T-628

A NEW SPACECRAFT AUTOPILOT

by

EDWARD VINCENT BERGMANN

B.S., Boston University

(1974)

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May, 1976

Signature of Author

Certified by

Accepted by

Department of Aeronautics and Astronautics

Thesis Supervisor

Chairman, Departmental Graduate Committee

Aero  

OCT 1 1976
A NEW SPACECRAFT AUTOPILOT

by

EDWARD VINCENT BERGMANN

Submitted to the Department of Aeronautics and Astronautics on May 12, 1976, in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

An autopilot is developed for rotation and translation control of a rigid spacecraft of arbitrary design, using reaction control jets as control effectors. The autopilot incorporates a six-dimensional phase space control law, and a linear programming algorithm for jet selection. The interaction of the control law and jet selection are investigated and a recommended configuration proposed. Simulations are performed to verify the performance of the new autopilot and comparisons are made with an existing spacecraft autopilot. The new autopilot is shown to require 35.4% fewer words of core memory, 20.5% less average CPU time, up to 65% fewer firings, and consume up to 25.7% less propellant for the cases tested. However, the time required to perform the jet selection computations may render the new autopilot unsuitable for existing flight computer applications, without modifications. Finally, the new autopilot is shown to be capable of performing attitude control in the presence of a large number of jet failures.

Thesis Supervisor: Dr. Donald C. Fraser
Title: Lecturer in Aeronautics and Astronautics
Director, Control and Flight Dynamics Division, Charles Stark Draper Laboratory
ACKNOWLEDGEMENTS

I wish to express my gratitude to Dr. Steven R. Croopnick of the Charles Stark Draper Laboratory, who originated the phase space control law, for numerous constructive discussions in the course of this research, and editorial assistance in preparing this report. I also wish to thank Dr. Donald C. Fraser of the Charles Stark Draper Laboratory for serving as my thesis supervisor, for his constructive advice, and for providing the opportunity to conduct this research. I would like to thank Mr. Craig C. Work of the Laboratory for allowing me to work with his implementation of the optimal linear jet selection algorithm and for the assistance and guidance he has given me throughout this work.

Dr. Bard S. Crawford is to be commended for his doctoral research into the jet selection problem, which was the basis of the optimal linear jet selection algorithm. Also to be commended is Mr. John Turkovich, whose initial implementation of the phase space control law was the starting point for the control law development in this thesis.

The Technical Publications Group prepared the figures included in this thesis. Their cooperation is appreciated.

Finally, I wish to thank my wife, Karen, for typing the final manuscript, and for her encouragement and support throughout this research.

This report was prepared under contract NAS 9-13809 with the National Aeronautics and Space Administration.
Publication of this report does not constitute approval by the Charles Stark Draper Laboratory or NASA of the findings or conclusions contained herein. It is published solely for the exchange of ideas.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>12</td>
</tr>
<tr>
<td>1.1</td>
<td>A Perspective on Spacecraft Attitude Control</td>
<td>12</td>
</tr>
<tr>
<td>1.2</td>
<td>History of Spacecraft Attitude Control</td>
<td>13</td>
</tr>
<tr>
<td>1.3</td>
<td>Summary of Contents</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>FUNDAMENTALS OF AUTOPILOT DESIGN</td>
<td>18</td>
</tr>
<tr>
<td>2.1</td>
<td>Spacecraft Dynamics</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>Control Effectors</td>
<td>32</td>
</tr>
<tr>
<td>2.3</td>
<td>Digital Spacecraft Control</td>
<td>37</td>
</tr>
<tr>
<td>2.4</td>
<td>Existing Attitude Control Laws</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>THE PHASE SPACE CONTROL LAW</td>
<td>50</td>
</tr>
<tr>
<td>3.1</td>
<td>The Velocity to be Gained Principle</td>
<td>50</td>
</tr>
<tr>
<td>3.2</td>
<td>The Phase Sphere Concept</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>The Phase Space Control Law</td>
<td>54</td>
</tr>
<tr>
<td>3.4</td>
<td>Application to Spacecraft Control</td>
<td>66</td>
</tr>
<tr>
<td>4</td>
<td>JET SELECTION</td>
<td>69</td>
</tr>
<tr>
<td>4.1</td>
<td>Table Lookup Jet Selection</td>
<td>69</td>
</tr>
<tr>
<td>4.2</td>
<td>Dot Product Jet Selection</td>
<td>71</td>
</tr>
<tr>
<td>4.3</td>
<td>Jet Selection by Linear Programming</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>THE NEW AUTOPILOT</td>
<td>90</td>
</tr>
<tr>
<td>5.1</td>
<td>The Supervisor</td>
<td>92</td>
</tr>
<tr>
<td>5.2</td>
<td>Operation of the New Autopilot</td>
<td>94</td>
</tr>
<tr>
<td>5.3</td>
<td>Implementation of the New Autopilot</td>
<td>95</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6</td>
<td>TESTING THE NEW AUTOPILOT</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>6.1 Constant Angular Rate</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>6.2 Attitude Maneuver</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>6.3 Jet Failures</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>6.4 Position Maneuver</td>
<td>107</td>
</tr>
<tr>
<td>7</td>
<td>TEST RESULTS AND COMPARISON</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>7.1 Fuel Usage and Firings</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>7.2 Computer Burden</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>7.3 Jet Failures</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>7.4 Accuracy</td>
<td>112</td>
</tr>
<tr>
<td>8</td>
<td>CONCLUSIONS, CONTRIBUTIONS, AND RECOMMENDATIONS</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>8.1 Conclusions</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>8.2 Contributions</td>
<td>115</td>
</tr>
<tr>
<td></td>
<td>8.3 Recommendations</td>
<td>116</td>
</tr>
</tbody>
</table>

Appendix

| A       | CONDITION 2 FOR PERFORMING A NEW JET SELECTION          | 117  |
|         | WHEN JETS ARE FIRING                                    |      |
| B       | EXAMPLE AUTOPILOT AND JET SELECT COMPUTATION            | 119  |
| C       | DETAILED DISCUSSION OF ATTITUDE MANEUVER SIMULATIONS    | 128  |
|         | LIST OF REFERENCES                                      | 153  |
LIST OF SYMBOLS

\( a \quad \) scalar parameter in definition of convex set
\( \overline{a} \quad \) linear acceleration with respect to inertial space
\( a_c \quad \) control acceleration
\( a_i \quad \) linear coefficients in optimal solution
\( a_j \quad \) acceleration (both linear and angular) due to jet \( j \) -- body coordinates
\( \overline{a}_d \quad \) disturbance acceleration
\( \overline{a}_{\text{desired}} \quad \) desired acceleration
\( a_{\text{min}} \quad \) minimum component of control acceleration
\( b \quad \) difference in radius of inner and outer phase spheres
\( b_i \quad \) basis vector
\( B \quad \) basis, a matrix whose columns are the basis vectors
\( c \quad \) convergence rate
\( c_i, c_2 \quad \) boundary conditions on double integrator problem
\( c_i \quad \) jet firing command for jet \( i \) (Section 4.2); objective function coefficient (Section 4.3)
\( C \quad \) limit cycle speed
\( C_k \quad \) limit cycle speed
\( c_k \quad \) acceleration due to jet cluster \( k \)
\( db^+, db^- \quad \) positive and negative deadband
\( db \quad \) minimum necessary benefit to justify further iterations in simplex algorithm
\( db_i \quad \) deadband on \( i \)th element of state vector
\( dm \quad \) mass element
\( D \quad \) objective dot product
\( e \quad \) weighting factor in jet select cost function
E attitude error
E angular rate error
E\textsubscript{i} external forces
f(x) objective function in linear programming (may be vector)
f\textsubscript{i} weighting factor for jet i
f\textsubscript{i}(x) set of linear functions defining a set
F net applied force
F(u) minimum fuel cost function
F\textsubscript{ap} applied control force along an axis
F\textsubscript{i} force applied to particle i
g weighting factor in jet select cost function
G parameter indicating tolerable deviation between extrapolated and measured jet select performance
h\textsubscript{i} moment of momentum of particle i
H angular momentum of group of particles
H Hamiltonian function
i\textsubscript{d} unit vector along disturbance acceleration
I inertia tensor
I\textsubscript{i} principle moment of inertia about axis i
i\textsubscript{j} unit vector along the j\textsuperscript{th} coordinate axis
J\textsubscript{ij} cross product of inertia, axes ij
m mass of body or vehicle
m\textsubscript{i} mass of particle i
M\textsubscript{ap} applied control torque about an axis
n number of particles in a system or body
P\textsubscript{i} linear momentum of particle i
P\textsubscript{1, 2} costate variables of double integrator problem
P\textsubscript{m} missile position (Chapter 2)
P\textsubscript{t} target position (Chapter 2)
P\textsubscript{0} constraint constants
P\textsubscript{F} equivalent point of application of resultant force
\begin{align*}
\text{\(R_i\)} & \quad \text{displacement of particle } i \\
\text{\(r_{ps}\)} & \quad \text{phase sphere radius} \\
\text{\(R_{cm}\)} & \quad \text{position of mass center} \\
\text{\(R_1, R_2, R_3, R_4\)} & \quad \text{state plane of phase plane regions} \\
\text{\(S_i\)} & \quad \text{internal forces} \\
\text{\(S_{ij}\)} & \quad \text{internal force on particle } i \text{ due to particle } j \\
\text{\(t, S_1, S_2, S_3, S_4\)} & \quad \text{phase plane switching curves} \\
\text{\(t\)} & \quad \text{time} \\
\text{\(t_1\)} & \quad \text{time to travel from initial state to reversal} \\
\text{\(t_2\)} & \quad \text{time to travel from reversal to phase sphere} \\
\text{\(t_c\)} & \quad \text{cycle period} \\
\text{\(T\)} & \quad \text{final time in double integrator problem} \\
\text{\(T_j\)} & \quad \text{thrust vector of jet } j \\
\text{\(u(t)\)} & \quad \text{control} \\
\text{\(v(t)\)} & \quad \text{magnitude of optimal control} \\
\text{\(v_{\text{desired}}\)} & \quad \text{desired velocity} \\
\text{\(v\)} & \quad \text{linear velocity} \\
\text{\(v_g\)} & \quad \text{velocity to be gained} \\
\text{\(v_i\)} & \quad \text{velocity of particle } i \\
\text{\(v_i\)} & \quad \text{velocity along } i^{\text{th}} \text{ axis} \\
\text{\(v_n\)} & \quad \text{current velocity} \\
\text{\(v_T^+\)} & \quad \text{set of allowable control magnitudes} \\
\text{\(v\)} & \quad \text{magnitude of velocity} \\
\text{\(W\)} & \quad \text{rate change request (Chapter 4)} \\
\text{\(x_1, x_2\)} & \quad \text{state variables of double integrator} \\
\text{\(x\)} & \quad \text{state, position (in context) (Chapter 3); solution of} \\
& \quad \text{linear programming problem (Chapter 4) or jet firing times} \\
& \quad \text{Chapter 4)} \\
\text{\(X_B\)} & \quad \text{state at point } B \\
\text{\(\dot{X}_B\)} & \quad \text{rate at point } B
\end{align*}
\( x_c \) state at point c
\( \dot{x}_c \) rate at point c
\( x_d \) desired state
\( \dot{x}_d \) desired rate
\( xe \) external point
\( xe \) state error
\( \tilde{xe} \) normalized state error
\( \tilde{xe} \) "translated" state error
\( \ddot{xe} \) rate error
\( \dot{x}_e \) rate to be gained
\( \dot{x}_i \) initial rate
\( \dot{x}_0 \) optimal solution
\( y(t) \) response
\( y_k \) coefficients of representation of \( x_k \) in current basis
\( (z_i - c_i) \) saving or improvement in cost function
\( \beta \) displacement of trajectory reversal point from origin parallel to disturbance
\( \gamma^+, \gamma^- \) fuel optimal switch curves for double integrator problem
\( \delta^+, \delta^- \) switch curve offsets
\( \delta_i \) integer denoting jet is operative
\( \delta\phi_j \) virtual angular displacement of particle j
\( \delta\xi_i \) virtual linear displacement of particle i
\( \theta(i) \) feasible solution of linear programming problem
\( \delta_k^+ \) unknown positive scalar
\( \theta, \phi, \psi \) Euler angles
\( \xi \) angular displacement about an axis
\( \omega \) angular rate
\( \Omega \) angular velocity of reference frame with respect to inertial space
\( (\ )_i \) denotes \( i^{th} \) component or rotation about \( i \) (where noted)
\[ m f(\cdot) \frac{dm}{dt} \] integration with respect to mass

\[ \sum_{i=1}^{n} \] summation over the range 1, n

\[ (\cdot)' \] differentiation with respect to time

\( (\cdot)', (\cdot)', (\cdot)'' \) quantity referred to transformed reference systems

\[ (\cdot)(i) \] denotes position in basis

\[ (\cdot)e \] extreme point

\[ (\cdot)m \] maximum

\[ (\cdot)\text{min} \] minimum

\[ (\cdot)_{0} \] initial

\[ (\cdot)f \] final

\[ (\cdot) \] measured

\[ (\cdot)_{\text{desired}} \] desired

\[ (\cdot) \] scaled or transformed

\[ (\cdot)_{g} \] to be gained

\[ (\cdot)_{\Delta} \] change in

\[ (\cdot)_{t\text{now}} \] current

\[ (\cdot)_{t\text{last}} \] from previous cycle

\( 0^{+}, 0^{-} \) offset curves on phase plane

\[ (\cdot)^{+} \] positive value of a pair

\[ (\cdot)^{-} \] negative value of a pair

\[ \text{sgn}(\cdot) \] sign function

\[ (\cdot)_{\text{ap}} \] applied

\[ \text{unit}(\cdot) \] unit vector along ( ), as in Eq.(3-2)

\[ l(\cdot) \] unit vector along ( )
CHAPTER 1

INTRODUCTION

1.1 A Perspective on Spacecraft Attitude Control

Spacecraft attitude control systems are generally implemented with fixed or bounded control effectors, such as reaction control jets, control moment gyros, or magnetic torquers. Control systems for spacecraft have usually required several simplifying assumptions about the vehicle's behavior to make the design problem tractable. These control systems process inputs from guidance algorithms, the spacecraft crew, sensors and navigation systems, and subsequently issue commands to spacecraft control effectors to achieve the desired motion. They are called spacecraft autopilots.

Rigorous control algorithms for only a limited class of spacecraft maneuvers exist. Most autopilots are based on ad hoc generalizations. Optimal maneuver trajectories have been obtained only for a very few specific maneuvers of axially symmetric spacecraft, and are of little use for general problems at this time. No easily implemented method exists for the design of optimal control of nonlinear systems and hence, of spacecraft.

This report offers a spacecraft autopilot which combines a simple control law, and an optimal linear jet selection algorithm, and is intended to control or maintain a spacecraft about a nominal trajectory as provided by guidance or some other driving algorithm. Although the jet selection is based on an optimal linear solution, this autopilot is
not intended to be optimal, but rather represents a new approach to spacecraft control.

The results of this report represent testing of a preliminary version of the new autopilot. Further verification and development must be performed before considering the new autopilot as a candidate for a specific application.

1.2 History of Spacecraft Attitude Control

The earliest spacecraft, e.g., Explorer and Sputnik, relied on geometrical and dynamic considerations for passive attitude stabilization. It was soon discovered that such stabilization was inadequate for spacecraft which had to meet the requirements of more sophisticated missions implying that some active means of stabilization would be necessary.

Frequent disturbances are normally encountered by spacecraft which cause undesired motions. Requirements may also exist for the spacecraft to rotate to a desired orientation or move to a desired orbit. A means of determining attitude and the trajectory is needed, as is a means of using these data to maintain the spacecraft at a desired attitude in a desired orbit.

Many means of attitude sensing exist; these include horizon sensors, star trackers and inertial reference devices. To determine the nature of the orbit, tracking by optical, radio, and radar means has been employed. Inertial navigation systems aid the spacecraft in performing onboard guidance and navigation functions as a prelude to control.

To use this information, two other portions of the system are necessary: a control algorithm (law) and control effectors. In many spacecraft, the control algorithm is realized via electromechanical switching and analog computation. Since the early 1960's, digital
computers have proven useful in the control of more complex spacecraft, performing control, navigation, and guidance in one hardware component. The computational abilities and memory of these digital computers led to vast increases in control law sophistication. In the electromechanical systems, wiring was provided to issue firing commands to command each jet (or pair of jets) based on a linear combination of attitude and angular rate. Apollo saw the development of the more sophisticated phase plane switching logic concept to generate acceleration requests based on attitude errors and angular rate errors. These requests were then implemented by a table lookup scheme which selected a combination of jets to fire. The phase planes and tables could be changed in flight as the vehicle configuration and mass properties change, and lead to fuel savings over simple electromechanical switching.

The digital flight computer has the flexibility to allow the autopilot to reconfigure to meet requirements for attitude hold and maneuvering, and to provide both manual and automatic control functions.

Several devices have been developed to serve as spacecraft control effectors. These include flexible appendages, momentum wheels, gyro torquers, gas jets and rockets, ranging widely in available force and versatility. Passive stabilization of a simple vehicle can often be realized by using gravity gradients and flexible appendages. Pointing and tracking are feasible with control moment gyros or magnetic torquers. Both systems can be operated on solar power, are versatile, and are entirely self contained. The use of reaction control, via compressed gas or small rockets, provides greater control authority, including translation.

Various combinations of sensors, control algorithms, and control effectors have been developed for certain applications to meet requirements of versatility, simplicity, reliability, accuracy, or cost. Most often, the design criterion is a combination of these.
In the case of Apollo, for example, control systems were developed for two vehicles of varying configurations including powered flight, rendezvous and docking maneuvers, lunar landing, and liftoff. The system had to be sufficiently simple to be readily inspected prior to flight, and operated and monitored by a crew with many other duties. Lunar trips required a week or more, over hundreds of thousands of miles and could tolerate few control failures, thereby having required great confidence in the control system. The vehicles had to be guided to within a few miles of a target on the moon over a 240,000 mile range. Only a limited amount of fuel could be carried, so the control systems had to be economical. Finally, the computer that performed control computations had several other functions, so that control was allotted only a limited fraction of the onboard computer capacity.

A broad range of missions have successfully been performed with vehicles weighing from a few to several thousand pounds, both manned and unmanned. New missions and spacecraft will require more sophisticated autopilots, operating with ever greater requirements.

The Space Shuttle represents an advance over prior spacecraft in terms of performance and capability at the expense of complexity. Each Shuttle orbiter must be capable of performing a variety of tasks, from short, fast maneuvers to long term attitude maintenance with mission durations of up to a month. The currently baselined on-orbit autopilot for the Space Shuttle employs a phase plane control law for attitude control and open loop compensation for translation control, as in Apollo. The jet selection is mechanized by a table lookup algorithm. Unlike earlier spacecraft, with the exception of the Apollo Lunar Module, the Shuttle's mass properties and reaction control jet configurations are not simple, or even symmetric about the roll, pitch, and yaw axes. Thus, jet commands about any of these axes typically cause accelerations about the other axes, and motion in all three translation axes.
The widely varying mass properties due to changing mission requirements, payload deployment and retrieval, and fuel consumption, as well as the changing reaction control system jet availability due to control constraints, fuel management, and failure status, prevent design of a simple autopilot. Schemes for recomputing phase plane switching curves and for switching between several jet selection algorithms add complication and increases the flight computer burden associated with the problems of autopilot design.

The new autopilot addresses these problems by applying a new approach to autopilot design. The phase space concept is used as a control law, and linear programming is applied to the jet selection problem. The resulting autopilot is more versatile, more fuel economical, and, for example, places a smaller burden on the flight computer than the existing Space Shuttle autopilot.

1.3 Summary of Contents

Chapter 2 presents background information on spacecraft control system design. A derivation of the general equations of spacecraft motion is presented. Reaction control jets as control effectors are next discussed. Next, the digital computer as a means of implementing the control system is examined. Finally, some existing control laws are discussed.

Chapter 3 introduces the phase space control law. The basic concept is introduced and its application to general systems are discussed. Finally, the development of the phase space control law for spacecraft control is examined in detail.

Chapter 4 contains the development of the optimal linear jet selection algorithm. Linear programming is first reviewed, and conditions for its applicability discussed. The application of linear programming to the problem of jet selection is then presented. Finally,
several modifications due to practical limitations and special cases are presented.

Chapter 5 discusses the design of the new autopilot. It begins with an overview of the autopilot structure and data flow. Next, operational considerations are addressed and configurations examined. Finally, the proposed implementation is presented.

