ANALYSIS OF OPTIMAL CONTROL OF A FOUR-GIMBAL SYSTEM

by

MICHAEL ANDREW GENNERT

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREES OF

BACHELOR OF SCIENCE

and

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1980

Signature redacted

Signature of Author ____________________
Department of Electrical Engineering and Computer Science
May 9, 1980

Signature redacted

Certified by ____________________
Nils R. Sandell, Jr.
Thesis Supervisor

Signature redacted

Certified by ____________________
Richard V. Spencer
Company Supervisor

Signature redacted

Accepted by ____________________
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

VI-A THESIS RELEASE LETTER

Department of Electrical Engineering and Computer Science
Room 38-444
Massachusetts Institute of Technology
Cambridge, Mass. 02139

Attention: Professor Arthur C. Smith

Subject: Master's Thesis of Michael A. Gennert

I have reviewed the attached Thesis of Michael Gennert on behalf of the General Electric Company. The report is within the scope of the Thesis Proposal as previously approved and does not contain any material that is objectionable to the General Electric Company. It is also approved for its technical content.

It is understood that this Thesis will be the permanent property of MIT and will be placed in the MIT library within one month after the date of submissiom. However, the thesis will not be given general circulation until a review by the Office of the Secretary of Defense has been completed and the thesis released by the sponsor for general use. I will inform the Department of the results of such review as soon as possible. It is also understood that Michael Gennert has granted MIT permission to reproduce the thesis and to distribute copies publicly.

____

Authorized Official of General Electric

ANALYSIS OF OPTIMAL CONTROL OF A FOUR-GIMBAL SYSTEM

by

Michael Andrew Gennert

Submitted to the Department of Electrical Engineering and Computer Science on May 9, 1980 in partial fulfillment of the requirements for the Degrees of Bachelor of Science and Master of Science.

ABSTRACT

This thesis investigates modelling and control of a four-gimbal inertial system. The system under study is used to stabilize an inertial platform and to isolate the platform from vibration and rotation of the vehicle in which the system is mounted.

A few simplifying assumptions are made about the gimbal system. Using these assumptions and Euler's torque equations for a rotating body, a set of linear equations is developed relating angular acceleration of the gimbal elements to torque motor voltage. Taking a state-space approach, a set of nonlinear differential equations is used to compute the orientations of the gimbal elements from the torque motor voltages. A novel approach to the incorporation of static friction is presented, which leads to a simplified set of equations in the presence of static friction. Coulomb friction is also taken into account.

Modern optimal control techniques are applied to a linearized discrete-time version of the state equations to produce an optimal control scheme. The gimbal system and controller are simulated on a digital computer using the FORTRAN programming language. A listing of the program is included in the appendix. Comparisons are made with an earlier control strategy showing the reduction of platform misorientation, reduction of required torque, and elimination of switching transients.

THESIS SUPERVISOR: Nils R. Sandell, Jr.
TITLE: Adjunct Professor of Electrical Engineering

COMPANY SUPERVISOR: Richard V. Spencer
TITLE: Lead Advanced Systems Development Engineer
General Electric Ordance Systems

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, Professor Nils Sandell, for supervising this thesis. I would also like to express my thanks to the General Electric Company, Ordance Systems Division in Pittsfield, Massachusetts where most of this work was carried out under the VI-A Cooperative Program. Particular thanks are due Gordon Fiddes for suggesting research in this area; Dick Spencer, without whose guidance this thesis would have been impossible; and Howard Jones for his invaluable assistance. Last but not least I would like to express my deepest gratitude to my parents, Joyce and Warren Galkin, for their inspiration, encouragement and love.

TABLE OF CONTENTS

List of Figures

List of Tables

I. Introduction

This thesis investigates modelling and control of a four-gimbal system. Gimbals are generally used for precise orientation and/or stabilization. Typical applications include: attachment of a rocket engine so that the engine may be aimed, suspension of a ship's compass in a horizontal position despite pitch and roll, mounting a radar to rapidly track a target, stabilization of an inertial platform and isolation of the platform from vibration. It is this last application that will be of concern to us in this paper.

Inertial guidance and navigation systems generally use gyroscopes and accelerometers as sensing devices. High performance inertial guidance systems usually have these sensors mounted on an inertial platform and a series of concentric gimbals connecting the platform to the case. Gyros on the platform sense rotations of the platform with respect to inertial space, and are used in feedback loops to maintain an inertial reference.

The inertial platform and gimbals are housed in the inertial measurement unit case. The case is rigidly affixed to a vehicle whose rotation rate will be changing with time. The rotation rate is not measured directly; it can be calculated from other quantities, as will be shown. The rotation can be viewed as an input to the gimbal system, uninfluenced by the behavior of the system. As such, the vehicle's motion provides a set of boundary conditions for the kinematic equations describing the behavior of the gimbals.

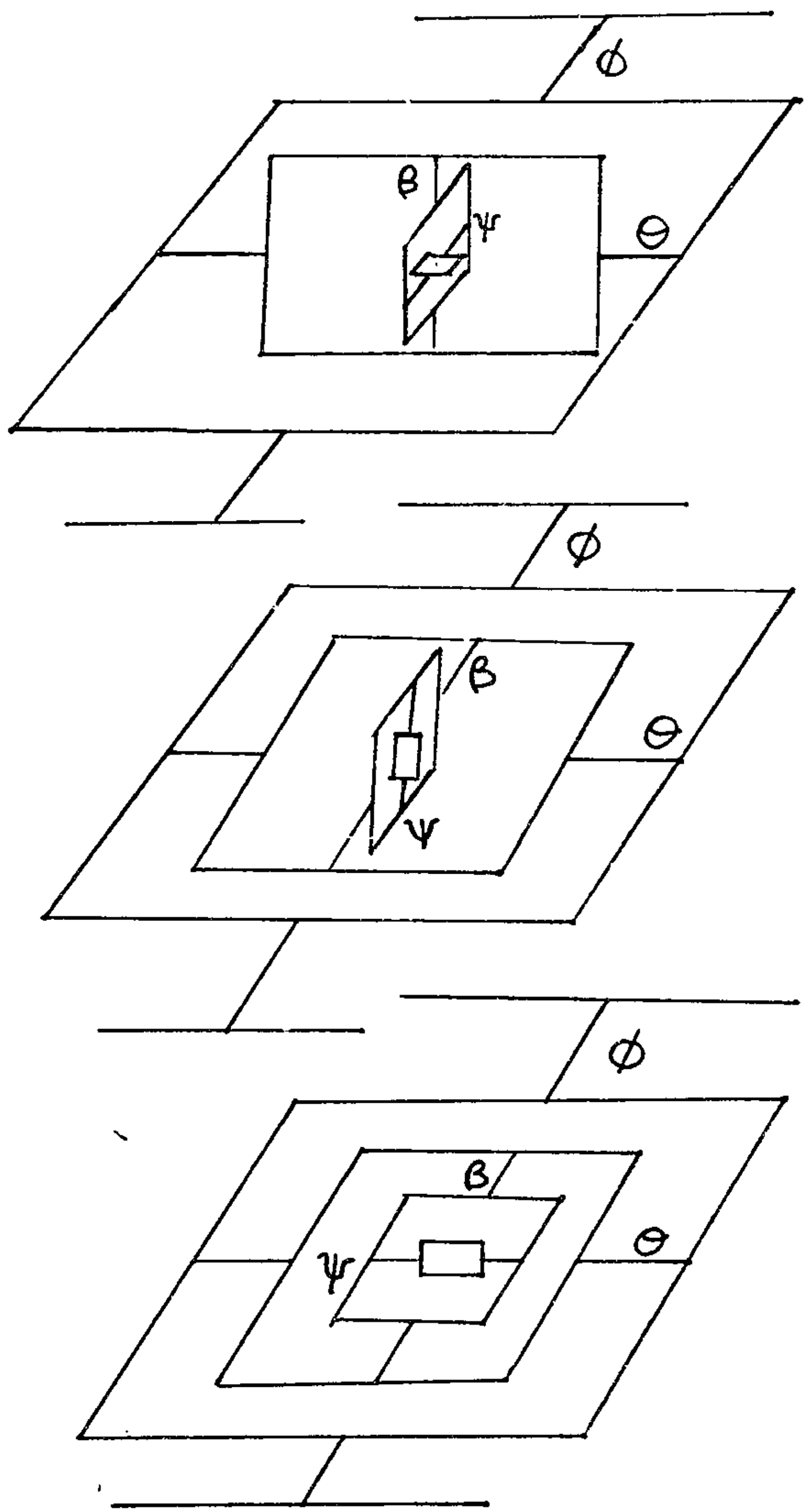
To fully isolate the inertial platform from vehicle motion requires a minimum of three gimbals, providing three degrees of freedom. It is possible for two of the gimbal axes to become parallel; "gimbal lock" is then said to occur and one degree of freedom is lost. If all three axes lie in one plane, rotation about an axis perpendicular to this plane is impossible. Clearly, gimbal lock must be avoided. However, it is not sufficient that the system stay out of gimbal lock; it must not even get close because, as gimbal lock is approached, increasingly high torque levels are required to keep the platform inertial[5]. If the required torque should exceed the maximum available torque, then the inertial platform may lose its inertial reference.

There are basically two strategies available for dealing with the gimbal lock problem. The simplest solution is to restrict the vehicle's motion so that gimbal lock cannot occur. Early guidance systems used exactly this restricted attitude scheme. The drawbacks are obvious. A present state-of-the-art all-attitude guidance system avoids gimbal lock by adding a fourth gimbal (Figure 1.1). The extra degree of freedom ensures that it will always be possible to avoid gimbal lock. If two gimbal axes are aligned there will still be three degrees of freedom. However, if the system is not properly controlled it is possible for all four axes to lie in one plane, a second degree of freedom will be lost, and gimbal lock will result. The problem then is one of allocation of control among the four gimbals to stabilize the inertial platform while avoiding gimbal lock given the vehicle's rotation rate.

Control is effected through torque motors mounted on the outer three gimbals and the case. The torquers are driven by saturating amplifiers, limiting the maximum available torque. Information on the state of the system is available from three sources. Gyroscope outputs indicate any deviation of the platform attitude from inertial, resolvers mounted on each gimbal indicate the angles between gimbals, and tachometers measure angular velocities.

Presently, the inner two gimbals are driven directly by gyroscopes, and control is switched between the two outer gimbals, depending on the two middle angles. The control law takes the form of decision rules, so that control is allocated based upon the zone in which the middle two angles reside. Although the zone control does avoid gimbal lock, it is not optimal. Large attitude errors and torque transients may occur when switching zones. The maximum torque requirements are excessive; by reducing them it will be possible to improve torque motor performance and/or reduce the torquer size, weight and cost. Furthermore, reductions in attitude errors resulting from optimization will contribute to overall system accuracy.

The approach taken is as follows. The mechanics of the gimbal system are discussed first. Simplifying assumptions and approximations are presented and justified. Based upon Euler's torque equations a set of equations are derived that characterize the system. We examine friction and its effects. Modern optimal control techniques are applied to a linearized discrete-time version of the torque equations to yield an optimal control scheme. Various methods of implementing the controller are suggested. The controller is realized as a simulation on a digital computer using the FORTRAN programming language. Results of the simulation are analyzed and compared with an earlier control strategy.



All angles zero

θ = 90

Gimbal lock

B = θ = 90

Figure 1.1

Gimbal Configurations

II. Nomenclature

A	Continuous-time dynamics matrix
A^*	Discrete-time dynamics matrix
B	Continous-time matrix from control signals to state derivative
B^*	Discrete-time matrix from control signals to state derivative
C	Case
$\vec{C}$	Continuous-time constant vector
$\vec{C}^*$	Discrete-time constant vector
C_j^k	Coordinate transformation from j to k system
D	State information compression matrix
$\vec{e}$	State error vector
E	Elevation gimbal = Inertial platform
$\vec{E}$	Optimal next state
$\vec{H}_j^k$	Angular momentum of gimbal j in the k frame
I	Inner gimbal
I	3 x 3 identity matrix
J	Cost function
J_k^l	Inertia tensor of gimbal k in the l reference frame
J_{kv}^l	Moment of inertia of gimbal k about its v-axis in the k frame
L	Matrix transforming acceleration to torques
M	Middle gimbal
M	Matrix transforming torques to accelerations = L^{-1}
O	Outer gimbal
Q	Symmetric state weight matrix in cost function
R	Symmetric torque weight matrix in cost function
S	Inertial space

$\vec{T}_j^k$	Total torque on gimbal j in the k frame
$\vec{T}_{kj}^l$	Torque on gimbal j supplied by gimbal k in the l frame
T_{jkv}^l	Component of T in the v direction
$\vec{U}$	Control vector
$\vec{W}_{kj}^l$	Rotation of gimbal j with respect to gimbal k in the l frame
$\vec{X}$	State vector
$\vec{Y}$	Vector composed of torques and torque-like terms
$\vec{Z}$	Angular acceleration vector
Υ	Angle between E and I
B	Angle between I and M
Θ	Angle between M and O
$\emptyset$	Angle between O and C
λ	Gimbal lock angle

III. System Description

The four-gimbal system is shown schematically in Figure 3.1. Pictured are the case (C), outer gimbal (O), middle gimbal (M), inner gimbal (I) and elevation gimbal (E). The terms "elevation gimbal" (1) and "inertial platform" refer to the same thing and will be used interchangably. "Case" and "vehicle" will also be used interchangably in the context of rotation and acceleration, although they do not refer to the same thing. The case is securely bolted to the vehicle and thus experiences the same velocity and acceleration.

The outer, middle and inner gimbals look much the same except for size. Two slipring assemblies connect each gimbal to the next innermost and next outermost gimbals. The slipring assemblies contain resolvers, tachometers and torque motors. The relative position and velocity of each gimbal pair may be directly observed (after filtering to remove noise). The torque motors are the sole actuators present in the system.

The elevation gimbal is totally different from the others. It is essentially a platform laden with sensors. The only sensors of concern to us here will be the gyroscopes. The

---

(1) The phrase "elevation gimbal" is carried over from three-gimbal system days when the elevation angle Y was exactly equal to the elevation of the vehicle with respect to the earth's surface. What is now the inner gimbal was then called the "azimuth gimbal." It is still occaisionally referred to by the older name. We will stick with "inner gimbal." The letter "B" used for the angle between the inner and middle gimbals reflects the fact that this angle equalled the bearing of the vehicle in the three-gimbal system.

gyroscopes will be treated as though there were three single degree of freedom (SDF) gyros. In fact, two two degree of freedom (TDF) gyros may be used, one degree of freedom being redundant. The gyros are aligned so that their input axes lie along X_E, Y_E and Z_E. Any rotation of the inertial platform will be sensed by one or more gyros. Any misalignment of the gyroscopes with respect to the inertial platform will be subject to compensation elsewhere in the guidance system and will not concern us.

Six different Cartesian coordinate systems may be defined. Four of these coordinate systems are fixed to the four gimbals, the fifth and sixth coordinate systems are associated with the case and inertial space (S). One may restate the purpose of the controller as being to keep the elevation gimbal coordinate frame and the inertial space coordinate frame as closely aligned as possible given the rotation rate of the case coordinate frame. The rotation rates of the case and gimbals with respect to inertial space coordinatized in the case and gimbal frames may be defined as follows:

$$\vec{W}^{C}_{SC} \triangleq \begin{vmatrix} W_{CX} \\ W_{CY} \\ W_{CZ} \end{vmatrix} \qquad \vec{W}^{O}_{SO} \triangleq \begin{vmatrix} W_{OX} \\ W_{OY} \\ W_{OZ} \end{vmatrix}$$

$$\vec{W}^{M}_{SM} \triangleq \begin{vmatrix} W_{MX} \\ W_{MY} \\ W_{MZ} \end{vmatrix} \qquad \vec{W}^{I}_{SI} \triangleq \begin{vmatrix} W_{IX} \\ W_{IY} \\ W_{IZ} \end{vmatrix} \qquad \vec{W}^{E}_{SE} \triangleq \begin{vmatrix} W_{EX} \\ W_{EY} \\ W_{EZ} \end{vmatrix}$$

The above vectors are interpreted as the rotation rate of the coordinate system denoted by the the right subscript with respect to the coordinate system denoted by the left subscript as seen from the coordinate system denoted by the superscript. This convention is discussed in more detail in Britting[3].

In order to relate the various coordinate frames it is necessary to define the angle between adjacent gimbals.

Angle name	Between	Also called
Υ	E and I	Elevation Angle
B	I and M	Inner Angle
Θ	M and O	Middle Angle
ϕ	O and C	Outer Angle

That only a single degree of freedom exists between gimbals simplifies the direction cosine matrices. Specifically:

$$C_C^O = \begin{vmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{vmatrix} \qquad (3.1)$$

$$C_O^M = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\Theta & \sin\Theta \\ 0 & -\sin\Theta & \cos\Theta \end{vmatrix} \qquad (3.2)$$

$$C_M^I = \begin{vmatrix} \cos B & \sin B & 0 \\ -\sin B & \cos B & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (3.3)$$

$$C_I^E = \begin{vmatrix} \cos\Upsilon & 0 & -\sin\Upsilon \\ 0 & 1 & 0 \\ \sin\Upsilon & 0 & \cos\Upsilon \end{vmatrix} \quad (3.4)$$

The above matrices are interpreted as a linear transform from the coordinate system denoted by the subscript to the coordinate system denoted by the superscript. Direction cosine matrices are treated in more detail in Appendix A. The special form of the direction cosine matrices is due to the fact that $\emptyset$ is measured around the outer gimbal and case y-axis, Θ is measured around the middle and outer z-axis, B is measured around the inner and middle x-axis, and Υ is measured around the elevation and inner y-axis. These definitions are entirely arbitrary but survive for historical reasons. The time derivatives of these angles are nothing but the relative rotation rates. That is:

$$\vec{W}^C_{CO} = \vec{W}^O_{CO} \triangleq \begin{vmatrix} 0 \\ \dot{\emptyset} \\ 0 \end{vmatrix} \qquad \vec{W}^O_{OM} = \vec{W}^M_{OM} \triangleq \begin{vmatrix} \dot{\Theta} \\ 0 \\ 0 \end{vmatrix}$$

$$\vec{W}^{M}_{MI} = \vec{W}^{I}_{MI} \hat{=} \begin{vmatrix} 0 \\ 0 \\ \dot{B} \end{vmatrix} \qquad \vec{W}^{I}_{IE} = \vec{W}^{E}_{IE} \hat{=} \begin{vmatrix} 0 \\ \dot{\Upsilon} \\ 0 \end{vmatrix}$$

It is now possible to relate the rotation of any gimbal to inertial space. This is necessary to express the torque equations later. Starting with the outer gimbal and applying equations (A.2) and (A.6) we have:

$$\vec{W}^{O}_{SO} = \vec{W}^{O}_{SC} + \vec{W}^{O}_{CO} = C^{O}_{C}\vec{W}^{C}_{SC} + \vec{W}^{O}_{CO}$$

$$= \begin{vmatrix} \cos\emptyset \ W_{CX} - \sin\emptyset \ W_{CZ} \\ W_{CY} + \dot{\emptyset} \\ \sin\emptyset \ W_{CX} + \cos\emptyset \ W_{CZ} \end{vmatrix} \qquad (3.5)$$

$$\vec{W}^{M}_{SM} = C^{M}_{O}\vec{W}^{O}_{SO} + \vec{W}^{M}_{OM}$$

$$= \begin{vmatrix} W_{OX} + \dot{\Theta} \\ \cos\Theta \ W_{OY} + \sin\Theta \ W_{OZ} \\ -\sin\Theta \ W_{OY} + \cos\Theta \ W_{OZ} \end{vmatrix} \qquad (3.6)$$

$$\vec{W}^{I}_{SI} = C^{I}_{M}\vec{W}^{M}_{SM} + \vec{W}^{I}_{MI}$$

$$= \begin{vmatrix} \cos B \ W_{MX} + \sin B \ W_{MY} \\ -\sin B \ W_{MX} + \cos B \ W_{MY} \\ W_{MZ} + \dot{B} \end{vmatrix} \qquad (3.7)$$

$$\vec{W}^{E}_{SE} = C^{E}_{I}\vec{W}^{I}_{SI} + \vec{W}^{E}_{IE}$$

$$= \begin{vmatrix} \cos\Upsilon\ W_{IX} - \sin\Upsilon\ W_{IZ} \\ W_{IY} + \dot{\Upsilon} \\ \sin\Upsilon\ W_{IX} + \cos\Upsilon\ W_{IZ} \end{vmatrix} \quad (3.8)$$

Equations (3.5) through (3.8) and (A.2) may be combined to compute the case rates.

$$\vec{W}^{E}_{SE} = C^{E}_{C}\vec{W}^{C}_{SC} + C^{E}_{O}\vec{W}^{O}_{CO} + C^{E}_{M}\vec{W}^{M}_{OM} + C^{E}_{I}\vec{W}^{I}_{MI} + \vec{W}^{E}_{IE} \quad (3.9)$$

Rearranging terms and multiplying by C^{C}_{E} yields:

$$\vec{W}^{C}_{SC} = C^{C}_{E}\vec{W}^{E}_{SE} - C^{C}_{E}C^{E}_{O}\vec{W}^{O}_{CO} - C^{C}_{E}C^{E}_{M}\vec{W}^{M}_{OM} - C^{C}_{E}C^{E}_{I}\vec{W}^{I}_{MI} - C^{C}_{E}\vec{W}^{E}_{IE}$$

$$= C^{C}_{E}\vec{W}^{E}_{SE} - C^{C}_{O}\vec{W}^{O}_{CO} - C^{C}_{M}\vec{W}^{M}_{OM} - C^{C}_{I}\vec{W}^{I}_{MI} - C^{C}_{E}\vec{W}^{E}_{IE}$$

$$\vec{W}^{C}_{SC} = C^{C}_{E}\vec{W}^{E}_{SE} - \vec{W}^{C}_{CO} - C^{C}_{O}\vec{W}^{O}_{OM} - C^{C}_{M}\vec{W}^{M}_{MI} - C^{C}_{I}\vec{W}^{I}_{IE} \quad (3.10)$$

The left hand side is the rotation rate of the case, which is to be determined; the right hand side is dependent only upon measurable quantities. We will want to relate torque to acceleration in the next section, so we may apply equation (A.8) to equations (3.5) through (3.8).

$$\dot{\vec{W}}^{O}_{SO} = C^{O}_{C}\dot{\vec{W}}^{C}_{SC} - \vec{W}^{O}_{CO} \times C^{O}_{C}\vec{W}^{C}_{SC} + \dot{\vec{W}}^{O}_{CO} \quad (3.11)$$

$$\dot{\vec{W}}^{M}_{SM} = C^{M}_{O}\dot{\vec{W}}^{O}_{SO} - \vec{W}^{M}_{OM} \times C^{M}_{O}\vec{W}^{O}_{SO} + \dot{\vec{W}}^{M}_{OM} \quad (3.12)$$

$$\dot{\vec{W}}^{I}_{SI} = C^{I}_{M}\dot{\vec{W}}^{M}_{SM} - \vec{W}^{I}_{MI} \times C^{I}_{M}\vec{W}^{M}_{SM} + \dot{\vec{W}}^{I}_{MI} \quad (3.13)$$

$$\dot{\vec{W}}^{E}_{SE} = C^{E}_{I}\dot{\vec{W}}^{I}_{SI} - \vec{W}^{E}_{IE} \times C^{E}_{I}\vec{W}^{I}_{SI} + \dot{\vec{W}}^{E}_{IE} \quad (3.14)$$

Unfortunately, equation (3.11) contains $\dot{\vec{W}}^{C}_{SC}$, the acceleration of the case, and a difficult quantity to measure.

It will be desirable to know $\dot{\vec{W}}^{C}_{SC}$ in order to predict the trajectory of $\vec{W}^{C}_{SC}$ and thereby optimize the performance of the gimbal system at some time in the future. For a massive vehicle such as the one under consideration here the rotation rate cannot change rapidly. Unable to measure the vehicle's acceleration directly to predict its behavior, a reasonable approach is to assume that it does not change at all. Therefore, throughout this paper it will be assumed that $\dot{\vec{W}}^{C}_{SC} = \vec{0}$. This is not such a bad assumption over a short time interval. Thus, equation (3.11) reduces to

$$\dot{\vec{W}}^{O}_{SO} = -\vec{W}^{O}_{CO} \times C^{O}_{C}\vec{W}^{C}_{SC} + \dot{\vec{W}}^{O}_{CO} \tag{3.15}$$

In theory it is possible to predict the vehicle's acceleration knowing the generated thrust and mass. It is preferable, though, to keep the four-gimbal controller as decoupled as possible from all other vehicular systems, including propulsion.

