which the vehicle is allowed to translate and rotate. The limit cycling constant, Γ, was empirically determined by commanding the Shuttle to fly along twenty six different vectors while maintaining a constant attitude. These vectors were defined from the center of a unit cube to each corner, the midpoint of each edge, and the center of each face. The average fuel use per second was defined as the limit cycling constant. Since this constant is a function of both dead banding limits and vehicle jet configuration, it is correct only for the Space Shuttle with this particular set of limits.

The last two parameters define how the the OEX autopilot achieves the desired rate changes. Although the jet select algorithm can command a jet on time of any duration, it is constrained to a maximum of 4.96 seconds. At the end of the firing, a subsequent velocity change including both the remainder of the requested velocity change plus a component to correct the accumulated position error is implemented. By doing this the 'doglegging' effects are minimized during large rate changes. The minimum cycle time of the jets is merely a relic of the digital autopilot. Since the current state is updated by the autopilot every eighty milliseconds, the jet status is changed to correct any state errors or to command burns only at these times.