Chapter 6 describes the tests performed to verify the design of the new autopilot and compares its performance to the baselined Space Shuttle on-orbit autopilot which is currently under development.

Chapter 7 contains the results of these tests. It is shown that the new autopilot is capable of six degree of freedom control. Comparisons are made between the new autopilot and the current Space Shuttle on-orbit autopilot based on fuel consumption, accuracy, jet failure tolerance, and flight computer burden.

Chapter 8 contains the conclusions drawn from this research and recommendations for further study.

The appendices contain material supportive to the text.
CHAPTER 2

FUNDAMENTALS OF AUTOPilot DESIGN

Fundamental to understanding spacecraft control systems is a grasp of spacecraft dynamics. Various forces and torques influence the motion of spacecraft, including gravitation, control forces and torques, dynamic coupling, bending, and so on. Often one or several of these can be ignored, resulting in some simplification of the equations of motion. Selection of a suitable reference frame, based on inertia properties, ease of motion sensing, and facility to resolve forces and torques is of great importance. The first section of this chapter presents a derivation of the general equations of motion of a rigid body, and their application to spacecraft dynamics.

Reaction control jets as control effectors are next discussed. Characteristics of these jets and their utility are considered, as familiarity with such jets is basic to the following chapters.

Digital computers as a means of implementation of a control system are reviewed. Briefly described are some benefits and constraints in this use of digital computers.

Finally, some existing control laws are discussed to indicate the present state of autopilot design.

2.1 Spacecraft Dynamics

The mass, \( m_i \), of a particle multiplied by its velocity with respect to an inertial frame, \( v_i \), defines its linear momentum \( p_i \).
\[ F_i = m_i v_i \quad (2-1) \]

Newton's second law relates a force \( F_i \) applied to a particle to its linear momentum by

\[ F_i = \frac{d}{dt} \left( m_i v_i \right) \quad (2-2) \]

and, if the position of a particle is represented by its displacement vector \( r_i \), the velocity may be represented by

\[ v_i = \frac{d}{dt} r_i \quad (2-3) \]

so that

\[ F_i = \frac{d}{dt} \left( m_i \frac{d}{dt} r_i \right) \quad (2-4) \]

or, if \( m_i \) is constant,

\[ F_i = m_i \mathbf{a}_i \quad (2-5) \]

For a group of \( n \) particles comprising a body, one can write

\[ \sum_{i=1}^{n} F_i = \sum_{i=1}^{n} (m_i \mathbf{a}_i) \quad (2-6) \]

where each \( F_i \) may include internal forces \( S_i \) and external forces \( E_i \).

\[ F_i = S_i + E_i \quad (2-7) \]

We will consider rigid bodies only. In rigid bodies, the separation between particles comprising the bodies is constant. From Newton's second law, the equilibrium condition for particle \( i \) is

\[ E_i + S_i = 0 \quad (2-8) \]
If the particle undergoes a virtual displacement $\delta r_i$, this condition expressed in terms of virtual work is

$$E_i \cdot \delta r_i + S_i \cdot \delta r_i = 0 \quad (2-9)$$

The internal force on particle $i$ due to particle $j$ is $S_{ij}$ and

$$S_i = \sum_{j=1}^{n} S_{ij} \quad (2-10)$$

By Newton's third law,

$$S_{ij} = -S_{ji} \quad (2-11)$$

In a rigid body, any virtual motion $\delta r_i$ must maintain a constant separation between particles. This is so if

(1) particle $i$ and particle $j$ undergo the same displacement $\delta r_i$ or

(2) particle $i$ and particle $j$ rotate relative to each other

In the general case, both are present: both particles are displaced $\delta r_i$, and particle $j$ is rotated through an angle $\delta \phi_j$ about particle $i$ (Figure 2-1). The work done through the rotation is zero, as the tangential motion of particle $j$ is perpendicular to $S_{ji}$. The work done through displacement $\delta r$ on particle $i$ is equal and opposite to that done

![Figure 2-1. General case of relative motion and forces of two particles in a rigid body.](image-url)
on particle \( j \) through the same displacement, as the forces are equal and opposite. Thus, the net virtual work on both particles is zero, showing that the resultant of internal forces in a rigid body is zero. We see that

\[
\sum_{i=1}^{n} F_i = \sum_{i=1}^{n} F_i + \sum_{i=1}^{n} \mathbf{g}_i \tag{2-12}
\]

The motion of the collection of particles in translation can be described by the motion of the center of mass, defined by

\[
\mathbf{R}_{cm} = \frac{\sum_{i=1}^{n} m_i \mathbf{r}_i}{\sum_{i=1}^{n} m_i} \tag{2-13}
\]

Multiplying Eq. (2-13) by \( \sum_{i=1}^{n} m_i \), and differentiating twice with respect to time, one obtains

\[
(\sum_{i=1}^{n} m_i) \ddot{\mathbf{R}}_{cm} = \sum_{i=1}^{n} m_i \ddot{\mathbf{r}}_i \tag{2-14}
\]

or, by Eq. (2-5),

\[
\mathbf{F} = m \ddot{\mathbf{R}}_{cm} \tag{2-15}
\]

where

\[
m = \sum_{i=1}^{n} m_i \tag{2-16}
\]

and

\[
\mathbf{F} = \sum_{i=1}^{n} F_i \tag{2-17}
\]

which can be expressed in orthogonal XYZ coordinates as
Taking the cross product of a particle's displacement with its momentum, one obtains the definition of the moment of momentum $h_i$

$$h_i = \mathbf{r}_i \times m_i \mathbf{v}_i$$  \hspace{1cm} (2-21)$$

Summing this over the group of particles,

$$\sum_{i=1}^{n} h_i = \sum_{i=1}^{n} (\mathbf{r}_i \times m_i \mathbf{v}_i) = H$$  \hspace{1cm} (2-22)$$

where $H$ is called the angular momentum of the group of particles. Taking the derivative with respect to time,

$$\frac{d}{dt} H = \sum_{i=1}^{n} (\mathbf{r}_i \times m_i \mathbf{v}_i) + \sum_{i=1}^{n} (\mathbf{r}_i \times m_i \dddot{\mathbf{r}}_i)$$  \hspace{1cm} (2-23)$$

Since the first term of the right hand side is zero, and by Eq.(2-5),

$$\frac{d}{dt} \left( \sum_{i=1}^{n} \mathbf{r}_i \times m_i \mathbf{v}_i \right) = \sum_{i=1}^{n} (\mathbf{r}_i \times \mathbf{F}_i)$$  \hspace{1cm} (2-24)$$

The discussion thus far has treated motion only for an inertial reference frame. In a reference frame experiencing constant linear acceleration $\mathbf{a}$ with respect to inertial space, where a particle's position in the reference frame is $\mathbf{r}$, absolute acceleration is $\ddot{\mathbf{r}} + \mathbf{a}$, where $\ddot{\mathbf{r}}$ is now the acceleration relative to the reference frame. Then, for a single particle,
\[ F_i = m_i (\ddot{r}_i + a) \quad (2-25) \]

and
\[
\sum_{i=1}^{n} (r_i \times F_i) = \sum_{i=1}^{n} (r_i \times m_i (\ddot{r}_i + a))
\quad (2-26)
\]
\[
= \sum_{i=1}^{n} (r_i \times m_i \ddot{r}_i) + \sum_{i=1}^{n} (r_i \times m_i a)
\quad (2-27)
\]

However, \( a \) is the same for all particles, so
\[
\sum_{i=1}^{n} (r_i \times F_i) = \frac{d}{dt} \sum_{i=1}^{n} (r_i \times m_i \ddot{r}_i) + \sum_{i=1}^{n} m_i (R_{cm} \times a)
\quad (2-28)
\]

Note that for an unaccelerated origin (\( a = 0 \)) or one moving with the body center of mass (\( R_{cm} = 0 \)) the last term is zero. A reference frame fixed to the center of mass of the spacecraft, with origin at the spacecraft center of mass is assumed for the autopilot design which follows.

Treating the collection of particles as a continuum, one can replace the summation over masses \( m_i \) with integrals over elements of mass \( dm \). By so doing, Eq. (2-28) becomes
\[
\mathbf{r}_f \times \mathbf{F} = \frac{d}{dt} \int m \mathbf{r} \times \mathbf{\ddot{r}} \, dm + \int m \mathbf{R}_{cm} \times \mathbf{a}
\quad (2-29)
\]

where \( \mathbf{r}_f \) is the equivalent point of application of the external force resultant \( \mathbf{F} \) (Figure 2-2) such that
\[
(\mathbf{r}_f \times \mathbf{F}) = \sum_{i=1}^{n} (r_i \times F_i)
\quad (2-30)
\]
For a reference frame fixed to a body,

\[ \ddot{r} = \omega \times \dot{r} \]  

(2-31)

(i.e., \( \ddot{r} = 0 \) where \( \dot{r} = \dot{r}_{r} + \omega \times r \)) where \( \omega \) is the body's angular velocity relative to that origin, \( H \) is given by

\[ H = \int m r \times (\omega \times r) \, dm \]  

(2-32)

and Eq.(2-23) becomes

\[ \sum_{i=1}^{n} (r_i \times F_i) = \sum_{i=1}^{n} \left( \frac{d}{dt} (r_i \times F_i) \right) + m(R_{cm} \times a) \]  

(2-33)

which is the generalized moment equation for a rigid body. Again with respect to a frame fixed at the body center of mass, the second term of the right hand side is zero. At this time, and for what follows, it is convenient to write out the terms of Eq.(2-33) in component form. Picking orthogonal axes XYZ with origin at the body center of mass and unit
vectors \( i, j, k \) the term \( \int_M \vec{r} \times (\vec{\omega} \times \vec{r}) \, dm \) expands to

\[
H_x = \omega_x \int_M (y^2 + z^2) \, dm - \omega_y \int_M xy \, dm - \omega_z \int_M xz \, dm \tag{2-34}
\]

\[
H_y = -\omega_x \int_M xy \, dm + \omega_y \int_M (z^2 + x^2) \, dm - \omega_z \int_M yz \, dm \tag{2-35}
\]

\[
H_z = -\omega_x \int_M xz \, dm - \omega_y \int_M yz \, dm + \omega_z \int_M (x^2 + y^2) \, dm \tag{2-36}
\]

The integrals in these equations are second moments of mass, and for axes fixed to a rigid body they are constant. We can define moments and products of inertia

\[
I_x = \int_M (y^2 + z^2) \, dm \tag{2-37}
\]

\[
I_y = \int_M (z^2 + x^2) \, dm \tag{2-38}
\]

\[
I_z = \int_M (x^2 + y^2) \, dm \tag{2-39}
\]

\[
J_{xy} = J_{yx} = \int_M xy \, dm \tag{2-40}
\]

\[
J_{yz} = J_{zy} = \int_M yz \, dm \tag{2-41}
\]

\[
J_{xz} = J_{zx} = \int_M xz \, dm \tag{2-42}
\]

so that by substituting Eq. (2-37) - (2-42) in Eq. (2-34) - (2-36) we obtain

\[
H_x = I_x \omega_x - J_{xy} \omega_y - J_{xz} \omega_z \tag{2-43}
\]

\[
H_y = -J_{yx} \omega_x + I_y \omega_y - J_{yz} \omega_z \tag{2-44}
\]

\[
H_z = -J_{zx} \omega_x - J_{zy} \omega_y + I_z \omega_z \tag{2-45}
\]
Now Eq. (2-33) is written in components along inertially fixed axes.

\[
\begin{align*}
(\mathbf{r}_F \times \mathbf{F})_x &= \frac{d}{dt} (I_{xx} \omega_x - J_{xy} \omega_y - J_{xz} \omega_z) \\
(\mathbf{r}_F \times \mathbf{F})_y &= \frac{d}{dt} (-J_{yx} \omega_x + I_{xy} \omega_y - J_{yz} \omega_z) \\
(\mathbf{r}_F \times \mathbf{F})_z &= \frac{d}{dt} (-J_{zx} \omega_x - J_{zy} \omega_y + I_{zz} \omega_z)
\end{align*}
\]

(2-46)  (2-47)  (2-48)

Coordinatizing Eq. (2-46)-(2-48) in rotating axes, one can obtain the form of these equations relevant to a rotating observer. If the axis system rotates with angular velocity \( \Omega \) relative to the inertially fixed frame,

\[
\begin{align*}
(\mathbf{r}_F \times \mathbf{F})_x &= \frac{d}{dt} H_x - \Omega_z H_y + \Omega_y H_z \\
(\mathbf{r}_F \times \mathbf{F})_y &= \frac{d}{dt} H_y - \Omega_x H_z + \Omega_z H_x \\
(\mathbf{r}_F \times \mathbf{F})_z &= \frac{d}{dt} H_z - \Omega_y H_x + \Omega_x H_y
\end{align*}
\]

(2-49)  (2-50)  (2-51)

However, if these rotating axes are fixed to the body,

\[
\Omega = \omega
\]

(2-52)

giving

\[
\begin{align*}
(\mathbf{r}_F \times \mathbf{F})_x &= \frac{d}{dt} (I_{xx} \omega_x - J_{xy} \omega_y - J_{xz} \omega_z) \\
&\quad - J_y \omega_x \omega_z + J_{yx} \omega_x^2 + J_{yx} \omega_x \omega_x \\
&\quad + J_z \omega_y \omega_z - J_{zy} \omega_y^2 - J_{zy} \omega_y \omega_y \\
&\quad + J_z \omega_x \omega_z - J_{zx} \omega_x \omega_y - J_{zy} \omega_y \omega_y
\end{align*}
\]

(2-53)
\[(r_x \times F)_y = \frac{d}{dt} (I_y \omega_y - J_{yz} \omega_z - J_{yx} \omega_x) - I_z \omega_z \omega_x + J_{zx} \omega_x^2 + J_{zy} \omega_y \omega_y + I_x \omega_x \omega_x - J_{xy} \omega_y \omega_x - J_{xz} \omega_x \omega_z + I_y \omega_y \omega_y - J_{yx} \omega_x \omega_z - J_{yx} \omega_x^2) \quad (2-54)\]

\[(r_x \times F)_z = \frac{d}{dt} (I_z \omega_z - J_{zx} \omega_x - J_{zy} \omega_y) - I_x \omega_x \omega_x + J_{xy} \omega_y \omega_y + J_{xz} \omega_z \omega_y + I_y \omega_y \omega_y - J_{yx} \omega_x \omega_z - J_{yx} \omega_x \omega_x + I_z \omega_z \omega_z - J_{xz} \omega_x \omega_z - J_{xz} \omega_x \omega_z) \quad (2-55)\]

where the moments and products of inertia are computed in body fixed axes. The moments and products of inertia are given by Eq. (2-37)-(2-42) with x, y, z referred to body fixed coordinates.

A great simplification is realized by rewriting these in principle axes, so that all the $J$ terms become zero, leaving

\[(r_x \times F)_x = I_x \frac{d}{dt} \omega_x - (I_y - I_z) \omega_y \omega_z \quad (2-56)\]

\[(r_x \times F)_y = I_y \frac{d}{dt} \omega_y - (I_z - I_x) \omega_z \omega_x \quad (2-57)\]

\[(r_x \times F)_z = I_z \frac{d}{dt} \omega_z - (I_x - I_y) \omega_x \omega_y \quad (2-58)\]

which are the well known Euler equations.

These equations which describe a body's rotation about its center of mass, and Eq. (2-15)-(2-17) describing the motion of the body center of mass relative to inertial space completely characterize the body's motion with the given assumptions. The extent to which the rotational equations are coupled depends on the magnitudes of the angular rates and the values of the principle moments of inertia.
The angular position of a vehicle can be described in several ways, including rotation matrices, euler angles, and quaternions. In flight vehicle control, the angular position of a vehicle relative to some reference or desired orientation is described by a rotation about each of three orthogonal vehicle fixed axes. These axes can be selected to be principle axes, as are conventional aircraft axes, so that Euler's equations can be used to describe vehicle angular motion. If we restrict our angular velocities to be small so that the products

\[ \omega_y \omega_z < \frac{d \omega_x}{dt} \quad (2-59) \]

\[ \omega_x \omega_z < \frac{d \omega_y}{dt} \quad (2-60) \]

\[ \omega_x \omega_y < \frac{d \omega_z}{dt} \quad (2-61) \]

or the principle moments of inertia are similar in magnitude so that

\[ I_y - I_z < I_x \quad (2-62) \]

\[ I_z - I_x < I_y \quad (2-63) \]

\[ I_x - I_y < I_z \quad (2-64) \]

A simple approximation to Euler's equations is

\[ (r_f \times F)_x = I_x \frac{d \omega_x}{dt} \quad (2-65) \]

\[ (r_f \times F)_y = I_y \frac{d \omega_y}{dt} \quad (2-66) \]

\[ (r_f \times F)_z = I_z \frac{d \omega_z}{dt} \quad (2-67) \]

that is, dynamical coupling is insignificant.
The transformation matrices representing angular displacements behave differently for large and small displacements. Consider rotation through three Euler angles $\theta$, $\phi$, $\psi$ starting with axes $XYZ$. The transformations are

rotate through $\theta$ about $X$ to $X'Y'Z'$

\[
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta & \sin \theta \\
0 & -\sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]  
(2-68)

rotate through $\phi$ about $Y'$ to $X''Y''Z''$

\[
\begin{bmatrix}
X'' \\
Y'' \\
Z''
\end{bmatrix} =
\begin{bmatrix}
\cos \phi & 0 & \sin \phi \\
0 & 1 & 0 \\
-\sin \phi & 0 & \cos \phi
\end{bmatrix}
\begin{bmatrix}
X' \\
Y' \\
Z'
\end{bmatrix}
\]  
(2-69)

rotate through $\psi$ about $Z''$ to $X'''Y'''Z'''$

\[
\begin{bmatrix}
X''' \\
Y''' \\
Z'''
\end{bmatrix} =
\begin{bmatrix}
\cos \psi & \sin \psi & 0 \\
-\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X'' \\
Y'' \\
Z''
\end{bmatrix}
\]  
(2-70)

The transformation from $XYZ$ to $X'''Y'''Z'''$ is obtained by substitution.

\[
\begin{bmatrix}
X''' \\
y'''
\end{bmatrix} =
\begin{bmatrix}
\cos \psi & \sin \psi & 0 \\
-\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\cos \phi & 0 & \sin \phi \\
0 & 1 & 0 \\
-\sin \phi & 0 & \cos \phi
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\]  
(2-71)
Multiplying through,

\[
\begin{align*}
X'' &= \begin{bmatrix} \cos \psi \cos \phi & (-\sin \theta \sin \phi \cos \psi) & (\sin \phi \cos \theta \cos \psi) \\ -\sin \psi \cos \phi & (\sin \theta \sin \phi \sin \psi) & (+ \sin \phi \cos \theta \sin \psi) \\ -\sin \phi & -\sin \theta \cos \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\
Y'' &= \begin{bmatrix} \cos \phi \cos \psi & (\cos \theta \cos \phi \sin \psi) & (\sin \theta \sin \psi \cos \phi) \\ -\sin \psi & + \cos \theta \cos \psi & \sin \theta \cos \psi \\ -\sin \phi \cos \psi & (-\cos \theta \sin \phi \sin \psi) & (+ \cos \theta \cos \phi) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\
Z'' &= \begin{bmatrix} -\sin \phi \cos \psi & (-\cos \theta \sin \phi \sin \psi) & (-\sin \theta \sin \phi \sin \psi) \\ -\sin \theta \cos \phi & -\sin \phi \cos \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}
\end{align*}
\]

(2-72)

Were we to change the order of rotation to \( \theta, \psi, \phi \) we would obtain

\[
\begin{align*}
X'' &= \begin{bmatrix} \cos \phi \cos \psi & (\cos \theta \cos \phi \sin \psi) & (\sin \theta \sin \psi \cos \phi) \\ -\sin \psi \cos \phi & (+ \cos \theta \cos \psi) & \sin \theta \cos \psi \\ -\sin \phi \cos \phi & (-\cos \theta \sin \phi \sin \psi) & (+ \cos \theta \cos \phi) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\
Y'' &= \begin{bmatrix} \cos \phi \cos \psi & (\cos \theta \cos \phi \sin \psi) & (\sin \theta \sin \psi \cos \phi) \\ -\sin \psi \cos \phi & (+ \cos \theta \cos \psi) & \sin \theta \cos \psi \\ -\sin \phi \cos \phi & (-\cos \theta \sin \phi \sin \psi) & (+ \cos \theta \cos \phi) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\
Z'' &= \begin{bmatrix} -\sin \phi \cos \psi & (-\cos \theta \sin \phi \sin \psi) & (-\sin \theta \sin \phi \sin \psi) \\ -\sin \theta \cos \phi & -\sin \phi \cos \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}
\end{align*}
\]

(2-73)

which is clearly a different transform. However, if \( \theta, \phi, \psi \) are small, we can approximate

\[
\begin{align*}
\sin \theta &\approx 0 \\
\sin \phi &\approx \phi \\
\sin \psi &\approx \psi \\
\cos \theta &\approx 1 \\
\cos \phi &\approx 1 \\
\cos \psi &\approx 1
\end{align*}
\]

(2-74)

(2-75)

so the transformation of Eq.(2-72) becomes

\[
\begin{align*}
X'' &= \begin{bmatrix} 1 & -\theta \phi + \psi & \phi + \theta \psi \\ -\psi & \theta \phi \psi + 1 & -\phi \psi + \theta \\ -\phi & -\theta & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\
Y'' &= \begin{bmatrix} 1 & -\theta \phi + \psi & \phi + \theta \psi \\ -\psi & \theta \phi \psi + 1 & -\phi \psi + \theta \\ -\phi & -\theta & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \\
Z'' &= \begin{bmatrix} 1 & -\theta \phi + \psi & \phi + \theta \psi \\ -\psi & \theta \phi \psi + 1 & -\phi \psi + \theta \\ -\phi & -\theta & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}
\end{align*}
\]

(2-76)
and that of Eq.(2-73) becomes

\[
\begin{bmatrix}
X'''

Y'''

Z'''
\end{bmatrix} =
\begin{bmatrix}
1 & \psi & \phi \\
-\psi & 1 & \theta \\
-\phi & -\phi\psi & 1
\end{bmatrix}
\begin{bmatrix}
X

Y

Z
\end{bmatrix}
\]

(2-77)

Further, by considering the products of angles negligible compared to the angles, and 1.0, we obtain from Eq.(2-76)

\[
\begin{bmatrix}
X'''

Y'''

Z'''
\end{bmatrix} =
\begin{bmatrix}
1 & \psi & \phi \\
-\psi & 1 & \theta \\
-\phi & -\phi\psi & 1
\end{bmatrix}
\begin{bmatrix}
X

Y

Z
\end{bmatrix}
\]

(2-78)

and from Eq.(2-77),

\[
\begin{bmatrix}
X'''

Y'''

Z'''
\end{bmatrix} =
\begin{bmatrix}
1 & \psi & \phi \\
-\psi & 1 & \theta \\
-\phi & -\phi\psi & 1
\end{bmatrix}
\begin{bmatrix}
X

Y

Z
\end{bmatrix}
\]

(2-79)

which are equal, showing that within the above assumptions the order of rotation through three angles from one frame to another does not affect the result. The transformations of each rotation are said to commute.