The vector angular acceleration equations, although compact, are of limited utility by themselves[10]. They need to be expressed in terms of scalar quantities. To this end, equations (3.12) through (3.15) will be expanded using equations (3.5) through (3.8).

$$\dot{\vec{W}}^{O}_{SO} = \begin{vmatrix} -\dot{\phi}\ W_{OZ} \\ \ddot{\phi} \\ \dot{\phi}\ W_{OX} \end{vmatrix} \tag{3.16}$$

$$\dot{W}^{M}_{SM} = \begin{vmatrix} \dot{W}_{OX} + \ddot{\theta} \\ \cos\theta\ \dot{W}_{OY} + \sin\theta\ \dot{W}_{OZ} + \dot{\theta}\ W_{MZ} \\ -\sin\theta\ \dot{W}_{OY} + \cos\theta\ \dot{W}_{OZ} - \dot{\theta}\ W_{MY} \end{vmatrix}$$

$$= \begin{vmatrix} -\dot{\phi}\ W_{OZ} + \ddot{\theta} \\ \cos\theta\ \ddot{\phi} + \sin\theta\ \dot{\phi}\ W_{OX} + \dot{\theta}\ W_{MZ} \\ -\sin\theta\ \ddot{\phi} + \cos\theta\ \dot{\phi}\ W_{OX} - \dot{\theta}\ W_{MY} \end{vmatrix} \qquad (3.17)$$

$$\dot{W}^{I}_{SI} = \begin{vmatrix} \cos B\ \dot{W}_{MX} + \sin B\ \dot{W}_{MY} + \dot{B}\ W_{IY} \\ -\sin B\ \dot{W}_{MX} + \cos B\ \dot{W}_{MY} - \dot{B}\ W_{IX} \\ \dot{W}_{MZ} + \ddot{B} \end{vmatrix}$$

$$= \begin{vmatrix} \cos B\ (-\dot{\phi}\ W_{OZ} + \ddot{\theta}) + \sin B\ (\cos\theta\ \ddot{\phi} \\ \quad + \sin\theta\ \dot{\phi}\ W_{OX} + \dot{\theta}\ W_{MZ}) + \dot{B}\ W_{IY} \\ \\ -\sin B\ (-\dot{\phi}\ W_{OZ} + \ddot{\theta}) + \cos B\ (\cos\theta\ \ddot{\phi} \\ \quad + \sin\theta\ \dot{\phi}\ W_{OX} + \dot{\theta}\ W_{MZ}) - \dot{B}\ W_{IX} \\ \\ -\sin\theta\ \ddot{\phi} + \cos\theta\ \dot{\phi}\ W_{OX} - \dot{\theta}\ W_{MY} + \ddot{B} \end{vmatrix} \qquad (3.18)$$

$$\dot{W}^{E}_{SE} = \begin{vmatrix} \cos\gamma\ \dot{W}_{IX} - \sin\gamma\ \dot{W}_{IZ} - \dot{\gamma}\ W_{EZ} \\ \dot{W}_{IY} + \ddot{\gamma} \\ \sin\gamma\ \dot{W}_{IX} + \cos\gamma\ \dot{W}_{IZ} + \dot{\gamma}\ W_{EX} \end{vmatrix}$$

$$\begin{vmatrix} \cos\gamma\ \{\cos B\ (-\dot{\phi}\ W_{OZ} + \ddot{\theta}) + \sin B\ (\cos\theta\ \ddot{\phi} \\ \quad + \sin\theta\ \dot{\phi}\ W_{OX} + \dot{\theta}\ W_{MZ}) + \dot{B}\ W_{IY}\} \\ \quad - \sin\gamma\ \{-\sin\theta\ \ddot{\phi} + \cos\theta\ \dot{\phi}\ W_{OX} - \dot{\theta}\ W_{MY}) \\ \quad + \ddot{B}\} - \dot{\gamma}\ W_{EZ} \\ \\ -\sin B\ (-\dot{\phi}\ W_{OZ} + \ddot{\theta}) + \cos B\ (\cos\theta\ \ddot{\phi} \\ \quad + \sin\theta\ \dot{\phi}\ W_{OX} + \dot{\theta}\ W_{MZ}) - \dot{B}\ W_{IX} + \ddot{\gamma} \\ \\ \sin\gamma\ \{\cos B\ (-\dot{\phi}\ W_{OZ} + \ddot{\theta}) + \sin B\ (\cos\theta\ \ddot{\phi} \\ \quad + \sin\theta\ \dot{\phi}\ W_{OX} + \dot{\theta}\ W_{MZ}) + \dot{B}\ W_{IY}\} \\ \quad + \cos\gamma\ \{-\sin\theta\ \ddot{\phi} + \cos\theta\ \dot{\phi}\ W_{OX} - \dot{\theta}\ W_{MY}) \\ \quad + \ddot{B}\} - \dot{\gamma}\ W_{EX} \end{vmatrix} \qquad (3.19)$$

These last four equations are second order differential equations. Note that the only place where second time derivatives appear is on angles. This is quite a propitious occurrence because, in a later section the state variables will be specified, and the angles will be among the state variables. We will want to express the highest order derivatives of the state variables as functions of lower order derivatives and other known quantities, and to do this we must separate the highest order derivatives from all other factors. Equations (3.16) through (3.19) show where the high order derivatives lie and this is a great help.

We now introduce three variables ΔSV, ΔJ and ΔSR. They represent the tilt (rotation) of the inertial platform with respect to inertial space. ΔSR is measured about the x-axis, ΔJ is measured about the y-axis and ΔSV is measured about the z-axis of the elevation gimbal. The tilts equal the angular displacement of the inertial platform as sensed by the gyros about the relevant axes. They may be described by differential equations by noting that the rate of change of the tilts must equal the rotation rate of the inertial platform. The rotation rate of the platform is merely $\vec{W}^{E}_{SE}$. Applying equation (3.8) we have:

$$\Delta \dot{SR} = \cos I \, W_{IX} - \sin I \, W_{IZ} \qquad (3.20)$$

$$\Delta \dot{J} = W_{IY} + \dot{I} \qquad (3.21)$$

$$\Delta \dot{SV} = \sin I \, W_{IX} + \cos I \, W_{IZ} \qquad (3.22)$$

Now to define the moments of inertia of the gimbals. Let J_k^l be the inertia tensor of gimbal k in the l coordinate system. The matrix representation of an inertia tensor transforms under similarity transformations, i.e., $J_k^m = C_l^m J_k^l C_m^l$. Because the gimbals are symmetric and have been evenly balanced, and because the gimbal-fixed coordinate systems are aligned with the principal axes, the inertia matrix will have zeros off the diagonal when coordinatized in the reference frame of that gimbal. Thus:

$$J_k^k = \begin{vmatrix} J_{kx} & 0 & 0 \\ 0 & J_{ky} & 0 \\ 0 & 0 & J_{kz} \end{vmatrix}$$

In general, this is true only when the inertia is coordinatized in the reference frame of that gimbal, and not true in most other reference frames. Thus, J_k^l, $l \neq k$ will, in general, have nonzero terms off the diagonal. Furthermore, the elevation gimbal is almost symmetric, so we may approximate $J_{EX} \approx J_{EY} \approx J_{EZ} \triangleq J_{EXYZ}$ The other three gimbals take the shape of bands, each having two roughly equal moments of inertia and a third distinct moment of inertia, the distinct inertia corresponding to the gimbal axis passing through the "hole" in the gimbal. For the given geometry:

$$J_{IY} \approx J_{IZ} \triangleq J_{IYZ}$$
$$J_{MX} \approx J_{MZ} \triangleq J_{MXZ}$$
$$J_{OX} \approx J_{OY} \triangleq J_{OXY}$$

These approximations will greatly simplify the torque equations. As shown in Table 3.1 the approximations are good ones. The largest error introduced is 8% for the E gimbal, 2.5% for the M gimbal and 0% for the other gimbals. The 8% E gimbal error will have a negligible effect because that gimbal should remain inertial and the exact value of its moment of inertia ought not to matter much.

It should be noted here that the symmetry of the E gimbal gives rise to some useful results.

$$J_E^E = \begin{vmatrix} J_{EX} & 0 & 0 \\ 0 & J_{EY} & 0 \\ 0 & 0 & J_{EZ} \end{vmatrix} = \begin{vmatrix} J_{EXYZ} & 0 & 0 \\ 0 & J_{EXYZ} & 0 \\ 0 & 0 & J_{EXYZ} \end{vmatrix} = J_{EXYZ} I \qquad (3.23)$$

Thus in any coordinate system k,

$$J_E^k = C_E^k J_E^E C_k^E = C_E^k (J_{EXYZ} I) C_E^{k^{-1}} = J_{EXYZ} I \qquad (3.24)$$

$$J_E^E = J_E^I = J_E^M = J_E^O \qquad (3.25)$$

Equation (3.25) has the following interpretation: J_E^E, J_E^I, J_E^M and J_E^O are all different tensors; they just happen to share the same matrix representation.

Table 3.1

Moments of Inertia

Gimbal	Axis	Name	Value
Elevation	X	J_{EX}	1.3
	Y	JEY	1.2
	Z	JEZ	1.1
Inner	X	JIX	1.7
	Y	JIY	1.3
	Z	JIZ	1.3
Middle	X	J_{MX}	2.2
	Y	JMY	3.0
	Z	JMZ	2.3
Outer	X	JOX	3.0
	Y	JOY	3.0
	Z	JOZ	3.9

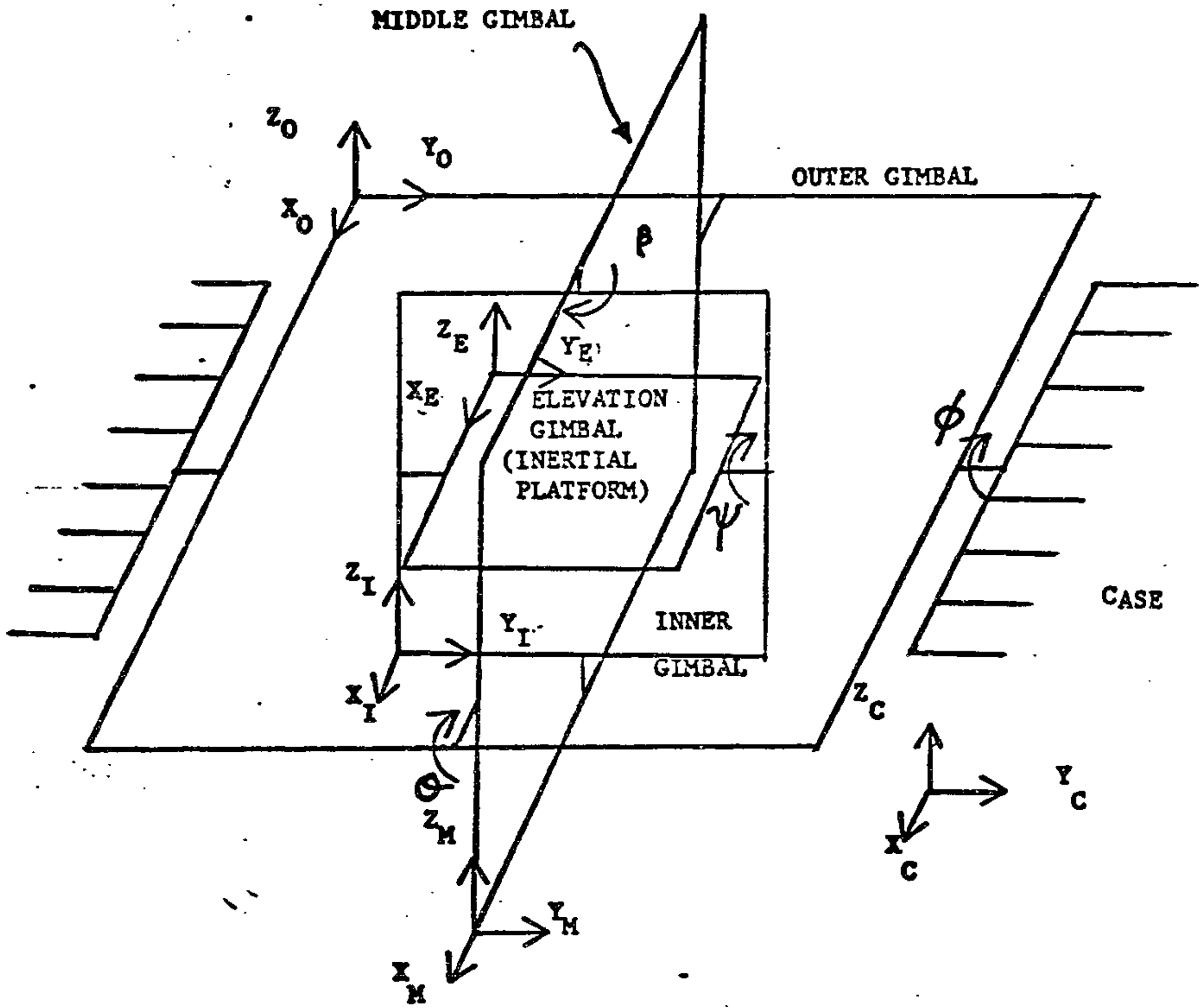


Figure 3.1

Four-Gimbal System

IV. Derivation of Torque-Acceleration Equations

Torque is the rate of change of angular momentum. For a four-gimbal system there will be four angular momentum vectors to consider, one for each gimbal. This will lead to four torque equations. These four torque equations will be solved for the four angular accelerations ($\ddot{Y}$, $\ddot{B}$, $\ddot{O}$, $\ddot{\emptyset}$). The angular accelerations can be integrated twice to solve for the angular velocities and the angles themselves, thus completely characterizing the system.

Torques are applied to gimbals through their pivot assemblies. Torques may be applied either along a torque motor axis or normal to a torque motor axis or both. Torques normal to a motor axis are coupled through the bearings; these forces are not controlled directly. Control is exerted directly only on the components of torque along the motor axes. There are four sources in all of torques about a motor axis. They are control voltage, back-emf, coulomb friction and static friction.

Let's examine the relationship between angular momentum and torque. The angular momentum of gimbal j with respect to inertial space (1) is

---

(1) Strictly speaking, angular momentum is only defined with respect to inertial space. Nonetheless it will be convenient to treat angular momentum like any other vector, especially as regards coordinate transformations. As long as we remember that torque is the rate of change of angular momentum in inertial space there will be no problem.

$$\vec{H}^S_j = J^S_j \vec{W}^S_{Sj} \tag{4.1}$$

In the j coordinate system equation (4.1) becomes:

$$\begin{aligned}\vec{H}^j_j &= C^j_S \vec{H}^S_j = C^j_S J^S_j \vec{W}^S_{Sj} = C^j_S J^S_j C^S_j C^j_S \vec{W}^S_{Sj} \\ &= J^j_j \vec{W}^j_{Sj}\end{aligned} \tag{4.2}$$

Torque is the time derivative of angular momentum in an inertial coordinate frame. Differentiating equation (4.2) and applying (A.2) and (A.8)

$$\begin{aligned}\vec{T}^j_j &= C^j_S \vec{T}^S_j = C^j_S d/dt(\vec{H}^S_j) = C^j_S d/dt(C^S_j \vec{H}^j_j) \\ &= C^j_S [C^S_j \dot{\vec{H}}^j_j - \vec{W}^S_{Sj} \times (C^S_j \vec{H}^j_j)] \\ &= \dot{\vec{H}}^j_j + C^j_S [\vec{W}^S_{Sj} \times (C^S_j \vec{H}^j_j)] \\ &= \dot{\vec{H}}^j_j + (C^j_S \vec{W}^S_{Sj}) \times (C^j_S C^S_j \vec{H}^j_j) \\ &= \dot{\vec{H}}^j_j + \vec{W}^j_{Sj} \times \vec{H}^j_j\end{aligned} \tag{4.3}$$

For a rigid body such as a gimbal, $d/dt(J^j_j) = 0$, so equation (4.3) becomes:

$$\vec{T}^j_j = J^j_j \dot{\vec{W}}^j_{Sj} + \vec{W}^j_{Sj} \times (J^j_j \vec{W}^j_{Sj}) \tag{4.4}$$

For the O gimbal we have:

$$\vec{T}^O_O = \vec{T}^O_{CO} + \vec{T}^O_{MO} = J^O_O \dot{\vec{W}}^O_{SO} + \vec{W}^O_{SO} \times (J^O_O \vec{W}^O_{SO}) \tag{4.5}$$

$$\vec{T}^O_{CO} = J^O_O \dot{\vec{W}}^O_{SO} + \vec{W}^O_{SO} \times (J^O_O \vec{W}^O_{SO}) + C^O_M \vec{T}^M_{OM} \tag{4.6}$$

$\vec{T}^O_{CO}$ represents the torque transmitted from the case to the outer gimbal as seen from the outer gimbal. The form of equation (4.6) will prove most useful. Similar equations can be written for the other gimbals.

$$\vec{T}^{M}_{OM} = J^{M}_{M}\dot{\vec{W}}^{M}_{SM} + \vec{W}^{M}_{SM} \times (J^{M}_{M}\vec{W}^{M}_{SM}) + C^{M}_{I}\vec{T}^{I}_{MI} \quad (4.7)$$

$$\vec{T}^{I}_{MI} = J^{I}_{I}\dot{\vec{W}}^{I}_{SI} + \vec{W}^{I}_{SI} \times (J^{I}_{I}\vec{W}^{I}_{SI}) + C^{I}_{E}\vec{T}^{E}_{IE} \quad (4.8)$$

$$\vec{T}^{E}_{IE} = J^{E}_{E}\dot{\vec{W}}^{E}_{SE} + \vec{W}^{E}_{SE} \times (J^{E}_{E}\vec{W}^{E}_{SE}) \quad (4.9)$$

Because the elevation gimbal is assumed to be symmetric, and $\vec{W}^{E}_{SE}$ is to be kept small, equation (4.9) reduces to:

$$\vec{T}^{E}_{IE} \simeq J^{E}_{E}\dot{\vec{W}}^{E}_{SE} \quad (4.10)$$

The torque motor force from the inner to the elevation gimbal is along the y-axis of both gimbals.

$$T^{E}_{IEY} = J_{EXYZ}\dot{W}_{EY}$$
$$= J_{EXYZ}\{\ddot{\Upsilon} - \sin B\,(-\dot{\emptyset}\, W_{OZ} + \ddot{\Theta}) + \cos B\,(\cos\Theta\, \ddot{\emptyset} + \sin\Theta\, \dot{\emptyset}\, W_{OX} + \dot{\Theta}\, W_{MZ}) - \dot{B}\, W_{IX}\} \quad (4.11)$$

Equation (4.11) can be rewritten as:

$$Y_E = J_{EXYZ}\ddot{\Upsilon} - \sin B\, J_{EXYZ}\ddot{\Theta} + \cos B \cos\Theta\, J_{EXYZ}\ddot{\emptyset} \quad (4.12)$$

Where $Y_E \triangleq$

$$T^{E}_{IEY} + J_{EXYZ}(-\sin B\, \dot{\emptyset}\, W_{OZ} - \cos B \sin\Theta\, \dot{\emptyset}\, W_{OX} - \cos B\, \dot{\Theta}\, W_{MZ} + \dot{B}\, W_{IX}) \quad (4.13)$$

Y_E is a quantity that contains all of the terms of the torque equation for T^{E}_{IEY} that do not contain an angular acceleration. Similar definitions will be made for the other gimbals. Proceeding in a parallel manner with the inner gimbal we repeat equation (4.8).

$$\vec{T}^{I}_{MI} = J^{I}_{I}\dot{\vec{W}}^{I}_{SI} + \vec{W}^{I}_{SI} \times (J^{I}_{I}\vec{W}^{I}_{SI}) + C^{I}_{E}\vec{T}^{E}_{IE} \quad (4.8)$$

Applying equations (3.14) and (4.13) to (4.8):

$$\begin{aligned}
\vec{T}^{I}_{MI} &= J^{I}_{I}\dot{\vec{W}}^{I}_{SI} + \vec{W}^{I}_{SI} \times (J^{I}_{I}\vec{W}^{I}_{SI}) + C^{I}_{E}J^{E}_{E} (C^{E}_{I}\dot{\vec{W}}^{I}_{SI} \\
&\quad - \vec{W}^{E}_{IE} \times (C^{E}_{I}\vec{W}^{I}_{SI}) + \dot{\vec{W}}^{E}_{IE}) \\
&= (J^{I}_{I} + C^{I}_{E}J^{E}_{E}C^{E}_{I})\, \dot{\vec{W}}^{I}_{SI} + \vec{W}^{I}_{SI} \times (J^{I}_{I}\vec{W}^{I}_{SI}) \\
&\quad + C^{I}_{E}J^{E}_{E}C^{E}_{I} (-\vec{W}^{I}_{IE} \times \vec{W}^{I}_{SI} + \dot{\vec{W}}^{I}_{IE}) \\
&= (J^{I}_{I} + J^{I}_{E})\, \dot{\vec{W}}^{I}_{SI} + \vec{W}^{I}_{SI} \times (J^{I}_{I}\vec{W}^{I}_{SI}) + J^{I}_{E} (-\vec{W}^{I}_{IE} \times \vec{W}^{I}_{IE} + \dot{\vec{W}}^{I}_{IE})
\end{aligned} \tag{4.14}$$

Recalling equation (3.25) and expanding (4.14):

$$\vec{T}^{I}_{MI} = \begin{vmatrix} J_{IX} + J_{EXYZ} & 0 & 0 \\ 0 & J_{IYZ} + J_{EXYZ} & 0 \\ 0 & 0 & J_{IYZ} + J_{EXYZ} \end{vmatrix} \begin{vmatrix} \dot{W}^{I}_{SIX} \\ \dot{W}^{I}_{SIY} \\ \dot{W}^{I}_{SIZ} \end{vmatrix}$$

$$+ \begin{vmatrix} W_{IY}\, W_{IZ}\, (J_{IYZ} - J_{IYZ}) \\ W_{IZ}\, W_{IX}\, (J_{IX} - J_{IYZ}) \\ W_{IX}\, W_{IY}\, (J_{IYZ} - J_{IX}) \end{vmatrix} + \begin{vmatrix} J_{EXYZ} & 0 & 0 \\ 0 & J_{EXYZ} & 0 \\ 0 & 0 & J_{EXYZ} \end{vmatrix} \begin{vmatrix} -\dot{Y}\, W_{IZ} \\ \ddot{Y} \\ \dot{Y}\, W_{IX} \end{vmatrix}$$

$$= \begin{vmatrix} (J_{IX} + J_{EXYZ})\, \dot{W}_{IX} - J_{EXYZ}W_{IZ}\, \dot{Y} \\ (J_{IYZ} + J_{EXYZ})\, \dot{W}_{IY} + J_{EXYZ}\ddot{Y} \\ (J_{IYZ} + J_{EXYZ})\, \dot{W}_{IZ} + J_{EXYZ}W_{IX}\, \dot{Y} + W_{IX}\, W_{IY}\, (J_{IYZ} - J_{IX}) \end{vmatrix} \tag{4.15}$$

When the elevation gimbal is inertial both W_{IX} and W_{IZ} should be small; ideally they will be zero. The product of such small terms will certainly be negligible. Therefore the term W_{IX} W_{IZ} $(J_{IX} - J_{IYZ})$ has been dropped from equation (4.15). Motor torque from the M to the I gimbal is along the z-axis.

$$T^{I}_{MIZ} = (J_{IYZ} + J_{EXYZ})\, \dot{W}_{IZ} + W_{IX} W_{IY} (J_{IYZ} - J_{IX}) + J_{EXYZ}\, \dot{\Upsilon}\, W_{IX}$$
$$= (J_{IYZ} + J_{EXYZ})\, (-\sin\Theta\, \ddot{\emptyset} + \cos\Theta\, \dot{\emptyset}\, W_{OX} - \dot{\Theta}\, W_{MX} + \ddot{B}) + W_{IX} W_{IY} (J_{IYZ} - J_{IX}) + J_{EXYZ}\, \dot{\Upsilon}\, W_{IX} \tag{4.16}$$

Equation (4.16) can be rewritten in a similar fashion to (4.11):

$$Y_I = (J_{IYZ} + J_{EXYZ})\, (\ddot{B} - \sin\Theta\, \ddot{\emptyset}) \tag{4.17}$$

Where $Y_I \hat{=} T^{I}_{MIZ} + (J_{IYZ} + J_{EXYZ})\, (\dot{\Theta}\, W_{MX} - \cos\Theta\, \dot{\emptyset}\, W_{OX})$

$$+ W_{IX} W_{IY} (J_{IX} - J_{IYZ}) - \dot{\Upsilon}\, J_{EXYZ}\, W_{IX} \tag{4.18}$$

Proceeding to the middle gimbal:

$$\vec{T}^{M}_{OM} = J^{M}_{M}\dot{\vec{W}}^{M}_{SM} + \vec{W}^{M}_{SM} \times (J^{M}_{M}\vec{W}^{M}_{SM}) + C^{M}_{I}\vec{T}^{I}_{MI} \tag{4.7}$$

$$= \begin{vmatrix} J_{MXZ} & 0 & 0 \\ 0 & J_{MY} & 0 \\ 0 & 0 & J_{MXZ} \end{vmatrix} \begin{vmatrix} \dot{W}_{MX} \\ \dot{W}_{MY} \\ \dot{W}_{MZ} \end{vmatrix} + \begin{vmatrix} W_{MY} W_{MZ} (J_{MXZ} - J_{MY}) \\ W_{MZ} W_{MX} (J_{MXZ} - J_{MXZ}) \\ W_{MX} W_{MY} (J_{MY} - J_{MXZ}) \end{vmatrix}$$

$$+ \begin{vmatrix} \cos B & -\sin B & 0 \\ \sin B & \cos B & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} (J_{IX} + J_{EXYZ})\, \dot{W}_{IX} - J_{EXYZ}\, W_{IZ}\, \dot{\Upsilon} \\ (J_{IYZ} + J_{EXYZ})\, \dot{W}_{IY} + J_{EXYZ}\, \ddot{\Upsilon} \\ (J_{IYZ} + J_{EXYZ})\, \dot{W}_{IZ} + W_{IX} W_{IY} (J_{IYZ} - J_{IX}) + J_{EXYZ}\, \dot{\Upsilon}\, W_{IX} \end{vmatrix} \tag{4.19}$$