This is important, as it allows small angular displacements to be treated as a vector, with components corresponding to angles about three axes, a key point in the sequel. From the definition of the derivative, we obtain a definition of angular velocity \( \omega \), about some axis

\[
\omega = \frac{d\xi}{dt} = \lim_{\Delta t \to 0} \frac{\xi(t + \Delta t) - \xi(t)}{\Delta t}
\]

(2-80)

where \( \xi(t) \) is the angular displacement about that axis at time \( t \). In the limit, \( \xi(t + \Delta t) - \xi(t) \) is an infinitesimal angle, so we can differentiate Eq.(2-78) term by term and find that transformations due to
angular rate also commute. Thus, angular velocity can also be treated as a vector.

The discussion presented in this section outlined the assumptions used in design of space vehicle control systems. Specifically, it is seen that while the rotational equations of spacecraft motion are non-linear, under simplifying assumptions they can be treated as linear, second order systems. The assumption that small angular displacements and angular velocities behave as vectors is helpful in the development of the new autopilot.

2.2 Control Effectors

Several types of devices have been used to apply control torques and forces to spacecraft. Among these are gyro torquers, magnetic torquers, thrust vector control of rocket engines, ion thrusters and reaction control jets. Only reaction control jets are discussed in detail, as they are the example control effectors used in this thesis by the new autopilot. Two types of reaction control jets are reviewed, their merits and drawbacks are compared and some existing jet configurations are reviewed.

2.2.1 Reaction Control Jets

Compressed gas, stored on board the vehicle and expelled through various nozzles, is the simplest type of reaction control jet. Nitrogen or helium are commonly used, stored in either the gaseous or liquid state. The gas is piped to each of several jet nozzles which are positioned about the spacecraft. The jets are turned on and off by simply opening or closing a valve in the nozzle. Total energy capacity for such a system is low for its weight compared to other types of reaction control. This is because the only useful work is done by the
expansion of the gas from the storage pressure to ambient pressure. Such systems, however, tend to be simplest and the most reliable of all reaction control systems.

A more efficient means of reaction control is realized by replacing the compressed gas system by a group of small rocket engines. Existing designs have employed monopropellants and hypergolic bipropellants, neither of which requires an igniter. In both cases the engines are turned on and off by simply opening and closing valves. However, there is the hazard of propellant leaks, or improper mixing which could lead to explosions. A great deal of effort has gone into minimizing explosion hazards of such systems. Rocket engines offer an improvement in performance over compressed gas systems with specific impulse values close to 300 sec being realized currently. Such increases in performance allow longer missions with greater maneuvering capability than had been previously possible.

Typical of the current state of the art are the reaction control jets designed for the Space Shuttle $^5,6$. These are bipropellant hypergolic jets, using a mixture of nitrogen tetraoxide and monomethyl hydrazine. The design specific impulse of the engines is 289 sec and the minimum impulse obtainable is 16 lbf·sec. These jets must be capable of firing for as little as 40 msec and as long as 150 sec over missions lasting up to 30 days. The operating life of each engine is expected to be 20,000 sec over 100 missions in 10 years.

2.2.2 Reaction Control Jet Configurations

On unmanned spacecraft, jet configurations often consist of only 12 jets, as in Mariner (Figure 2-3)$^7$. Each jet is simply a signed roll, pitch, or yaw jet. Since pure couples with both positive and negative sense about each axis are required, the number of jets becomes three
Figure 2-3. Examples of multi-jet systems.
axes times two jets per couple times two senses per axis or 12. The Apollo command module employs a similar twelve-jet scheme, as had the Mercury spacecraft. Several spacecraft also required translation capabilities in all directions. To achieve this with reaction control, spacecraft such as the Apollo Service Module and Lunar Module (Figure 2-3) employed configurations of 16 jets, providing full control, and some redundancy at the expense of degraded control.

In all these spacecraft, the reaction control design was intended primarily for exoatmospheric use and provided little or no redundancy. The Space Shuttle, as it is both spacecraft and glider, must compromise aerodynamic fairing of reaction jets and utility, and must offer a high level of redundancy due to its long manned missions. While the Apollo Command Module and Mercury spacecraft also traveled through the atmosphere on entry, their shape (generally a frustrum) was such that the compromise between aerodynamics and control was quite simple. The Space Shuttle, however, is shaped more like an aircraft, with some aerodynamic constraints on reaction control jet design, sacrificing the simplicity of design found in other spacecraft. Additionally, while other spacecraft were one-shot affairs, the Shuttle is intended to be flown repeatedly, requiring greater jet redundancy.

These considerations led the Shuttle designers to a 44-jet configuration (Figure 2-4) with no pure couples and no pure translation capability whatsoever. Thus, jets cannot be selected purely on an axis by axis basis, but on the basis of the complete six elements of an acceleration request. Jet failures may render unavailable certain jets desirable for a specific request, so that alternate jets must be selected. Thus, jet control torques are not repeatable on a case by case basis, but vary with the nature of the request and fail status of the jets.
Figure 2-4. Space Shuttle Orbiter reaction control system.
2.3 Digital Spacecraft Control

Since the middle 1960's digital computers of a type practical for in-flight application have been available. Flight computers must be small and light enough to be carried within the spacecraft, and draw little power, as the resources of a spacecraft are limited. The first use of onboard digital computers as spacecraft control elements was in Apollo. An entirely new digital computer was developed expressly for the Apollo program, capable of guidance, navigation, and control tasks, involving complex computation and decisions. The great success of this computer control system has led to a new generation of autopilots built around digital computers. Several types of digital flight computers have since been introduced for onboard applications in aircraft, missiles, and spacecraft.

The Space Shuttle, for example, employs four IBM AP-101 computers in the primary flight control system and one AP-101 for system management and backup flight control. The flexibility and the computational capability of the AP-101 is an advancement over the Apollo computers, and have enabled the implementation of several additional functions in the flight computer. The direction of research in computer technology indicates that even more sophisticated flight computers, capable of handling tasks of increasing complexity, and allowing the control engineer much more freedom in the design of control algorithms, will be developed.

2.4 Existing Attitude Control Laws

Existing control laws for reaction control of spacecraft treat the spacecraft as a linear system, with no coupling between axes (even for non-orthogonal axes). They are generally based on "bang-bang" control principles or phase plane switching logic. Herein is a brief review of the bang-bang control problem, and phase plane control law.
2.4.1 Double Integrator Problem

Control of a double integrator is a classical problem of control theory. However, when the control is bounded, and an optimal control law sought, the problem is no longer one of classical control, but rather a so-called "bang-bang" control problem. Consider a system with the governing differential equation

\[ \ddot{y}(t) = u(t) \quad (2-81) \]

where the output variable is \( y \), and \( u \) represents the control input.

This might be acceleration control of a particle, or angular acceleration of a spacecraft. Suppose it is desired to drive \( y \) to a desired value at time \( T, y_d \). We can define the state variables \( x_1, x_2 \) by

\[ y = x_1 \quad (2-82) \]
\[ x_2 = \dot{x}_1 \quad (2-83) \]

so that Eq. (2-79) becomes

\[ \dot{x}_1 = x_2 \quad (2-84) \]
\[ \dot{x}_2 = u(t) \quad (2-85) \]

Assume finally that the control is bounded, i.e.,

\[ |u(t)| \leq 1 \quad (2-86) \]

It is desired to drive this system from some state \( (\xi_1, \xi_2) \) to \( (0, 0) \) while minimizing the fuel

\[ F(u) = \int_0^T |u(t)| \, dt \quad (2-87) \]

over an unspecified interval \( (0, T) \).

The approach taken in finding the optimal control is to find the control which absolutely minimizes the Hamiltonian associated with the
system. Here, the Hamiltonian, $H$, is given by

$$H = |u(t)| + x_2(t)p_1(t) + u(t)p_2(t)$$  \hspace{1cm} (2-88)

where $p_1(t)$ and $p_2(t)$ are the system costate variables. Athans and Falb\textsuperscript{9} show that the control which absolutely minimizes the Hamiltonian is given by

$$u(t) = 0 \text{ if } |p_2(t)| < 1$$  \hspace{1cm} (2-89)

$$u(t) = -\text{sgn}(p_2(t)) \text{ if } |p_2(t)| > 1$$  \hspace{1cm} (2-90)

$$0 \leq u(t) \leq 1 \text{ if } p_2(t) = -1$$  \hspace{1cm} (2-91)

$$-1 \leq u(t) \leq 0 \text{ if } p_2(t) = +1$$  \hspace{1cm} (2-92)

The costate variables $p_1(t)$ and $p_2(t)$ are given by

$$\dot{p}_1(t) = \frac{-\partial H}{\partial x_1(t)} = 0$$  \hspace{1cm} (2-93)

and

$$\dot{p}_2(t) = \frac{-\partial H}{\partial x_2(t)} = -p_1(t)$$  \hspace{1cm} (2-94)

from which

$$p_1(t) = c_1$$  \hspace{1cm} (2-95)

$$p_2(t) = c_2 - c_1t$$  \hspace{1cm} (2-96)

where

$$c_1 = p_1(0)$$  \hspace{1cm} (2-97)

and

$$c_2 = p_2(0)$$  \hspace{1cm} (2-98)
If we have the condition
\[ c_1 = 0 \] (2-99)
and
\[ |c_2| = 1 \] (2-100)
then
\[ |p_2(t)| = 1, \ t \in [0, T] \] (2-101)

Suppose \( v(t) \in v^+_T \), where the set \( v^+_T \) is given by
\[ v^+_T = \left\{ v(t): 0 \leq v(t) \leq 1, \ \text{all} \ t \in [0, T], \ v(t) \neq 0 \right\} \] (2-102)
then a possible fuel optimal control is
\[ u(t) = -\text{sgn}(c_2) v(t), \ v(t) \in v^+_T \] (2-103)

If \( c_1 \neq 0 \), then only nine control sequences, \((0), (+1), (-1), (+1, 0), (-1, 0), (0, +1), (0, -1), (+1, 0, -1), (-1, 0, +1)\) are candidates for fuel optimal control.

To obtain fuel optimal solutions, and to ascertain when no fuel optimal solutions exist, one can create an \( x_1x_2 \) state plane, and divide it into four regions, as in Figure 2-5, by curves \( \gamma^+, \gamma^- \) defined by Athans and Falb as
\[ \gamma^+ = \left\{ (x_1, x_2): \ x_1 = \frac{1}{2} x_2^2; \ x_2 \leq 0 \right\} \] (2-104)
\[ \gamma^- = \left\{ (x_1, x_2): \ x_1 = -\frac{1}{2} x_2^2; \ x_2 \geq 0 \right\} \] (2-105)
The curve \( \gamma^+ \) is simply the locus of points which can be forced to the origin of the state plane by the control \( u = -1 \). A fuel optimal control law for this problem is then given by
\[ u = +1 \text{ for all } (x_1, x_2) \in \gamma^+ \]  
(2-106)

\[ u = -1 \text{ for all } (x_1, x_2) \in \gamma^- \]  
(2-107)

\[ u = 0 \text{ for all } (x_1, x_2) \in R_2 \cup R_4 \]  
(2-108)

and no fuel optimal control exists for \((x_1, x_2) \in R_1 \cup R_3\). In practical systems based on this law, control is applied as soon as possible after the state enters region \(R_1\) or \(R_3\), to minimize fuel and time expended in driving the state into region \(R_3\) or \(R_4\), as time and fuel expenditure generally increase with the \(X_2\) component of phase plane displacement required of control. Inspection of Figure 2-5 shows that this distance increases as the state is allowed to drift along the \(X_1\) axis away from the origin.

![Figure 2-5. \(X_1X_2\) state plane divided into four regions.](image)

41
Figure 2-6. Phase plane for spacecraft control problem.

What has been derived is an optimal "bang-bang" control law for the double integrator problem. By considering the "control" to be acceleration applied by control effectors such as reaction control jets, and the state variables $x_1$ and $x_2$ to be angular position and angular rate, the phase plane control law shall be developed for spacecraft attitude control.

2.4.2 Phase Plane Control Law

For the single axis translational or rotational motion of a spacecraft in the absence of coupling (see Section 2.1),

\[
\begin{align*}
\ddot{\theta} &= M_{ap} \\
\ddot{x} &= F_{ap}
\end{align*}
\]

Here, $I$ is the moment of inertia about the axis being considered, $\theta$ is the angular position about that axis from a reference orientation, and $M_{ap}$ is the applied torque. In the second equation $m$ is the vehicle mass, $x$ is the displacement along the axis from a reference position, and $F_{ap}$
is the applied force. Both Eq.(2-107) and Eq.(2-108) are in the same form, so only Eq.(2-107) will be studied, the results directly applicable to Eq.(2-108).

Given a desired attitude $\theta_d$ and angular rate $\dot{\theta}_d$, the attitude and rate errors are

$$E = \theta - \theta_d \tag{2-111}$$

and

$$\dot{E} = \dot{\theta} - \dot{\theta}_d \tag{2-112}$$

The dynamical equation can then be written

$$\ddot{E} = \frac{d\dot{E}}{dt} = \frac{d\dot{\theta}}{dt} - \frac{d}{dt} (\dot{\theta}_d) \tag{2-113}$$

$$= \ddot{\theta} - \frac{d}{dt} (\dot{\theta}_d) \tag{2-114}$$

By Eq.(2-107), this becomes

$$\frac{d\dot{E}}{dt} = \frac{M_{ap}}{I} - \frac{d}{dt} (\dot{\theta}_d) \tag{2-115}$$

Differentiating Eq.(2-109) gives

$$\dot{E} = \frac{dE}{dt} = \frac{d\theta}{dt} - \frac{d\dot{\theta}_d}{dt} \tag{2-116}$$

On the state plane of the previous section, we plot $E$ vs. $\dot{E}$, giving a phase plane. The trajectories $\gamma^+$ and $\gamma^-$ are thus constant torque trajectories. To obtain expressions for these, divide Eq.(2-113) by Eq.(2-114).

$$\frac{d\dot{E}}{dE} = \frac{(M_{ap}/I) - (d\dot{\theta}_d/dt)}{\dot{E}} \tag{2-117}$$

If we constrain $\dot{\theta}_d$ to be a constant, which is typically the case, then
\[
\frac{d\dot{E}}{dE} = \frac{M_{ap}}{I\dot{E}}
\]

or
\[
\ddot{E} \dot{E} = \frac{M_{ap}}{I} \dot{E}
\]

Integrating,
\[
\int_{E_0}^{E_f} \dot{E} \dot{E} = \int_{E_0}^{E_f} \frac{M_{ap}}{I} \dot{E}
\]

which gives
\[
\frac{1}{2} (E_f^2 - E_0^2) = \frac{M_{ap}}{I} (E_f - E_0)
\]

or
\[
E_f = \frac{I}{2M_{ap}} (E_f^2 - E_0^2) + E_0
\]

which is a parabola. Thus, for the spacecraft control problem, the phase plane can be divided into four regions by two parabolas given by Eq. (2-119), one with \( \dot{E}_0 > 0 \) and one with \( \dot{E}_0 < 0 \). The optimal control is then given by

\[
u = +1 \text{ for all } (E_0, \dot{E}_0) \in \gamma^+
\]

\[
u = -1 \text{ for all } (E_0, \dot{E}_0) \in \gamma^-
\]

\[
u = 0 \text{ for all } (E_0, \dot{E}_0) \in R_2 \cup R_4
\]

with regions \( R_2 \) and \( R_4 \) as in Figure 2-6.

It is necessary to consider what control to apply in regions \( R_1 \) and \( R_3 \). No control would result in divergence of \( E_0 \) as in Figure 2-7. A convergent, but suboptimal scheme is to apply constant control immediately upon entering these regions. This will drive the state into \( R_2 \) or
Figure 2-7. No control trajectories in regions $R_1, R_3$.

$R_4$, for which the optimal control is a zero torque, so the trajectories of Figure 2-8 are realized. Thus, a convergent, non-optimal control law is realized. Other practical modifications to this control law, i.e., limit cycling, hysteresis, are necessary, but the basic idea is nonetheless sound.

Figure 2-8. Constant torque trajectories in regions $R_1, R_3$. 
The intent of this control system is to drive the phase plane trajectory to the origin, corresponding to zero attitude error and zero rate error. With disturbances present, and the finite control granularity of the jets, it is impractical to drive the trajectory exactly to the origin. Instead, allowable deadbands on $E$, $db^+$ and $db^-$ are defined, and the trajectory is driven to within these deadbands. To do so, the curves $\gamma^+$ and $\gamma^-$ are replaced by four switching curves $S_1$, $S_2$, $S_3$, and $S_4$, as in Figure 2-9, defined as follows.

$$S_1 = \{(E, \dot{E}): E = -\frac{1}{2} \dot{E}^2 + \delta^+, \dot{E} \geq 0\}$$ (2-127)

$$S_2 = \{(E, \dot{E}): E = -\frac{1}{2} \dot{E}^2 + \delta^+, \dot{E} < 0\}$$ (2-128)

$$S_3 = \{(E, \dot{E}): E = \frac{1}{2} \dot{E}^2 - \delta^-, \dot{E} \geq 0\}$$ (2-129)

$$S_4 = \{(E, \dot{E}): E = \frac{1}{2} \dot{E}^2 - \delta^-, \dot{E} < 0\}$$ (2-130)

Figure 2-9. Phase plane switching curves.
These curves define three regions, $R_1$, $R_2$, and $R_3$. In region $R_1$, a negative torque is applied to drive the state to curve $S_2$. In region $R_2$, no control is applied, so that the state drifts toward $S_1$ if $\dot{E} > 0$, or $S_4$ if $\dot{E} < 0$ (obviously, if $\dot{E} = 0$, the state does not move on the phase plane). In region $R_3$, positive torque is applied to drive the state to curve $S_3$.

An "offset" ($\delta^+, \delta^-$) is built into the curves to ensure that the control will not drive the state far beyond the $E$ axis when the state has just emerged from $R_1$ or $R_3$ (Figure 2-10). Rather, the state is allowed to overshoot the $E$ axis by a minimal amount, control is set to zero, and the state then coasts toward the $E$ axis.

![Figure 2-10](image)

*Figure 2-10. Overshoot of $E$ axis; ---- denotes trajectory.*

The disturbances and jet granularity also prevent the rate error $\dot{E}$ from being driven or maintained exactly at zero. When the attitude error is within the deadband it is desired to keep $|\dot{E}|$ as small as possible, thereby minimizing the frequency of deadband limit cycle firings,
the trajectory will thus follow a limit cycle as shown in Figure 2-11. The state coasts at a small \( \dot{E} \), until reaching either \( DB^+ \) or \( DB^- \), at which point a small impulse is applied to arrive at a small \( \dot{E} \) with sign reversed.

![Figure 2-11. Limit cycle trajectory.](image)

Two example phase plane trajectories appear in Figure 2-12. In the first, the state starts in region \( R_2 \) at point a. It drifts toward \( +E \), at constant \( \dot{E} \), intercepting curve \( S_1 \) at b. Upon hitting \( S_1 \), negative torque is applied to drive the state into the offset at c, where the control torque is removed, and a limit cycle ensues. In the second example, the state starts in region \( R_1 \) at point d. Negative torque is applied, driving the state into curve \( S_2 \) at point e. Here, the state coasts (with no applied torque) to \( S_4 \) at f. At point f, positive torque is applied to drive the state into the offset at g, where a limit cycle is begun. The trajectory for state starting in region \( R_3 \) is similar to that of the second example.

Existing autopilots used in Apollo and the Space Shuttle orbiter are based on these concepts. These autopilots assume that spacecraft control can be performed independently by axis, and that the control torques are constant and known a priori. While these are reasonable
assumptions for such spacecraft as Mariner and Apollo, it is seen elsewhere in this chapter that they are not readily applicable to such vehicles as the Space Shuttle orbiter, and possibly to future spacecraft. The new autopilot incorporates a new philosophy which obviates these assumptions.
A new control law, called the phase space control law, is the basis of the new autopilot. This control law, first conceived by Steven R. Croopnick\textsuperscript{11} uses the classical velocity to be gained principle, and the new concept of a "phase sphere". In this chapter, the velocity to be gained principle is first reviewed. Next, the phase sphere concept is introduced for \( n \)th order systems. Finally, the application of the phase space control law to six degree of freedom spacecraft control is developed in detail.

3.1 Velocity to be Gained Principle

Most familiar in its application to guidance, the velocity to be gained principle is an algorithm to generate a desired rate change or "velocity to be gained" based on the current state and on a target state.

Consider a system with a position state vector \( x \), which has an explicitly computable rate derivative \( \dot{x} \). It is desired to drive the system to a target state \( x_d \), and rate \( \dot{x}_d \). The velocity to be gained principle defines a desired rate change \( \dot{x}_g \) by

\[
\dot{x}_g = c \text{ unit} \left( x_d - x \right) + \left( \dot{x}_d - \dot{x} \right) \tag{3-1}
\]

The first term, \( c \text{ unit} \left( x_d - x \right) \), is a component of rate which drives the state to the target value, a convergent velocity. The scalar \( c \) is called the convergence rate, and unit \( (x_d - x) \) is defined in the conventional manner.
The second term, \( \dot{x}_d - \dot{x} \), represents the correction to the current rate, and the negative of this term, \( \dot{x} - \dot{x}_d \), is called the relative rate.

To appreciate the significance of these terms, consider four simple cases of an example problem. Suppose it is desired to guide a missile from position \( P_m \), measured in a convenient reference frame, to a target at position \( P_t \), in that same frame. Suppose further that the missile is traveling at velocity \( v_n \) and it is desired to have it arrive at the target with velocity \( v_{\text{desired}} \), both with respect to the frame in which \( P_m \) and \( P_t \) are measured. The velocity to be gained principle gives the desired velocity increment \( v_g \) as

\[
v_g = \frac{c(P_t - P_m)}{\sqrt{(P_t - P_m) \cdot (P_t - P_m)}} \cdot (v_{\text{desired}} - v_n) \quad (3-3)
\]

**CASE 1:**

\[
v_{\text{desired}} = v_n \quad (3-4)
\]

\[
P_m \neq P_t \quad (3-5)
\]

The missile is moving at the desired rate (which may be toward or away from the target) but is not at the target. A small rate change,

\[
v_g = \frac{c(P_t - P_m)}{\sqrt{(P_t - P_m) \cdot (P_t - P_m)}} + 0 \quad (3-6)
\]

is obtained from the principle. This is a convergent velocity, which will drive the state toward the target.
CASE 2:

\[ P_m = P_t \]  \hspace{1cm} (3-7)
\[ \dot{v}_{\text{desired}} \neq \dot{v}_n \]  \hspace{1cm} (3-8)

The state is at the target, but moving at the wrong rate. Here velocity-to-be-gained gives

\[ \dot{v}_g = 0 + (\dot{v}_{\text{desired}} - \dot{v}_n) \]  \hspace{1cm} (3-9)

which is simply a rate correction.

CASE 3:

\[ P_m = P_t \]  \hspace{1cm} (3-10)
\[ \dot{v}_{\text{desired}} = \dot{v}_n \]  \hspace{1cm} (3-11)

The missile is at the target with the desired velocity. The rate change is zero. One must note, however, that here, and in Case 2, the first term of the principle actually diverges, due to the zero in the denominator, but is taken as zero. In practical applications, it is necessary to ignore this term or inhibit its computation at the target.

CASE 4:

\[ P_m \neq P_t \]  \hspace{1cm} (3-12)
\[ \dot{v}_{\text{desired}} \neq \dot{v}_n \]  \hspace{1cm} (3-13)

Here we obtain the full law, generating a convergent rate as in Case 1, and a rate correction as in Case 3.

We have seen a modification to this law at the target position. In general, it is not possible to determine when one is exactly at the target, and errors exist in the rate computation. Thus, it is not precisely known when each of the four cases obtains. Further, it is not
feasible to apply the rate change instantaneously, so that frequent re-
computation of the velocity to be gained law will give a rapidly changing
request. It is therefore necessary to incorporate deadbands and hyste-
resis into any control law using this principle. The method used in the
new autopilot is described in the following sections.

3.2 The Phase Sphere Concept

Consider a system described by a state error vector, $\mathbf{x}_e$, given by

$$\mathbf{x}_e = \mathbf{x} - \mathbf{x}_d$$  \hspace{1cm} (3-14)

This is a $j$-dimensional vector, and it is necessary to control each
element within its particular deadband, $\mathbf{db}_i$, which may have a different
value and different units for each element. To decide whether an ele-
ment $x_{e_i}$ is within its deadband, it is only necessary to compare two
numbers, $|x_{e_i}|$ and $\mathbf{db}_i$, if the deadbands are symmetric, i.e., the positive
deadband $\mathbf{db}^+$ has magnitude equal to that of the negative deadband $\mathbf{db}^-$. If not, the variable and its deadbands can be "translated" so a similar
test is performed. A new vector, $\mathbf{\alpha}_e$, is created by

$$\mathbf{\alpha}_e = |\mathbf{x}_{e_i} - \frac{1}{2} (\mathbf{db}^+ - \mathbf{db}^-)|$$  \hspace{1cm} (3-15)

which centers the range of the variable between $\mathbf{db}^+$ and $\mathbf{db}^-$. This is
compared to the halfwidth of this range, i.e., $\frac{1}{2} (\mathbf{db}^+ - \mathbf{db}^-)$. The locus
of points described by the $j$ equations

$$\mathbf{db}_i = \pm \text{constant}_i$$  \hspace{1cm} (3-16)

or

$$\mathbf{db}_i = \pm \frac{1}{2} (\mathbf{db}^+ - \mathbf{db}^-)$$  \hspace{1cm} (3-17)

is a $j$-dimensional prism, centered on $\mathbf{x}_{d_i}$ in those variables where the
deadbands are symmetric and on $\frac{1}{2} (\mathbf{db}^+ - \mathbf{db}^-)$ in those variables where
they are not. The above comparisons with the deadbands are equivalent to determining whether the tip of \( \mathbf{x}_e \) lies within the prism.

By normalizing the vector space of \( \mathbf{x} \), it is possible to replace this prism by a hypercube. The \( i \)th dimension of the prism is twice \( db_i \). All of these can be made numerically equal in a space of \( \mathbf{x}_e \), with \( \mathbf{x}_{e_i} \) given by

\[
\frac{\mathbf{x}_{e_i}}{r_{ps}} = \frac{x_{e_i}}{db_i}
\]

(3-18)

where \( 2r_{ps} \) is the size of a side of the hypercube. A slightly more restrictive, but simpler test for the tip of \( \mathbf{x}_e \) in this hypercube is to check if the tip lies within the inscribed hypersphere. This will be so if

\[
\sqrt{\mathbf{x}_e \cdot \mathbf{x}_e} < r_{ps}
\]

(3-19)

so that the variables are tested for an out-of-deadband condition by comparison of two scalars. The above-mentioned hypersphere will be referred to as a "phase sphere" and is a simple means of providing deadbands and hysteresis.

### 3.3 The Phase Space Control Law

The phase space control law brings together the velocity to be gained principle and the phase sphere concept into a simple control law. The primary interaction of the two is in the determination of the convergence rate, \( c \). Should the state lie within a phase sphere described by the deadbands, case 2 or 3 applies, and the problem of determining a convergent velocity is solved by setting \( c \) very small, or zero. Otherwise, in case 1 or 4 \( c \) is typically set equal to a larger value. Due to response time requirements, or other considerations, this value may be scheduled in any of several ways. For example, \( c \) may be a linear...
function of $|x_e|$. In the following section, it will be seen that the new autopilot uses a second phase sphere, and sets $c$ to one of two discrete values, depending on whether the state is within the inner sphere, outside the outer sphere, or between the two spheres.

In several applications, the numerical values of $db_i$ are greatly different. For example, in the spacecraft problem, an attitude deadband may be 0.1 degrees, and a position deadband 50 feet. Generating the convergent velocity along unit ($x_d - x$) will direct the rate change along the numerically largest variable, rather than that furthest beyond its deadband. By generating this velocity along -unit ($x_e$), the rate change is directed along the variable largest in relation to its deadband. Otherwise, serious overcontrolling of some variables (such as position above) and undercontrolling of others (attitude) may result.

Thus, the phase space control law uses a modified velocity to be gained principle,

$$\dot{x}_g = -c \text{ unit } (x_e) + (\dot{x}_d - \dot{x}_n) \quad (3-20)$$

where the value of $c$ is determined by consideration of whether $x_e$ is in the phase sphere, as outlined above. The following are some specific developments of this basic concept, addressing particular aspects of system behavior.

3.3.1 **Limit Cycles - Hysteresis**

Limit cycling is the tendency of a system to oscillate about a desired state, generally due to the system's inability to exactly achieve a target state with zero relative rate, or to maintain that state due to constraints or imperfections in control, disturbances, or feedback uncertainties. Spacecraft, for example, can be driven to a desired attitude but will drift from that attitude due to gravity gradients, outgassing,
thermal effects, and atmospheric drag. Control is applied to overcome this drift, and to drive the attitude toward the desired attitude. The vehicle may overshoot the desired attitude, or disturbances may drive the spacecraft from this attitude, so that further control must be applied. A typical result is that the vehicle may oscillate about the desired attitude. This is attitude limit cycling.

When a system's state is within allowable bounds (i.e., $\dot{x}_e$ lies within the phase sphere) but its relative rate is nonzero, it is prone to limit cycling. When one of the position state variables approaches its bound, it is necessary to reverse the component of relative rate driving that variable, i.e., to change the sign of its derivative. To do so at the exact point when that variable hits the limit is impractical as the variable will overshoot the limit during the finite interval necessary to reverse the appropriate component of relative rate. In this case hysteresis is desirable.

A second phase sphere provides a simple means to avoid this problem. If the original phase sphere is centered on $x_d$, with radius $r_{ps}$, the second is placed concentric to it, with radius $(r_{ps} - b)$. When the state of the vehicle drifts from the desired state, it intercepts the inner sphere before any of the state variables reaches its limit. The parameter $b$ is then chosen to assure that the appropriate component of relative rate can be reversed before any state variable reaches its deadband. An expression for $b$ might be

$$b = \frac{c_{\dot{h}C}^2}{a_{\min}}$$

where $c_{\dot{h}C}$ is a desired limit cycle rate and $a_{\min}$ is the minimum component of available acceleration in any of the state variable directions. Since the magnitude of available control will generally be different along each
of the state variables, \( b \) actually is a function of the component of relative rate being reversed. As a result, the inner phase sphere would actually be a more complicated surface. By making it a sphere with the radius given in Eq. (3-21) some simplicity is gained, at the cost of more precise control than is required in some state variables.

The convergence rate, \( c \), will be set to one of two values depending on the location of \( x_e \) relative to the phase spheres. If it is outside the outer phase sphere, the value of \( c \) will be such as to ensure sufficiently rapid convergence on the desired state. If the state is within the outer sphere, \( c \) will be set to a reasonable limit cycle rate.

Figure 3-1 is a two-dimensional picture of a limit cycle trajectory. Starting at \( a \), with a small relative rate, the state coasts until it hits the inner phase sphere at \( b \). Control is applied to reverse the relative rate, and the state reenters the inner phase sphere at \( c \). Again the state coasts until leaving the inner sphere at \( d \), where control is again applied to reverse the relative rate, and the state reenters the inner sphere at \( e \).

Figure 3-2 shows a convergent trajectory. Starting with a large state error at \( a \), the state is driven at the chosen convergent rate toward the desired state. Upon intercepting the outer sphere at \( b \), the value of \( c \) in the velocity to be gained expression is changed to the limit cycle rate, and control applied to achieve this new rate at point \( c \), where a limit cycle is initiated.

3.3.2 Rate Hysteresis

Control of the rate of a system is often imprecise and the implementation of rate change requests will generally lag the requests. For the case of these requests changing significantly faster than the rate response, rate hysteresis is necessary. This hysteresis takes the form
Figure 3-1. Two-dimensional limit cycle trajectory; *'s indicate equal time intervals.

Figure 3-2. Convergent trajectory; *'s indicate equal time intervals.
of a phase sphere applied to the rate variables. In Figure 3-3 is shown a rate change request, where \(-c_1x_E\) is the desired convergent velocity, \(\dot{x}_n\) is the actual rate, and \(v_g\) is the resultant velocity to be gained. At times, \(v_g\) will be smaller than the granularity of the control effectors, or within the noise level of the overall system. In such cases, it is not reasonable to implement the request, so the control law must zero it. To do so, the magnitude of \(v_g\) is checked, and if it is smaller than some predetermined value, it is ignored. This is equivalent to controlling rate to within some tolerance, rather than exactly.

![Figure 3-3. Rate change request.](image)

The system being controlled may have nonlinear control effectors or other nonlinearities; however, a \(v_g\) request such as that generated by the phase space control law is based on linear approximations to obtain the desired rate change. This is especially significant when system errors are large (recall the discussion of rotation matrices in Chapter 2). Also, during the interval required to drive the system to the target state, control anomalies or disturbances may alter \(v_g\), as
in Figure 3-4. A saving in control cost will generally be realized by the application of rate hysteresis. When the system has large state errors, the precision of the control law may be relaxed and coarse control may be applied to the system to drive it toward the target state. As the actual state converges on the target state, the nonlinearities and disturbance errors grow less significant and a more rigorous control can be applied to drive the state to a close tolerance about the target.

Figure 3-4. State trajectory and rate change requests in 2 dimensions; trajectory is curved due to system nonlinearities and disturbances, and requests change due to these effects.
3.3.3 Disturbance Accelerations

In some instances, a disturbance may exist which will remain constant or nearly so for a large number of autopilot cycles. Such disturbances may be due to venting, gravity gradient torques, or thrust vector offsets. An onboard state estimator will often be capable of identifying such disturbances, so that appropriate corrective action may be taken.

With no change to the current algorithm, a disturbance will cause an offset limit cycle, as in Figure 3-5. The state is driven away from the target by the disturbance until reaching the phase sphere. Depending on the magnitude of the disturbance relative to the available control

Figure 3-5. Limit cycle with disturbance acceleration and no acceleration compensation.
forces or torques, the trajectory may be temporarily reversed by the control effectors, establishing an offset limit cycle with jet firings every few seconds, or continue away from the target, hence loss of control.

When sufficient control authority exists, a desirable limit cycle can be established. The principal features of this desired limit cycle are that the errors never exceed their deadbands, and a long period. By reducing the frequency of jet firings, fuel is conserved. Further, a longer period will require longer jet firings than a short period, resulting in greater jet efficiency, as discussed in Section 2.2. A means of establishing and maintaining such a limit cycle, based on a technique suggested by Turkovich\textsuperscript{12}, follows.

Consider the situation of Figure 3-6. The vehicle state is indicated by $\mathbf{x}_i$ and the disturbing acceleration by $\mathbf{a}_d$. In this section, all vectors will denote actual physical quantities, rather than the normalized quantities of Section 3.2. It is required to determine the initial rate $\dot{\mathbf{x}}_i$ relative to the target state such that the state follows trajectory $\mathbf{x}_{iBC}$ of Figure 3-6. This trajectory carries the state toward a point $B$, selected so that the state will then coast to a point $C$, described by the intersection of the point $\mathbf{a}_d$ and the phase sphere.

Turkovich\textsuperscript{12} shows that for zero average state error in the ensuing limit cycle trajectory, the component of displacement of point $B$ from the target parallel to $\mathbf{a}_d$ is given by

$$\beta = \frac{r_{ps}}{3} \left( 1 + \frac{|\mathbf{a}_d|}{\mathbf{a}_c \cdot \mathbf{i}_d} \right)$$

(3-22)

where $r_{ps}$ is the radius of the phase sphere, $\mathbf{a}_d$ the disturbing acceleration, $\mathbf{a}_c$ the control acceleration, and $\mathbf{i}_d$ the unit vector along $\mathbf{a}_d$. In the case of different deadbands for each axis, $r_{ps}$ would become a function
of the direction of \( \mathbf{a}_d \), depending on the relative values of the deadbands in each axes. The component of displacement of point B perpendicular to \( \mathbf{a}_d \) is selected to cause the state to arrive at point C, rather than overshoot to point D.

![Diagram](image.png)

**Figure 3-6. Desired disturbance trajectory.**

By so targeting the state, a limit cycle on or quite near the trajectory CEC is obtained, using less propellant than one along AED due to the lower component of rate perpendicular to \( \mathbf{a}_d \).