Torque from the O to the M gimbal is along the x-axis. Using equations (3.17) and (3.18) the x component of (4.19) can be expanded as follows:

$$T^{M}_{OMX} = J_{MXZ}\, \dot{W}_{MX} + W_{MY} W_{MZ} (J_{MXZ} - J_{MY})$$
$$+ \cos B\, \{(J_{IX} + J_{EXYZ})\, \dot{W}_{IX} - J_{EXYZ}\, W_{IZ}\, \dot{\Upsilon}\}$$
$$- \sin B\, \{(J_{IYZ} + J_{EXYZ})\, \dot{W}_{IY} + J_{EXYZ}\, \ddot{\Upsilon}\}$$

$$= J_{MXZ}\,\dot{W}_{MX} + W_{MY}\,W_{MZ}\,(J_{MXZ} - J_{MY})$$
$$+ \cos B\,(J_{IX} + J_{EXYZ})\,(\cos B\,\dot{W}_{MX} + \sin B\,\dot{W}_{MY} + \dot{B}\,W_{IY})$$
$$- \cos B\,J_{EXYZ}\,W_{IZ}\,\dot{\Upsilon}$$
$$- \sin B\,(J_{IYZ} + J_{EXYZ})\,(-\sin B\,\dot{W}_{MX} + \cos B\,\dot{W}_{MY} - \dot{B}\,W_{IX})$$
$$- \sin B\,J_{EXYZ}\,\ddot{\Upsilon}$$
$$= (J_{MXZ} + \cos^2 B\,J_{IX} + \sin^2 B\,J_{IYZ} + J_{EXYZ})\,(\ddot{\Theta} - \dot{\emptyset}\,W_{OZ})$$
$$+ W_{MY}\,W_{MZ}\,(J_{MXZ} - J_{MY})$$
$$+ \sin B\,\cos B\,(J_{IX} - J_{IYZ})\,(\cos\Theta\,\ddot{\emptyset} + \sin\Theta\,\dot{\emptyset}\,W_{OX} + \dot{\Theta}\,W_{MZ})$$
$$+ \dot{B}\,\{\cos B\,(J_{IX} + J_{EXYZ})\,(-\sin B\,W_{MX} + \cos B\,W_{MY})$$
$$+ \sin B\,(J_{IYZ} + J_{EXYZ})\,(\cos B\,W_{MX} + \sin B\,W_{MY})\}$$
$$- \cos B\,J_{EXYZ}\,W_{IZ}\,\dot{\Upsilon} - \sin B\,J_{EXYZ}\,\ddot{\Upsilon} \tag{4.20}$$

Equation (4.20) can be rearranged like this:

$$Y_M = -\sin B\,J_{EXYZ}\,\ddot{\Upsilon}$$
$$+ (J_{MXZ} + \cos^2 B\,J_{IX} + \sin^2 B\,J_{IYZ} + J_{EXYZ})\,\ddot{\Theta}$$
$$+ \sin B\,\cos B\,\cos\Theta\,(J_{IX} - J_{IYZ})\,\ddot{\emptyset} \tag{4.21}$$

Where $Y_M \mathrel{\hat{=}}$

$$T^M_{OMX}$$
$$+ (J_{MXZ} + \cos^2 B\,J_{IX} + \sin^2 B\,J_{IYZ} + J_{EXYZ})\,\dot{\emptyset}\,W_{OX}$$
$$+ \sin B\,\cos B\,(J_{IYZ} - J_{IX})\,(\sin\Theta\,\dot{\emptyset}\,W_{OX} + \dot{\Theta}\,W_{MZ} - \dot{B}\,W_{MX})$$
$$- (\cos^2 B\,J_{IX} + \sin^2 B\,J_{IYZ} + J_{EXYZ})\,\dot{B}\,W_{MY}$$
$$+ \cos B\,J_{EXYZ}\,W_{IZ}\,\dot{\Upsilon} \tag{4.22}$$

Finally, for the outer gimbal:

$$\vec{T}^{O}_{CO} = J^{O}_{O}\dot{\vec{W}}^{O}_{SO} + \vec{W}^{O}_{SO} \times (J^{O}_{O}\vec{W}^{O}_{SO}) + C^{O}_{M}\vec{T}^{M}_{OM} \qquad (4.6)$$

The component of interest here is along the y-axis since that is where the torque motor is. The algebra required is extremely tedious, and little insight is obtained. We will not go through the entire derivation. A rigorous derivation is given in [6]. The resulting equation for Y_O (which is really all we want) is:

$$\begin{aligned} Y_O = & \cos B \cos\theta \, J_{EXYZ} \ddot{Y} \\ & - \sin\theta \, (J_{IYZ} + J_{EXYZ}) \ddot{B} \\ & + \sin B \cos B \cos\theta \, (J_{IX} - J_{IYZ}) \ddot{\theta} \\ & + (J_{OXY} + \sin^2\theta \, J_{MXZ} + \cos^2\theta \, J_{MY} + \sin^2\theta \, J_{IYZ} \\ & + \sin^2 B \cos^2\theta \, J_{IX} + \cos^2 B \cos^2\theta \, J_{IYZ} + J_{EXYZ}) \ddot{\emptyset} \end{aligned} \qquad (4.23)$$

Where $Y_O \triangleq$

$$\begin{aligned} & T^{O}_{COY} \\ & + W_{OX} W_{OY} (J_{OZ} - J_{OXY}) \\ & + \sin B \cos B \cos\theta \, (J_{IX} - J_{IYZ}) (\dot{\emptyset} W_{OZ} - \dot{B} W_{MY}) \\ & + \sin\theta \cos\theta \, (J_{MXZ} - J_{MY} + \sin^2 B \{J_{IYZ} - J_{IX}\}) (\dot{\emptyset} W_{OX} - \dot{\theta} W_{OY}) \\ & - (\sin^2\theta \, J_{MXZ} + \cos^2\theta \, J_{MY} + \sin^2\theta \, J_{IYZ} + \sin^2 B \cos^2\theta \, J_{IX} \\ & \quad + \cos^2 B \cos^2\theta \, J_{IYZ} + J_{EXYZ}) \dot{\theta} W_{OZ} \\ & + \sin\theta \, W_{MY} W_{MX} (J_{MY} - J_{MXZ}) \\ & + \sin\theta \, W_{IY} W_{IX} (J_{IYZ} - J_{IX}) \\ & + \sin B \cos\theta \, J_{EXYZ} \dot{Y} W_{IZ} \\ & + \sin\theta \, J_{EXYZ} \dot{Y} W_{IX} \end{aligned} \qquad (4.24)$$

Equations (4.12), (4.17), (4.21) and (4.23) may be combined into a single matrix equation.

$$\vec{Y} = L\,\vec{Z} \qquad (4.25)$$

Where $\vec{Y} \triangleq (Y_E, Y_I, Y_M, Y_O)^T$

$\vec{Z} \triangleq (\ddot{Y}, \ddot{B}, \ddot{\theta}, \ddot{\phi})^T$

$$L = \begin{vmatrix} L_{11} & L_{12} & L_{13} & L_{14} \\ L_{21} & L_{22} & L_{23} & L_{24} \\ L_{31} & L_{32} & L_{33} & L_{34} \\ L_{41} & L_{42} & L_{43} & L_{44} \end{vmatrix}$$

With $L_{11} = J_{EXYZ}$ (4.26)

$L_{12} = L_{21} = 0$ (4.27)

$L_{13} = L_{31} = -\sin B\ J_{EXYZ}$ (4.28)

$L_{14} = L_{41} = \cos B \cos\theta\ J_{EXYZ}$ (4.29)

$L_{22} = J_{IYZ} + J_{EXYZ}$ (4.30)

$L_{23} = L_{32} = 0$ (4.31)

$L_{24} = L_{42} = -\sin\theta\ (J_{IYZ} + J_{EXYZ})$ (4.32)

$L_{33} = J_{MXZ} + \cos^2 B\ J_{IX} + \sin^2 B\ J_{IYZ} + J_{EXYZ}$ (4.33)

$L_{34} = L_{43} = \sin B \cos B \cos\theta\ (J_{IX} - J_{IYZ})$ (4.34)

$$L_{44} = J_{OXY} + \sin^2\theta\ J_{MXZ} + \cos^2\theta\ J_{MY} + \sin^2\theta\ J_{IYZ} + \sin^2 B \cos^2\theta\ J_{IX} + \cos^2 B \cos^2\theta\ J_{IYZ} + J_{EXYZ} \qquad (4.35)$$

A term lying on the diagonal of L, L_{ii}, is the effective moment of inertia of gimbal i and those gimbals inside it as seen looking into the pivot axis of gimbal i. For example, if all four gimbals are treated as a single unit, then the inertia along the y-axis of the outer gimbal is just L_{44}. Similarly, L_{33} is the inertia of the three innermost gimbals along the x-axis of the middle gimbal. The off-diagonal terms of L are a consequence of the fact that an inertia matrix may no longer be diagonal if coordinatized in a coordinate system not attached to the appropriate gimbal.

L contains information about the geometry of the gimbals. We have already assumed that the elevation gimbal is symmetric. This implies that the orientation of the elevation gimbal is not relevant to the overall geometry of the system and therefore we would not expect the elevation angle Υ to appear in equations (4.26) through (4.35). The outer angle $\emptyset$ also should not affect the gimbal geometry, so we would not expect $\emptyset$ to appear in equations (4.26) through (4.35) either. These expectations are realized. The gimbal configuration as defined by the matrix L is only a function of B and Θ.

Note that L is symmetric. This is an instance of a reciprocity relationship between torque and angular acceleration. A torque applied at angle i will produce a response at angle j equal to the response at angle i to a torque at angle j.

The actual torque values are nestled into the Y vector together with a great many other terms having the same dimensions as torque. These other terms for the most part resemble Coriolis forces, although their exact interpretation is not always obvious. In any event, for reasonable gimbal rates and reasonable torque levels the torque terms will dominate the Coriolis forces.

Equation (4.25) allows the computation of torque given acceleration. In actuality we know the torque since the controller will be supplying the control signals; it is the acceleration we wish to compute. So we may take the inverse of equation (4.25) to ccme up with:

$$\vec{Z} = L^{-1}\vec{Y} = M\,\vec{Y} \qquad \text{where } M \triangleq L^{-1} \tag{4.36}$$

M will of course be symmetric since L is. The computation of M is aided by repeated application of the following matrix identity:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]^{-1} = \left[\begin{array}{c|c} A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1} \\ \hline -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1} \end{array}\right] \tag{4.37}$$

Before presenting the terms of M it is helpful to define a quantity called DENOM. DENOM is the determinant of $(L_{44}-L_{43}L_{33}^{-1}L_{34})$ obtained when using formula (4.37). Since this quantity appears in each element of M it will be much easier to define DENOM once than to write it out in full each time.

$$\begin{aligned} \text{DENOM} = {} & [J_{OXY} + \sin^2\theta\, J_{MXZ} + \cos^2\theta\, (J_{MY} + \cos^2 B\, J_{IYZ} + \sin^2 B\, \{J_{IX} \\ & + J_{EXYZ}\})]\, [J_{MXZ} + \cos^2 B\, (J_{IX} + J_{EXYZ}) + \sin^2 B\, J_{IYZ}] \\ & - \cos^2 B \sin^2 B \cos^2\theta\, [J_{IYZ} - J_{IX} - J_{EXYZ}]^2 \end{aligned} \tag{4.38}$$

$$\begin{aligned} M_{11} = {} & \{[J_{OXY} + \sin^2\theta\, J_{MXZ} + \cos^2\theta\, (J_{MY} + J_{IX} + J_{EXYZ})] \\ & [J_{EXYZ} + \sin^2 B\, J_{IYZ} + \cos^2 B\, J_{IX} + J_{MXZ}] \\ & - \cos^2 B \cos^2\theta\, [J_{IX} - J_{IYZ}]\, [J_{MXZ} + J_{IX} + J_{EXYZ}]\} \\ & / \text{ DENOM } / J_{EXYZ} \end{aligned} \tag{4.39}$$

$$M_{12} = -\cos B \sin\theta \cos\theta\, [J_{EXYZ} + J_{IX} + J_{MXZ}] \;/\; \text{DENOM} \tag{4.40}$$

$$\begin{aligned} M_{13} = {} & \sin B\, [J_{OXY} + \sin^2\theta\, J_{MXZ} \\ & + \cos^2\theta\, (J_{MY} + J_{IX} + J_{EXYZ})] \;/\; \text{DENOM} \end{aligned} \tag{4.41}$$

$$M_{14} = -\cos B \cos\theta\, [J_{MXZ} + J_{IX} + J_{EXYZ}] \;/\; \text{DENOM} \tag{4.42}$$

$$M_{22} = \{\cos^2 B \cos^2\Theta \, [J_{IYZ} - J_{IX} - J_{EXYZ}] \, [J_{EXYZ} + J_{IX} + J_{MXZ}] + [J_{MXZ} + \cos^2 B \, (J_{IX} + J_{EXYZ}) + \sin^2 B \, J_{IYZ}] \, [J_{OXY} + \cos^2\Theta \, (J_{MY} + J_{IX}) + \sin^2\Theta \, (J_{MXZ} + J_{IYZ}) + J_{EXYZ}]\} \, / \, [J_{IYZ} + J_{EXYZ}] \, / \, DENOM \quad (4.43)$$

$$M_{23} = \sin B \cos B \sin\Theta \cos\Theta \, [J_{IYZ} - J_{IX} - J_{EXYZ}] \, / \, DENOM \quad (4.44)$$

$$M_{24} = \sin\Theta \, [J_{MXZ} + \cos^2 B \, (J_{IX} + J_{EXYZ}) + \sin^2 B \, J_{IYZ}] \, / \, DENOM \quad (4.45)$$

$$M_{33} = [J_{OXY} + \sin^2\Theta \, J_{MXZ} + \cos^2\Theta \, (J_{MY} + \cos^2 B \, J_{IYZ} + \sin^2 B \, \{J_{IX} + J_{EXYZ}\})] \, / \, DENOM \quad (4.46)$$

$$M_{34} = \cos B \sin B \cos\Theta \, [J_{IYZ} - J_{IX} - J_{EXYZ}] \, / \, DENOM \quad (4.47)$$

$$M_{44} = [J_{MXZ} + \cos^2 B \, (J_{IX} + J_{EXYZ}) + \sin^2 B \, J_{IYZ}] \, / \, DENOM \quad (4.48)$$

The above equations are rather difficult to manipulate and verify. By writing a computer program to numerically multiply L amd M it was found that M is indeed the inverse of L.

V. Torque and Friction

The torque produced by a given torque motor is proportional to the current through it. The constant of proportionality is Kt/r. This current will equal the applied voltage, in this case the control signal, minus the back-emf generated by the motor, divided by the resistance of the motor windings. Back-emf is created when a torque motor acts like an electric generator, putting out a voltage proportional to the relative rotation rate of the rotor and stator, tending to cancel any rotation of the gimbals. The constant of proportionality is denoted by Kv. These torque motor parameters will differ from gimbal to gimbal. Inductive effects in the motors are negligible.

Anathema to designers of precision guidance equipment, friction is nonetheless a force to be reckoned with, or at least accounted for. It is a major factor in the four-gimbal system; much of the torque supplied by the torque motors is used to overcome friction. In fact, in the absence of friction there would be almost no forces acting to perturb the inertial platform except in the neighborhood of gimbal lock.

There are essentially two types of friction: static friction and Coulomb friction. Although they originate in the same intermolecular forces the analysis of the two types of friction is substantially different. We deal first with Coulomb friction.

Gimbals in relative motion will be subject to Coulomb friction. We will use a very simple model for friction in the simulation.

$$\text{Tcoulomb} = -\text{sgn (relative gimbal rate) X Tcoulomb-limit} \quad (5.1)$$

$$\text{Where } \text{sgn}(x) = \begin{cases} -1 & x<0 \\ 0 & x=0 \\ 1 & x>0 \end{cases}$$

This simple model has adequately predicted Coulomb friction in earlier simulations. It has the advantage of requiring only a single parameter for each gimbal. Gully[7] goes into more sophisticated models. The net torque at each pivot can now be determined in terms of control signals and rotation rates.

$$T^{E}_{IEY} = (Kt/r)_E \{U_E - (Kv)_E \dot{I}\} - \text{sgn}(\dot{I})\ Tcl_E \quad (5.2)$$
$$T^{I}_{MIZ} = (Kt/r)_I \{U_I - (Kv)_I \dot{B}\} - \text{sgn}(\dot{B})\ Tcl_I \quad (5.3)$$
$$T^{M}_{OMX} = (Kt/r)_M \{U_M - (Kv)_M \dot{\theta}\} - \text{sgn}(\dot{\theta})\ Tcl_M \quad (5.4)$$
$$T^{O}_{COY} = (Kt/r)_O \{U_O - (Kv)_O \dot{\emptyset}\} - \text{sgn}(\dot{\emptyset})\ Tcl_O \quad (5.5)$$

Static friction or stiction as it is often called, is the force tending to prevent adjacent bodies from moving at all relative to one another once they have stopped moving. Static friction is in general stronger than Coulomb friction, the latter being effective only after the onset of relative motion. Static friction is quite annoying from the viewpoint of the four-gimbal controller. It means that a comparatively large torque must be applied to get a stuck gimbal pair unstuck.

The model used for static friction here is extremely simple. Others are certainly possible and ought to be analyzable in the same framework. The model used here is characterized by a single parameter, the static friction torque limit. The static friction torque limit will differ from gimbal to gimbal. The model works as follows:

Whenever two adjacent gimbals are not in relative motion (i.e. their relative rotation rate is zero) they will be considered stuck until the magnitude of the torque supplied by a torque motor from one gimbal to the other exceeds the static friction torque limit. If a greater amount of torque is applied, then the gimbals will be free to rotate subject to Coulomb friction. If the relative rotation rate is nonzero, no matter how small in magnitude, then the gimbals will not be stuck.

This may cause some difficulty in the computer simulation of the system. Because of numerical considerations it is unlikely that the relative rate of any gimbal pair will exactly equal zero in the simulation. The approach taken then is to check if the relative rotation rate about any axis has recently passed through zero (i.e. changed sign). If so, then a comparison of applied torque with the static friction limit is made as though the rotation rate were exactly zero, and the system is treated accordingly.

When two gimbals are stuck they will travel together. Neither a relative velocity nor a relative acceleration will be experienced, despite any applied torque up to the static friction torque limit. This causes problems in applying equation (4.25). We no longer know the net torque being supplied between the stuck gimbals. The motor torque is known, but not the amount of stiction. Static friction will be just adequate to prevent motion along the affected axis, but it is not possible to predict a priori. Calculation of angular acceleration by means of equation (4.36) is thereby rendered impossible. Some other method is required.

The method used is to go back to equation (4.25). If all net torques were known then (4.25) could be inverted as was done in (4.36). But the net torque will not be known at a stuck gimbal pair. So a constraint will have been lost from equation (4.25) and the system will be indeterminate. However, another constraint may be added, namely that the acceleration of the affected angle will be zero. This can be best expressed by rearranging and partitioning the elements of equation (4.25)

Let $\vec{Y}_1$ be a vector containing those elements of $\vec{Y}$ not affected by stiction. $\vec{Y}_1$ can be computed since the net torque is readily computable in the absence of stiction. Let $\vec{Z}_1$ be a vector containing the angular accelerations in $\vec{Z}$ not affected by stiction. These are the values we wish to compute. Similarly, let $\vec{Y}_2$ be a vector containing the elements of $\vec{Y}$ that are affected by static friction. Even though the torque motor contributions to $\vec{Y}_2$ will be known, the static friction contributions will not, as was discussed above. Lastly, let $\vec{Z}_2$ be a vector containing those angular accelerations that are affected by static friction. $\vec{Z}_2$ will be identically zero. Introduce a new matrix L' whose elements are permuted elements of L such that:

$$\begin{vmatrix} \vec{Y}_1 \\ \hline \vec{Y}_2 \end{vmatrix} = L' \begin{vmatrix} \vec{Z}_1 \\ \hline \vec{Z}_2 \end{vmatrix} \tag{5.6}$$

L' can be partitioned like so:

$$L' = \left|\begin{array}{c|c} L'_{11} & L'_{12} \\ \hline L'_{21} & L'_{22} \end{array}\right| \tag{5.7}$$

Equations (5.6) and (5.7) can be combined as follows:

$$\vec{Y}_1 = L'_{11}\,\vec{Z}_1 + L'_{12}\,\vec{Z}_2$$

$$= L'_{11}\,\vec{Z}_1 \tag{5.8}$$

$$\vec{Z}_1 = (L'_{11})^{-1}\,\vec{Y}_1 \tag{5.9}$$

So that

$$\vec{Z} = \left|\begin{array}{c} \vec{Z}_1 \\ \hline \vec{Z}_2 \end{array}\right| = \left|\begin{array}{c} (L'_{11})^{-1}\,\vec{Y}_1 \\ \hline \vec{0} \end{array}\right| \tag{5.10}$$

This is what we wanted. The presence of stiction leads to a smaller set of equations to solve. The exact contribution of static friction was not needed. If stiction is present and equation (5.10) is used, or stiction is absent and equation (4.36) is used, the angular accelerations, and thus the angles can be correctly determined.

VI. Optimal Control of the Four-Gimbal System

Up to this point, differential equations have been derived that relate acceleration of the gimbals to control signals. These have all been scalar equations although they may be considered selected components of a set of vector equations of the type exemplified by (4.3). Rearranging the scalar equations into state-space form will aid the application of modern optimal control theory to the four-gimbal problem. Define an 11-dimensional state vector $\vec{X}$ and a 4-dimensional control vector $\vec{U}$ by:

$$\vec{X} = (\Upsilon,\ B,\ \Theta,\ \emptyset,\ \dot{\Upsilon},\ \dot{B},\ \dot{\Theta},\ \dot{\emptyset},\ \Delta SR,\ \Delta J,\ \Delta SV)^T$$

$$\vec{U} = (U_E,\ U_I,\ U_M,\ U_O)^T$$

$\vec{X}$ is composed of the gimbal angles and velocities plus the inertial platform tilts ΔSV, ΔJ, ΔSR. The entire dynamics of the four-gimbal system can be compressed into a single nonlinear vector differential equation by writing:

$$\dot{\vec{X}} = \vec{f}\,(\vec{X},\ \vec{U},\ \vec{W}^{C}_{SC}) \tag{6.1}$$

Explicitly, $\dot{\vec{X}}$ may be expanded using equations (3.20)-(3.22) and (4.36) to yield:

$$\dot{X}_1 = X_5 \tag{6.2}$$

$$\dot{X}_2 = X_6 \tag{6.3}$$

$$\dot{X}_3 = X_7 \tag{6.4}$$

$$\dot{X}_4 = X_8 \tag{6.5}$$

$$\dot{X}_5 = M_{11}Y_1 + M_{12}Y_2 + M_{13}Y_3 + M_{14}Y_4 \tag{6.6}$$

$$\dot{X}_6 = M_{21}Y_1 + M_{22}Y_2 + M_{23}Y_3 + M_{24}Y_4 \quad (6.7)$$

$$\dot{X}_7 = M_{31}Y_1 + M_{32}Y_2 + M_{33}Y_3 + M_{34}Y_4 \quad (6.8)$$

$$\dot{X}_8 = M_{41}Y_1 + M_{42}Y_2 + M_{43}Y_3 + M_{44}Y_4 \quad (6.9)$$

$$\dot{X}_9 = \cos X_1 \; W_{IX} - \sin X_1 \; W_{IZ} \quad (6.10)$$

$$\dot{X}_{10} = W_{IY} + X_5 \quad (6.11)$$

$$\dot{X}_{11} = \sin X_1 \; W_{IX} + \cos X_1 \; W_{IZ} \quad (6.12)$$

The M's are functions of B and Θ, or X_2 and X_3. The W's are functions of case rates, gimbal angles and angular rates, and so can be expressed in terms of $\vec{W}^C_{SC}$ and X's. The Y's are also functions of X's and U's. Only the states, controls and case rates appear on the right hand side of equations (6.2)-(6.12) in accordance with the formulation (6.1).

One advantageous aspect of this formulation of the system relates to sensors. Each state variable has a unique sensor associated with it. X_1 through X_4 are measured by resolvers, X_5 through X_8 are measured by tachometers and X_9 through X_{11} are measured by gyros. There can be no question as to whether or not the system is observable. Measurement noise does complicate the picture somewhat, but filtering of the sensor data should suffice to provide accurate estimates of the state variables. The oft-quoted Separation Theorem permits issues of estimation to be considered separately from issues of control for linear systems. The system under study is not linear, but as we will shortly see, it can be approximated by linear equations. Henceforth we will not be concerned with estimation of state except insofar as it relates to the validity of simulation studies.

The nonlinear equations embodied in (6.1) are fine for numerical analysis and simulation. They allow for numerical integration of the dynamical equations given any inputs to predict the trajectory of the system. As far as optimal control is concerned, equation (6.1) is horrendous. The theory of nonlinear optimal control is difficult to apply to actual real-time processes. For this reason linear quadratic optimal control will be applied to a linearized discrete-time version of the state equations.

Start by looking at the system at time t0 and at short intervals thereafter. Over a short enough interval, tens of milliseconds for example, the system will not change state much and the dynamics may be faithfully described by linear equations.