To obtain the desired rate \( \dot{x}_B \), begin with the expression for the rate \( \dot{x}_B \) at point B.

\[
\dot{x}_B = \dot{x}_i + \mathbf{a}_d t_1
\]

(3-23)

where \( t_1 \) is the time required for the state to travel from \( x_i \) to point B, \( x_B \). Taking the component parallel to \( \mathbf{a}_d \),
\[ \dot{x}_B \cdot i_d = (\dot{x}_i + a_d t_1) \cdot i_d \quad (3-24) \]

Noting the trajectory reversal at point B, we want this component of \( \dot{x}_B \) to be zero, giving

\[ \dot{x}_i \cdot i_d = -a_d t_1 \cdot i_d \quad (3-25) \]

from which

\[ t_1 = -\frac{\dot{x}_i \cdot i_d}{a_d \cdot i_d} \quad (3-26) \]

\[ = -\frac{\dot{x}_i \cdot i_d}{|a_d|} \quad (3-27) \]

The state at point B, \( x_B \), is given by

\[ x_B = x_i + \dot{x}_i t_1 + \frac{1}{2} a_d t_1^2 \quad (3-28) \]

Inserting Eq. (3-27), this becomes

\[ x_B = x_i - \dot{x}_i \left[ \frac{\dot{x}_i \cdot i_d}{|a_d|} \right] + \frac{1}{2} a_d \left[ \frac{\dot{x}_i \cdot i_d}{|a_d|} \right]^2 \quad (3-29) \]

However, the component of \( x_B \) parallel to \( a_d \) is given by Eq. (3-22), thus

\[ -\beta = x_B \cdot i_d = x_i \cdot i_d + \frac{1}{2} (a_d \cdot i_d) \left[ \frac{\dot{x}_i \cdot i_d}{|a_d|} \right]^2 \]

\[ - (\dot{x}_i \cdot i_d) \left[ \frac{\dot{x}_i \cdot i_d}{|a_d|} \right] \quad (3-30) \]

which reduces to

\[ [\beta + (x_i \cdot i_d)]|a_d| = \frac{1}{2} (\dot{x}_i \cdot i_d)^2 \quad (3-31) \]
giving

\[
\begin{align*}
\mathbf{v}_i \cdot \mathbf{i}_d &= \sqrt{2|a_d|[(\mathbf{x}_i \cdot \mathbf{i}_d) + \beta]} \\
\mathbf{x}_C &= \mathbf{x}_B + \mathbf{x}_B t_2 + \frac{1}{2} a_d t_2^2 \\
(\mathbf{x}_C \cdot \mathbf{i}_d) &= (\mathbf{x}_B \cdot \mathbf{i}_d) + (\mathbf{x}_B \cdot \mathbf{i}_d) + \frac{1}{2} |a_d| t_2^2
\end{align*}
\] (3-32)

Placing Eq. (3-32) into Eq. (3-27),

\[
t_1 = \sqrt{2|\mathbf{x}_i \cdot \mathbf{i}_d| + \beta} |a_d|
\] (3-33)

The trajectory from B to C, \( \mathbf{x}_C \), is described by

\[
\mathbf{x}_C = \mathbf{x}_B + \mathbf{x}_B t_2 + \frac{1}{2} a_d t_2^2
\] (3-34)

Taking the component of this trajectory parallel to \( a_d \) gives

\[
(\mathbf{x}_C \cdot \mathbf{i}_d) = (\mathbf{x}_B \cdot \mathbf{i}_d) + (\mathbf{x}_B \cdot \mathbf{i}_d) + \frac{1}{2} |a_d| t_2^2
\] (3-35)

but from Figure 3-6,

\[
\mathbf{x}_C \cdot \mathbf{i}_d = r_{ps}
\] (3-36)

where \( r_{ps} \) is the radius of the phase sphere, and

\[
\mathbf{x}_B \cdot \mathbf{i}_d = -\beta
\] (3-37)

and it was previously noted that

\[
\mathbf{x}_B \cdot \mathbf{i}_d = 0
\] (3-38)

so

\[
r_{ps} = -\beta + 0 + \frac{1}{2} |a_d| t_2^2
\] (3-39)

giving
Substituting Eq. (3-28) and (3-23) into Eq. (3-34), we obtain

\[ x_C = x_1 + \frac{1}{2} a_d t_1^2 + \dot{x}_1 t_1 + (\dot{x}_1 + a_d t_1) t_2 + \frac{1}{2} a_d t_2^2 \]  

(3-41)

and noting that

\[ x_C = -r_{ps} i_d \]  

(3-42)

one obtains

\[ x_i \]  

\[ \frac{1}{(t_1 + t_2)} \left[ -r_{ps} i_d - x_i - a_d \left( \frac{1}{2} (t_1^2 + t_2^2) + t_1 t_2 \right) \right] \]  

(3-43)

Thus, when a disturbance is identified by the state estimator, the rate change will be given by

\[ V_g = \dot{x}_i + (\dot{x}_d - \dot{x}) \]  

(3-44)

and the calls to jet select inhibited until point C is attained.

This technique, though it achieves improved fuel economy over the basic algorithm, is suboptimal. It is based on achieving zero average rate, rather than fuel optimization. A subject for continued study on the phase space autopilot is the optimization of fuel consumption in disturbance limit cycles.

3.4 Application to Spacecraft Control

The equations of motion of a rigid body are given in Chapter 2. Under the assumptions stated, uncoupled linear equations can describe spacecraft motion, and form a basis for application of the phase space...
control law to spacecraft control. Six state variables are used, three
describing rotation with respect to some inertial nonrotating reference
axes and three describing translation also with respect to inertial non-
rotating reference axes. It is convenient to choose body fixed reference
axes, as the control effectors have fixed acceleration vectors in these
axes; also, the crew typically thinks in these axes. The error state
vector is given as

\[
\mathbf{x}_e = \begin{bmatrix}
\theta_{d1} - \hat{\theta}_1 \\
\theta_{d2} - \hat{\theta}_2 \\
\theta_{d3} - \hat{\theta}_3 \\
x_{d1} - \dot{x}_1 \\
x_{d2} - \dot{x}_2 \\
x_{d3} - \dot{x}_3 \\
\end{bmatrix}
\]

(3-32)

where \( \theta_{d1} \) is the desired attitude about the \( i^{th} \) axis, \( \hat{\theta}_1 \) is the measured
attitude about the \( i^{th} \) axis, \( x_{d1} \) is the desired position on the \( i^{th} \) axis,
and \( \dot{x}_1 \) is the measured position on the \( i^{th} \) axis. Each element of this
vector is to be controlled to within a separate deadband \( db_i \). The relative
rate vector is

\[
-(\ddot{x}_{d} - \dot{x}) = -\begin{bmatrix}
\omega_{d1} - \dot{\omega}_1 \\
\omega_{d2} - \dot{\omega}_2 \\
\omega_{d3} - \dot{\omega}_3 \\
v_{d1} - \dot{v}_1 \\
v_{d2} - \dot{v}_2 \\
v_{d3} - \dot{v}_3 \\
\end{bmatrix}
\]

(3-33)
where \( \omega_{d_i} \) is the desired angular velocity about the \( i^{th} \) axis, \( \hat{\omega}_i \) is the measured angular velocity about the \( i^{th} \) axis, \( v_{d_i} \) is the desired linear velocity along the \( i^{th} \) axis, and \( \hat{v}_i \) is the measured linear velocity along the \( i^{th} \) axis. The velocity to be gained expression becomes

\[
\begin{align*}
\mathbf{v}_g &= \begin{bmatrix}
\Delta \omega_1 \\
\Delta \omega_2 \\
\Delta \omega_3 \\
\Delta v_1 \\
\Delta v_2 \\
\Delta v_3 
\end{bmatrix} = \mathbf{c} \text{ unit} \begin{bmatrix}
\hat{\omega}_1 \\
\hat{\omega}_2 \\
\hat{\omega}_3 \\
\hat{v}_1 \\
\hat{v}_2 \\
\hat{v}_3 
\end{bmatrix} + \begin{bmatrix}
\omega_{d_1} - \hat{\omega}_1 \\
\omega_{d_2} - \hat{\omega}_2 \\
\omega_{d_3} - \hat{\omega}_3 \\
v_{d_1} - \hat{v}_1 \\
v_{d_2} - \hat{v}_2 \\
v_{d_3} - \hat{v}_3 
\end{bmatrix} \\
(3-34)
\end{align*}
\]

Appropriate radii for the phase spheres are selected as are the normalizations based on these radii. The phase space control law is then implemented by the control effectors, be they magnetic torquers, gyro torquers, or RCS jets. If the control effectors are jets, the problem of determining which jets are to be fired, and their firing intervals, is the so-called jet selection problem, and is the subject of the next chapter.
CHAPTER 4

JET SELECTION

The phase space control law requests rate changes to be produced by the control effectors. The logic involved in translating these requests into a sequence of jet firings, when the effectors are reaction jets, is called jet selection. Current spacecraft autopilots typically use either table lookup or dot product schemes, which will be covered briefly in the first two sections of this chapter. In his doctoral thesis, Bard S. Crawford investigated linear programming to solve the jet selection problem. A practical algorithm for using linear programming for jet selection was developed by Craig Work, of the Charles Stark Draper Laboratory. The balance of this chapter is a discussion of this algorithm, and developments to it arising in the course of this research.

4.1 Table Lookup Jet Selection

Many spacecraft designers attempt to have jets carefully placed to provide pure couples and pure translations when fired in the correct groups. Thus, the same set of jets can always be used to obtain a certain angular acceleration about a body fixed axis, and linear acceleration along a body fixed axis. The problem of jet selection is then a simple matter of firing appropriate jets for certain intervals to satisfy the control law request. An example system is shown in Figure 4-1 for controlling rotation about a body fixed y axis (out of page) and translation along the x and z axes, as shown. A jet select table for this system is given in Table 4-1.
Figure 4-1. Simplified spacecraft jet configuration.

Table 4-1. Jet select table and resultant accelerations for system of Figure 4-1.

\[ m = \text{vehicle mass} \]
\[ I_y = \text{moment of inertia about vehicle Y axis} \]

<table>
<thead>
<tr>
<th>Direction of Linear Acceleration</th>
<th>Jets Fired</th>
<th>Resultant Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>+X</td>
<td>5</td>
<td>( T_5 /m )</td>
</tr>
<tr>
<td>-X</td>
<td>6</td>
<td>( T_6 /m )</td>
</tr>
<tr>
<td>+Z</td>
<td>2,4</td>
<td>( (T_2 + T_4) /m )</td>
</tr>
<tr>
<td>-Z</td>
<td>1,3</td>
<td>( (T_1 + T_3) /m )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Direction of Angular Acceleration</th>
<th>Jets Fired</th>
<th>Resultant Acceleration</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Y</td>
<td>2,3</td>
<td>( \left[ (r_2 \times T_2) + (r_3 \times T_3) \right] / I_y )</td>
</tr>
<tr>
<td>+Y</td>
<td>1,4</td>
<td>( \left[ (r_1 \times T_1) + (r_4 \times T_4) \right] / I_y )</td>
</tr>
</tbody>
</table>

Table lookup jet selection algorithms are simple to implement, and require relatively little CPU time for flight computer operation. However, for more complicated systems than that of the example, they grow large to account for a larger number of possible commands and jets.
They tend to be inflexible, however, requiring additional tables to accommodate jet failures, changing vehicle mass properties or firing policy restrictions such as nose jets only, no tail jets under certain conditions, etc. While they are quite suitable for simple jet configurations, jet select tables become extremely burdensome for more complex vehicles.

4.2 Dot Product Jet Selection

On some spacecraft, the assignment of jets to fixed control directions is not simple, and may be impractical to implement as a table. For example, the engines may gimbal, be skew to the control axes, or suffer impingement effects. Flexibility in jet selection may be required to account for jet failures, changing vehicle mass properties, or firing policy.

One approach to solving some of these problems is a dot product jet selection. In this scheme, a dot product $D$, 

$$ D = \mathbf{a}_j \cdot \mathbf{a}_{\text{desired}} \quad (4-1) $$

where $\mathbf{a}_j$ is the vehicle acceleration due to jet firings, and $\mathbf{a}_{\text{desired}}$ is the desired acceleration, is maximized. Expanding to include several jets, failure status and crude fuel minimization, one might examine

$$ D = \mathbf{a}_{\text{desired}} \cdot \left[ \frac{\# \text{ jets}}{i=1} \mathbf{a}_i \delta_i f_i c_i \right] \quad (4-2) $$

where $\mathbf{a}_i$ is the vehicle acceleration produced by firing jet $i$, $f_i$ is a weighting factor for jet $i$ based on fuel consumption and other factors (i.e., response time),

71
\[ \delta_i = \begin{cases} 1.0 & \text{if jet } i \text{ is operative} \\ 0 & \text{if jet } i \text{ is failed} \end{cases} \quad (4-3) \\
\]

and

\[ c_i = \begin{cases} 1.0 & \text{if jet } i \text{ is commanded on} \\ 0 & \text{if jet } i \text{ is commanded off} \end{cases} \quad (4-5) \]

The dot product jet selection provides more versatility than the table lookup and does not require implementation of several detailed tables. Methods must be devised for seeking the maximum of this dot product given certain constraints, costing greater flight computer burden. Implementations of Eq. (4-2) do exist, as in the Space Shuttle vernier RCS jet select.

For a system with a large number of jets the dot product may also become cumbersome. Many iterations are needed to find the best jet combinations, and these may change rapidly, causing unacceptable burdens on the flight computer. Moreover, the components of \( a_j \) not along \( a_{\text{desired}} \) may be troublesome or undesirable.

4.3 Jet Selection by Linear Programming

The new autopilot uses linear programming to solve the jet selection problem to minimize fuel costs. In this section, linear programming problems are first presented. The application of linear programming to jet selection is then discussed and several practical modifications to the basic scheme are introduced.

4.3.1 Linear Programming

Linear Programming is a technique of mathematical optimization applicable to a certain class of problems; these problems have three characteristics in common:
Each problem has a linear objective function which is to be either minimized or maximized.

Each problem has a set of linear constraints. These constraints are characteristically inequalities, although some may be equalities. Their number is unrestricted except for the practical consideration of the computational load.

Each of the variables is bounded to be non-negative. This is a consequence of the nature of the problems for which linear programming was originally developed.

The general linear programming problem, then, is to maximize (or minimize) a function,

$$f(x) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$  \hspace{1cm} (4-7)

subject to a set of constraints

$$Bx \leq p_0$$  \hspace{1cm} (4-8)

and

$$x_i \geq 0, \text{ for all } i$$  \hspace{1cm} (4-9)

Two important observations are to be made concerning linear programming problems. First, the solutions to the constraints (Eq.(4-8) and (4-9)) form a convex set. Second, the optimal solution which minimizes or maximizes the objective function of Eq.(4-2) is an extreme point of this convex set. An extreme point is the optimal solution, if such exists.

To understand the first, we consider a set of points (vectors) in n-space. Given two such points $x_1$, $x_2$, a segment joining $x_1$, $x_2$ is the locus of points $x$ given by

$$x = ax_1 + (1 - a)x_2$$  \hspace{1cm} (4-10)

73
where \( a \) is a scalar such that

\[
0 < a < 1 \quad (4-11)
\]

A convex set is defined as a set of points such that the segment joining any two points in the set is also in the set. The first observation follows from this definition, as a set of inequalities can only define a convex set. For a set defined by Eq. (4-10), and a set of linear functions,

\[
f_i(x) > 0 \quad (4-12)
\]

Satisfied by \( x_1, x_2 \) in the set defined by Eq. (4-10), convexity requires

\[
f_i(x_3) = f_i(ax_1 + (1 - a)x_2) = af_i(x_1) + (1 - a)f_i(x_2) \quad (4-13)
\]

but

\[
f_i(x_1), f_i(x_2) > 0 \quad (4-14)
\]

so \( f_i(x_3) \) is always positive if

\[
a > 0 \quad (4-15)
\]

and

\[
(1 - a) > 0 \quad (4-16)
\]

implying that

\[
0 < a < 1 \quad (4-17)
\]

For example, a set of inequalities can define the two-dimensional set in Figure 4-2a but not that in Figure 4-2b. Further, two points A, B can be found in Figure 4-2b, such that the line segment joining them is not within the set. No such pair of points can be found in Figure 4-2a.
Figure 4-2a. An example convex set.

Figure 4-2b. An example of a set which is not a convex set.
An extreme point of a convex set is any point in the set which is not on any segment between two other points in the set. Mathematically, if \( x_1 \) is an extreme point in a set and \( x_2 \) any other point in that set, then the only point \( x_3 \) in that set which satisfies

\[
x_1 = ax_3 + (1 - a)x_2
\]

(4-19)

is

\[
x_3 = x_1, \quad a = 1
\]

(4-20)

which is the extreme point. Extreme points in \( n \) space are the intersections of \( n \) hyperplanes; such intersections are called corners.

To show that the optimal solution of a linear programming problem lies at an extreme point, assume \( x_0 \) to be the maximal solution. This maximal solution can be expressed as a linear combination of the extremum solutions

\[
x_0 = a_1x_1^e + a_2x_2^e + \ldots + a_tx_t^e
\]

(4-21)

where

\[
0 \leq a_i \leq 1
\]

(4-22)

and

\[
\sum_{i=1}^{t} a_i = 1
\]

(4-23)

and the \( x_i^e \) are the extreme point solutions. Since the cost function \( f(x) \) is linear,

\[
f(x_0) = f(a_1x_1^e + a_2x_2^e + \ldots + a_tx_t^e)
\]

\[
= a_1f(x_1^e) + a_2f(x_2^e) + \ldots + a_tf(x_t^e)
\]

(4-24)

(4-25)
If one assumes that $f(x_e)$ is the largest of the $f(x)$, since all the $a_i$ are non-negative,

\[
a_1 f(x_1^e) + a_2 f(x_2^e) + \ldots + a_L f(x_L^e) 
\leq a_1 f(x_m^e) + a_2 f(x_m^e) + \ldots + a_L f(x_m^e) \quad (4-26)
\]

\[
\leq (a_1 + a_2 + \ldots + a_L) f(x_m^e) \quad (4-27)
\]

but by Eq.(4-23),

\[
(a_1 + a_2 + \ldots + a_L) f(x_m^e) = f(x_m^e) \quad (4-28)
\]

and by substituting Eq.(4-28) into Eq.(4-24)-(4-27), one finds

\[
f(x_0) \leq f(x_m^e) \quad (4-29)
\]

However, $f(x_0)$ is the maximum, so $f(x_m^e)$ must also be the maximum, and equal $f(x_0)$. Thus, although the extreme points are not the only optimal points, they are a complete set of optimal points. The search for an optimal solution need only consider those points which satisfy the constraints and are extrema (called feasible points), reducing the number of points to be checked to a finite number. In many problems, the number of extreme points is often too large to check one by one. Fortunately, techniques such as the simplex method further reduce the number of points checked, as it proceeds from a given extremum only to those of lower cost. The simplex method, in modified form, is used by the new autopilot to solve the linear programming jet select problem.
4.3.2 The Simplex Method

The simplex method is a means of solving linear programming problems. A few concepts are necessary before describing the algorithm: i.e., types of solutions, a vector basis, and an independent subset.

A solution is any vector $\mathbf{x}$ which satisfies the equality constraints in Eq. (4-8). If it also satisfies the inequality constraints of Eq. (4-8) it is a feasible solution. A basic feasible solution is a feasible solution $\mathbf{c}$, with no more nonzero components than equality constraints imposed on the problem, and the activity vectors $\mathbf{x}_i$ associated with these nonzero components form a basis of a finite dimensional vector space if they span the space. That is, any vector $\mathbf{x}$ in that space can be represented as a linear combination of the basis vectors.

Starting with any basic feasible solution, the Simplex method seeks to replace vectors in the basis with other vectors not in the basis. Those vectors which are to be replaced are selected to maintain the feasibility of the solution, while those to be admitted to the basis are determined on the basis of their contribution to improving cost. The process is repeated until an optimal solution is found, or it is decided that no other feasible solution exists.