It is necessary to choose a nominal operating point about which to perform the linearization. One could choose $\vec{X} = \vec{X}(t0)$, $\vec{U} = \vec{U}(t0)$ and $\vec{W} = \vec{W}^{C}_{SC}(t0)$. This is valid if $\vec{X}$, $\vec{U}$, and $\vec{W}$ are slowly time-varying. It has already been assumed that $\vec{W}$ is. $\vec{X}$ is also slowly changing on the time scale of interest here. But $\vec{U}$ need not be so constrained. $\vec{U}$, the control vector, is a quantity that ultimately will be minimized. Since U ideally will be near zero we will use $\vec{X} = \vec{X}(t0)$, $\vec{U} = \vec{0}$ and $\vec{W} = \vec{W}^{C}_{SC}(t0)$ as a nominal operating point. Assuming constant case velocity, equation (6.1) can be approximated by

$$\Delta\dot{\vec{X}} = (d\vec{f}/d\vec{X})\,\Delta\vec{X} + (d\vec{f}/d\vec{U})\,\Delta\vec{U} \tag{6.13}$$

$$\dot{\vec{X}}(t) - \dot{\vec{X}}(t0) \approx (d\vec{f}(\vec{X}, \vec{U}, \vec{W})/d\vec{X})\Big|_{\substack{\vec{X} = \vec{X}(t0) \\ \vec{U} = \vec{0}_C \\ \vec{W} = \vec{W}_{SC}}} \{\vec{X}(t) - \vec{X}(t0)\}$$

$$+ (d\vec{f}(\vec{X}, \vec{U}, \vec{W})/d\vec{U})\Big|_{\substack{\vec{X} = \vec{X}(t0) \\ \vec{U} = \vec{0}_C \\ \vec{W} = \vec{W}_{SC}}} \vec{U}(t) \qquad (6.14)$$

Equation (6.14) can be rewritten as:

$$\dot{\vec{X}}(t) = A\,\vec{X}(t) + B\,\vec{U}(t) + \dot{\vec{X}}(t0) - A\,\vec{X}(t0) \qquad (6.15)$$

$$\text{Where } A = d\vec{f}(\vec{X}, \vec{0}, \vec{W})/d\vec{X} \qquad (6.16)$$

$$B = d\vec{f}(\vec{X}, \vec{U}, \vec{W})/d\vec{U} \qquad (6.17)$$

Equation (6.15) is a linear continuous-time approximation to the four-gimbal system. Computation of A and B is extremely complex. Unfortunately, we do not have at our disposal a computer that can exactly simulate in a finite amount of time the continuous behavior of the system that is implicit in (6.15). It is appropriate to ask what the state of the system will be at time $t0 + \Delta t$ given the state and control at time t0. Simulating samples of the state will relieve the computational burden required for a continuous solution. Assuming $\vec{U}(t)$ to be constant in the interval [t0, t] and A to be nonsingular, the solution to the dynamical equation (6.15) is:

$$\vec{X}(t) = e^{A\,(t-t0)}\vec{X}(t0)$$
$$+ A^{-1}[e^{A\,(t-t0)} - I][B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)] \qquad (6.18)$$

Equation (6.18) can be differentiated versus time to check that it does solve the dynamical equation. Plugging in t0 for t allows us to check the initial conditions, too.

$$\begin{aligned}\dot{\vec{X}}(t) &= Ae^{A(t-t0)}\,\vec{X}(t0) + A^{-1}Ae^{A(t-t0)}\,[B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)]\\ &= Ae^{A(t-t0)}\,\vec{X}(t0) + e^{A(t-t0)}\,[B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)]\\ &\quad - [B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)] + [B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)]\\ &= A\{e^{A(t-t0)}\,\vec{X}(t0) + A^{-1}[e^{A(t-t0)} - I][B\,\vec{U} + \dot{\vec{X}}(t0)\\ &\quad - A\,\vec{X}(t0)]\} + B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)\\ &= A\,\vec{X}(t) + B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0) \qquad (6.19)\end{aligned}$$

$$\begin{aligned}\vec{X}(t0) &= e^{A(t0-t0)}\vec{X}(t0) + A^{-1}[e^{A(t0-t0)} - I][B\,\vec{U}\\ &\quad + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)]\\ &= \vec{X}(t0) \qquad (6.20)\end{aligned}$$

Denoting t by $t0 + \Delta t$, equation (6.18) can be rewritten as:

$$\begin{aligned}\vec{X}(t0+\Delta t) &= e^{A\Delta t}\vec{X}(t0) + A^{-1}[e^{A\Delta t} - I][B\,\vec{U} + \dot{\vec{X}}(t0) - A\,\vec{X}(t0)]\\ &= e^{A\Delta t}\vec{X}(t0) + A^{-1}[e^{A\Delta t} - I]\,B\,\vec{U}\\ &\quad + A^{-1}[e^{A\Delta t} - I][\dot{\vec{X}}(t0) - A\,\vec{X}(t0)] \qquad (6.21)\end{aligned}$$

Equation (6.21) can be put in discrete form as:

$$\vec{X}[n+1] = A^{*}\vec{X}[n] + B^{*}\vec{U}[n] + \vec{C}^{*} \qquad (6.22)$$

Where

$$A^{*} = e^{A\Delta t} \qquad (6.23)$$

$$B^{*} = A^{-1}[e^{A\Delta t} - I]B \qquad (6.24)$$

$$C^{*} = A^{-1}[e^{A\Delta t} - I][\dot{\vec{X}}(t0) - A\,\vec{X}(t0)] \qquad (6.25)$$

Because Δt is assumed small equations (6.23) through (6.25) may be approximated to second order:

$$A^{*} \approx I + A\Delta t/1! + A^{2}\Delta t^{2}/2! \qquad (6.26)$$

$$B^{*} = A^{-1}[I + A\Delta t/1! + A^{2}\Delta t^{2}/2! - I]B$$
$$= [I\Delta t + A\Delta t^{2}/2!]B \qquad (6.27)$$
$$C^{*} = [I\Delta t + A\Delta t^{2}/2!][\dot{\vec{X}}(t0) - A\ \vec{X}(t0)] \qquad (6.28)$$

A cost function will be used as a measure of performance. The performance index will be a quadratic function of those parameters to be minimized by the controller. They are motor torques, inertial platform misorientation as sensed by the gyros, inertial platform rotation rate, and proximity to gimbal lock. Only proximity to gimbal lock remains to be expressed mathematically.

The angle from the inner gimbal x-axis, X_I, to the outer gimbal plane defined by X_O and Y_O is a convenient measure of gimbal lock. This angle is called λ. λ can be shown to obey the following equation:

$$\sin\lambda = \sin B \sin\Theta \qquad (6.29)$$

Gimbal lock occurs when λ equals ±90 degrees. Equation (6.29) requires that both B equal ±90 degrees and Θ equal ±90 degrees for this to happen. In keeping with a philosophy of linearizing and sampling the equations, the gimbal lock contribution to performance is approximately:

$$\sin\lambda[n+1] = \cos B \sin\Theta\ \Delta B + \sin B \cos\Theta\ \Delta\Theta + \sin\lambda[n+1] \qquad (6.30)$$

The inertial platform rotation rates can be handled in similar fashion.

$$W_{EX}[n+1] = W_{EX}[n] + \sum_i (dW_{EX}/dX_i)\ \Delta X_i \qquad (6.31)$$

$$W_{EY}[n+1] = W_{EY}[n] + \sum_i (dW_{EY}/dX_i)\ \Delta X_i \qquad (6.32)$$

$$W_{EZ}[n+1] = W_{EZ}[n] + \sum_i (dW_{EZ}/dX_i)\ \Delta X_i \qquad (6.33)$$

Equations (6.30) through (6.33) can be combined into a single equation.

$$\vec{e}[n+1] = D[n]\ \vec{X}[n+1] + \vec{E}[n] \qquad (6.34)$$

Where $\vec{e}[n+1] = (\sin\lambda[n+1], \Delta SR[n+1], \Delta J[n+1], \Delta SV[n+1],$

$W_{EX}[n+1], W_{EY}[n+1], W_{EZ}[n+1])$ (6.35)

D[n] is a matrix of derivatives with respect to state

$\vec{E}[n]$ is a vector containing those terms in (6.30) through (6.33) not explicitly dependent on $\vec{X}[n+1]$

A quadratic cost function was chosen because of a desire to penalize large misorientations of the inertial platform over small ones. Perhaps it would be more appropriate to minimize the maximum torque rather than minimize the RMS torque, but the latter approach is compatible with a quadratic cost function and is certainly more tractable. The one-step performance index is given by:

$$J[n] = \vec{e}[n+1]^T Q\ \vec{e}[n+1] + \vec{U}[n]^T R\ \vec{U}[n] \qquad (6.36)$$

Where J[n] is a measure of system performance

$\vec{e}[n+1]$ is given by (6.35)

Q is a positive definite symmetric matrix reflecting the cost associated with any state

$\vec{U}[n]$ is the control vector

R is a positive definite symmetric matrix reflecting the cost associated with any control

The cost function in equation (6.36) can be rewritten using matrix trace. Equations (6.34) and (6.22) can then be used to express the cost in terms of $\vec{U}$.

$$J[n] = \vec{e}[n+1]^T Q\, \vec{e}[n+1] + \vec{U}[n]^T R\, \vec{U}[n] \qquad (6.36)$$

$$= Tr\{Q\, \vec{e}[n+1]\, \vec{e}[n+1]^T + R\, \vec{U}[n]\, \vec{U}[n]^T\}$$

$$= Tr\{Q\, (D[n](A^*\vec{X}[n] + B^*\vec{U}[n] + \vec{C}^*) + \vec{E}[n]) (D[n](A^*\vec{X}[n] + B^*\vec{U}[n] + \vec{C}^*) + \vec{E}[n])^T + R\, \vec{U}[n]\vec{U}[n]^T\} \qquad (6.37)$$

Applying the Matrix Minimum Principle[1,2] and taking the gradient of equation (6.37) with respect to $\vec{U}$ yields:

$$dJ/d\vec{U} = 2(B^{*T}D[n]^T Q\, D[n]\, B^* + R)\, \vec{U} + 2B^{*T}D[n]^T Q\, (D[n](A^*\vec{X}[n] + \vec{C}^*) + \vec{E}[n]) \qquad (6.38)$$

Setting equation (6.38) to $\vec{0}$ and solving for $\vec{U}opt$ while keeping in mind that things are really dependent on n gives:

$$\vec{U}opt = -(B^{*T}D^TQ\, D\, B^* + R)^{-1}B^{*T}D^TQ\, (D(A^*\vec{X} + \vec{C}^*) + \vec{E}) \qquad (6.39)$$

This can be expressed as:

$$\vec{U}opt = K_1\vec{X} + \vec{K}_2 \qquad (6.40)$$

Where $K_1 = -(B^{*T}D^TQ\, D\, B^* + R)^{-1}B^{*T}D^TQ\, D\, A^*$ (6.41)

$$\vec{K}_2 = -(B^{*T}D^TQ\, D\, B^* + R)^{-1}B^{*T}D^TQ\, (D\vec{C}^* + \vec{E}) \qquad (6.42)$$

Equations (6.40) through (6.42) immediately suggest an implementation like that depicted in Figure 6.2. Here the state vector is multiplied by gain matrix K_1 to produce an intermediate control signal. The intermediate control is corrected by $\vec{K}_2$ before driving the actuators. K_1 and $\vec{K}_2$ are functions of the state, so there are two feedback loops operating here.

Alternatively equation (6.40) can be written as:

$$\vec{U}opt = K_3(\vec{K}_4 - \vec{X}) \tag{6.43}$$

Where $K_3 = (B^{*T}D^TQ\ D\ B^* + R)^{-1}B^{*T}D^TQ\ D\ A^*$ (6.44)

$$\vec{K}_4 = -A^{*-1}(D^T(D\ D^T)^{-1}\vec{E} + \vec{C}^*) \tag{6.45}$$

These equations, although representing the same system as (6.40) through (6.42) suggest a different implementation shown in Figure 6.3. The controller should behave the same way regardless of which implementation is chosen. It is obvious that a great deal of effort is required to compute $K_1,\vec{K}_2$ or $K_3,\vec{K}_4$ since they are complicated functions of complicated functions. Their calculation poses an immense computational burden. Some way should be found to reduce the amount of work necessary.

One method is to update $K_1,\vec{K}_2$ or $K_3,\vec{K}_4$ less often. The relatively simple calculation of the control vector could be performed very frequently whereas it might be possible to update the gain matrix at a lower rate without sacrificing either performance or stability. Such an analysis has yet to be undertaken.

Another strategy for coping with the complexity of computation would be to simplify the torque equations by ignoring high order effects. This would hopefully not degrade performance, but might enable more frequent calculation of the gain matrix and offset vector. Carried to an extreme one could ignore everything in the Y's except for torque, and approximate L and M by constant matrices. Simplifications will get propagated through A, B, A^*, B^*, $\vec{C}^*$ etc. leading to more tractable formulas for the K's. In practice, some combination of both strategies may be most feasible.

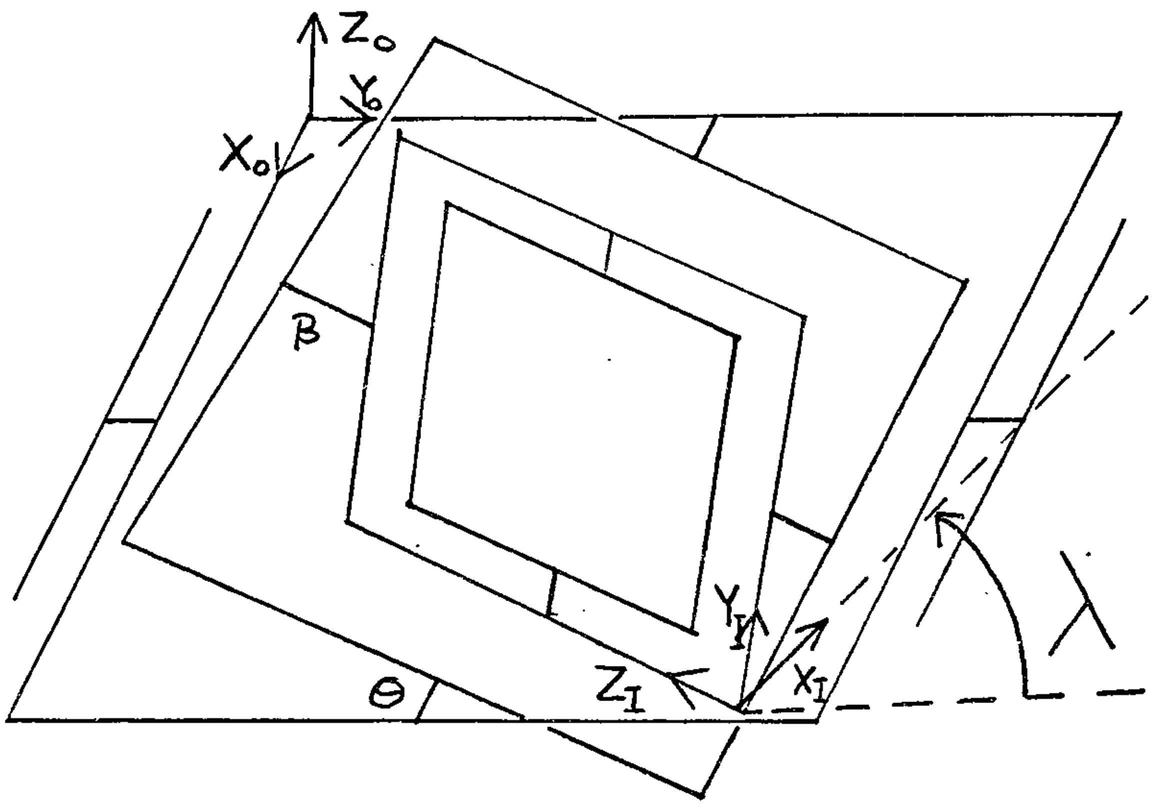


Figure 6.1

Gimbal Lock Angle

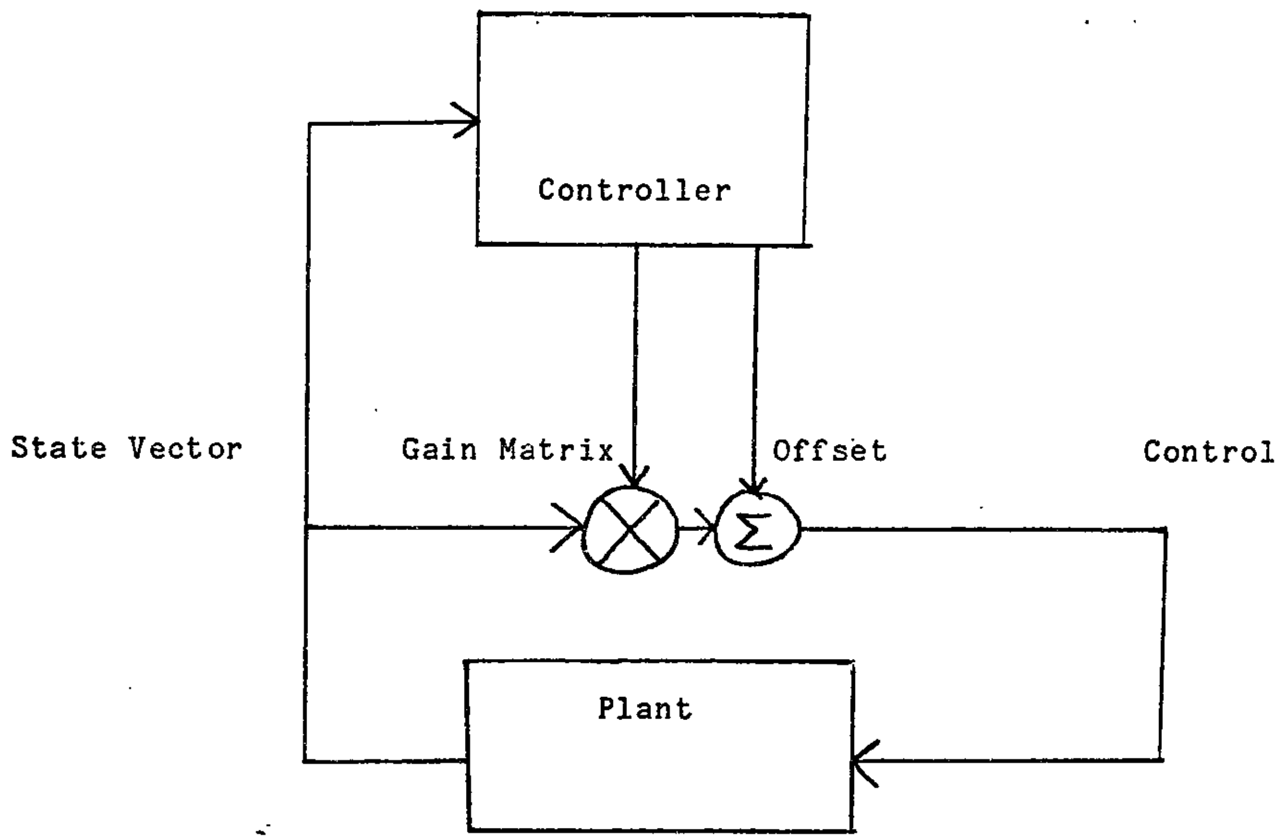


Figure 6.2

Proposed Controller Configuration 1

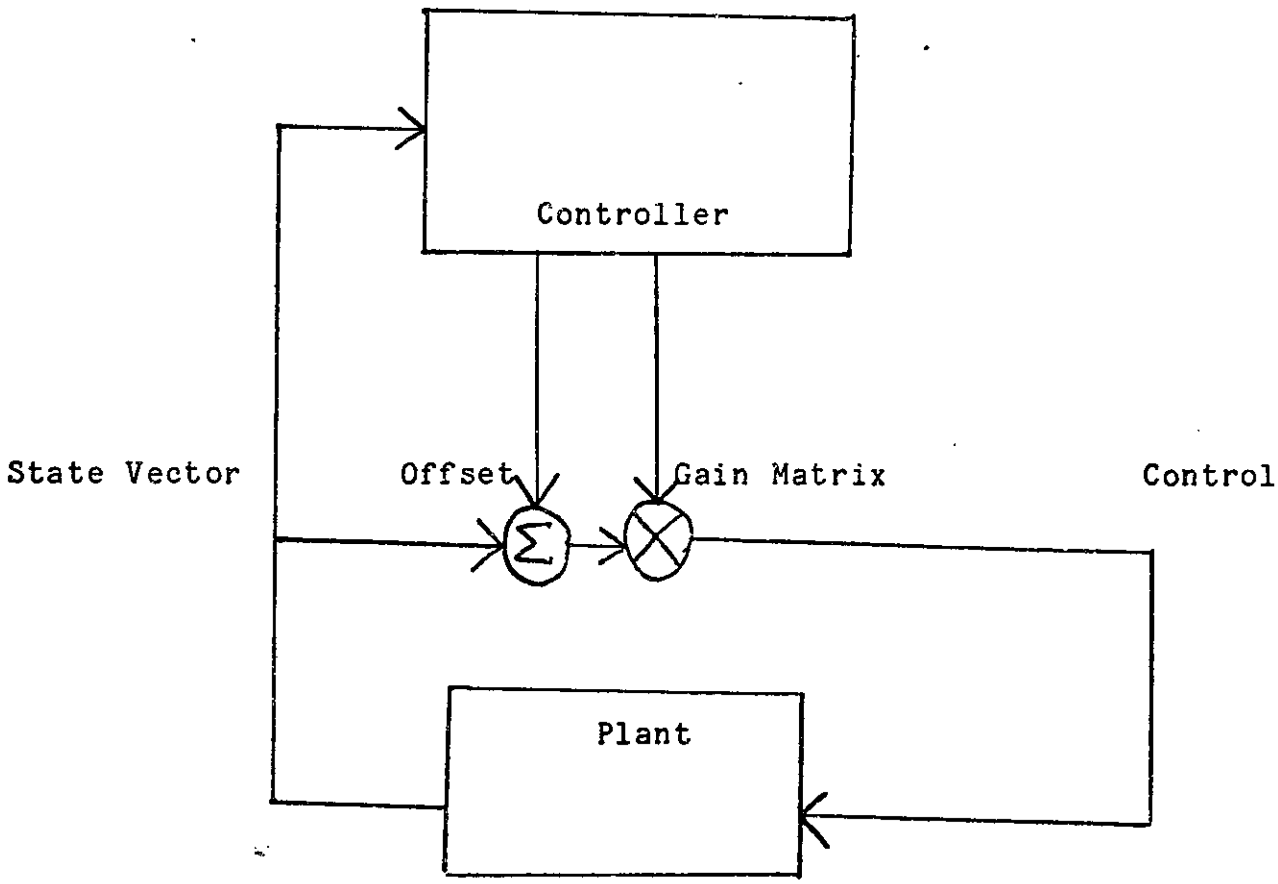


Figure 6.3

Proposed Controller Configuration 2

VII. Results

Embedded in the R and Q matrices of the previous section are four parameters called TORQWT, LOCKWT, TILTWT and RATEWT. They are the weights assigned to torque motor control signals, gimbal lock proximity, inertial platform tilt and inertial platform rate respectively in the cost function. These weights were not assigned in any specific fashion. Rather, a trial and error approach was taken to get results that look good. The simulation was run with various values for the weights and performance was judged on the basis of low control voltage, gimbal lock avoidance, small inertial platform tilts and rates, and stability of the controller. The four parameters were tweaked until the controller exhibited the desired behavior. It may be possible to further improve performance by further refining the weights but it is not clear that any significant amelioration will result. In any event, the cost function weights were not chosen in any formal way.

Before examining the performance of the optimal gimbal controller let us see what it replaces. The currently implemented controller uses a zone control scheme. In this scheme the B-Θ plane is divided into 16 regions (Figure 7.1). Torque motor control signals are generated based on the current zone. The idea is to steer clear of gimbal lock by staying within the numbered zones and avoiding those that include the gimbal lock condition. This is done by driving the elevation and inner gimbals from two of the gyros, and using the third gyro to control either the middle or outer gimbal depending on the zone. The remaining redundant gimbal is used to assist in some sensible fashion. Essentially it is a three-gimbal controller modified for an extra gimbal. Additionally, two of the physical gyros are replaced in the controller by "computed" gyros. The computed gyros, ΔR and ΔV, lie in the same plane as ΔSR and ΔSV. However, they point in the same direction as X_I and Z_I respectively. Equation (3.4) can be used to show that:

$$\Delta R = \Delta SR \cos I - \Delta SV \sin I \qquad (7.1)$$

$$\Delta V = \Delta SR \sin I + \Delta SV \cos I \qquad (7.2)$$

The zone control works fairly well until a zone switch is necessary. When a zone switch occurs, large transient effects arise. Large torque levels may be required to keep the platform inertial. Inertial platform misorientations are greatest immediately following zone changes. The decision rules are:

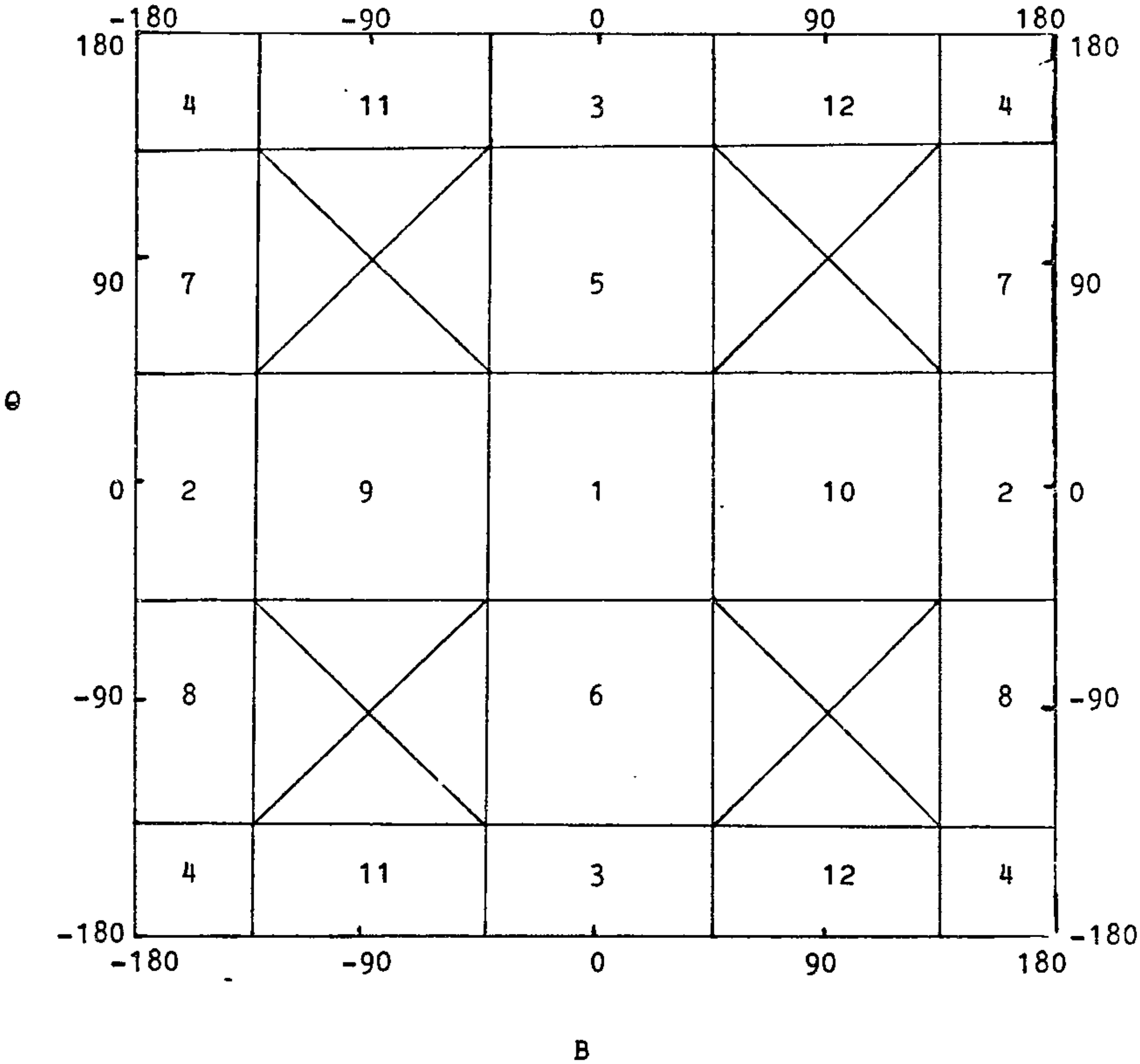
Zones 1-4 ΔR drives the middle gimbal
(Iactual - Icommanded) drives the outer gimbal

Zones 5-8 ΔR drives the middle gimbal
sinB drives the outer gimbal

Zones 9-12 ΔR drives the outer gimbal
sinθ drives the middle gimbal

All zones ΔJ drives the elevation gimbal
ΔV drives the inner gimbal

The next several pages compare the optimal controller with the zone control over a variety of orientations and case rates. In all examples the optimal controller exhibits much smaller gyro errors, plus lower RMS and peak torques while avoiding gimbal lock at least as well as the zone control. It wouldn't be optimal otherwise! Much of the apparent advantage of the optimal controller stems from the elimination of zone switch transients. Examples provided courtesy of H. M. Jones. For all examples the time between control updates is 5 milliseconds for the optimal controller, whereas the zone control is simulated as a continous system using a fourth order Runge-Kutta numerical integration technique with a time interval of 1 millisecond.



Zone Control Zones

Figure 7.1

Table 7.1

Optimal vs. Zone Control Run 1

Case Rates (deg/sec)		Initial Angles (deg)	
Roll	-30.0	Ø	0.0
Pitch	0.0	Θ	0.0
Yaw	-90.0	B	60.0
		Ɣ	0.0

Peak Torque (ft-lbs)	Optimal	Zone
Ø	0.328	0.601
Θ	0.203	0.460
B	0.172	0.160
Ɣ	0.223	0.121
RMS Torque (ft-lbs)		
Ø	0.111	0.218
Θ	0.112	0.135
B	0.100	0.100
Ɣ	0.090	0.090
Peak Gyro Errors (milliradians)		
ΔSR	0.03	0.51
ΔJ	0.26	0.42
ΔSV	0.06	0.38

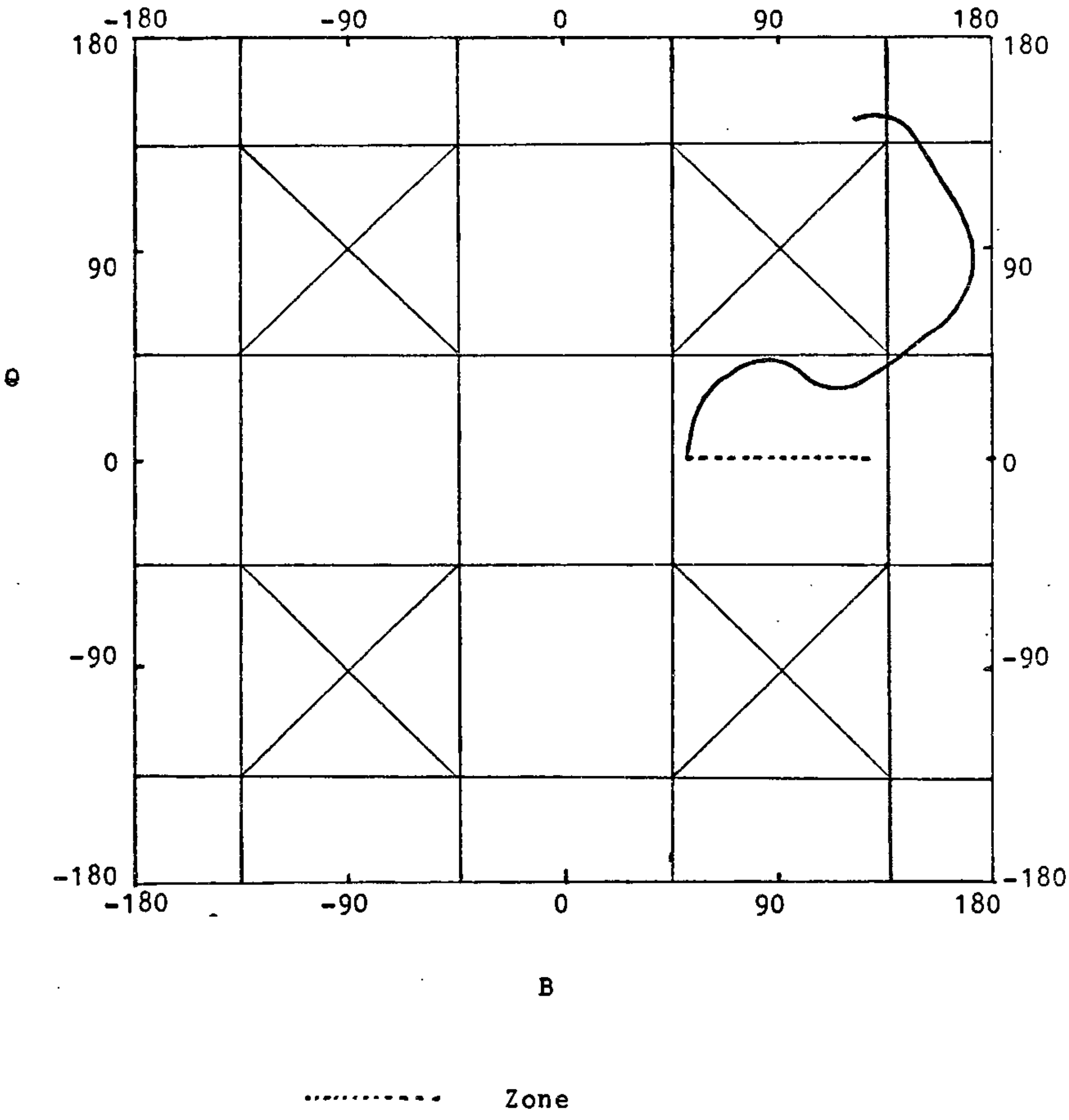


Figure 7.2

Optimal vs. Zone Control Trajectory 1

Table 7.2

Optimal vs. Zone Control Run 2

Case Rates (deg/sec)		Initial Angles (deg)	
Roll	0.0	Ø	0.0
Pitch	0.0	Θ	45.0
Yaw	-90.0	B	90.0
		Υ	0.0

Peak Torque (ft-lbs)	Optimal	Zone
Ø	0.0	0.0
Θ	0.258	0.763
B	0.0	0.0
Υ	0.196	0.117
RMS Torque (ft-lbs)		
Ø	0.0	0.0
Θ	0.119	0.413
B	0.0	0.0
Υ	0.090	0.090
Peak Gyro Errors (milliradians)		
ΔSR	0.0	0.0
ΔJ	0.098	0.436
ΔSV	0.0	0.0

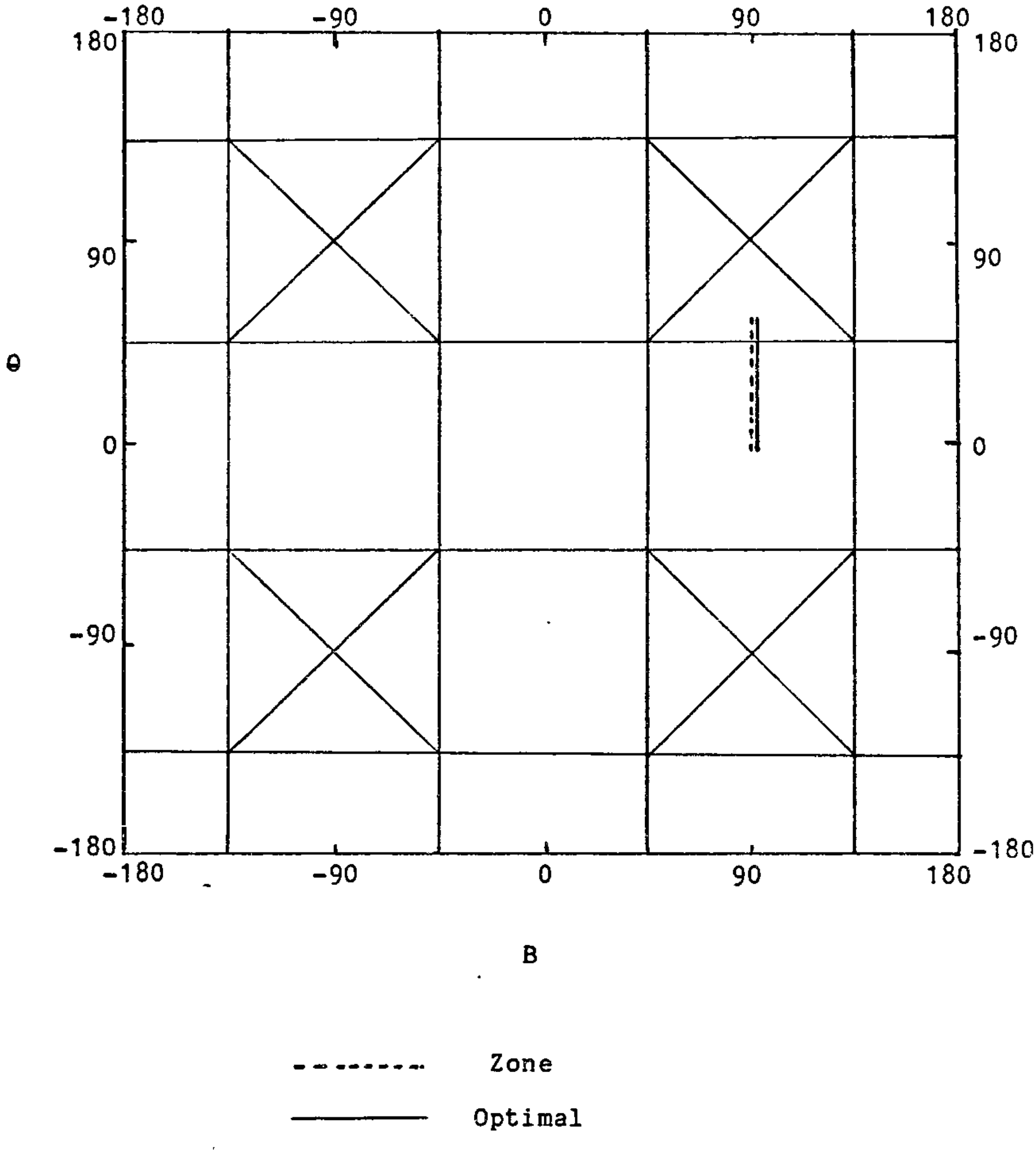


Figure 7.3

Optimal vs. Zone Control Trajectory 2

Table 7.3

Optimal vs. Zone Control Run 3

Case Rates (deg/sec)		Initial Angles (deg)	
Roll	0.0	Ø	135.0
Pitch	0.0	Θ	4.1
Yaw	90.0	B	41.2
		Υ	0.0

Peak Torque (ft-lbs)	Optimal	Zone
Ø	0.395	1.13
Θ	0.133	0.673
B	0.322	0.209
Υ	0.131	0.115
RMS Torque (ft-lbs)		
Ø	0.150	0.532
Θ	0.110	0.194
B	0.101	0.101
Υ	0.090	0.090
Peak Gyro Errors (milliradians)		
ΔSR	0.133	0.460
ΔJ	0.049	0.260
ΔSV	0.117	0.518

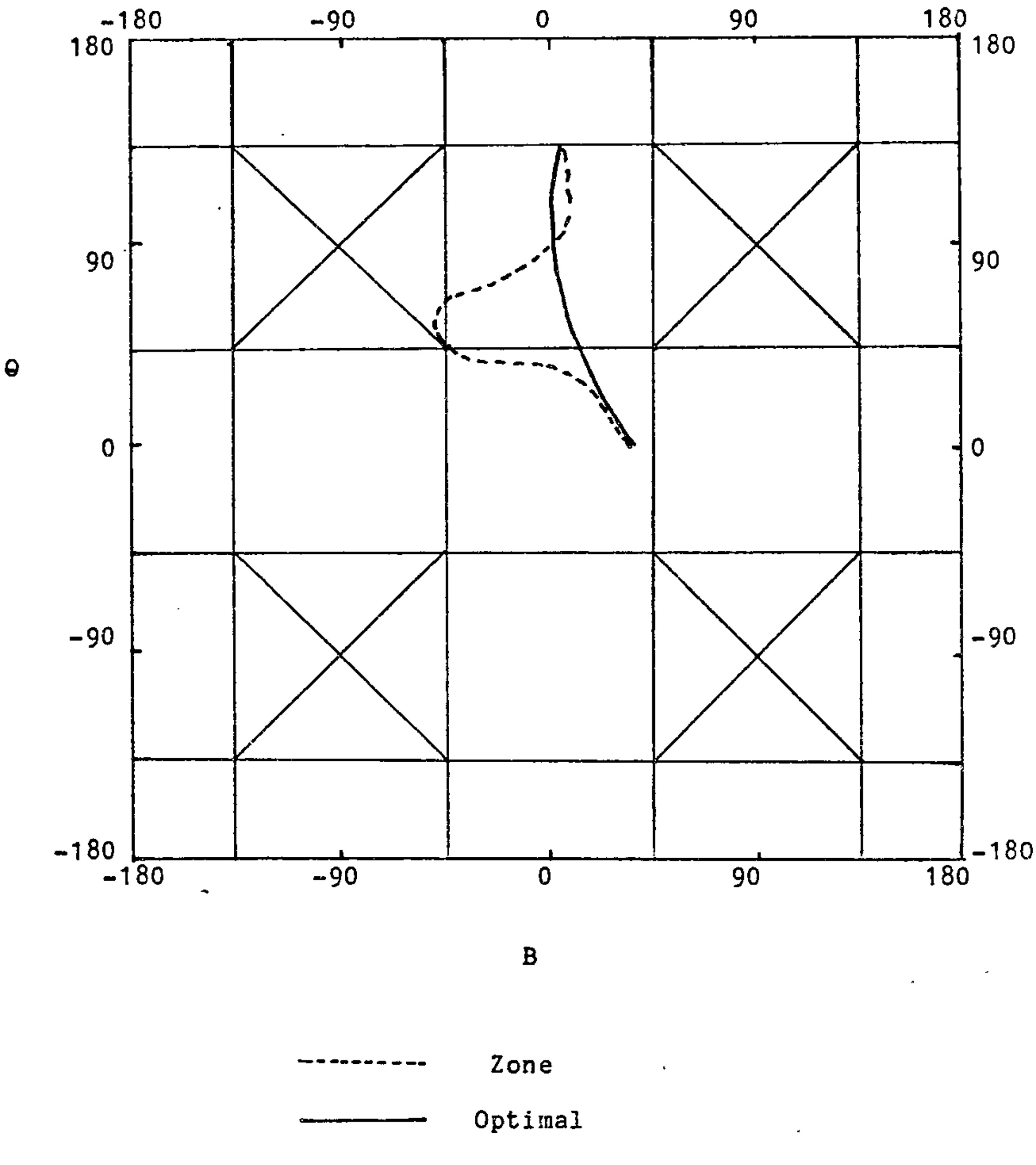


Figure 7.4

Optimal vs. Zone Control Trajectory 3

Table 7.4

Optimal vs. Zone Control Run 4

Case Rates (deg/sec)		Initial Angles (deg)	
Roll	-90.0	Ø	0.0
Pitch	0.0	Θ	0.0
Yaw	-90.0	B	0.0
		ϒ	0.0

Peak Torque (ft-lbs)	Optimal	Zone
Ø	0.75	1.63
Θ	0.36	1.23
B	0.22	0.138
ϒ	0.26	0.157
RMS Torque (ft-lbs)		
Ø	0.169	0.238
Θ	0.075	0.192
B	0.082	0.100
ϒ	0.015	0.085
Peak Gyro Errors (milliradians)		
ΔSR	0.05	1.06
ΔJ	0.18	0.84
ΔSV	0.05	0.08

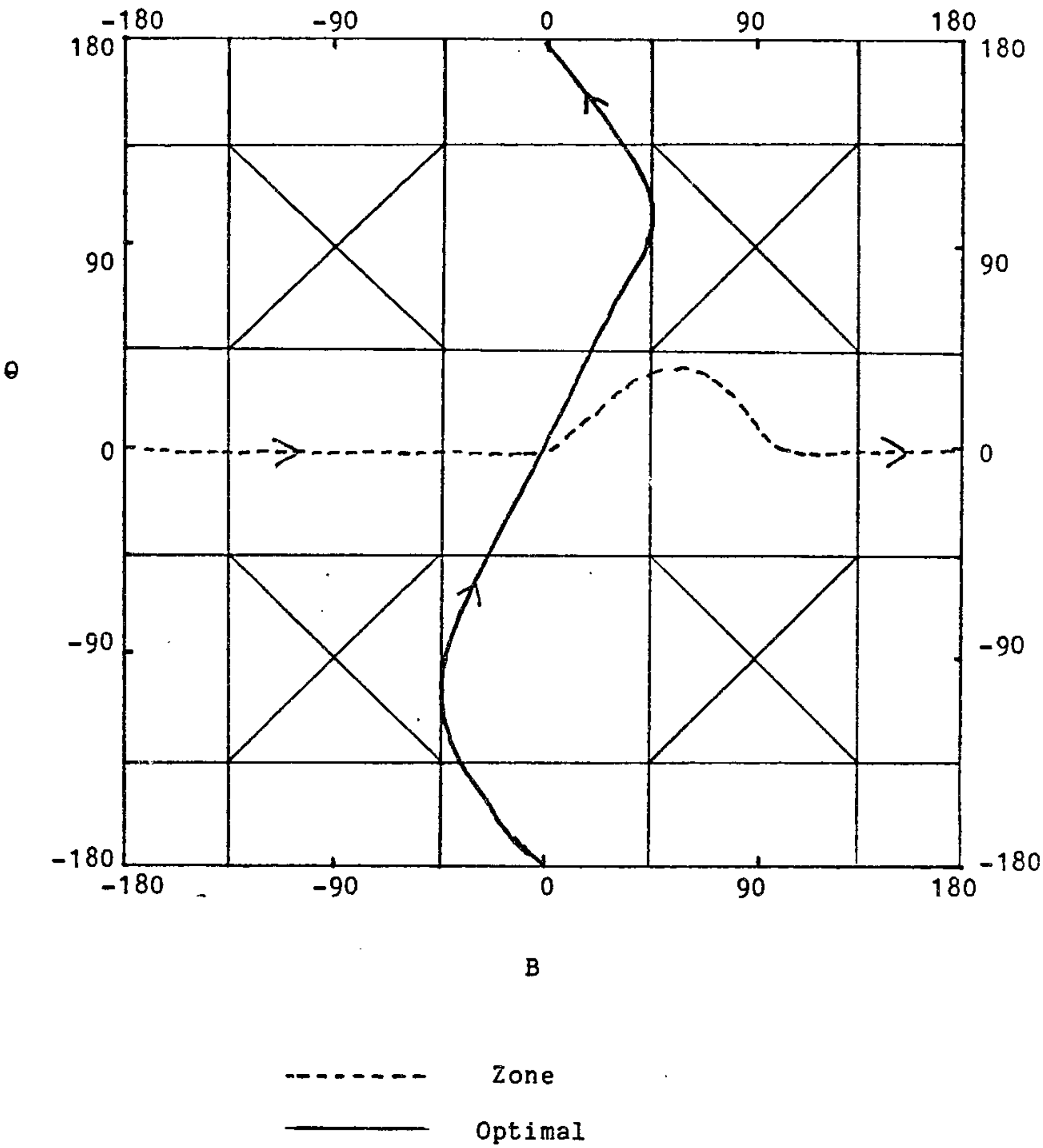


Figure 7.5

Optimal vs. Zone Control Trajectory 4

Table 7.5

Optimal vs. Zone Control Run 5

Case Rates (deg/sec)		Initial Angles (deg)	
Roll	0.0	Ø	180.0
Pitch	0.0	Θ	-105.0
Yaw	90.0	B	43.5
		Y	0.0

Peak Torque (ft-lbs)	Optimal	Zone
Ø	0.939	1.64
Θ	1.24	1.23
B	0.234	0.184
Y	0.415	0.177
RMS Torque (ft-lbs)		
Ø	0.276	0.503
Θ	0.174	0.173
B	0.100	0.099
Y	0.093	0.091
Peak Gyro Errors (milliradians)		
ΔSR	0.029	0.411
ΔJ	0.305	0.902
ΔSV	0.086	1.440

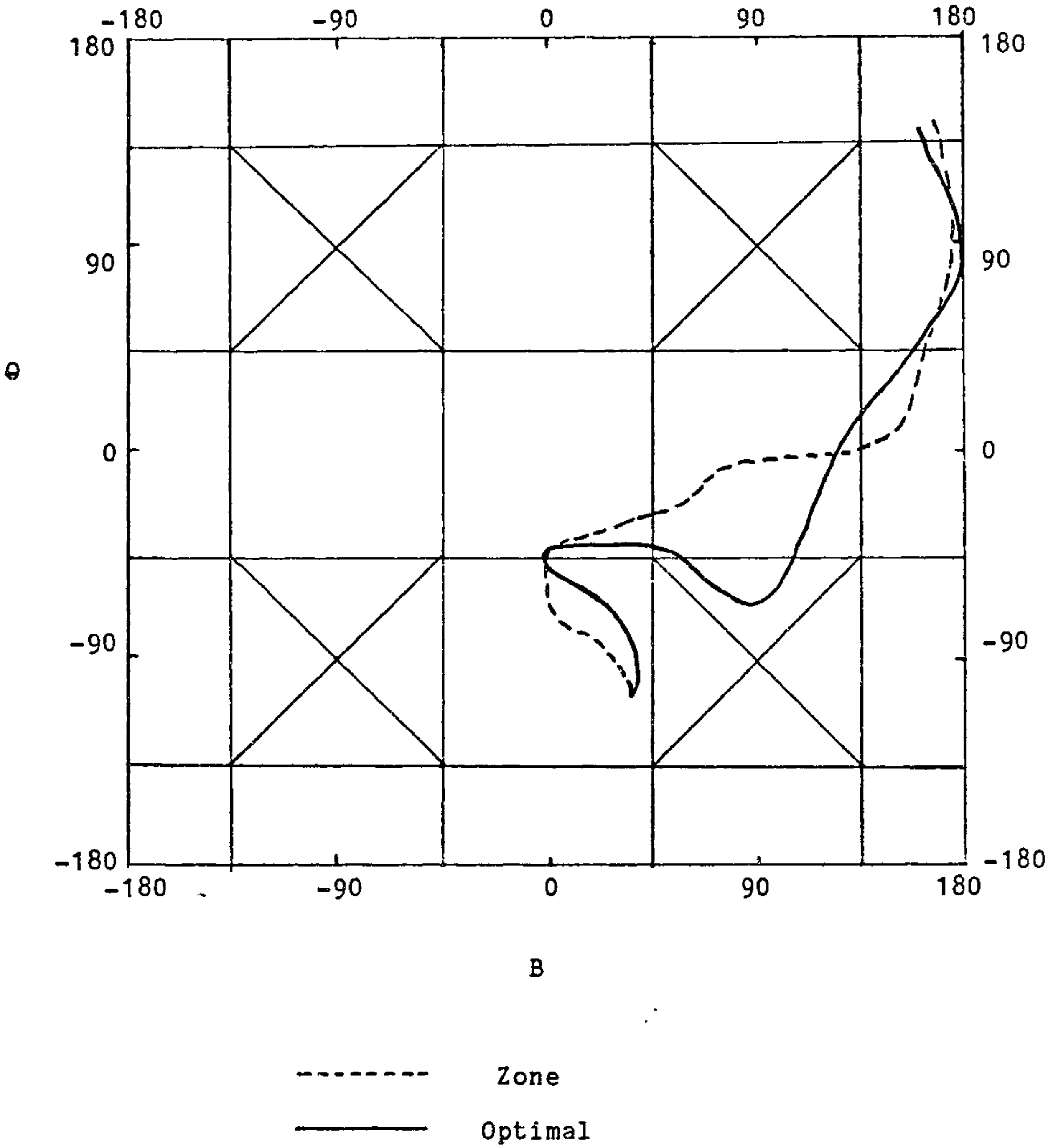


Figure 7.6

Optimal vs. Zone Control Trajectory 5

Table 7.6

Optimal vs. Zone Control Run 6

Case Rates (deg/sec)		Initial Angles (deg)	
Roll	0.0	Ø	135.0
Pitch	0.0	Θ	0.0
Yaw	90.0	B	-45.0
		Ɪ	0.0

Peak Torque (ft-lbs)	Optimal	Zone
Ø	0.637	1.29
Θ	0.230	1.23
B	0.301	0.138
Ɪ	0.183	0.111
RMS Torque (ft-lbs)		
Ø	0.252	0.390
Θ	0.110	0.300
B	0.099	0.087
Ɪ	0.089	0.081
Peak Gyro Errors (milliradians)		
ΔSR	0.029	0.887
ΔJ	0.139	0.236
ΔSV	0.117	0.232