Two basic rules govern this scheme, the first treating the generation of new basic feasible solutions, the second concerned with the selection of vectors to be added to the basis. The constraints on the linear programming problem can be written

$$\sum_{i=1}^{n} b_i x_i + \sum_{i=n+1}^{m} b_i x_i = p_0$$

(4-30)

where $x_{n+1} \ldots x_m$ represent slack variables and $b_{n+1} \ldots b_m$ corresponding slack vectors. Assume a basis, $B$, exists.

$$B = [b_{(1)}, b_{(2)}, \ldots b_{(m)}]$$

(4-31)
where the \( x_{(i)} \) are an independent subset (i.e., a subset of all \( x_i \) which are linearly independent) such that

\[
\sum_{i=1}^{m} \theta(i) b(i) = b_0 \tag{4-32}
\]

and

\[
\theta(i) \geq 0 \tag{4-33}
\]

for all \( i \). Then the basic feasible solution is

\[
x_{(i)} = \theta(i) \quad i = 1 \text{ to } n \tag{4-34}
\]

with

\[
x_{(i)} = 0 \quad i = n + 1 \text{ to } m \tag{4-35}
\]

Any vector \( x_k \) not in the basis can be expressed by

\[
x_k = \sum_{i=1}^{m} y(i)_k b(i) \tag{4-36}
\]

or

\[
x_k = BY_k \tag{4-37}
\]

where

\[
x_k = \begin{bmatrix}
    y(1)_k \\
    y(2)_k \\
    y(3)_k \\
    \vdots \\
    y(m)_k
\end{bmatrix} \tag{4-38}
\]
so that

\[ y_k = B^{-1}x_k \]  

(4-39)

The first rule states that a new basis and a new feasible solution can be obtained by including \( x_k \) in the basis in place of some \( b_i \) in the basis, if \( b_i \) is chosen to maintain feasibility of the solution. To show this, multiply Eq. (4-36) by an unknown positive scalar \( \theta_k \), and add to Eq. (4-32).

\[ \sum_{i=1}^{m} \theta(i)b(i) + \theta_k x_k = E_0 + \sum_{i=1}^{m} \theta_k y_i b(i) \]  

(4-40)

or

\[ \sum_{i=1}^{m} (\theta(i) - \theta_k y_i) b(i) + \theta_k x_k = E_0 \]  

(4-41)

A basic feasible solution has no more than \( m \) positive values among the \( n \) variables. We have \( m + 1 \) terms, one too many, in the left hand side of Eq. (4-41). It is necessary to make one of the coefficients \( (\theta(i) - \theta_k y_i) \) equal zero. This is accomplished by finding the minimal value of \( \theta_k \) which will make one of the coefficients the first to go to zero, to keep the other coefficients non-negative. If this is the \( i^{th} \) coefficient, then

\[ \theta_k \]  

(4-42)

\[ = \min \left( \frac{\theta(i)}{y(i)_k} \right) \]  

(4-43)

Having thus determined \( \theta_k \), the new basis variables \( \theta(i) \) can be computed to give a new basic feasible solution.
The second rule governs the selection of the vector to be admitted to the basis. Given the original solution

\[ \mathbf{x}(i) = \theta(i), \quad i = 1 \text{ to } m \]  \hspace{1cm} (4-44)

where

\[ \theta'(i) = \theta(i) - \frac{\theta(k)}{y(i)_k} y(i)_k', \quad i = (1), (2), \ldots (m) \]  \hspace{1cm} (4-45)

and

\[ \theta'(k) = \theta'(k) \]  \hspace{1cm} (4-46)

\[ = \frac{\theta(k)}{y(k)_k} \]  \hspace{1cm} (4-47)

The second rule governs the selection of the vector to be admitted to the basis. Given the original solution

\[ \mathbf{x}(i) = \theta(i), \quad i = 1 \text{ to } m \]  \hspace{1cm} (4-48)

with corresponding cost

\[ f(x) = \sum_{i=1}^{m} c(i) \theta(i) \]  \hspace{1cm} (4-49)

and a new basis vector \( \mathbf{x}_j \). From Eq.(4-45) and (4-46), \( \theta'_j \) is known, and the new cost \( f(x') \) is

\[ f(x') = \sum_{i=1}^{n} c(i) \left[ \theta(i) - \theta'_j y(i)_j \right] + c_j \theta'_j \]  \hspace{1cm} (4-50)

\[ = \sum_{i=1}^{n} \left[ c(i) \theta(i) - c(i) y(i)_j \theta'_j \right] + c_j \theta'_j \]  \hspace{1cm} (4-51)

\[ = f(x) - \theta'_j (z_j - c_j) \]  \hspace{1cm} (4-52)
where
\[ z_j - c_j = \sum_{i=1}^{n} \left( c_i y_i j \right) - c_j \] (4-53)

The second term of the right hand side of Eq.(4-52) is often called the saving. If \( f(x) \) is to be maximized,
\[ f(x') > f(x) \] (4-54)
implying
\[ \theta_j (x_j - c_j) < 0 \] (4-55)
However, \( \theta_j \) is greater than zero, so
\[ (z_j - c_j) < 0 \] (4-56)
then \( x_k \) is one of the \( x_j \) not in the basis for which the corresponding \( z_j - c_j \) satisfies Eq.(4-56). Several such vectors often exist, and in the interest of computational efficiency, the first \( x_j \) for which the above holds is taken. Were the problem to minimize \( f(x) \),
\[ f(x') < f(x) \] (4-57)
so that
\[ (z_j - c_j) > 0 \] (4-58)

The solution is optimal when no quantity \( (z_j - c_j) \) is less than zero for minimization, or greater than zero for maximization. As a practical matter, this test is relaxed slightly, by considering the solution optimal when there is no \( (z_j - c_j) \) such that
\[ |z_j - c_j| > db \] (4-59)
where \( db \) is determined by the uncertainties in the computation.
Summarizing the steps in the Simplex method, one proceeds as follows: To minimize $f(x)$, given a basic feasible solution $X$ and its associated basis, compute $y$ by Eq. (4-39) and

1. Compute $z_j - c_j$ for all vectors not in the basis. If all $|z_j - c_j| < db$, or all $(z_j - c_j) \leq 0$, we are finished. Otherwise, continue.

2. Select the $x_j$ associated with the most positive $(z_j - c_j)$; call it $x_k$.

3. Select the basis vector to be replaced by $x_k$: call it $x_{(k)}$.

4. Compute the new solution by Eq. (4-44)-(4-47) and compute the new $f(x)$.

5. Compute $y$; return to 1.

The problem remains to find an initial basic feasible solution. Most often this is based on other considerations of the problem and is not covered here.

4.3.3 Application of Linear Programming to Jet Selection

Linear programming as a means of jet selection has been studied as early as 1968. Discussed in this section is a practical implementation of linear programming as a means of jet selection.

Jet selection seeks to achieve a six-dimensional rate change $w$ requested by the control law. Posed as a linear programming problem, the jets are modeled by their activity vectors, which will form the basis. A jet activity vector, $a_j$, is defined to be the rate change obtained by firing jet $j$ for unit time. For jet $j$, with thrust vector $T_j$, displaced by $r_j$ from the vehicle mass center, $a_j$ is given by
\[
\mathbf{a}_j = \begin{bmatrix}
\mathbf{I}^{-1}(\mathbf{r}_j \times \mathbf{T}_j) \\
\mathbf{T}_j / m
\end{bmatrix}
\] (4-60)

in the coordinate system of \( r_j \) and \( T_j \). Here \( \mathbf{I}^{-1} \) is the inverse of the vehicle's inertia tensor and \( m \) is the vehicle mass. \( \mathbf{a}_j \) is a six-dimensional vector with the three components above the partition rotational rate change, and those below the partition representing translational rate change. This representation assumes that the second-order terms for \( \mathbf{a}_j \) are negligible. Since the state estimator closes the loop at short intervals, this is a safe approximation to reality.

Several cost functions could be selected depending on the requirements levied on the system. A minimum fuel system may use fuel consumed as the cost function \( f(x) \), i.e.,

\[
f(x) = \sum_{i=1}^{\text{jets}} x_i f_i
\] (4-61)

where \( f_i \) is the rate of fuel consumption for jet \( i \), and \( x_i \) is its firing time. A truly minimum fuel solution, however, is not to fire any jets and satisfy the request over an infinite interval. Thus, it is necessary to either constrain or minimize response time. Thus, inequality constraints on firing time may be posed in the form,

\[
x_i \leq \text{max response time, all } i
\] (4-62)

and

\[
x_i > 0, \text{ some } i
\] (4-63)

Alternatively, one may seek to minimize response time, such as

\[
f(x) = \max(x_i)
\] (4-64)
or a linear combination of response time and fuel consumption,

\[ f(x) = e \sum_{i=1}^{\text{# jets}} x_i f_i + g \max(x_i) \quad (4-65) \]

where \( e \) and \( g \) are weighting factors, determined by the nature of the problem.

Equality constraints for the problem are based on the requirement that the rate change be correctly implemented.

\[ W = \sum_{i=1}^{\text{# jets}} a_i x_i \quad (4-66) \]

Clearly, the jets cannot fire for a negative interval. Mathematically, this observation corresponds to the requirement that the coefficients of the basis vectors be non-negative, posing additional constraints.

\[ x_i \geq 0, \text{ all } i \quad (4-67) \]

The jet selection problem, posed as a linear programming problem, is to minimize a cost function such as that of Eq.(4-61), (4-62) or that of Eq.(4-63) subject to the equality constraints of Eq.(4-66) and the inequality constraints of Eq.(4-67) and possibly those of Eq.(4-62) and (4-63). To be tractable, this problem must have the following properties:
(1) The cost function must be linear and have a minimum.
(2) The equality constraints must admit one or several solutions.
(3) The inequality constraints must hold for one or more of these solutions.

The first assumption insures that it is reasonable and possible to minimize the cost function via linear programming. The second requires that it is possible to provide the rate change with the control effectors. The third states first that the sense of the activity vectors is correct (i.e., not all jets fire against the request), and that the maneuver can be performed in a reasonable time.

The above discussion is the basis of a linear programming jet select. However, practical implementation of the algorithm calls for close scrutiny of the vehicle to be controlled, the control effectors, and the applicability of the algorithm to these systems. In the following subsections are some practical modifications to the basic algorithm developed in the course of applying it to a real spacecraft.

4.3.3.1 Selection from a Large Number of Jets

Finding a solution to the linear programming jet select problem involves testing each of the available jets not in the basis to determine which jet to bring into the basis. A cost reduction for each jet is computed and that jet which maximizes the savings is brought into the basis. For most spacecraft, sixteen or fewer jets need be checked in so doing. The Space Shuttle uses 44 jets, and other spacecraft may use similarly large numbers of jets. Checking each of the jets would consume a large amount of time and effort each time a new selection is made. Large arrangements of jets are often redundant, or nearly so, so that they can be represented by clusters of jets, where a cluster is defined as a group of jets with numerically similar activity vectors.
A representative jet from each cluster is selected as the jet which maximizes the dot product $D$.

$$D = a_i \cdot c_k$$  \hspace{1cm} (4-68)

where $a_i$ is the activity vector of jet $i$, and

$$c_k = \sum_{i=1}^{\# \text{jets in cluster}} a_i$$  \hspace{1cm} (4-69)

is the cluster activity vector. The selection is then performed on the cluster representatives, or alternately on the $c_k$'s, significantly reducing the number of activity vectors to be checked. Implementation of the request is achieved by firing the representative or the cluster.

4.3.3.2 Minimum Impulse Jet Select

When the spacecraft is limit cycling, it is desired to maintain the longest limit cycle period, hence the lowest rates, to conserve fuel. The linear programming scheme very often selects two or three jets to fire or fails to meet a minimum on-time requirement when the request is on the order of the jet firing interval granularity. This granularity is due to a minimum firing time, on the order of a few milliseconds, imposed on the jets by startup and tailoff transients. The total impulse of this minimum duration burn is the jet's minimum impulse.

When the linear programming scheme generates on-times shorter than this minimum impulse time, on-times are rounded up to the minimum on-time, or down to zero. In the interest of minimizing jet cycling, the algorithm is constrained to selecting one jet when requests are on the order of jet granularity. To do so, the starting value of the cost function is taken as $10^6$ times the request, a six-element vector. The first
jet will be selected to maximize the saving in cost, by having the largest impact on the maximum component of the request. This demands that the jet activity vector be as closely coaligned with the request as possible. The jet so chosen is then fired for the minimum on-time.

4.3.3.3 Deletion of Certain Jet Firings

The linear programming algorithm for jet selection is constrained to satisfy exactly the rate change request. Since the control law is based on approximations to the equations governing spacecraft behavior, this request is inexact, particularly when state errors are large. It is thus unnecessary to exactly satisfy the request, and a benefit obtains from not doing so.

Making a jet selection, the linear jet select assigns firing times to six or fewer jets or jet clusters. Often, the firing is dominated by one or two jets with long firing times which closely approximate the request. The remaining jets have significantly shorter firing times, and "trim" the rate change to exactly meet the request.

The jet selection is modified when the state error is larger than some criterion, such as being outside the second phase sphere. In this case, the algorithm is used to compute the selection and firing times. These firing times are then compared to the maximum, and jets with firing times less than some fraction of the maximum are not fired. A saving results from this coarse control, as the nature of the approximations in the control law, or accelerating disturbances may cancel the value of some of the shorter firings, or render them unnecessary.

When the state error is within the larger phase sphere, the full selection is fired for more precise control. This is justifiable, as the control law approximations are most valid when the state errors are small. Further, the time to implement requests in this region is also small, reducing the likelihood of disturbances cancelling the benefit derived from a given firing.
4.3.3.4 Ignore Option

In certain situations it is desirable to precisely control the spacecraft about some axes, but not others. For example, it may be required to rotate a spacecraft at a precise rate for pointing, but large amounts of translation drift are acceptable. The jet selection problem, then, is one of fewer than six dimensions. If the uncontrolled axes are given a zero request component, the unnecessary constraint to keep rate change zero in those axes results. Means of reformulating the problem in these cases to one of fewer than six dimensions are desirable to save unnecessary computational effort.

A method employed by the new autopilot is to assume rotational control is always desired, but translational control is optional. The jet selection problem is then one of satisfying a three-dimensional request at times when rotation only is controlled, and a six-dimensional request when translation is also to be controlled. A substantial saving in computational burden is obtained, as the three-dimensional problem is much more quickly solved than the six-dimensional problem. Fuel savings also follow as three or fewer jets are used for the three-dimensional problem, whereas up to six may be used for the six-dimensional problem, particularly in the presence of jet coupling, where some jets are in the basis purely to cancel coupling.

A more sophisticated approach is to ignore those axes for which no specific request is made (i.e., rate is correct or can be controlled), and recast the problem with only the remaining axes. The jet select problem then can have as few as one dimension or up to six, but computations are performed only for those axes for which they are necessary. This promising concept is currently under study.
CHAPTER 5

THE NEW AUTOPilot

The new autopilot has been developed to incorporate the phase space control law and the linear programming jet select. Its function is to receive input commands from a guidance system or flight crew as well as vehicle state measurements, and to issue firing commands to reaction control jets to implement the input commands. An overall block diagram of the vehicle with the new autopilot is shown in Figure 5.1.

Figure 5-1. Overview of spacecraft/autopilot system

In Figure 5-2 is shown the structure of the new autopilot which consists of four major functions: a supervisor, a control law, a jet select, and error computation. The error computation and control law are developed in Chapter 3, and the jet select is developed in Chapter 4. The supervisor and the operation of the new autopilot make up the remainder of this chapter.
Figure 5-2. Overview of the new autopilot.
5.1 The Supervisor

The new autopilot will be implemented as a program in a spacecraft flight computer. Along with other tasks, such as guidance, navigation, and system management, the new autopilot is cycled at regular intervals, typically of 10 ms to 1 sec duration. Error and control computation is performed each of these cycles. However, new input commands are typically issued at longer, irregular intervals. Interaction of the control law and jet selection in the new autopilot are different from that of current autopilots. Current control laws generate an acceleration request on a pass by pass basis, and jet selection algorithms command jets from a table on a pass by pass basis. The control law of the new autopilot generates a rate change request on a pass by pass basis, and the jet select precomputes jet firing times to satisfy the request. A new jet selection is only to be made in response to a genuinely new rate change request from the control law, as it will completely satisfy that request.

Under certain conditions, to be defined, such a request is passed to the jet selection, and the implementation conditions of that request computed. On subsequent cycles, the control law will be updating its request due to the gradual implementation of the request, disturbances, and new input commands. If the autopilot is so configured that the jet select is performed each cycle, it would be essentially reduced to an acceleration request jet select. That is, the jet selection would be performed anew each pass. It is not equivalent to a table lookup in that the selection will not be fixed, but rather be the real time solution of the linear programming jet select problem, with the constraints of Eq.(4-62) and (4-63) replaced by the constraints

\[ x_i = 0 \text{ or } t_c, \text{ all } i \]  

(5-1)

where \( t_c \) is the autopilot cycle period.
In that jet selection is not performed each cycle, the autopilot is not strictly closed loop. In general, rate change requests are not being passed to jet select. The jet selection algorithm is either inactive or implementing a prior request. The feedback, error computation, and control law are acting as observers, producing new requests, which are not implemented. When a new request is passed to jet select, it is passed for one cycle only.

Certain tests are performed by the supervisor to determine if the request is new, and need be passed to the jet select. These criteria determine if the control law has generated a genuinely new request, or an unmodeled disturbance has occurred, necessitating computation of a new jet selection. Two sets of criteria exist, one for when the jets are on; that is, the jet select is implementing a new request, and one for when the jets are off; the vehicle is coasting. They are:

If jets are firing and

(1) the guidance system or crew sets a flag indicating a new input command or

(2) the rate change prediction of the jet select disagrees with state measurement or

(3) the state error has crossed from one region of phase space into another region.

(4) jet fail status changes

Condition (2) is discussed in detail in Appendix A, and (3) is discussed in Chapter 3.
Or, if no jets are firing and

(1) the guidance system or crew has set a flag indicating a new input command or

(2) the magnitude of the state error is increasing and is not within the innermost sphere or

(3) the state error has crossed from the outermost region of phase space into the buffer region.

Thus, the new autopilot typically performs jet selection at a rate much lower than that at which error and control computations are performed.

5.2 Operation of the New Autopilot

Implemented as a separate program in a digital flight computer, the operation of the new autopilot is controlled by the internal supervisor, discussed above, and an external supervisor/interface. Initially, the new autopilot requires the vehicle mass and inertia tensor, and the jet thrust and position vectors to generate jet activity vectors. Deadbands on the three rotation axes and three translation axes are then set, and an initial target state chosen.

Parameters of the phase sphere can be selected depending on the vehicle constraints and desired performance. In operation, the new autopilot receives measurements of the state in body coordinates; commands to the new autopilot are also made in body components. These commands consist of target attitude, target angular rate, target position, and target velocity. A flag must be set each time the targets change. The new autopilot processes these and computes the jet firing times. Through an appropriate interface, the new autopilot turns on all the jets in the selection. Capability must exist to schedule a jet
sequencer on the basis of a computer clock. This sequencer is scheduled to turn off the jets at the appropriate times, through the appropriate interface.

Failure detection and identification of the jets must exist and be rapid. Otherwise, the jet selection could be erroneously executed at a high rate due to the incorrect rate change from jet failures. Jets can be inhibited by simply setting their failure discretes, and reinstated by turning off the failure discretes.

The cycle period of the new autopilot must be determined by vehicle constraints, mission requirements, and desired performance. In general, the shorter the cycle period, the more precise the control. The lower limit could be the RCS jet response time. Typical cycle periods for the Space Shuttle are 40 ms. As an example, performance, computer burden, and fuel usage of the new autopilot applied to the Space Shuttle are described in the following chapters.

5.3 Implementation of the New Autopilot

The new autopilot has been implemented as a set of computer programs in the HAL/S and HAL/F languages. HAL/S and HAL/F are higher level languages intended for design verification and actual implementation of flight computer software. The HAL/S compiler produces code suitable for IBM 360 or 370 computers, and the HAL/F compiler produces code suitable for the IBM AP 101 flight computer, five of which are used to control the Space Shuttle.

An overview of the software is presented here. The outermost block of the new autopilot, the Digital Flight Control System (DFCS), shown in Figure 5-3a, is called by the flight computer executive on a cyclic basis. This program contains the code supervising the

95
IF RESTART INDICATED

OBTAIN VEHICLE MASS PROPERTIES, INITIAL GIMBAL ANGLES

READ CURRENT GIMBAL ANGLES, VEHICLE VELOCITY

IF NEW DEADBAND SET

UPDATE DEADBAND GAINS

PERFORM CONTROL LOGIC

RETURN

Figure 5-3a. Flowchart of DFCS.
operation of the new autopilot. DFCS first checks for a restart indication due to startup of the autopilot or restarting after a computer failure. The current gimbal angles and vehicle velocity are then obtained from hardware or appropriate filters. If a change in deadbands is indicated, new deadband gains, to be used in normalizing the state error, are generated. Next, the actual autopilot (Figure 5-3b) is executed. The error computation (Figure 5-3c) is performed, followed by the phase space control law (Figure 5-3d). The conditions for performing a new jet select are examined, and if met, the linear programming jet select is called.

The jet select (Figure 5-3e) first checks if the jet fail status or vehicle inertia properties have changed. If so, new jet activity vectors are computed, and new jet cluster representatives chosen. The request is then compared to the last request, and if it is within tolerance the previous selection implemented. Otherwise, a modified simplex algorithm is used to solve the linear programming jet selection problem of Chapter 4, resulting in a set of jets to be fired and associated firing times. Implementation of the selection is achieved by initially turning all the jets on. Code is then scheduled to execute at the time when the first jet is to be shut off. At that time, it does so and reschedules itself for the time when the next jet is to be turned off. This cycle continues until either all the firings in the selection are complete or a new selection is made. During the firings, a routine is executed each autopilot cycle to predict the rate change due to jet firings, and used to determine when to call jet select.
Figure 5-3b. Flowchart of new autopilot.
POSITION ERROR = POSITION - DESIRED POSITION

FIRST THREE COMPONENTS OF STATE ERROR = POSITION ERROR

DESIRED GIMBAL ANGLE = MODULAR ADD OF CURRENT GIMBAL ANGLE, GIMBAL ANGLE INCREMENT

GIMBAL ANGLE ERROR = MODULAR SUBTRACTION OF DESIRED GIMBAL ANGLE FROM CURRENT GIMBAL ANGLE

TRANSFORM GIMBAL ANGLE ERROR TO BODY ANGLE ERROR

SECOND THREE COMPONENTS OF STATE ERROR = BODY ANGLE ERROR

FIND MAGNITUDE, UNIT VECTOR OF STATE ERROR

RELATIVE RATE = RATE - DESIRED RATE (6 COMPONENTS)

RETURN

Figure 5-3c. Flowchart of error computation.
\[
\dot{x}_g = -C \text{ UNIT (NORMALIZED STATE ERROR)} - \text{RELATIVE RATE}
\]

**Figure 5-3d. Flowchart of phase space control la.**
IF JET FAIL STATUS OR VEHICLE INERTIA PROPERTIES HAVE CHANGED then RECOMPUTE ACTIVITY VECTORS

IF THIS REQUEST = LAST REQUEST then NEW SELECTION = LAST SELECTION
else COMPUTE NEW SELECTION

IMPLEMENT NEW SELECTION

RETURN

Figure 5-3e. Flowchart of jet selection.
To verify the performance of the new autopilot, tests were made of the phase space control law and the linear jet selection separately, then of the entire autopilot, using a simulation which includes spacecraft dynamics. The spacecraft dynamics for these tests were simulated by the Charles Stark Draper Laboratory (CSDL) Statement Level Simulator\textsuperscript{14,15}. Briefly, the Statement Level Simulator is a collection of computer programs which can simulate the Space Shuttle vehicle dynamics, control effectors, sensors, and environment. It is structured so that a user can provide computer code to simulate any function of the flight computer, specify mission phase, initial conditions, simulated failures and disturbances, to enable computer simulations of vehicle response to this particular design of one or several flight computer functions.