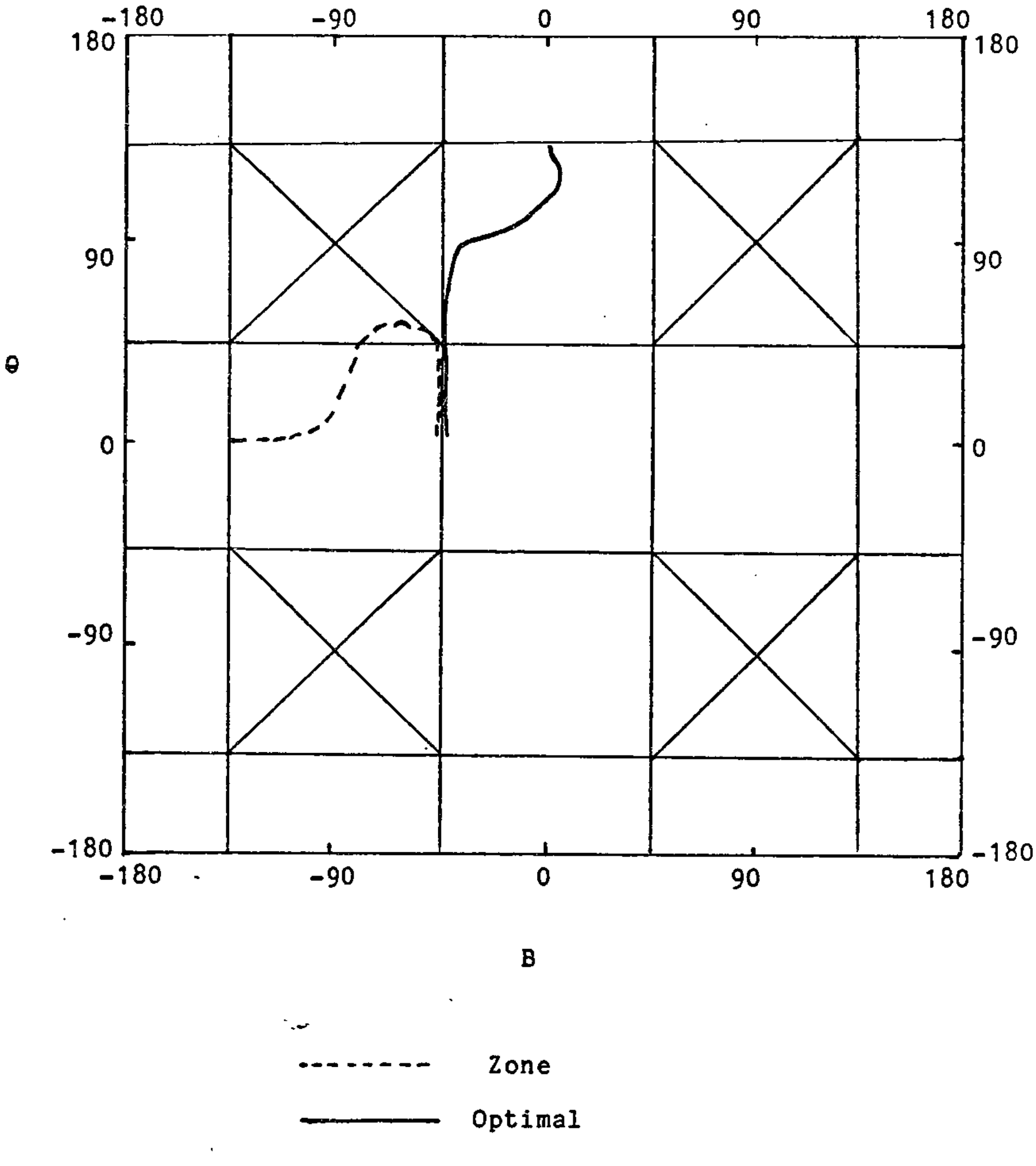


Figure 7.7

Optimal vs. Zone Control Trajectory 6

Table 7.7

Optimal vs. Zone Control Run 7

Case Rates (deg/sec)		Initial Angles (deg)	
Roll	0.0	Ø	180.0
Pitch	0.0	Θ	0.0
Yaw	90.0	B	-45.0
		Υ	0.0

Peak Torque (ft-lbs)	Optimal	Zone
Ø	0.637	1.10
Θ	0.230	0.401
B	0.301	0.158
Υ	0.183	0.149
RMS Torque (ft-lbs)		
Ø	0.252	0.430
Θ	0.110	0.123
B	0.099	0.085
Υ	0.089	0.081
Peak Gyro Errors (milliradians)		
ΔSR	0.029	0.278
ΔJ	0.139	0.846
ΔSV	0.117	0.312

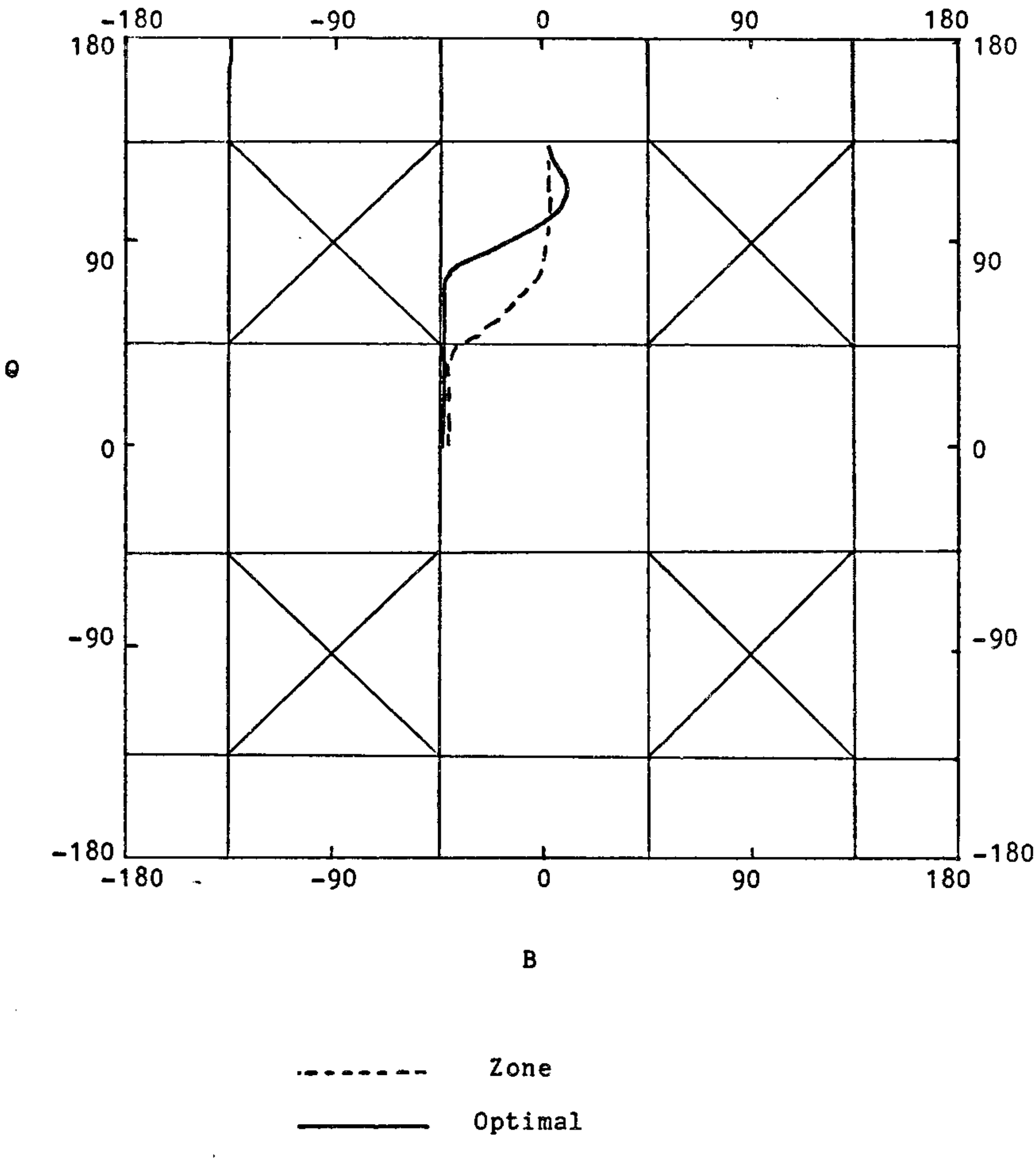


Figure 7.8

Optimal vs. Zone Control Trajectory 7

VIII. Conclusions

Modern optimal control provides a useful framework in which to analyze and improve the performance of feedback systems. Many untapped applications exist for this powerful theory. Unfortunately, it is not always used to advantage. This thesis has attempted to relieve this situation for one particular system. Simulation studies indicate great success. The optimal controller for a four-gimbal system potentially far outperforms an earlier nonoptimal controller.

This improvement in performance does not come free. A significant computational burden is imposed by optimization. Some techniques for reducing the load have been suggested. Work remains to be done actually implementing the proposed controller. Final judgement on its feasibility awaits.

There is no reason to be content even with an optimal controller. Under different optimality criteria it is conceivable that a controller could be designed with more desirable operating characteristics. A bang-bang controller is one worth considering. By applying full torque in short pulses it may be possible to further reduce platform tilts.

Leaving such speculation aside, the fact remains that with a suitable model developed, optimal control can be applied to components of inertial guidance equipment. One can only hope that deployment precludes actual use.

Appendix A. Coordinate Transformations and Notation

The notation used here is based on work by Britting[3]. This notation is helpful for representing orientations, rotations and coordinate transformations. The reference frame of a vector is indicated by a superscript. $\vec{r}^j$ is a vector coordinatized in the j reference frame. Any vector in the j frame can be expressed in the k frame by premultiplying the original vector by a coordinate transformation C_j^k. The subscript indicates the original reference frame and the superscript denotes the new reference frame. Thus, for the example given:

$$\vec{r}^k = C_j^k \vec{r}^j \tag{A.1}$$

Note that the original superscript has been canceled by the subscript of C_j^k. For Cartesian coordinate systems, in which the basis vectors are orthonormal, the entries of a coordinate transformation matrix are direction cosines. Direction Cosine Matrix (DCM) is a term often used to describe such a matrix. The direction cosine from the m-axis of reference frame j to the n-axis of frame k is the mn<u>th</u> entry of C_j^k. DCM's exhibit many interesting properties. Some follow:

$$C_k^l C_j^k = C_j^l \quad \text{but} \quad C_j^k C_k^l \neq C_j^l \tag{A.2}$$

$$C_j^j = I \tag{A.3}$$

$$C_j^k = (C_k^j)^{-1} \tag{A.4}$$

$$C_j^k = (C_k^j)^T \tag{A.5}$$

Rotations satisfy the same superscript convention as other vectors. In addition rotation vectors have two subscripts. The sense of rotation is from the left subscript to the right subscript. To be precise, coordinate systems rotate, not subscripts. $\vec{W}^{l}_{kj}$ would be the rotation rate of system j with respect to system k as seen from the l reference frame. Rotations add vectorially. When they do, subscripts cancel.

$$\vec{W}^{l}_{ki} = \vec{W}^{l}_{kj} + \vec{W}^{l}_{ji} \tag{A.6}$$

It follows that:

$$\vec{W}^{l}_{kj} = -\vec{W}^{l}_{jk} \tag{A.7}$$

The superscripts must be the same for these relations to hold. Differention of vectors is no longer simple in rotating reference frames. For any vector $\vec{r}^{i}$ we have the following equivalent expressions:

$$\begin{aligned} \dot{\vec{r}}^{i} &= C^{i}_{j}\dot{\vec{r}}^{j} - C^{i}_{j}(\vec{W}^{j}_{ji} \times \vec{r}^{j}) \\ &= C^{i}_{j}\dot{\vec{r}}^{j} - C^{i}_{j}\vec{W}^{j}_{ji} \times C^{i}_{j}\vec{r}^{j} \\ &= C^{i}_{j}\dot{\vec{r}}^{j} - \vec{W}^{i}_{ji} \times \vec{r}^{i} \\ &= C^{i}_{j}\dot{\vec{r}}^{j} + \vec{W}^{i}_{ij} \times \vec{r}^{i} \end{aligned} \tag{A.8}$$

Appendix B. Summary of Computer Routines Used

Main Program

1. Calls INITLZ routine
2. Calls DERIVE routine
3. Calls OUTPUT routine
4. Calls UPDATE routine
5. Loops to 2.

INITLZ (X, DXDT, W, OLDRATE, TOTALT, DELTAT)

1. Clears out storage areas
2. Initializes state, case rates and other parameters

OUTPUT (X, DXDT, U, W, I, TIME)

1. Prints output 1 out of J invocations else returns
2. Prints state, derivative, control and case rate vectors

UPDATE (X, DXDT, U, W, OLDRATE, DELTAT)

1. Updates state via 4th order Runge-Kutta Integration
2. Calls DERIVE during computation

DERIVE (X, DXDT, U, W, OLDRATE)

1. Computes friction as described in section V.
2. Derives torque-acceleration equations as per section VI.
3. Solves for angular accelerations using SIMQ
4. Returns state derivative in DXDT

SIMQ (A, B, N, KS)

1. Solves system of equations of form AX=B
2. Returns solution in B

MINV (A, N, D, L, M)

1. Inverts a matrix
2. Returns result in A

MATMPY (A, B, C)

1. Computes C=AB

CONTRL (X, U, W, OLDRATE, DELTAT)

1. Computes linear discrete-time equations as in section VII
2. Calls MINV and MATMPY to perform matrix manipulations
3. Returns control in U

Appendix C. Computer Simulation of the Four-Gimbal System

```
C      SIMUL4G -- 4 GIMBAL SYSTEM SIMULATION
C      MICHAEL A. GENNERT
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
       PARAMETER  IXX=11,IDR=7
       DIMENSION X(11), DXDT(11), U(4), W(3), OLDRATE (4)
C
C      DELTAT = TIME INCREMENT SIZE
C      TOTALT = TOTAL SIMULATION TIME
C      X      = STATE VECTOR
C      DXDT   = TIME RATE OF CHANGE OF STATE VECTOR
C      U      = CONTROL VECTOR
C      W      = CASE RATE VECTOR
C      OLDRATE= VECTOR CONTAINING PREVIOUS VALUES OF GIMBAL ANGLE RATES
C               USED TO DETERMINE FRICTION EFFECTS IN DERIVE ROUTINE
C
C      X(1)   = PSI      (E)
C      X(2)   = BETA     (I)
C      X(3)   = THETA    (M)
C      X(4)   = PHI      (O)
C      X(5)   = PSIDOT   (dE/dt)
C      X(6)   = BETADOT  (dI/dt)
C      X(7)   = THETADOT (dM/dt)
C      X(8)   = PHIDOT   (dO/dt)
C      X(9)   = DELTASR
C      X(10)  = DELTAJ
C      X(11)  = DELTASV
C
C      U(1)   = CONTROL ON E GIMBAL
C      U(2)   = CONTROL ON I GIMGAL
C      U(3)   = CONTROL ON M GIMBAL
C      U(4)   = CONTROL ON O GIMBAL
C
C      W(1)   = X AXIS CASE RATE (WCX)
C      W(2)   = Y AXIS CASE RATE (WCY)
C      W(3)   = Z AXIS CASE RATE (WCZ)
C
C
C      INITIALIZE
C
       CALL INITLZ (X, DXDT, W, OLDRATE, TOTALT, DELTAT)
       N = TOTALT/DELTAT
       DO 100 I = 1, N+2
C
C      DETERMINE CONTROL SIGNAL TO BE APPLIED
C
       CALL CONTRL (X, U, W, OLDRATE, DELTAT)
C
C      COMPUTE STATE DERIVATIVE VECTOR PRIOR TO OUTPUT
C
       CALL DERIVE (X, DXDT, U, W, OLDRATE)
C
C      PRINT STATE AND CONTROL INFORMATION
C
       II = I-1
       CALL OUTPUT (X, DXDT, U, W, II,  DBLE(FLOAT(II))*DELTAT)
C
C      UPDATE STATE EQUATIONS AND INTEGRATE
C
       CALL UPDATE (X, DXDT, U, W, OLDRATE, DELTAT)
   100 CONTINUE
       STOP
       END
C
C      INITIALIZE SUBROUTINE
C
       SUBROUTINE INITLZ (X, DXDT, W, OLDRATE, TOTALT, DELTAT)
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
       PARAMETER  IXX=11
```

```
      DIMENSION X (1), DXDT (1), W (1), OLDRATE (1)
      XK = 3.14159265358979323846D0/180.D0
C
C     CLEAR STATE VECTOR
C
      DO 100 I = 1, IXX
      X(I)     = 0.D0
  100 CONTINUE
C
C     E, I, M, O FOLLOW IN DEGREES
C
      X (1) = 0.D0*XK
      X(2)=60.D0*XK
      X(3)=0.D0*XK
      X (4) = 0.D0*XK
C
C     dE/dt, dI/dt, dM/dt, dO/dt FOLLOW IN DEGREES/SECOND
C
      X (5) = 0.D0*XK
      X(6)=30.D0*XK
      X(7)=90.D0*XK
      X(8) = 0.D0*XX
C
C     SET OLDRATE TO ANGLEDOT FOR FRICTION COMPUTATION
C
      DO 110 I = 1, 4
      OLDRATE (I) = X (I + 4)
  110 CONTINUE
C
C     CALCULATE CASE RATES IN RADIANS/SECOND
C
      SB = DSIN (X (2))
      CB = DCOS (X (2))
      ST = DSIN (X (3))
      CT = DCOS (X (3))
      SF = DSIN (X (4))
      CF = DCOS (X (4))
      WIY = -X (5)
      WMX = -SB*WIY
      WMY = CB*WIY
      WMZ = -X (6)
      WOX = WMX-X (7)
      WOY = CT*WMY-ST*WMZ
      WOZ = ST*WMY+CT*WMZ
      W (1) = CF*WOX+SF*WOZ
      W (2) = WOY-X (8)
      W (3) = -SF*WOX+CF*WOZ
C
C     SET UP TIME PARAMETERS IN SECONDS
C
      TOTALT = 1.D0
      DELTAT=1.D0/3000.D0
      RETURN
      END
C
C     OUTPUT SUBROUTINE
C
      SUBROUTINE OUTPUT (X, DXDT, U, W, I, TIME)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION X (1), DXDT (1), U (1), W(1)
      J=30
      IPRINT = 6
      XK = 3.14159265358979323846D0/180.D0
C
C     PRINT EVERY Jth TIME, RETURN THE OTHER J-1 OCCURANCES
C
      IF (I.NE.(I/J)*J) RETURN
      WRITE (IPRINT,900) TIME
```

```
 900 FORMAT (//" TIME =", F8.3, " SECONDS")
     WRITE (IPRINT,901)
 901 FORMAT (18X, "DEG", 11X, "DEG/SEC", 7X, "DEG/SEC/SEC", 7X, "CONTROL")
     WRITE (IPRINT,902) "E", X (1)/XK, X (5)/XK, DXDT (5)/XK, U (1)
 902 FORMAT (6X, 1A1, 4(8X, F8.3))
     WRITE (IPRINT,902) "I", X (2)/XK, X (6)/XK, DXDT (6)/XK, U (2)
     WRITE (IPRINT,902) "M", X (3)/XK, X (7)/XK, DXDT (7)/XK, U (3)
     WRITE (IPRINT,902) "O", X (4)/XK, X (8)/XK, DXDT (8)/XK, U (4)
     WRITE (IPRINT,903)
 903 FORMAT (18X, "MRAD", 9X, "MRAD/SEC", 25X, "DEG/SEC")
     WRITE (IPRINT,904) "SR", X (9)*1E3, DXDT (9)*1E3, "WCX", W (1)/XK
 904 FORMAT (6X, "DELTA", A2, F10.3, 6X, F10.3, 16X, A3, 5X, F8.3)
     WRITE (IPRINT,904) "J ", X (10)*1E3, DXDT (10)*1E3, "WCY", W (2)/XK
     WRITE (IPRINT,904) "SV", X (11)*1E3, DXDT (11)*1E3, "WCZ", W (3)/XK
     RETURN
     END
C
C    UPDATE SUBROUTINE
C    CALLS DERIVE WHICH COMPUTES DERIVATIVE, THEN
C    EMPLOYS RUNGE-KUTTA 4th ORDER INTEGRATION
C
     SUBROUTINE UPDATE (X, DXDT, U, W, OLDRATE, DELTAT)
     IMPLICIT DOUBLE PRECISION(A-H,O-Z)
     PARAMETER IXX=11
     DIMENSION X (1), DXDT (1), U (1), W (1), OLDRATE (1)
     DIMENSION Q (IXX), XSTOR (IXX)
C
C    STORE STATE VECTOR IN XSTOR
C
     DO 100 I = 1, IXX
     XSTOR (I) = X (I)
 100 CONTINUE
C
C    COMPUTE DERIVATIVE AND MAKE 1st APPROXIMATION
C
     CALL DERIVE (XSTOR, DXDT, U, W, OLDRATE )
     DO 110 I = 1, IXX
     Q (I) = DXDT (I)
     XSTOR (I) = X (I) + .5D0 * DELTAT * DXDT (I)
 110 CONTINUE
C
C    2nd APPROXIMATION
C
     CALL DERIVE (XSTOR, DXDT, U, W, OLDRATE)
     DO 120 I = 1, IXX
     Q(I) = Q (I) + 2.D0 * DXDT (I)
     XSTOR (I) = X (I) + .5D0 * DELTAT * DXDT (I)
 120 CONTINUE
C
C    3rd APPROXIMATION
C
     CALL DERIVE (XSTOR, DXDT, U, W, OLDRATE)
     DO 130 I = 1, IXX
     Q (I) = Q (I) + 2.D0 * DXDT (I)
     XSTOR (I) = X (I) + DELTAT * DXDT (I)
 130 CONTINUE
C
C    FINAL APPROXIMATION
C
     CALL DERIVE (XSTOR, DXDT, U, W, OLDRATE)
C
C    STORE OLD VALUES OF ANGLE RATES FOR FRICTION COMPUTATION
C
     DO 140 I = 1, 4
     OLDRATE (I) = X (I+4)
 140 CONTINUE
     DO 150 I = 1, IXX
     DXDT (I) = (Q (I) + DXDT (I)) / 6.D0
```

```
      X (I) = X (I) + DELTAT * DXDT (I)
  150 CONTINUE
      RETURN
      END
C
C     DERIVATIVE SUBROUTINE
C     COMPUTES DXDT GIVEN X, U AND W
C
      SUBROUTINE DERIVE (X, DXDT, U, W, OLDRATE)
      IMPLICIT DOUBLE PRECISION(A-Z)
      LOGICAL STUCK
      INTEGER I,II,FLAG,KS
      DIMENSION X (1), DXDT (1), U (1), W (1), OLDRATE (1)
      DIMENSION STUCK (4), TORQUE (4), KTR (4), KV (4), FSTATC (4)
      DIMENSION FCOULM (4)
      DIMENSION L (4,4), Y (4)
      DATA JE, JIX, JIYZ, JMXZ, JMY, JOXY, JOZ
     &    /1.2D-2, 1.7D-2, 1.3D-2, 2.25D-2, 3.0D-2, 3.0D-2, 3.9D-2/
      DATA FSTATC /.09D0,.1D0,.11D0,.165D0 /
      DATA FCOULM /.09D0,.1D0,.11D0,.165D0 /
      DATA KTR /1.9D-2,1.9D-2,4.1D-2,5.13D-2/
      DATA KV /6.1D-1,6.1D-1,7.9D-1,1.25D0/
C     THIS ROUTINE SOLVES THE FOLLOWING MATRIX EQUATON. Y1 IS A FUNCTION
C     ONLY OF GIMBAL ANGLES, GIMBAL RATES, AND TIEY. Y2 IS A FUNCTION
C     ONLY OF GIMBAL ANGLES, GIMBAL RATES, AND TMIZ. Y3 IS A FUNCTION
C     ONLY OF GIMBAL ANGLES, GIMBAL RATES, AND TOMX. Y4 IS A FUNCTION
C     ONLY OF GIMBAL ANGLES, GIMBAL RATES, AND TCOY.
C
C     I Y1  I    I L11 L12 L13 L14 I I PSIDOUBLEDOT   I
C     I Y2  I    I L21 L22 L23 L24 I I BETADOUBLEDOT  I
C     I Y3  I =  I L31 L32 L33 L34 I I THETADOUBLEDOT I
C     I Y4  I    I L41 L42 L43 L44 I I PHIDOUBLEDOT   I
C
C     STUCK  = ONE FLAG FOR EACH GIMBAL. VALUE IS TRUE IF THE SPECIFIED
C              GIMBAL IS STUCK TO THE NEXT OUTER GIMBAL DUE TO STATIC
C              FRICTION. VALUE IS FALSE IF THE GIMBALS ARE NOT STUCK.
C     FSTATC = STATIC FRICTION TORQUE LIMIT. SPECIFIES THE STATIC FRICTION
C              LEVELS THAT MUST BE OVERCOME TO FREE A STUCK GIMBAL.
C     FCOULM = COULOMB FRICTION TORQUE LIMIT. SPECIFIES THE FRICTION
C              MAGNITUDE WHEN THE GIMBALS ARE UNSTUCK.
C     KTR    = CONVERSION CONSTANTS FROM TORQUE MOTOR VOLTAGES TO TORQUES
C     KV     = PROPORTIONALITY CONSTANTS FROM ANGLEDOTS TO BACK EMFS
C     JE     = INERTIA ABOUT ANY AXIS OF THE ELEVATION GIMBAL
C     JIX    = INERTIA ABOUT THE X AXIS OF THE INNER GIMBAL
C     JIYZ   = INERTIA ABOUT THE Y AND Z AXES OF THE INNER GIMBAL
C     JMXZ   = INERTIA ABOUT THE X AND Z AXES OF THE MIDDLE GIMBAL
C     JMY    = INERTIA ABOUT THE Y AXIS OF THE MIDDLE GIMBAL
C     JOXY   = INERTIA ABOUT THE X AND Y AXES OF THE OUTER GIMBAL
C     JOZ    = INERTIA ABOUT THE Z AXIS OF THE OUTER GIMBAL
C
C     ASSOCIATE VARIABLES WITH ARRAY ELEMENTS
C     COMPUTE REQUIRED TRIGONOMETRIC  FUNCTIONS
C
      PSI   = X (1)
      BETA  = X (2)
      THETA = X (3)
      PHI   = X (4)
      SP = DSIN (PSI)
      CP = DCOS (PSI)
      SB = DSIN (BETA)
      CB = DCOS (BETA)
      ST = DSIN (THETA)
      CT = DCOS (THETA)
      SF = DSIN (PHI)
      CF = DCOS (PHI)
      SB2 = SB * SB
      CB2 = CB * CB
      ST2 = ST * ST
```

```
      CT2 = CT * CT
C
C     DEFINE L11 THROUGH L44
C     FOR THE SIGNIFICANCE OF THESE QUANTIES REFER TO THE
C     GIMBAL TORQUE EQUATION DERIVATIONS
C
      L (1,1) = JE
      L (1,2) = 0.D0
      L (1,3) = -SB * JE
      L (1,4) = CB * CT * JE
      L (2,1) = 0.D0
      L (2,2) = JIYZ + JE
      L (2,3) = 0.D0
      L (2,4) = -ST * (JIYZ + JE)
      L (3,1) = -SB * JE
      L (3,2) = 0.D0
      L (3,3) = (JE + SB2 * JIYZ + CB2 * JIX + JMXZ)
      L (3,4) = SB * CB * CT * (JIX - JIYZ)
      L (4,1) = CB * CT * JE
      L (4,2) = -ST * (JIYZ + JE)
      L (4,3) = SB * CB * CT * (JIX - JIYZ)
      L (4,4) = JOXY + CT2 * JMY + ST2 * (JMXZ + JIYZ) + SB2 * CT2 * JIX
     &          + CB2 * CT2 * JIYZ + JE
C
C     ROUTINE TO CONVERT CONTROL SIGNALS TO TORQUES, INCLUDING FRICTION
C
      DO 120 I = 1, 4
      TORQUE (I) = KTR (I) * (U (I) - KV (I) * X (I+4))
C
C     IF THE MAGNITUDE OF THE APPLIED TORQUE DOES NOT EXCEED THE STATIC
C     FRICTION LIMIT AND THE GIMBAL RATE IS PASSING THROUGH ZERO (ie.
C     ANGLEDOT CHANGES SIGN) THEN THE GIMBALS WILL BE STUCK TOGETHER
C
      IF (ABS (TORQUE (I)).LE.FSTATC (I).AND.((X (I+4)*OLDRATE (I)).
     &   LT.0.D0.OR.X(I+4).EQ.0.D0)) GOTO 100
C
C     GIMBALS NOT STUCK TOGETHER -- CLEAR STUCK FLAG, SUBTRACT FRICTION
C
      STUCK (I) = .FALSE.
      TORQUE (I) = TORQUE (I)-SIGN (FCOULM (I), X (I+4))
      GOTO 120
C
C     GIMBALS STUCK TOGETHER -- SET STUCK FLAG, SET ANGLEDOT TO ZERO
C     SET Ith ROW AND Ith COLUMN OF [L] TO ZERO, SET L (I,I) = 1.
C
  100 STUCK (I) = .TRUE.
      X (I+4) = 0.D0
      DO 110 II = 1, 4
      L (I,II) = 0.D0
      L (II,I) = 0.D0
  110 CONTINUE
C
C     SET L (I,I) = 1. SO AS NOT TO HAVE A SINGULAR MATRIX
C
      L(I,I)=1.D0
  120 CONTINUE
C
C     DEFINE GIMBAL RATES
C
      PSIDOT   = X (5)
      BETADOT  = X (6)
      THETADOT = X (7)
      PHIDOT   = X (8)
      WCX = W (1)
      WCY = W (2)
      WCZ = W (3)
      WOX = CF * WCX - SF * WCZ
      WOY = WCY + PHIDOT
```

```
      WOZ = SF * WCX + CF * WCZ
      WMX = WOX + THETADOT
      WMY = CT * WOY + ST * WOZ
      WMZ =-ST * WOY + CT * WOZ
      WIX = CB * WMX + SB * WMY
      WIY =-SB * WMX + CB * WMY
      WIZ = WMZ + BETADOT
      WEX = CP * WIX - SP * WIZ
      WEY = WIY + PSIDOT
      WEZ = SP * WIX + CP * WIZ
C
C     DEFINE TORQUES
C
      TIEY = TORQUE (1)
      TMIZ = TORQUE (2)
      TOMX = TORQUE (3)
      TCOY = TORQUE (4)
C
C     TORQUE EQUATIONS FOR THE FOUR GIMBALS
C
      Y (1) = TIEY + JE * (-SB * PHIDOT * WOZ - CB * ST * PHIDOT * WOX
     &        - CB * THETADOT * WMZ + BETADOT * WIX)
      Y (2) = TMIZ + (JIYZ + JE) * (THETADOT * WMY - CT * PHIDOT * WOX)
     &        + WIX * WIY * (JIX-JIYZ) - PSIDOT * JE * WIX
      Y (3) = TOMX + (JMXZ + CB2 * JIX + SB2 * JIYZ + JE) * PHIDOT * WOZ
     &        - SB * CB * (JIX-JIYZ ) * (ST * PHIDOT * WOX + THETADOT
     &        * WMZ - BETADOT * WMX) + WMY * WMZ * (JMY - JMXZ) - (CB2
     &        * JIX + SB2 * JIYZ + JE) * BETADOT * WMY + CB * JE * PSIDOT
     &        * WIZ
      Y (4) = TCOY - WOX * WOZ * (JOXY - JOZ) - (CT2 * (JMY + SB2 * JIX)
     &        + ST2 * (JMXZ + JIYZ) + JE) * THETADOT * WOZ + ST * CT
     &        * (JMY - JMXZ + SB2 * (JIX-JIYZ)) * (THETADOT * WOY
     &        - PHIDOT * WOX) + SB * CB * CT * (JIX - JIYZ) * (PHIDOT
     &        * WOZ - BETADOT * WMY) + (CT * (SB2 * JIX + CB2 * JIYZ + JE)
     &        * BETADOT + ST * WMY * (JMY- JMXZ)) * WMX + SB * CT * JE
     &        * PSIDOT * WIZ + ST * (JE * PSIDOT + WIY * (JIYZ - JIX))
     &        * WIX
C
C     CALL SIMQ TO SOLVE FOR ACCELERATIONS
C
      CALL SIMQ (L, Y, 4, KS)
C
C     SET TO ZERO THE ACCELERATION OF ANY GIMBAL THAT IS STUCK
C
      DO 130 II = 1, 4
      IF (STUCK (II)) Y (II) = 0.D0
  130 CONTINUE
C
C     SET DXDT TO THE COMPUTED DERIVATIVE
C
      DXDT (1)  = PSIDOT
      DXDT (2)  = BETADOT
      DXDT (3)  = THETADOT
      DXDT (4)  = PHIDOT
      DXDT (5)  = Y (1)
      DXDT (6)  = Y (2)
      DXDT (7)  = Y (3)
      DXDT (8)  = Y (4)
      DXDT (9)  = WEX
      DXDT (10) = WEY
      DXDT (11) = WEZ
      RETURN
      END
C
C     SUBROUTINE TO SOLVE SYSTEMS OF SIMULTANEOUS LINEAR EQUATIONS
C
      SUBROUTINE SIMQ (A, B, N, KS)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
```

```
      DIMENSION A (1), B (1)
C
C     SOLVE SET OF EQUATIONS AX=B
C     A = MATRIX OF COEFFICIENTS STORED COLUMNWISE.  THESE ARE DESTROYED
C         IN THE COMPUTATION. THE SIZE OF A IS N BY N.
C     B = VECTOR OF ORIGINAL CONSTANTS (LENGTH N).  THESE ARE REPLACED
C         BY FINAL SOLUTION VALUES, VECTOR X.
C     N = NUMBER OF EQUATIONS AND VARIABLES.
C     KS= OUTPUT DIGIT.  0 FOR NORMAL SOLUTION, 1 FOR A SINGULAR SYSTEM.
C
      TOL = 0.D00
      KS = 0
      JJ = -N
      DO 65 J = 1, N
      JY = J + 1
      JJ = JJ + N +.1
      BIGA = 0.D0
      IT = JJ - J
      DO 30 I = J, N
      IJ = IT + I
      IF (ABS (BIGA) - ABS (A (IJ))) 20, 30, 30
   20 BIGA = A (IJ)
      IMAX = I
   30 CONTINUE
      IF (ABS (BIGA) - TOL) 35, 35, 40
   35 KS = 1
      RETURN
   40 I1 = J + N * (J - 2)
      IT = IMAX - J
      DO 50 K = J, N
      I1 = I1 + N
      I2 = I1 + IT
      SAVE = A (I1)
      A (I1) = A (I2)
      A (I2) = SAVE
   50 A (I1) = A (I1) / BIGA
      SAVE = B (IMAX)
      B (IMAX) = B (J)
      B (J) = SAVE / BIGA
      IF (J - N) 55, 70, 55
   55 IQS = N * (J - 1)
      DO 65 IX = JY, N
      IXJ = IQS + IX
      IT = J - IX
      DO 60 JX = JY, N
      IXJX = N * (JX - 1) + IX
      JJX = IXJX + IT
   60 A (IXJX) = A (IXJX) - A (IXJ) * A (JJX)
   65 B (IX) = B (IX) - B (J) * A (IXJ)
   70 NY = N -1
      IT = N  * N
      DO 80 J = 1, NY
      IA = IT - J
      IB = N - J
      IC = N
      DO 80 K = 1, J
      B (IB) = B (IB) - A (IA) * B (IC)
      IA = IA - N
   80 IC = IC - 1
      RETURN
      END
C
C     SUBROUTINE TO INVERT A MATRIX
C
      SUBROUTINE MINV (A, N, D, L, M)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A (1), L (1), M (1)
C
```

```
C     INVERT A MATRIX
C     A = INPUT ARRAY, DESTROYED IN COMPUTATION AND REPLACED BY INVERSE
C     N = ORDER OF MATRIX A
C     D = RESULTANT DETERMINANT. A ZERO DETERMINANT INDICATES A
C           SINGULAR MATRIX
C     L = WORK VECTOR OF LENGTH N
C     M = WORK VECTOR OF LENGTH N
C
      D=1.D0
      NK = -N
      DO 80 K = 1, N
      NK = NK + N
      L (K) = K
      M (K) = K
      KK = NK + K
      BIGA = A (KK)
      DO 20 J = K, N
      IZ = N * (J - 1)
      DO 20 I = K, N
      IJ = IZ + I
   10 IF (ABS (BIGA) - ABS (A (IJ))) 15, 20, 20
   15 BIGA = A (IJ)
      L (K) = I
      M (K) = J
   20 CONTINUE
      J = L (K)
      IF (J - K) 35, 35, 25
   25 KI = K - N
      DO 30 I = 1, N
      KI = KI + N
      HOLD = -A (KI)
      JI = KI - K + J
      A (KI) = A (JI)
   30 A (JI) = HOLD
   35 I = M (K)
      IF (I - K) 45, 45, 38
   38 JP = N * (I - 1)
      DO 40 J = 1, N
      JK = NK + J
      JI = JP + J
      HOLD = -A (JK)
      A (JK) = A (JI)
   40 A (JI) = HOLD
   45 IF (BIGA) 48, 46, 48
   46 D = 0.D0
      RETURN
   48 DO 55 I = 1, N
      IF (I - K) 50, 55, 50
   50 IK = NK + I
      A (IK) = A (IK) / (-BIGA)
   55 CONTINUE
      DO 65 I = 1, N
      IK = NK + I
      HOLD = A (IK)
      IJ = I - N
      DO 65 J = 1, N
      IJ = IJ + N
      IF (I - K) 60, 65, 60
   60 IF (J - K) 62, 65, 62
   62 KJ = IJ - I + K
      A (IJ ) = HOLD * A (KJ) + A (IJ)
   65 CONTINUE
      KJ = K - N
      DO 75 J = 1, N
      KJ = KJ + N
      IF (J - K) 70, 75, 70
   70 A (KJ) = A (KJ) /BIGA
   75 CONTINUE
```

```
      D = D * BIGA
      A(KK)=1.D0/BIGA
   80 CONTINUE
      K = N
  100 K = (K - 1)
      IF (K) 150, 150, 105
  105 I = L (K)
      IF (I - K) 120, 120, 108
  108 JQ = N * (K - 1)
      JR = N * (I - 1)
      DO 110 J = 1, N
      JK = JQ + J
      HOLD = A (JK)
      JI = JR + J
      A (JK) = -A (JI)
  110 A (JI) = HOLD
  120 J = M (K)
      IF (J - K) 100, 100, 125
  125 KI = K - N
      DO 130 I = 1, N
      KI = KI + N
      HOLD = A (KI)
      JI = KI - K + J
      A (KI) = -A (JI)
  130 A (JI) = HOLD
      GO TO 100
  150 RETURN
      END
C
C     MATRIX MULTIPLICATION SUBROUTINE
C
      SUBROUTINE MATMPY (A, B, C, L1, L2, M1, M2, N1, N2, IFLAG)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION A (1), B (1), C (1)
C
C     A = L1 BY L2 INPUT MATRIX
C     B = M1 BY M2 INPUT MATRIX
C     C = N1 BY N2 OUTPUT MATRIX
C
C     IF IFLAG = 1 THEN C = A * B
C              = 2        = TRANSPOSE (A) * B
C              = 3        = A * TRANSPOSE (B)
C              = 4        = TRANSPOSE (A) * TRANSPOSE (B)
C
      GOTO (100, 110, 120, 130),IFLAG
  100 I1 = L1
      I2 = M2
      I3 = L2
      GOTO 140
  110 I1 = L2
      I2 = M2
      I3 = L1
      GOTO 140
  120 I1 = L1
      I2 = M1
      I3 = L2
      GOTO 140
  130 I1 = L2
      I2 = M1
      I3 = L1
  140 DO 200 I = 1, I1
      DO 200 II = 1, I2
      TEMP = 0.D0
      DO 190 III = 1, I3
      GOTO (150, 160, 170, 180),IFLAG
  150 TEMP = TEMP + A (I + L1 * (III - 1)) * B (M1 * (II - 1) + III)
      GOTO 190
  160 TEMP = TEMP + A (L1 * (I - 1) + III) * B (M1 * (II - 1) + III)
```

```
      GOTO 190
  170 TEMP = TEMP + A (I + L1 * (III - 1)) * B (II + M1 * (III - 1))
      GOTO 190
  180 TEMP = TEMP + A (L1 * (I - 1) + III) * B (II + M1 * (III - 1))
  190 CONTINUE
      C (I + N1 * (II - 1)) = TEMP
  200 CONTINUE
      RETURN
      END
C
C     CONTROL SIGNAL GENERATION SUBROUTINE
C
      SUBROUTINE CONTRL (X, U, W, OLDRATE, DELTAT)
      IMPLICIT DOUBLE PRECISION(A-H,J-Z)
      PARAMETER IXX=11,IDR=7
      DIMENSION AAA(IXX,IXX),AAAA(IXX,IXX)
      DIMENSION X (1), U (1), W (1), OLDRATE (1)
      DIMENSION ASTAR (IXX,IXX), BSTAR (IXX,4), CSTAR (IXX)
      DIMENSION A (IXX,IXX), B (IXX,4), XDOT0 (IXX)
      DIMENSION AA (IXX,IXX), DIA (IXX,IXX), XDOTAX (IXX)
      DIMENSION ZERO (4), IWORK1 (4), IWORK2 (4)
      DIMENSION D (IDR,IXX), E (IDR), Q (IDR,IDR), R (4,4)
      DIMENSION DB (IDR,4), BDQ (4,IDR), BDQDB (4,4), RBDQDB (4,4)
      DIMENSION AX (IXX), AXC (IXX), DAXC (IDR), DAXCE (IDR), BDQDAXCE (4)
      DIMENSION KTR (4), KV (4), FCOULM (4)
      DIMENSION ASTRX(IXX),XNEW(IXX)
      DIMENSION XTEMP(IXX)
      DATA JE, JIX, JIYZ, JMXZ, JMY, JOXY, JOZ
     &     /1.2D-2, 1.7D-2, 1.3D-2, 2.25D-2, 3.0D-2, 3.0D-2, 3.9D-2/
      DATA FCOULM /.09D0,.1D0,.11D0,.165D0 /
      DATA KTR /1.9D-2,1.9D-2,4.1D-2,5.13D-2/
      DATA KV /6.1D-1,6.1D-1,7.9D-1,1.25D0/
      DATA ZERO /4*0.D0/
      DATA TORQWT, LOCKWT, TILTWT, RATEWT /1.D-13, 1.D-13, 1.D6,1.D-13/
C
C     ZERO   = ZERO VECTOR USED TO COMPUTE DERIVATIVE WITH U = 0
C     TORQWT = WEIGHT ASSIGNED TO TORQUE REQUIREMENT IN COST FUNCTION
C     LOCKWT = WEIGHT ASSIGNED TO GIMBAL LOCK PROXIMITY IN COST FUNCTION
C     TILTWT = WEIGHT ASSIGNED TO ELEVATION GIMBAL TILT IN COST FUNCTION
C     RATEWT = WEIGHT ASSIGNED TO ELEVATION GIMBAL RATE IN COST FUNCTION
C     FCOULM = COULOMB FRICTION TORQUE LIMIT. SPECIFIES THE FRICTION
C              MAGNITUDE WHEN THE GIMBALS ARE UNSTUCK.
C     KTR    = CONVERSION CONSTANTS FROM TORQUE MOTOR VOLTAGES TO TORQUES
C     KV     = PROPORTIONALITY CONSTANTS FROM ANGLEDOTS TO BACK EMFS
C     JE     = INERTIA ABOUT ANY AXIS OF THE ELEVATION GIMBAL
C     JIX    = INERTIA ABOUT THE X AXIS OF THE INNER GIMBAL
C     JIYZ   = INERTIA ABOUT THE Y AND Z AXES OF THE INNER GIMBAL
C     JMXZ   = INERTIA ABOUT THE X AND Z AXES OF THE MIDDLE GIMBAL
C     JMY    = INERTIA ABOUT THE Y AXIS OF THE MIDDLE GIMBAL