This simulation was selected for several reasons:

1. The Space Shuttle reaction control system is complex since individual jets typically couple into more than one control axis.

2. An autopilot for reaction jet control system of the Space Shuttle exists, and is an available basis of comparison to the current design.

3. The Statement Level Simulator allows simulation with actual AP 101 flight computer code, enabling a comparative determination of flight software core size and timing.
Since the proposed design is intended to be a complete orbital reaction control autopilot for a spacecraft, it must demonstrate several basic capabilities. These include:

1. Hold vehicle attitude and position in the presence of disturbances.
2. Perform attitude and position maneuvers within prespecified accuracy and time constraints.
3. Maintain control (i.e., 1 and 2) in the presence of jet failures.
5. Place minimal burden on the flight computer.

Spacecraft autopilots are designed with regard to these general requirements, specific details of which depend upon the spacecraft and its intended mission. The Space Shuttle requires all of these over a mission of one week to one month. Specific requirements are documented elsewhere. Representative features of these requirements as they apply to testing and verification of the current autopilot and the base-lined Shuttle autopilot are as follows:

1. Attitude and position hold capability:
   Attitude shall be held, when so requested, within the following allowable deadbands:

<table>
<thead>
<tr>
<th>AXIS</th>
<th>ATTITUDE DEADBAND</th>
<th>MAXIMUM LIMIT CYCLE RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(deg per axis)</td>
<td>(deg/sec per axis)</td>
</tr>
<tr>
<td>roll</td>
<td>±0.1 to ±20</td>
<td>±0.01 - 0.1</td>
</tr>
<tr>
<td>pitch</td>
<td>±0.1 to ±20</td>
<td>±0.01 - 0.1</td>
</tr>
<tr>
<td>yaw</td>
<td>±0.1 to ±20</td>
<td>±0.01 - 0.1</td>
</tr>
</tbody>
</table>
Although specific requirements have not been written concerning position holding, a typical set of requirements might be:

<table>
<thead>
<tr>
<th>AXIS</th>
<th>POSITION DEADBAND (meters)</th>
<th>MAXIMUM LIMIT CYCLE RATE (m/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>±0.2 to ±10</td>
<td>±0.01 - 0.1</td>
</tr>
<tr>
<td>Y</td>
<td>±0.2 to ±10</td>
<td>±0.01 - 0.1</td>
</tr>
<tr>
<td>Z</td>
<td>±0.2 to ±10</td>
<td>±0.01 - 0.1</td>
</tr>
</tbody>
</table>

Typical disturbances may result from gravity gradients, venting procedures, payload or fuel shifting, or jet "on" failures. A rough maximum figure might be that obtained by considering two jets failed on at the same station. These jets could produce an acceleration of 0.07 m/sec\(^2\) and an angular acceleration of 0.8 deg/sec\(^2\). Typically, onfailures require two seconds to detect.

(2) Attitude and position maneuvers:

There are several attitude and position drivers for the Space Shuttle, both manual and automatic. The requirements below are representative.

**Attitude:** The control system must be capable of converging to within ±0.75 deg and ±0.10 deg/sec of a desired attitude and angular rate. It must be capable of so doing from an initial value of ±20 deg from desired attitude and rates up to ±2 deg/sec, within 25 seconds.
Position: The specific manual translation capability thus far required entails rate change only. These modes require rate change capability from 0.03 ft/sec to 0.8 ft/sec. For automatic position maneuvers a reasonable requirement might also be levied from docking considerations. For a docking from six meters it might be required to control position to within approximately 0.5 meter at the docking drogue. This value is approximately the radius of a docking drogue, so that such a control law could be used for close up inspection and docking. A reasonable time for this might be 30 seconds.

(3) The specifications on the current Shuttle autopilot require it to maintain controllability (albeit degraded) in the presence of two jet failures or manifold isolation value closures, implying up to eight jets shut down. While the current requirements call for slightly degraded acceleration capability, it is the intent of this section to show that the current design can meet all the above requirements with far greater failure capabilities.

(4) The Space Shuttle may be required to perform missions of up to one month. During this time, the RCS may be called upon to perform extended attitude hold, attitude maneuvers, and docking. The vehicle carries approximately 3360 kg of RCS propellant, providing an RCS total impulse of 591,000 kg-sec. It will be necessary to conserve this propellant as much as possible to provide the vehicle with the greatest flexibility.

(5) The Space Shuttle flight computers serve several purposes, as can be inferred from Chapter 2. Core storage of each computer is approximately 64,000 words, in which must reside programs and data to perform all the necessary tasks. This
requires that the programs and data of the current design be as small as possible. Additionally, cycle time limitations of the computer require these functions to be executed in the least possible time.

By testing the current design, it is intended to demonstrate that it satisfies these requirements. Additional runs are made with the baselined Space Shuttle autopilot to compare the performance of the new design with the baselined autopilot. Specific tests were performed to demonstrate each of these capabilities. Below are presented four key tests, demonstrating these abilities: a constant angular rate test, an attitude maneuver test, an attitude maneuver with failed jets, and a position maneuver test.

6.1 Constant Angular Rate

A constant rate of 2.0 deg/sec in roll is commanded. All position and attitude errors were initially set to zero to test the rate response of both control laws.

6.2 Attitude Maneuver

A 20 degree yaw maneuver is commanded with a desired rate of 1.0 deg/sec, and attitude deadbands of 0.75 deg. A yaw maneuver was selected, as "yaw jets" couple significantly into roll. Thus, the test would be an indicator of each autopilot's ability to perform attitude maneuvers in the presence of strong jet coupling. Position deadbands were set to 1000 m in the phase space autopilot, to cause it to control attitude only, as does the comparison autopilot. The maneuver is simulated with both autopilots for 24 seconds.
6.3 Jet Failures

The 20 degree yaw of Section 6.2 is repeated with all aft RCS jets failed. Such a test verifies the ability of the autopilot to perform attitude maneuvers in the presence of critical jet failures. The simulation was allowed to run for 34 seconds.

6.4 Position Maneuver

The autopilot is required to translate the vehicle 1.5 meters along the Z axis, while holding attitude within a 2.5 deg deadband. Position deadbands are set equal to 0.06 meter, all three axes.
CHAPTER 7

TEST RESULTS AND COMPARISON

7.1 Fuel Usage and Firings

Fuel usage and firings were tabulated by the Statement Level Simulator (SLS) during simulations. Additionally, a fuel penalty of 0.0226 kg/firing was added to the figures to represent incremental fuel loss in jet turn on and turn off. Results are shown in Table 7-1.

Performing the constant rate maneuver, the new autopilot commanded three firings, for a total ontime of 4.048 seconds, using 5.5878 kg of propellant. The phase plane autopilot commanded eight firings for a total ontime of 3.967 seconds using 5.5928 kg of propellant to attain the same rate. The new autopilot saved 0.005 kg of propellant over the phase plane.

For the 20 degree yaw maneuver with no jets failed, the new autopilot commanded 14 jet firings, for a total ontime of 8.318 sec, using 11.674 kg of propellant. The phase plane autopilot performing the same maneuver commanded 18 firings, for a total ontime of 11.202 seconds using 15.703 kg of propellant. The phase space thus commanded four fewer firings, and saved 4.029 kg of propellant.

Since the phase plane autopilot failed to perform the 20 degree yaw with aft jets failed, no comparison is made on that test. In performing the position maneuver, the new autopilot commanded nine firings, for a total of 11.382 seconds, consuming 15.722 kg of propellant. No comparable maneuver could be performed with the phase plane autopilot.
No detailed testing was performed to determine fuel usage during long term attitude hold. With the minimum impulse option, the fuel consumption will generally be equal to or less than that of the baseline Space Shuttle autopilot for the same task. The new autopilot uses only one jet firing for a minimum duration to reverse the limit cycle rate, while the baseline Space Shuttle autopilot uses two or three, requiring more fuel. Limit cycle periods are thus typically longer for the new autopilot, resulting in a lower firing frequency. Tests to determine the fuel consumption of the new autopilot in limit cycling are currently underway.

<table>
<thead>
<tr>
<th>Maneuver</th>
<th>Phase Plane Autopilot</th>
<th>New Autopilot</th>
<th>Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>2°/sec constant rate</td>
<td>5.5928 kg</td>
<td>5.5878 kg</td>
<td>0.005 kg</td>
</tr>
<tr>
<td>20° yaw</td>
<td>15.703 kg</td>
<td>11.674 kg</td>
<td>4.029 kg</td>
</tr>
<tr>
<td>20° yaw aft jets failed</td>
<td>unacceptable roll coupling</td>
<td>21.05258 kg</td>
<td>--</td>
</tr>
<tr>
<td>1.5 meter translation along Z axis</td>
<td>failed</td>
<td>15.722</td>
<td>--</td>
</tr>
</tbody>
</table>

7.2 **Computer Burden**

The computer burden of the new autopilot and the phase plane autopilot are based on HAL/F compilations for sizing and HAL/S compilations for timing. These figures are based on running the autopilot in the IBM AP 101 computer, baselined for the Space Shuttle.

The core requirements for the phase plane autopilot in AP 101 wholewords are shown in Table 7-2 for a total of 3289 words of core memory.
Table 7-2. Core requirements for the phase plane autopilot (AP 101 wholewords).

<table>
<thead>
<tr>
<th>Component</th>
<th>Program Words</th>
<th>Data Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jet Selection</td>
<td>1836</td>
<td>855</td>
</tr>
<tr>
<td>Phase plane controller</td>
<td>436</td>
<td>36</td>
</tr>
<tr>
<td>Jet firing interface</td>
<td>20</td>
<td>6</td>
</tr>
<tr>
<td>Error computation</td>
<td>34</td>
<td>7</td>
</tr>
<tr>
<td>Automatic translation control</td>
<td>44</td>
<td>15</td>
</tr>
<tr>
<td>total</td>
<td>2370</td>
<td>919</td>
</tr>
</tbody>
</table>

The core requirements of the new autopilot in AP 101 wholewords are shown in Table 7-3 for a total of 2125 words of core storage. The new autopilot saves 1164 wholewords of core burden over the phase plane autopilot.

Table 7-3. Core requirements for the new autopilot (AP 101 wholewords).

<table>
<thead>
<tr>
<th>Component</th>
<th>Program Words</th>
<th>Data Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear jet selection</td>
<td>831</td>
<td>645</td>
</tr>
<tr>
<td>Jet firing supervisor</td>
<td>94</td>
<td>40</td>
</tr>
<tr>
<td>Rate change prediction</td>
<td>66</td>
<td>10</td>
</tr>
<tr>
<td>Phase space control law</td>
<td>82</td>
<td>6</td>
</tr>
<tr>
<td>Error computation</td>
<td>179</td>
<td>26</td>
</tr>
<tr>
<td>Control law sequencer</td>
<td>87</td>
<td>59</td>
</tr>
<tr>
<td>total</td>
<td>1339</td>
<td>786</td>
</tr>
</tbody>
</table>
The worst case execution time required for specific parts of the phase plane autopilot is as follows:

<table>
<thead>
<tr>
<th>Component</th>
<th>time (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase plane controller</td>
<td>3.84</td>
</tr>
<tr>
<td>(three passes)</td>
<td></td>
</tr>
<tr>
<td>Jet selection</td>
<td>4.663</td>
</tr>
<tr>
<td>total</td>
<td>8.503 msec</td>
</tr>
</tbody>
</table>

Both must be cycled at 25 hz, giving a 21.3% CPU burden.

The new autopilot requires:

<table>
<thead>
<tr>
<th>Component</th>
<th>time (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase space control law</td>
<td>0.7802</td>
</tr>
<tr>
<td>Control law supervisor</td>
<td>1.68</td>
</tr>
<tr>
<td>Linear jet select</td>
<td>31.69</td>
</tr>
<tr>
<td>total</td>
<td>34.1502 msec</td>
</tr>
</tbody>
</table>

Here, it is necessary to cycle the phase space control law and the control law supervisor at 25 hz. However, the linear jet select is only performed under certain conditions determined by the supervisor, described in Section 5.1. An estimate of its CPU burden can be determined by noting that the linear jet selection was performed six times over the 24 seconds of the attitude maneuver test. This gives a CPU burden of 0.79%. Combining this with a 6.15% CPU burden for the phase space control law and supervisor gives a 6.94% total CPU burden for the new autopilot.
7.3 Jet Failures

To compare the response of the new autopilot and the phase space autopilot to jet failures, the 20 degree yaw described in Section 6.4 was repeated with the aft jets failed. The phase plane autopilot yawed 20 degrees, but accumulated an unacceptable 9.6 degree roll error (Figures C-31 to C-36). The new autopilot was able to perform the 20 degree yaw and maintain roll and pitch error within the allowable deadband (Figures C-19 to C-30).

7.4 Accuracy

Neither autopilot reached the allowable error at the end of the simulation of the 20 degree yaw. Each was within 0.2 deg of the deadband, and coasting toward the commanded attitude at the termination of the simulation (Figures C-1 to C-18). The final errors in degrees are shown in Table 7-4. When the maneuver was repeated with aft jets failed the new autopilot drove the attitude to within the deadbands, but the phase plane autopilot accumulated a -9.6 deg roll error (Figure C-32). The errors are shown in Table 7-4. For the two deg/sec roll rate command, the new autopilot drove the rate error below .01 deg/sec in all three axes, as did the phase plane autopilot. Finally, in performing the position maneuver, the new autopilot drove the position error within .06 meter in all translation axes, maintaining attitude within the 2.5 degree deadband in all rotation axes.

The two autopilots show roughly equal accuracy in general, but the phase plane autopilot fails when the aft jets are failed, whereas the new autopilot retains its accuracy.
Table 7-4. Comparison of final attitude errors (deg)

<table>
<thead>
<tr>
<th>Maneuver</th>
<th>Phase plane</th>
<th>New autopilot</th>
</tr>
</thead>
<tbody>
<tr>
<td>$20^\circ$ yaw</td>
<td>(.5313, .2719, .810)</td>
<td>(-.766, -.256, .914)</td>
</tr>
<tr>
<td>$20^\circ$ yaw</td>
<td>(-9.607, -.651, .03)</td>
<td>(.076, -.196, .512)</td>
</tr>
<tr>
<td>aft jets failed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.1 Conclusions

A new spacecraft autopilot, capable of controlling vehicles of arbitrary design with changing mass properties and jet failure status, has been developed. The phase space control law concept forms the basis of the control law, and a linear programming algorithm is used for jet selection. This autopilot has successfully been applied to the Space Shuttle orbiter, controlling both translation and rotation with the reaction jets as control effectors.

Preliminary tests comparing the new autopilot with the currently baselined Space Shuttle autopilot indicate that some software benefits and fuel savings may be realized with comparable performance for the cases studied herein. The new autopilot requires 35.4% fewer words of core storage, and places an average 20.5% lower CPU burden on the flight computer; however, since a single pass through jet selection takes approximately 31 msec, the new autopilot lacks in response time. For some maneuvers tested, a 25.6% saving in RCS propellant is realized.

The new autopilot incorporates greater flexibility than current autopilots, incorporating adaptive logic to adjust to changing vehicle mass properties and jet configurations. Thus, the new autopilot should be capable of controlling a spacecraft over the range of its inertia properties (due to payload deployment, fuel usage, etc.) and in the presence of jet failures. Specifically, failure of 28 of the 44 jets on
the Shuttle does not prevent the new autopilot from performing certain maneuvers, but disables the current baselined autopilot.

It must be emphasized that these preliminary conclusions are drawn from only a limited set of tests. A complete comparison of the new autopilot with any other autopilot would require much more exhaustive testing.

8.2 Contributions

Contributions of this thesis are of a developmental nature, based on the existing phase space concept and linear programming. These contributions are:

(1) Extension of the three-dimensional phase space concept to n dimensions, each with separate deadbands, and application of this concept to six-degree-of-freedom spacecraft control.

(2) Improvements on the linear jet select algorithm in the light of practical considerations. These include the minimum impulse logic, and the concept of deleting from the selection those firings with total impulse on the order of nonlinearities or disturbances.

(3) Development of the new autopilot. This is a new class of autopilot, as the control law requests rate changes rather than fixed accelerations. The jet select is not performed each cycle, instead being performed when certain criteria are met.

(4) Testing of the new autopilot, and comparison with an existing autopilot.
8.3 **Recommendations**

The design of the new autopilot herein presented is the result of a year's research. Several aspects of the design are ad hoc, or suggested by specific situations, and it may be fruitful to formalize these. Among these aspects are several parameters such as the cycle period, the allowable deviation of the predicted and actual rate change during a jet firing, the size of the buffer region, and the convergence and limit cycle rates. Further work will be needed to determine the fuel optimal control in the presence of disturbances. Other promising areas of research may be the application of the phase space and linear jet select concepts to control of other spacecraft, or other systems or processes.

Finally, the phase space control law is a linear approximation to the desired control of a nonlinear system. Simple extensions of this concept to a nonlinear control law may exist. Also, it is suboptimal, even for linear systems, but performance may be improved by proper selection of such parameters as convergent rate and buffer region size.
APPENDIX A

CONDITION 2 FOR PERFORMING A NEW JET
SELECTION WHEN JETS ARE FIRING

Consider a single component rate change request \( \dot{x}_g \) and the actual rate \( \dot{x} \). The desired rate is \( \dot{x} + \dot{x}_g \). At time \( t_0 \), the jet select turns on a set of jets to implement the rate change request, with firings lasting through time \( t_1 \). Assume that during the interval \([t_0, t_1]\) the state has not crossed either of the phase spheres, so that the value of the convergence rate \( c \) in the velocity to be gained principle,

\[
\dot{x}_g = -c(x_d - \dot{x}) - (\dot{x}_d - \dot{x}) \quad (A-1)
\]

is constant, so that \( \dot{x}_g \) changes as a function of relative rate only. Clearly, the relative rate is decreasing in magnitude with time as the request is implemented. This observation applies similarly to the magnitude of \( n \) dimensional vectors, so that in controlling any system, the request changes during its implementation.

Since the jet activity vectors in general do not coalign with the request, and vehicle dynamics are nonlinear, the state trajectory during a set of jet firings will not be a straight line. That is, the sum \( \sum_{i=1}^{\# \text{jets firing}} a_i \) for the jets firing changes during implementation of a request as the jets are shut off one by one and is generally not parallel to \( \dot{x}_g \). The combination of jets selected to fire often includes one or two jets nearly aligned with the request, and one to four others fired to bring the net rate change in line with the request. Since all the
jets are turned on at once, the rate change achieved when the first few jets are shut off will not be aligned with the request, but the additional rate change of the remaining jets added to this will cause the net rate change again to coalign with and satisfy the request.

Actual rate change due to jet firings must be monitored and compared to some anticipated rate change. Unfortunately, since the rate vector rotates with time during a firing, checking that acceleration coaligns with the request will not work. Rather, it must be assumed that the solution of the linear jet select problem will produce a rate change ultimately satisfying the request, though instantaneously misaligned with the request. Prediction of the rate change on a per cycle basis, as

$$\Delta \dot{x} = t_c \sum_i a_i$$  \hspace{1cm} (A-2)

for all the jets $i$ that are firing and $t_c$ is the autopilot cycle period, is performed and compared with actual measurements of the vehicle rate change. The comparison is based on whether

$$t_c \sum_i a_i \cdot (\dot{x}^\text{now} - \dot{x}^\text{last}) < G |t_c \sum_i a_i|^2$$ \hspace{1cm} (A-3)

where $x^\text{now}$ is the current measurement of vehicle rate, $x^\text{last}$ is the measurement of vehicle rate from the last cycle, and $G$ is a gain. If this condition is true, the actual rate change is either not closely coaligned with the prediction or significantly smaller than the prediction. Both cases indicate that a new selection and possibly a recomputation of jet activity vectors is required: the jet select is called.

If the condition is false, the actual rate change is coaligned with the prediction or greater in magnitude than the prediction. The first is desirable, indicating the selection is being correctly implemented. The second is undesirable, resulting in either overcontrol or divergence of the errors. Either case will be detected by the control law, and corrective action taken.
APPENDIX B

EXAMPLE AUTOPilot AND JET SELECT COMPUTATION

Typical autopilot and jet selection computations of the new autopilot are discussed in this section. Code was added to the new autopilot and jet select to print out certain parameters during a simulated maneuver. The printout of a few cycles of one such run are shown in Figure B-1 to Figure B-6.

Figure B-1 shows the startup cycle and input command to the new autopilot. At A, the simulation is begun, and inputs read. Position and velocity relative to the target are zero, as no command has been received and no disturbances exist. At B, the phase space scale factors of Eq.(3-18) are computed. These correspond to 2.5 deg attitude deadbands in all three axes, and 0.06 position deadbands in all three axes, with rate deadbands of .0008 rad/sec and .008 m/sec. The desired gimbal angles are shown in radians and all errors are zero. New region 1 corresponds to the inside of the smaller phase sphere. At C is shown the zero rate change command (DELTA RATE) and a zero rate change, as no jets are fired. A new position command is issued at D, corresponding to a translation of 1.5 meters along the +Z axis. At E, the control law has found the magnitude of the normalized state error vector to be 0.2175, as the state error is -1.5 meters in -Z shown at F, r = .0087, putting the state error outside the outer sphere (new region = 3). The convergent rate is 0.017 m/sec along +Z at G, and since the relative rate is zero, this is the rate change command, shown at H. A call is made to the jet selection,
Figure B-1. Startup cycle and input to the new autopilot.
Figure B-2. Call to jet selection and computation of cluster activity vectors.
Figure B-3. Computation of individual jet activity vectors
<table>
<thead>
<tr>
<th>Time (S)</th>
<th>Jet</th>
<th>Select Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>-13 9.445694E-01</td>
</tr>
<tr>
<td>1.000000E+03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2.000000E+02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3.000000E+00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4.000000E+02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5.000000E+03</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Figure B-4. Jet select computations.**
Figure B-6. Implementation of jet selection and subsequent autopilot cycle.
which realizes it is starting up, performs computation of jet activities, and selects cluster representatives before performing a selection. At J, the jet select verifies vehicle mass and inertia properties, and at K, lists the cluster acceleration and cost. At L, individual jet activity vectors are listed, along with the jet's cluster number. Actual computation of the selection begins at M, in Figure B-4. The statement "INITLIZE CCWLJS" indicates that a starting basis follows.

The starting basis is obtained by creating six imaginary jets, each having an activity vector of unit magnitude along one of the vehicle axes (i.e. roll, pitch, yaw, X, Y, Z) so that the basis is the identity matrix, and firing them for a period numerically equal to the component of the request in each axis (FTIME). The cost for each of these is set at 1000 so any selection will give a substantial saving, and the jet i.d. is negative, indicating it is imaginary. Simplex solution of the jet select problem begins. At N, jet 13 is invited into the basis, to replace jet -1. The inverse of the basis (B^{-1}) is printed, along with the new costs and firing times. This process is continued down to 0, where six real jets are in the basis, and the benefit for proceeding is negligible. The jets selected and their firing times are:

<table>
<thead>
<tr>
<th>Jet #</th>
<th>ontime (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1.711</td>
</tr>
<tr>
<td>9</td>
<td>1.521</td>
</tr>
<tr>
<td>3</td>
<td>1.679</td>
</tr>
<tr>
<td>11</td>
<td>0.0358</td>
</tr>
<tr>
<td>5</td>
<td>0.00775</td>
</tr>
<tr>
<td>12</td>
<td>0.00432</td>
</tr>
</tbody>
</table>

At P, we see that only jets 13, 9, and 3 are fired, as the firing times of the others are below the limit of 30% of the maximum ontime. Notice that the jet numbers for the simulation differ from those of the autopilot. This is because
13, 9, and 3 are cluster numbers, as the selection is performed on 20
cluster representatives, and jets 6, 23, and 35 are the associated
representatives. At R, the rate change prediction due to the operation
of these jets is shown. This rate change is over an interval starting
with the jet firings and ending with the autopilot execution, as this is
shorter than the cycle period.
APPENDIX C

DETAILED DISCUSSION OF ATTITUDE MANEUVER SIMULATIONS

In this appendix are detailed presentations of the simulated attitude maneuvers which form the basis of the performance comparisons in Chapters 6 and 7. Specifically, the 20 degree yaw maneuvers with all jets operative and with aft jets failed are discussed. The initial conditions for all simulations are given in Table C-1.

C.1 Yaw Manuever, No Jets Failed

Twenty degree yaw maneuvers are commanded, driven by the three axis attitude maneuver routine\(^\text{17}\). Briefly, the three axis attitude maneuver routine computes an equivalent axis of single rotation, here the yaw axis, and maneuver angle about that axis. A steering procedure is cycled each 1.25 sec, computing desired gimbal angles, gimbal angle increments, and desired body rates, used to drive the autopilot through the maneuver. At the time when the maneuver would complete given instantaneous rate response, the routine commands attitude hold at the target attitude.

For the new autopilot performing this maneuver, the phase plane trajectory is shown in Figures C-1 through C-6, and state error cross plots in Figures C-7 through C-12. The jet firing history appears in Table C-2. For the current Shuttle autopilot performing the same maneuver, phase plane trajectories and switching curves are shown in Figures C-13 through C-18, and jet firing history in Table C-3.
Table C-1. Initial conditions for all simulations presented.

<table>
<thead>
<tr>
<th>ORIG SLS REL</th>
<th>SS POSITION MANEUVER</th>
<th>LSCMAPF</th>
<th>TSIMAPR</th>
<th>MANSROY</th>
<th>BEGMANN</th>
<th>REL</th>
<th>BLMAN</th>
<th>TRANS MANEUVER</th>
<th>SIMTIME</th>
<th>EXECUTED: AREA INDEX VALUE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS101101 GS11</td>
<td>PATCHES EXECUTED</td>
<td>AREA INDEX VALUE</td>
<td>VALUE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATAFILE 287031</td>
<td>C.0000000000 E 0</td>
<td>3.350999999 E -2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATAFILE 287033</td>
<td>C.0000000000 E 0</td>
<td>2.690999999 E -2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS101201 INITIAL MISSION PHASE:</td>
<td>CK-ORBIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

INTER METER COORDS (METERS, W/SEC) = -7.1200000000 04 0.9330000000 06 3.0750000000 06 |
| VEL, PITCH, ROLL ANGLES, RELATIVE TO LOCAL HORIZONTAL COORDINATES (DEG) = 0.0 0.0 0.0 |
| CRANURF = OFF (ORBITAL INSERTION MANEUVER NOT YET PERFORMED) |
| PAYLOAD = ON (ORBITER VEHICLE PAYLOAD INCLUDED) |
Table C-1. Initial conditions for all simulations presented (cont.).

<table>
<thead>
<tr>
<th>Initial Conditions</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Mass (kg)</td>
<td>73111.378</td>
</tr>
<tr>
<td>Payload Mass (kg)</td>
<td>24562.564</td>
</tr>
<tr>
<td>CS Propellant</td>
<td>8910.441</td>
</tr>
<tr>
<td>ACPS Propellant</td>
<td>1542.574</td>
</tr>
<tr>
<td>Total Vehicle Mass</td>
<td>112148.599</td>
</tr>
<tr>
<td>Center of Gravity</td>
<td>9.31164000000 00 1.2099999999 -02 3.1242000000 -01</td>
</tr>
<tr>
<td>Inertia Matrix</td>
<td>1.1295282027864 05 -1.3655403341 00 3.78272207564 05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Mass</td>
<td>112148.599</td>
</tr>
<tr>
<td>Payload Mass</td>
<td>24562.564</td>
</tr>
<tr>
<td>CS Propellant</td>
<td>8910.441</td>
</tr>
<tr>
<td>ACPS Propellant</td>
<td>1542.574</td>
</tr>
<tr>
<td>Total Vehicle Mass</td>
<td>112148.599</td>
</tr>
<tr>
<td>Center of Gravity</td>
<td>9.31164000000 00 1.2099999999 -02 3.1242000000 -01</td>
</tr>
<tr>
<td>Inertia Matrix</td>
<td>1.1295282027864 05 -1.3655403341 00 3.78272207564 05</td>
</tr>
</tbody>
</table>
Table C-1. Initial conditions for all simulations presented (cont.).

**DIM0061 INERTIAL MEASUREMENT UNIT (IMU) INITIAL CONDITIONS:**

<table>
<thead>
<tr>
<th>NO INSTRUMENT ERRORS</th>
<th>INITIAL REFERENCE TO STABLE MEMBER TRANSFORMATION</th>
<th>IMU3</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMU1</td>
<td>IMU2</td>
<td>IMU3</td>
</tr>
<tr>
<td>1.00000000 0.00000000 0.00000000 1.00000000 0.00000000 0.00000000 1.00000000 0.00000000 0.00000000 0.00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00000000 1.00000000 0.00000000 0.00000000 1.00000000 0.00000000 0.00000000 1.00000000 0.00000000 0.00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00000000 0.00000000 1.00000000 0.00000000 1.00000000 0.00000000 0.00000000 1.00000000 0.00000000 0.00000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**INITIAL EULER ANGLES (DEGREES)**

<table>
<thead>
<tr>
<th>IMU1</th>
<th>IMU2</th>
<th>IMU3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000000 0.000000 0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000 0.000000 0.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000000 0.000000 0.000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIM0061 SIMULATION EXECUTION BEGINS WITH 998.8 CPU SECONDS REMAINING.**
Figure C-1. Roll phase plane, new autopilot, 20 deg yaw.

Figure C-2. Pitch phase plane, new autopilot, 20 deg yaw.
Figure C-3. Yaw phase plane, new autopilot, 20 deg yaw.

Figure C-4. X translation phase plane, new autopilot, 20 deg yaw.
Figure C-5. Y translation phase plane, new autopilot, 20 deg yaw.

Figure C-6. Z translation phase plane, new autopilot, 20 deg yaw.
Figure C-7. Cross plot of roll error vs. pitch error, new autopilot, 20 deg yaw.

Figure C-8. Cross plot of roll error vs. yaw error, new autopilot, 20 deg yaw.
Figure C-9. Cross plot of pitch error vs. yaw error, new autopilot, 20 deg yaw.

Figure C-10. Cross plot of X translation error vs. Y translation error, new autopilot, 20 deg yaw.
Figure C-11. Cross plot of X translation error vs. Z translation error, new autopilot, 20 deg yaw.

Figure C-12. Cross plot of Y translation error vs. Z translation error, new autopilot, 20 deg yaw.
Table C-2. Jet history, new autopilot performing 20 deg yaw.

<table>
<thead>
<tr>
<th>Jet #</th>
<th>time on* (sec)</th>
<th>time off* (sec)</th>
<th>duration (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>1.0411</td>
<td>1.6540</td>
<td>0.6129</td>
</tr>
<tr>
<td>35</td>
<td>1.0442</td>
<td>1.6410</td>
<td>0.5968</td>
</tr>
<tr>
<td>5</td>
<td>1.0462</td>
<td>3.3479</td>
<td>2.3017</td>
</tr>
<tr>
<td>5</td>
<td>3.3773</td>
<td>3.7136</td>
<td>0.3363</td>
</tr>
<tr>
<td>12</td>
<td>11.7918</td>
<td>12.1170</td>
<td>0.3252</td>
</tr>
<tr>
<td>13</td>
<td>13.4647</td>
<td>13.5055</td>
<td>0.0408</td>
</tr>
<tr>
<td>26</td>
<td>13.5991</td>
<td>13.6399</td>
<td>0.0408</td>
</tr>
<tr>
<td>5</td>
<td>13.6011</td>
<td>13.6421</td>
<td>0.0410</td>
</tr>
<tr>
<td>6</td>
<td>13.6589</td>
<td>13.6997</td>
<td>0.0408</td>
</tr>
<tr>
<td>26</td>
<td>13.6616</td>
<td>13.7024</td>
<td>0.0408</td>
</tr>
<tr>
<td>5</td>
<td>13.6637</td>
<td>13.7044</td>
<td>0.0407</td>
</tr>
<tr>
<td>38</td>
<td>20.5731</td>
<td>21.1665</td>
<td>0.5934</td>
</tr>
<tr>
<td>23</td>
<td>20.5757</td>
<td>21.2404</td>
<td>0.6647</td>
</tr>
<tr>
<td>12</td>
<td>20.5780</td>
<td>23.2201</td>
<td>2.6421</td>
</tr>
</tbody>
</table>

* from start of maneuver

C.2 Yaw Maneuver, Aft Jets Failed

A yaw maneuver identical to that of Section C.1 was repeated with the Shuttle's aft jets failed. For the new autopilot, the resulting phase plane trajectories are shown in Figures C-19 through C-24 and state error cross plots in Figures C-25 through C-31. The jet firing history appears in Table C-4. For the current Shuttle autopilot, the phase plane trajectories appear in Figures C-32 through C-36, and jet firing histories in Table C-5.
Table C-3. Jet firing history, current autopilot performing 20 deg yaw

<table>
<thead>
<tr>
<th>Jet #</th>
<th>time on* (sec)</th>
<th>time off* (sec)</th>
<th>duration (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>.1790</td>
<td>1.5797</td>
<td>1.4007</td>
</tr>
<tr>
<td>32</td>
<td>.1797</td>
<td>1.5803</td>
<td>1.4006</td>
</tr>
<tr>
<td>35</td>
<td>1.5007</td>
<td>2.5395</td>
<td>1.0388</td>
</tr>
<tr>
<td>26</td>
<td>1.5008</td>
<td>2.5396</td>
<td>1.0388</td>
</tr>
<tr>
<td>4</td>
<td>1.7000</td>
<td>1.7397</td>
<td>0.0397</td>
</tr>
<tr>
<td>32</td>
<td>1.7006</td>
<td>1.7403</td>
<td>0.0397</td>
</tr>
<tr>
<td>4</td>
<td>1.9400</td>
<td>1.9797</td>
<td>0.0398</td>
</tr>
<tr>
<td>32</td>
<td>1.9406</td>
<td>1.9803</td>
<td>0.0397</td>
</tr>
<tr>
<td>4</td>
<td>2.1799</td>
<td>2.2197</td>
<td>0.0431</td>
</tr>
<tr>
<td>32</td>
<td>2.1806</td>
<td>2.2203</td>
<td>0.0430</td>
</tr>
<tr>
<td>4</td>
<td>2.4599</td>
<td>2.5030</td>
<td>0.2396</td>
</tr>
<tr>
<td>32</td>
<td>2.4606</td>
<td>2.5036</td>
<td>0.2395</td>
</tr>
<tr>
<td>23</td>
<td>8.9400</td>
<td>9.1796</td>
<td>1.6405</td>
</tr>
<tr>
<td>38</td>
<td>8.9403</td>
<td>9.1798</td>
<td>1.6405</td>
</tr>
<tr>
<td>11</td>
<td>20.3392</td>
<td>21.9797</td>
<td>1.1195</td>
</tr>
<tr>
<td>20</td>
<td>20.3396</td>
<td>21.9801</td>
<td>1.1196</td>
</tr>
<tr>
<td>23</td>
<td>21.3402</td>
<td>22.4597</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>21.3404</td>
<td>22.4600</td>
<td></td>
</tr>
</tbody>
</table>

* from start of maneuver
Figure C-13. Roll phase plane, current autopilot, 20 deg yaw.

Figure C-14. Pitch phase plane, current autopilot, 20 deg yaw.
Figure C-15. Yaw phase plane, current autopilot, 20 deg yaw.

Figure C-16. X translation phase plane, current autopilot, 20 deg yaw.
Figure C-17. Y translation phase plane, current autopilot, 20 deg yaw.

Figure C-18. Z translation phase plane, current autopilot, 20 deg yaw.
### Table C-4. Jet history, new autopilot, aft jets failed.

<table>
<thead>
<tr>
<th>Jet #</th>
<th>time on* (sec)</th>
<th>time off* (sec)</th>
<th>duration (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1.0156</td>
<td>1.7199</td>
<td>0.7043</td>
</tr>
<tr>
<td>9</td>
<td>1.0178</td>
<td>5.2622</td>
<td>4.2444</td>
</tr>
<tr>
<td>6</td>
<td>1.0199</td>
<td>4.2695</td>
<td>3.2496</td>
</tr>
<tr>
<td>9</td>
<td>6.3348</td>
<td>6.5900</td>
<td>0.2552</td>
</tr>
<tr>
<td>6</td>
<td>6.3368</td>
<td>6.4867</td>
<td>0.1499</td>
</tr>
<tr>
<td>13</td>
<td>21.7472</td>
<td>24.3233</td>
<td>2.5761</td>
</tr>
<tr>
<td>6</td>
<td>21.7493</td>
<td>23.7653</td>
<td>2.016</td>
</tr>
<tr>
<td>12</td>
<td>21.7915</td>
<td>22.3851</td>
<td>0.6339</td>
</tr>
<tr>
<td>13</td>
<td>24.3350</td>
<td>24.6334</td>
<td>0.2984</td>
</tr>
<tr>
<td>9</td>
<td>25.4575</td>
<td>25.8442</td>
<td>0.3867</td>
</tr>
<tr>
<td>6</td>
<td>25.4595</td>
<td>25.9850</td>
<td>0.5255</td>
</tr>
<tr>
<td>12</td>
<td>25.4618</td>
<td>25.5828</td>
<td>0.1210</td>
</tr>
<tr>
<td>9</td>
<td>25.8602</td>
<td>25.9019</td>
<td>0.0417</td>
</tr>
</tbody>
</table>

* from start of maneuver

### Table C-5. Jet history, phase plane, aft jets failed.

<table>
<thead>
<tr>
<th>Jet #</th>
<th>time on* (sec)</th>
<th>time off* (sec)</th>
<th>duration (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>.1790</td>
<td>3.2596</td>
<td>3.0806</td>
</tr>
<tr>
<td>11</td>
<td>7.1796</td>
<td>8.1398</td>
<td>0.9602</td>
</tr>
<tr>
<td>11</td>
<td>20.1395</td>
<td>23.5792</td>
<td>3.4397</td>
</tr>
<tr>
<td>13</td>
<td>21.4594</td>
<td>21.7393</td>
<td>0.2799</td>
</tr>
<tr>
<td>9</td>
<td>21.4594</td>
<td>21.7393</td>
<td>0.2799</td>
</tr>
<tr>
<td>4</td>
<td>26.4186</td>
<td>28.2582</td>
<td>1.8396</td>
</tr>
</tbody>
</table>

* from start of maneuver
Figure C-19. Roll phase plane, new autopilot, aft jets failed.

decoupling after initial firing

Figure C-20. Pitch phase plane, new autopilot, aft jets failed.
Figure C-21. Yaw phase plane, new autopilot, aft jets failed.

Figure C-22. X translation phase plane, new autopilot, aft jets failed.
Figure C-23. Y translation phase plane, new autopilot, aft jets failed.

Figure C-24. Z translation phase plane, new autopilot, aft jets failed.
Figure C-25. Cross plot of roll error vs. pitch error, new autopilot, aft jets failed.

Figure C-26. Cross plot of roll error vs. yaw error, new autopilot, aft jets failed.
Figure C-27. Cross plot of pitch error vs. yaw error, new autopilot, aft jets failed.

Figure C-28. Cross plot of X translation error vs. Y translation error, new autopilot, aft jets failed.
Figure C-29. Cross plot of X translation error vs. Z translation error, new autopilot, aft jets failed.

Figure C-30. Cross plot of Y translation error vs. Z translation error, new autopilot, aft jets failed.
final large error, rate such that it continues to diverge

Figure C-31. Roll phase plane, current autopilot, aft jets failed.

fortunately, this could be decoupled, and is converging at end of run

coupling of first burn

Figure C-32. Pitch phase plane, current autopilot, aft jets failed.
Figure C-33. Yaw phase plane, current autopilot, aft jets failed.

Figure C-34. X translation phase plane, current autopilot, aft jets failed.
Figure C-35. Y translation phase plane, current autopilot, aft jets failed.

Figure C-36. Z translation phase plane, current autopilot aft jets failed.