C     JOXY   = INERTIA ABOUT THE X AND Y AXES OF THE OUTER GIMBAL
C     JOZ    = INERTIA ABOUT THE Z AXIS OF THE OUTER GIMBAL
C
C     THE LINEARIZED CONTINUOUS TIME SYSTEM EQUATIONS ARE OF THE FORM
C     .                                           .
C     X(t) = A(t0) (X(t) - X(t0)) + B(t0) U(t) + X(t0)
C
C     THE LINEARIZED DISCRETE TIME SYSTEM EQUATIONS ARE OF THE FORM
C                   *          *         *
C     X[n+1] = A[n] X[n] + B[n] U[n] + C[n]
C
C     THE COST FUNCTION TAKES THE FORM
C                                  T                                T
C     J = (D[n] X[n+1] + E[n] ) Q (D[n] X[n+1] + E[n] ) + U[n] R U[n]
C
C     SOLUTION IS
C                   *T T    *-1*T T      *      *
C     U[n] = -(R + B D Q D B) B D Q (D (A X + C) + E)
C
```

```
C     ASSOCIATE VARIABLES WITH ARRAY ELEMENTS
C     COMPUTE REQUIRED TRIGONOMETRIC  FUNCTIONS
C
      DO 90 I=1,IXX
      XTEMP(I)=X(I)
   90 CONTINUE
      PSI   = X (1)
      BETA  = X (2)
      THETA = X (3)
      PHI   = X (4)
      SP = DSIN (PSI)
      CP = DCOS (PSI)
      SB = DSIN (BETA)
      CB = DCOS (BETA)
      ST = DSIN (THETA)
      CT = DCOS (THETA)
      SF = DSIN (PHI)
      CF = DCOS (PHI)
      SB2 = SB * SB
      CB2 = CB * CB
      ST2 = ST * ST
      CT2 = CT * CT
      PSIDOT   = X (5)
      BETADOT  = X (6)
      THETADOT = X (7)
      PHIDOT   = X (8)
      WCX = W (1)
      WCY = W (2)
      WCZ = W (3)
      WOX = CF * WCX - SF * WCZ
      WOY = WCY + PHIDOT
      WOZ = SF * WCX + CF * WCZ
      WMX = WOX + THETADOT
      WMY = CT * WOY + ST * WOZ
      WMZ =-ST * WOY + CT * WOZ
      WIX = CB * WMX + SB * WMY
      WIY =-SB * WMX + CB * WMY
      WIZ = WMZ + BETADOT
      WEX = CP * WIX - SP * WIZ
      WEY = WIY + PSIDOT
      WEZ = SP * WIX + CP * WIZ
C
C     I PSIDOT   I    I M11 M12 M13 M14 I I Y1 I
C     I BETADOT  I    I M21 M22 M23 M24 I I Y2 I
C    -I THETADOT I =  I M31 M32 M33 M34 I I Y3 I
C     I PSIDOT   I    I M41 M42 M43 M44 I I Y4 I
C
C     TORQUE EQUATIONS FOR THE FOUR GIMBALS EXCLUDING CONTROL SIGNALS
C
      Y1    = -KTR (1) * KV (1) * PSIDOT
     &         + JE * (-SB * PHIDOT * WOZ - CB * ST * PHIDOT * WOX
     &         - CB * THETADOT * WMZ + BETADOT * WIX)
      Y2    = -KTR (2) * KV (2) * BETADOT
     &         + (JIYZ + JE) * (THETADOT * WMY - CT * PHIDOT * WOX)
     &         + WIX * WIY * (JIX-JIYZ) - PSIDOT * JE * WIX
      Y3    = -KTR (3) * KV (3) * THETADOT
     &         + (JMXZ + CB2 * JIX + SB2 * JIYZ + JE) * PHIDOT * WOZ
     &         - SB * CB * (JIX-JIYZ ) * (ST * PHIDOT * WOX + THETADOT
     &         * WMZ - BETADOT * WMX) + WMY * WMZ * (JMY - JMXZ) - (CB2
     &         * JIX + SB2 * JIYZ + JE) * BETADOT * WMY + CB * JE * PSIDOT
     &         * WIZ
      Y4    = -KTR (4) * KV (4) * PHIDOT
     &         - WOX * WOZ * (JOXY - JOZ) - (CT2 * (JMY + SB2 * JIX)
     &         + ST2 * (JMXZ + JIYZ) + JE) * THETADOT * WOZ + ST * CT
     &         * (JMY - JMXZ + SB2 * (JIX-JIYZ)) * (THETADOT * WOY
     &         - PHIDOT * WOX) + SB * CB * CT * (JIX - JIYZ) * (PHIDOT
     &         * WOZ - BETADOT * WMY) + (CT * (SB2 * JIX + CB2 * JIYZ + JE)
     &         * BETADOT + ST * WMY * (JMY- JMXZ)) * WMX + SB * CT * JE
```

```
     &          * PSIDOT * WIZ + ST * (JE * PSIDOT + WIY * (JIYZ - JIX))
     &          * WIX
C
C     M MATRIX -- THIS IS THE INVERSE OF THE L MATRIX ABOVE
C     BECAUSE M IS SYMMETRIC ONLY THE UPPER HALF NEED BE COMPUTED
C
      DENOM = (JOXY+ST2*JMXZ+CT2*(JMY+CB2*JIYZ+SB2*(JIX+JE)))*(JMXZ
     &        +CB2*(JIX+JE)+SB2*JIYZ)-CB2*SB2*CT2*(JIYZ-JIX-JE)**2
      M11 = ((JOXY+ST2*JMXZ+CT2*(JMY+JIX+JE))*(JE+SB2*JIYZ
     &      +CB2*JIX+JMXZ)-CB2*CT2*(JIX-JIYZ)*(JMXZ+JIX+JE))/JE/DENOM
      M12 = -CB*ST*CT*(JE+JIX+JMXZ)/DENOM
      M13 = SB*(JOXY+ST2*JMXZ+CT2*(JMY+JIX+JE))/DENOM
      M14 = -CB*CT*(JMXZ+JIX+JE)/DENOM
      M22 = (CB2*CT2*(JIYZ-JIX-JE)*(JE+JIX+JMXZ)+(JMXZ+CB2*(JIX+JE)+SB2*
     &      JIYZ)*(JOXY+CT2*(JMY+JIX)+ST2*(JMXZ+JIYZ)+JE))/(JIYZ+JE)/DENOM
      M23 = SB*CB*ST*CT*(JIYZ-JIX-JE)/DENOM
      M24 = ST*(JMXZ+CB2*(JIX+JE)+SB2*JIYZ)/DENOM
      M33 = (JOXY+ST2*JMXZ+CT2*(JMY+CB2*JIYZ+SB2*(JIX+JE)))/DENOM
      M34 = CB*SB*CT*(JIYZ-JIX-JE)/DENOM
      M44 = (JMXZ+CB2*(JIX+JE)+SB2*JIYZ)/DENOM
C
C     PARTIAL DERIVATIVES OF Y AND M WITH RESPECT TO X FOLLOW
C     DY1DB = dY1/dBETA, DY1DFD = dY1/dPHIDOT ETC.
C
      DY1DB  = JE*(-CB*PHIDOT*WOZ+SB*ST*PHIDOT*WOX+SB*THETADOT*WMZ+
     &         BETADOT*WIY)
      DY1DT  = JE*CB*(THETADOT*WMY-CT*PHIDOT*WOX)+SB*BETADOT*WMZ*JE
      DY1DF  = JE*(-SB*PHIDOT*WOX+CB*ST*PHIDOT*WOZ-CB*CT*THETADOT*WOX
     &         +BETADOT*(SB*ST*WOX-CB*WOZ))
      DY1DPD = -KTR (1)*KV (1)
      DY1DBD = JE*WIX
      DY1DTD = -JE*CB*WMZ+JE*CB*BETADOT
      DY1DFD = JE*(-SB*WOZ-CB*ST*WOX+CB*ST*THETADOT+SB*CT*BETADOT)
      DY2DB  = (WIY**2-WIX**2)*(JIX-JIYZ)-PSIDOT*JE*WIY
      DY2DT=WMZ*((JIYZ+JE)*THETADOT+(SB*WIY+CB*WIX)*(JIX-JIYZ)-
     &         PSIDOT*SB*JE)+(JIYZ+JE)*ST*PHIDOT*WOX
      DY2DF=(JIYZ+JE)*(ST*THETADOT*WOX+CT*PHIDOT*WOZ)+(-CB*WOZ+SB*ST
     &         *WOX)*(WIY*(JIX-JIYZ)-PSIDOT*JE)+WIX*(SB*WOZ+CB*CT*WOX)
     &         *(JIX-JIYZ)
      DY2DPD = -JE*WIX
      DY2DBD = -KTR (2)*KV (2)
      DY2DTD = (JIYZ+JE)*WMY+(JIX-JIYZ)*(CB*WIY-SB*WIX)-CB*JE*PSIDOT
      DY2DFD=CT*(JIYZ+JE)*(THETADOT-WOX)-CT*(JIX-JIYZ)*WMX-SB*CT*JE
     &         *PSIDOT
      DY3DB  = 2*SB*CB*(JIYZ-JIX)*(PHIDOT*WOZ-BETADOT*WMY)+(SB2-CB2)
     &         *(JIX-JIYZ)*(ST
     &         *PHIDOT*WOX+THETADOT*WMZ-BETADOT*WMX)-SB*JE*PSIDOT*WIZ
      DY3DT  = SB*CB*(JIX-JIYZ)*(-CT*PHIDOT*WOX+THETADOT*WMY)+(WMZ**2
     &         -WMY**2)*(JMY-JMXZ)-(CB2*JIX+SB2*JIYZ+JE)*BETADOT*WMZ-CB
     &         *JE*PSIDOT*WMY
      DY3DF  = (JMXZ+CB2*JIX+SB2*JIYZ+JE)*PHIDOT*WOX-SB*CB*(JIX-JIYZ)
     &         *(-ST*PHIDOT*WOZ+THETADOT*CT*WOX+BETADOT*WOZ)+(ST*WMZ
     &         +CT*WMY)*WOX*(JMY-JMXZ)-(CB2*JIX+SB2*JIYZ+JE)*BETADOT
     &         *ST*WOX+CB*JE*PSIDOT*CT*WOX
      DY3DPD = CB*JE*WIZ
      DY3DBD = SB*CB*(JIX-JIYZ)*WMX-(CB2*JIX+SB2*JIYZ+JE)*WMY+CB*JE*PSIDOT
      DY3DTD = -KTR (3)*KV (3)+SB*CB*(JIX-JIYZ)*(BETADOT-WMZ)
      DY3DFD = (JMXZ+CB2*JIX+SB2*JIYZ+JE)*WOZ+SB*CB*ST*(JIX-JIYZ)*(THETADOT
     &         -WOX)+(CT*WMZ-ST*WMY)*(JMY-JMXZ)-CT*(CB2*JIX+SB2*JIYZ+JE)
     &         *BETADOT-CB*ST*JE*PSIDOT
      DY4DB  = 2*SB*CB*(-CT2*JIX*THETADOT*WOZ+(JIX-JIYZ)*(THETADOT*WOY
     &         *ST*CT-ST*CT*PHIDOT*WOX+CT*BETADOT*WMX))+(CB2-SB2)*CT
     &         *(JIX-JIYZ)*(PHIDOT*WOZ-BETADOT*WMY)+CB*CT*JE*PSIDOT*WIZ
     &         -ST*WIX**2*(JIYZ-JIX)+ST*(JE*PSIDOT+WIY*(JIYZ-JIX))*WIY
      DY4DT  = 2*CT*ST*(JMY+SB2*JIX-JMXZ-JIYZ)*THETADOT*WOZ+(CT2-ST2)
     &         *(JMY-JMXZ+SB2*(JIX-JIYZ))*(THETADOT*WOY-PHIDOT*WOX)+SB
     &         *CB*(JIX-JIYZ)*(-ST*(PHIDOT*WOZ-BETADOT*WMY)-CT*BETADOT
     &         *WMZ)-ST*(SB2*JIX+CB2*JIYZ+JE)*BETADOT*WMX+(CT*WMY+ST*WMZ)
```

```
     &          *(JMY-JMXZ)*WMX-SP*JE*PSIDOT*(ST*WIZ+CT*WMY)+CT*(JE*PSIDOT
     &          +WIY*(JIYZ-JIX))*WIX+CB*ST*WMZ*(JIYZ-JIX)*WIX+ST*(JE
     &          *PSIDOT+WIY*(JIYZ-JIX))*SB*WMZ
      DY4DF  = (WOZ**2-WOX**2)*(JOXY-JOZ)-(CT2*(JMY+SB2*JIX)+ST2*(JMXZ
     &          +JIYZ)+JE)*THETADOT*WOX+ST*CT*(JMY-JMXZ+SB2*(JIX-JIYZ))
     &          *PHIDOT*WOZ+SB*CB*CT*(JIX-JIYZ)*(PHIDOT*WOX-BETADOT*ST
     &          *WOX)+ST2*WOX*(JMY-JMXZ)*WMX-(CT*(SB2*JIX+CB2*JIYZ+JE)
     &          *BETADOT+ST*WMY*(JMY-JMXZ))*WOZ+SB*CT2*JE*PSIDOT*WOX+ST
     &          *WIX*(JIYZ-JIX)*(CP*ST*WOX+SB*WOZ)+ST*(JE*PSIDOT+WIY
     &          *(JIYZ-JIX))*(SB*ST*WOX-CB*WOZ)
      DY4DPD = SB*CT*JE*WIZ+ST*JE*WIX
      DY4DBD = -SB*CB*CT*(JIX-JIYZ)*WMY+CT*(SB2*JIX+CB2*JIYZ+JE)*WMX
     &          +SB*CT*JE*PSIDOT
      DY4DTD = -(CT2*(JMY+SB2*JIX)+ST2*(JMXZ+JIYZ)+JE)*WOZ+ST*CT*(JMY
     &          -JMXZ+SB2*(JIX-JIYZ))*WOY+ST*WMY*(JMY-JMXZ)+ST*(JIYZ-JIX)
     &          *(CB*WIY-SB*WIX)+CT*(SB2*JIX+CB2*JIYZ+JE)*BETADOT+CB*ST
     &          *JE*PSIDOT
      DY4DFD = -KTR (4)*KU (4)+ST*CT*(JMY-JMXZ+SB2*(JIX-JIYZ))*(THETADOT
     &          -WOX)+SB*CB*CT*(JIX-JIYZ)*(WOZ-CT*BETADOT)+ST*CT*(JMY-JMXZ)
     &          *WMX+ST*CT*(JIYZ-JIX)*(CB*WIX+SB*WIY)
C
      DDENDB = 2*SB*CB*(JIYZ-JIX-JE)*(JOXY+ST2*JMXZ+CT2*(JMY-JMXZ))
      DDENDT = 2*ST*CT*((JMXZ-JMY-CB2*JIYZ-SB2*(JIX+JE))*(JMXZ+CB2*(JIX+JE)
     &          +SB2*JIYZ)+SB2*CB2*(JIYZ-JIX-JE)**2)
      DM11DB = 2*SB*CB*(JOXY+ST2*JMXZ+CT2*(JMY-JMXZ))*(JIYZ-JIX)/JE/DENOM
     &          -DDENDB*M11/DENOM
      DM11DT = 2*ST*CT*((JMXZ-JMY-JIX-JE)*(JE+SB2*JIYZ+CB2*JIX+JMXZ)
     &          +CB2*(JIX-JIYZ)*(JMXZ+JIX+JE))/JE/DENOM-DDENDT*M11/DENOM
      DM12DB = SB*ST*CT*(JE+JIX+JMXZ)/DENOM-DDENDB*M12/DENOM
      DM12DT = CB*(ST2-CT2)*(JE+JIX+JMXZ)/DENOM-DDENDT*M12/DENOM
      DM13DB = CB*(JOXY+ST2*JMXZ+CT2*(JMY+JIX+JE))/DENOM-DDENDB*M13/DENOM
      DM13DT = 2*SB*ST*CT*(JMXZ-JMY-JIX-JE)/DENOM-DDENDT*M13/DENOM
      DM14DB = SB*CT*(JMXZ+JIX+JE)/DENOM-DDENDB*M14/DENOM
      DM14DT = CB*ST*(JMXZ+JIX+JE)/DENOM-DDENDT*M14/DENOM
      DM22DB = 2*SB*CB*(JIYZ-JIX-JE)*(JOXY+CT2*(JMY-JMXZ)+ST2*(JIYZ
     &          +JMXZ+JE))/(JIYZ+JE)/DENOM-DDENDB*M22/DENOM
      DM22DT = 2*CT*ST*(CB2*(JIX-JIYZ+JE)*(JE+JIX+JMXZ)+(JMXZ+CB2*(JIX
     &          +JE)+SB2*JIYZ)*(JMXZ+JIYZ-JMY-JIX))/(JIYZ+JE)/DENOM
     &          -DDENDT*M22/DENOM
      DM23DB = (CB2-SB2)*ST*CT*(JIYZ-JIX-JE)/DENOM-DDENDB*M23/DENOM
      DM23DT = SB*CB*(CT2-ST2)*(JIYZ-JIX-JE)/DENOM-DDENDT*M23/DENOM
      DM24DB = 2*SB*CB*ST*(JIYZ-JIX-JE)/DENOM-DDENDB*M24/DENOM
      DM24DT = CT*(JMXZ+CB2*(JIX+JE)+SB2*JIYZ)/DENOM-DDENDT*M24/DENOM
      DM33DB = 2*SB*CB*CT2*(JIX+JE-JIYZ)/DENOM-DDENDB*M33/DENOM
     -DM33DT = 2*ST*CT*(JMXZ-JMY-CB2*JIYZ-SB2*(JIX+JE))/DENOM
     &          -DDENDT*M33/DENOM
      DM34DB = (CB2-SB2)*CT*(JIYZ-JIX-JE)/DENOM-DDENDB*M34/DENOM
      DM34DT = SB*CB*ST*(JIX+JE-JIYZ)/DENOM-DDEN[illegible]NOM
      DM44DB = 2*SB*CB*(JIYZ-JIX-JE)/DENOM-DDENDB*M44/DENOM
      DM44DT = -DDENDT*M44/DENOM
C
C
C     DERIVATIVES OF WE WITH RESPECT TO X FOLLOW
C     DWXDFD = dWEX/dPHIDOT ETC.
C
      DWXDP  = -WEZ
      DWXDB  = CP * WIY
      DWXDT  = CP * SB * WMZ + SP * WMY
      DWXDF  = CP * SB * ST * WOX - CP * CB * WOZ - SP * CT * WOX
      DWXDBD = -SP
      DWXDTD = CP * CB
      DWXDFD = CP * SB * CT + SP * ST
      DWYDB  = -WIX
      DWYDT  = CB * WMZ
      DWYDF  = SB * WOZ + CB * ST * WOX
      DWYDPD=1.D0
      DWYDTD = -SB
      DWYDFD = CB * CT
```

```
      DWZDP  = WEX
      DWZDB  = SP * WIX
      DWZDT  = SP * SB * WMZ - CP * WMY
      DWZDF  = SP * SB * ST * WOX - SP * CB * WOZ + CP * CT * WOX
      DWZDBD = CP
      DWZDTD = SP * CB
      DWZDFD = SP * SB * CT - CP * ST
C
C     SET UP MATRIX A
C
C     CLEAR OUT A
C
      DO 100 I = 1, IXX
      DO 100 II = 1, IXX
      A (I,II) = 0.D0
  100 CONTINUE
C
      DO 110 I = 1, 4
     A(I,I+4)=1.D0
  110 CONTINUE
      A (5,2)  = DM11DB * Y1 + M11 * DY1DB + DM12DB * Y2 + M12 * DY2DB
     &           + DM13DB * Y3 + M13 * DY3DB + DM14DB * Y4 + M14 * DY4DB
      A (5,3)  = DM11DT * Y1 + M11 * DY1DT + DM12DT * Y2 + M12 * DY2DT
     &           + DM13DT * Y3 + M13 * DY3DT + DM14DT * Y4 + M14 * DY4DT
      A (5,4)  = M11 * DY1DF  + M12 * DY2DF  + M13 * DY3DF  + M14 * DY4DF
      A (5,5)  = M11 * DY1DPD + M12 * DY2DPD + M13 * DY3DPD + M14 * DY4DPD
      A (5,6)  = M11 * DY1DBD + M12 * DY2DBD + M13 * DY3DBD + M14 * DY4DBD
      A (5,7)  = M11 * DY1DTD + M12 * DY2DTD + M13 * DY3DTD + M14 * DY4DTD
      A (5,8)  = M11 * DY1DFD + M12 * DY2DFD + M13 * DY3DFD + M14 * DY4DFD
      A (6,2)  = DM12DB * Y1 + M12 * DY1DB + DM22DB * Y2 + M22 * DY2DB
     &           + DM23DB * Y3 + M23 * DY3DB + DM24DB * Y4 + M24 * DY4DB
      A (6,3)  = DM12DT * Y1 + M12 * DY1DT + DM22DT * Y2 + M22 * DY2DT
     &           + DM23DT * Y3 + M23 * DY3DT + DM24DT * Y4 + M24 * DY4DT
      A (6,4)  = M12 * DY1DF  + M22 * DY2DF  + M23 * DY3DF  + M24 * DY4DF
      A (6,5)  = M12 * DY1DPD + M22 * DY2DPD + M23 * DY3DPD + M24 * DY4DPD
      A (6,6)  = M12 * DY1DBD + M22 * DY2DBD + M23 * DY3DBD + M24 * DY4DBD
      A (6,7)  = M12 * DY1DTD + M22 * DY2DTD + M23 * DY3DTD + M24 * DY4DTD
      A (6,8)  = M12 * DY1DFD + M22 * DY2DFD + M23 * DY3DFD + M24 * DY4DFD
      A (7,2)  = DM13DB * Y1 + M13 * DY1DB + DM23DB * Y2 + M23 * DY2DB
     &           + DM33DB * Y3 + M33 * DY3DB + DM34DB * Y4 + M34 * DY4DB
      A (7,3)  = DM13DT * Y1 + M13 * DY1DT + DM23DT * Y2 + M23 * DY2DT
     &           + DM33DT * Y3 + M33 * DY3DT + DM34DT * Y4 + M34 * DY4DT
      A (7,4)  = M13 * DY1DF  + M23 * DY2DF  + M33 * DY3DF  + M34 * DY4DF
      A (7,5)  = M13 * DY1DPD + M23 * DY2DPD + M33 * DY3DPD + M34 * DY4DPD
      A (7,6)  = M13 * DY1DBD + M23 * DY2DBD + M33 * DY3DBD + M34 * DY4DBD
      A (7,7)  = M13 * DY1DTD + M23 * DY2DTD + M33 * DY3DTD + M34 * DY4DTD
      A (7,8)  = M13 * DY1DFD + M23 * DY2DFD + M33 * DY3DFD + M34 * DY4DFD
      A (8,2)  = DM14DB * Y1 + M14 * DY1DB + DM24DB * Y2 + M24 * DY2DB
     &           + DM34DB * Y3 + M34 * DY3DB + DM44DB * Y4 + M44 * DY4DB
      A (8,3)  = DM24DT * Y1 + M14 * DY1DT + DM24DT * Y2 + M24 * DY2DT
     &           + DM34DT * Y3 + M34 * DY3DT + DM44DT * Y4 + M44 * DY4DT
      A (8,4)  = M14 * DY1DF  + M24 * DY2DF  + M34 * DY3DF  + M44 * DY4DF
      A (8,5)  = M14 * DY1DPD + M24 * DY2DPD + M34 * DY3DPD + M44 * DY4DPD
      A (8,6)  = M14 * DY1DBD + M24 * DY2DBD + M34 * DY3DBD + M44 * DY4DBD
      A (8,7)  = M14 * DY1DTD + M24 * DY2DTD + M34 * DY3DTD + M44 * DY4DTD
      A (8,8)  = M14 * DY1DFD + M24 * DY2DFD + M34 * DY3DFD + M44 * DY4DFD
      A (9, 1)  = DWXDP
      A (9, 2)  = DWXDB
      A (9, 3)  = DWXDT
      A (9, 4)  = DWXDF
      A (9, 6)  = DWXDBD
      A (9, 7)  = DWXDTD
      A (9, 8)  = DWXDFD
      A (10, 2) = DWYDB
      A (10, 3) = DWYDT
      A (10, 4) = DWYDF
      A (10, 5) = DWYDPD
      A (10, 7) = DWYDTD
```

```
      A (10, 8) = DWYDFD
      A (11, 1) = DWZDP
      A (11, 2) = DWZDB
      A (11, 3) = DWZDT
      A (11, 4) = DWZDF
      A (11, 6) = DWZDBD
      A (11, 7) = DWZDTD
      A (11, 8) = DWZDFD
C
C     SET UP ARRAY B
C
C     CLEAR OUT B
C
      DO 120 I = 1, IXX
      DO 120 II = 1, 4
      B (I, II) = 0.D0
  120 CONTINUE
C
      B (5, 1) = KTR (1) * M11
      B (5, 2) = KTR (2) * M12
      B (5, 3) = KTR (3) * M13
      B (5, 4) = KTR (4) * M14
      B (6, 1) = KTR (1) * M12
      B (6, 2) = KTR (2) * M22
      B (6, 3) = KTR (3) * M23
      B (6, 4) = KTR (4) * M24
      B (7, 1) = KTR (1) * M13
      B (7, 2) = KTR (2) * M23
      B (7, 3) = KTR (3) * M33
      B (7, 4) = KTR (4) * M34
      B (8, 1) = KTR (1) * M14
      B (8, 2) = KTR (2) * M24
      B (8, 3) = KTR (3) * M34
      B (8, 4) = KTR (4) * M44
C
C     SET UP VECTOR XDOT0
C
      CALL DERIVE (XTEMP, XDOT0, ZERO, W, OLDRATE)
C
C     4 GIMBAL SYSTEM DYNAMIC EQUATIONS ARE NOW COMPLETELY LINEARIZED
C
C     THE DISCRETE TIME APPROXIMATIONS FOLLOW
C      *                        2   2
C     A = I + DELTAT * A + DELTAT * A / 2!
C     -
      DELTA2 = .5D0 * DELTAT ** 2
      DELTA3=DELTAT*DELTA2/3.D0
      DELTA4=DELTAT*DELTA3/4.D0
      DELTA5=DELTAT*DELTA4/5.D0
C
      CALL MATMPY (A, A, AA, IXX, IXX, IXX, IXX, IXX, IXX, 1)
      CALL MATMPY(A,AA,AAA,IXX,IXX,IXX,IXX,IXX,IXX,1)
      CALL MATMPY(A,AAA,AAAA,IXX,IXX,IXX,IXX,IXX,IXX,1)
      DO 130 I = 1, IXX
      DO 130 II = 1, IXX
      ASTAR (I, II) = DELTAT * A (I, II) + DELTA2 * AA (I, II)
     &+DELTA3*AAA(I,II)+DELTA4*AAAA(I,II)
      IF (I .EQ .II) ASTAR (I, II) = ASTAR (I, II) + 1.D0
  130 CONTINUE
C      *                    2
C     B = (DELTAT * I + DELTAT * A / 2!) B
C
      DO 140 I = 1, IXX
      DO 140 II = 1, IXX
      DIA (I, II) = DELTA2 * A (I, II)
     &+DELTA3*AA(I,II)+DELTA4*AAA(I,II)+DELTA5*AAAA(I,II)
      IF (I .EQ. II) DIA (I, II) = DIA (I, II) + DELTAT
  140 CONTINUE
```

```
      CALL MATMPY (DIA, B, BSTAR, 11, 11, 11, 4, 11, 4, 1)
C      *                         2         .
C     C = (DELTAT * I + DELTAT * A / 2!) (X (t0) - A X (t0))
C
      DO 160 I = 1, IXX
      TEMP = XDOT0 (I)
      DO 150 II = 1, IXX
      TEMP = TEMP - A (I, II) * XTEMP (II)
  150 CONTINUE
      XDOTAX(I)=TEMP
  160 CONTINUE
      CALL MATMPY (DIA, XDOTAX, CSTAR, IXX, IXX, IXX, 1, IXX, 1, 1)
C
C     THE MATRIX R REFLECTS THE WEIGHT OF THE CONTROL SIGNALS IN
C     THE COST FUNCTION
C
      DO 170 I = 1, 4
      DO 170 II = 1, 4
      R (I, II) = 0.D0
      IF (I .EQ. II) R (I, II) = TORQWT
  170 CONTINUE
C
C     THE MATRIX Q REFLECTS THE WEIGHT OF THE STATE IN THE COST FUNCTION
C
      DO 180 I = 1, IDR
      DO 180 II = 1, IDR
      Q (I, II) = 0.D0
      IF (I .EQ. II .AND. I .GE. 2) Q (I, II) = TILTWT
      IF(I.EQ.II.AND.I.GE.5)Q(I,II)=RATEWT
  180 CONTINUE
      Q (1, 1) = LOCKWT
C
C     THE MATRIX D COMPRESSES THE STATE INFORMATION AND LINEARIZES
C     THE GIMBAL LOCK COST
C
      DO 190 I = 1, IDR
      DO 190 II = 1, IXX
      D (I, II) = 0.D0
  190 CONTINUE
      DO 200 I = 1, 3
      D (I + 1, I + 8) = 1.D0
  200 CONTINUE
      D(1,2)=CB*ST
      D(5,1)=DWXDP
     -D(5,2)=DWXDB
      D(5,3)=DWXDT
      D(5,4)=DWXDF
      D(5,6)=DWXDBD
      D(5,7)=DWXDTD
      D(5,8)=DWXDFD
      D(6,2)=DWYDB
      D(6,3)=DWYDT
      D(1,3)=SB*CT
      D(6,4)=DWYDF
      D(6,5)=DWYDPD
      D(6,7)=DWYDTD
      D(6,8)=DWYDFD
      D(7,1)=DWZDP
      D(7,2)=DWZDB
      D(7,3)=DWZDT
      D(7,4)=DWZDF
      D(7,6)=DWZDBD
      D(7,7)=DWZDTD
      D(7,8)=DWZDFD
C
C     THE MATRIX E EXPRESSES THE OPTIMAL LINEARIZED NEXT STATE
C
      E(1)=SB*ST-BETA*CB*ST-THETA*SB*CT
```

```
      DO 210 I = 2, 4
      E (I) = 0.D0
  210 CONTINUE
      E(5)=WEX-PSI*DWXDP-BETA*DWXDB-THETA*DWXDT-PHI*DWXDF
     &     -BETADOT*DWXDBD-THETADOT*DWXDTD-PHIDOT*DWXDFD
      E(6)=WEY-BETA*DWYDB-THETA*DWYDT-PHI*DWYDF-PSIDOT*
     &     DWYDPD-THETADOT*DWYDTD-PHIDOT*DWYDFD
      E(7)=WEZ-PSI*DWZDP-BETA*DWZDB-THETA*DWZDT-PHI*DWZDF-
     &     BETADOT*DWZDBD-THETADOT*DWZDTD-PHIDOT*DWZDFD
C
C     COMPUTE U
C
      CALL MATMPY (D, BSTAR, DB, IDR, IXX, IXX, 4, IDR, 4, 1)
      CALL MATMPY (DB, Q, BDQ, IDR, 4,IDR, IDR,4,IDR, 2)
      CALL MATMPY (BDQ, DB, BDQDB, 4, IDR, IDR, 4, 4, 4, 1)
      DO 220 I = 1, 4
      DO 220 II = 1, 4
      RBDQDB (I, II) = R (I, II) + BDQDB (I, II)
  220 CONTINUE
      CALL MINV (RBDQDB, 4, DETERM, IWORK1, IWORK2)
      CALL MATMPY (ASTAR, XTEMP, AX, IXX, IXX, IXX, 1, IXX, 1, 1)
      DO 230 I = 1, IXX
      AXC (I) = AX (I) + CSTAR (I)
  230 CONTINUE
      CALL MATMPY (D, AXC, DAXC, IDR, IXX, IXX, 1, IDR, 1, 1)
      DO 240 I = 1, IDR
      DAXCE (I) = DAXC (I) + E (I)
  240 CONTINUE
      CALL MATMPY (BDQ, DAXCE, BDQDAXCE, 4, IDR, IDR, 1, 4, 1, 1)
      CALL MATMPY (RBDQDB, BDQDAXCE, U, 4, 4, 4, 1, 4, 1, 1)
      DO 250 I = 1, 4
      U(I)=-U(I)
  250 CONTINUE
      RETURN
      END
```

Bibliography

1. M. Athans, The Matrix Minimum Principle, Information and Control, Vol. 11, 1967

2. M. Athans and F. C. Schweppe, Gradient Matrices and Matrix Calculations, MIT Lincoln Laboratory Technical Note 1965-53, 17 Nov 1965

3. K. R. Britting, Inertial Navigation Systems Analysis, Wiley-Interscience, 1971

4. C. T. Chen, Introduction to Linear System Theory, Holt, Rinehart and Winston, 1970

5. G. Fiddes and D. Mislivec, Transient Torque Requirements for a 4 Gimbal Inertial Guidance Platform, General Electric Company, TIS 66 POD 8

6. M. A. Gennert, Torque Equations for a Four Gimbal System: A State Space Approach, General Electric Company, January 1980

7. S. W. Gully, The Effects of Friction and Base Motion on High Performance Servos, General Electric Company, TIS 69 POD 10

8. R. W. Reeves, Characterizing a Four Gimbal Inertial Platform, General Electric Company, August 30 1974

9. A. K. Sen, Tools of Kinematics: A Critical Review, IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-15 No. 1, January 1979

10. R. V. Spencer, Development of Preliminary Torque Equations for Optimal Gimbal Control, General Electric Company, April 12 1980