T-977

A FEEDBACK LINEARIZATION APPROACH TO SPACECRAFT CONTROL USING MOMENTUM EXCHANGE DEVICES

by

John Edward Dzielski

April 1988

Doctor of Philosophy Thesis
Massachusetts Institute of Technology

The Charles Stark Draper Laboratory, Inc.
555 Technology Square
Cambridge, Massachusetts 02139

A Feedback Linearization Approach to Spacecraft Control Using Momentum Exchange Devices

by

John Edward Dzielski

S.M., Mechanical Engineering, Massachusetts Institute of Technology (1984)
B.S., Mechanical Engineering, Carnegie-Mellon University (1982)

SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
DOCTOR OF PHILOSOPHY IN MECHANICAL ENGINEERING
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 1988

Signature of Author ______________________
Department of Mechanical Engineering
May 1988

Certified by ______________________
Professor Derek Rowell, Thesis Supervisor
Department of Mechanical Engineering

Certified by ______________________
Professor David Wormley, Thesis Supervisor
Head, Department of Mechanical Engineering

Certified by Edward V. Bergmann
Edward Bergmann, Technical Supervisor
Section Chief, C. S. Draper Lab., Inc.

Certified by ______________________
Joseph Paradiso, Technical Supervisor
C. S. Draper Lab., Inc.

Accepted by ______________________
Professor Ain Sonin
Chairman, Departmental Graduate Committee

A Feedback Linearization Approach to Spacecraft Control Using Momentum Exchange Devices

by

John Edward Dzielski

Submitted to the Department of Mechanical Engineering
on April 29, 1988 in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy in Mechanical Engineering

ABSTRACT

Recent developments in the area of nonlinear control theory have shown how coordinate changes in the state and input spaces can be used with nonlinear feedback to transform certain nonlinear ordinary differential equations into equivalent linear equations. These feedback linearization techniques are applied to resolve two problems arising in the control of spacecraft equipped with control moment gyroscopes (CMGs). The first application involves the computation of rate commands for the gimbals that rotate the individual gyroscopes to produce commanded torques on the spacecraft. The second application is to the long-term management of stored momentum in the system of control moment gyroscopes using environmental torques acting on the vehicle.

An approach to distributing control effort among a group of redundant actuators is described that uses feedback linearization techniques to parameterize sets of controls which influence a specified subsystem in a desired way. The approach is adapted for use in spacecraft control with double-gimballed gyroscopes to produce an algorithm that avoids problematic gimbal configurations by approximating sets of gimbal rates that drive CMG rotors into desirable configurations. The algorithm solves a normed approximation problem to match the desired rates using an affine parameterization of all gimbal rates that produce a specified acceleration of the spacecraft.

The momentum management problem is stated as a trajectory optimization problem with a nonlinear dynamical constraint. Feedback linearization and collocation are used to transform this problem into an unconstrained nonlinear program. The procedure is developed in a general fashion for application to a broad class of linearizable systems. The approach to trajectory optimization is fast and robust to starting conditions in comparison to variational methods. A number of examples are presented showing applications to the proposed NASA space station.

Thesis Advisor: Professor Derek Rowell
Department of Mechanical Engineering
Technical Supervisor: Edward Bergmann
C. S. Draper Laboratory, Inc.

Acknowledgement

I would like to begin by thanking my two technical supervisors, Ed Bergmann and Joe Paradiso, for their guidance and support throughout my stay at Draper Laboratory. Ed encouraged me to pursue a number of personal inerests that ultimately lead to the formulation of a thesis topic that I have enjoyed studying. His suggestions and advice helped to maintain a focus on the important practical issues that needed to be addressed. Joe was always a ready sounding board for some ideas that, more often than not, hadn't been worked out to the point that I could state them clearly. I always appreciated his willingness to listen to me ramble about the latest insights and developments as my work proceeded. Also, I enjoyed our many lunches together, and the lengthy conversations we've had on all manner of subjects. In addition, Joe read drafts of the thesis with special detail and attention, the final draft surely benefited from his many comments.

I would also like to thank the members of my committee from MIT. Professors Wormley and Rowell were always interested and supportive, and I appreciated their help and suggestions. Prof. John Wyatt deserves special mention both for his enthusiasm for the work, and the initial help he provided in directing me to an understanding of certain theoretical concepts.

There are a large number of people here at Draper who either helped me directly with my work or were simply there to make the days more entertaining. Naz Bedrossian, Brent Appleby, and a number of Draper Fellows, past and present, were the sources of many stimulating and humorous discussions. I hope that they

learned as much from me as I learned from them.

I would also like to thank the unknown people in the Education and Personnel Offices that sheltered me from the myriad of paperwork that probably had to be done to maintain my tuition and funding.

I would like my parents to know that I greatly appreciate their support ever the years. In my youth they stood by me during difficult times, and their financial support in recent years has made day-to-day life much easier than it might have been.

Finally, I would like to extend my warmest love and deepest gratitude to my wife, Johanna, and children, Mary, Jane, and Dylan. The past five years have not been particularly easy or comfortable for my family, and they too have had to sacrifice so that I could pursue my doctorate. Their presence in my life is a constant source of reward and enrichment.

This document was prepared at The Charles Stark Draper Laboratory, Inc., under contract NAS9-17560 with the National Aeronautics and Space Administration.

Publication of this document does not constitute approval by the Draper Laboratory or NASA of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

John Dzielski

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The need to regulate the attitude of an orbiting spacecraft defines an important part of the overall design problem for a spacecraft control system. For example, the proposed NASA space station is required to provide stable inertial and Earth-pointing platforms for scientific instruments, and a platform capable of tracking objects. It is also necessary to prevent large attitude excursions during Space Shuttle docking and during re-boost maneuvers. The space station is being designed to orbit the Earth at a fixed attitude with respect to the local horizontal. In order to meet a wide variety of mission requirements, there are limits to the allowable deviations from the design attitude. These limits must be enforced even in the presence of uncertain environmental effects and on-board disturbances. Examples of environmental effects include aerodynamic and gravity torques, and on-board disturbances include crew motion, motion of the Mobile Manipulator, and the motion of the solar panels as they track the sun. As a consequence, it is necessary to develop algorithms which can utilize a set of available actuators to apply torques to the vehicle in order to maintain the attitude within the specified limits.

There are a variety of actuators available to generate torques on a spacecraft for the purpose of attitude control. While propulsive actuators such as reaction control jets are probably the most common, there is another class of actuators which operate by transferring momentum to and from the spacecraft via a momentum "storage" system. The two types of actuators which operate in this way are reaction wheels and control moment gyroscopes (CMGs). A reaction wheel consists of a flywheel whose axis of rotation is fixed with respect to the spacecraft. Torques are applied to the vehicle by speeding up or slowing down the spin rate of the flywheels. A CMG is constructed with a rotor that spins at a constant speed and is mounted in a gimbal system which can reorient the flywheel. CMGs can be used to apply torques to a spacecraft by rotating the angular momentum vectors of the individual rotors. Moving the rotors changes the net angular momentum of the group of CMGs. In the absence of external torques, momentum conservation implies a corresponding change in the vehicle momentum, or equivalently, the angular rate of the vehicle.

There are a number of other devices which can be used to apply torques to orbiting spacecraft. Magnetic torquers produce torques on a vehicle through the interaction of the Earth's magnetic field with currents in coils of wire. It is also possible to adjust panels attached to the spacecraft to alter the distribution of aerodynamic forces on the vehicle caused by the thin atmosphere encountered in low Earth orbit [1]. These last two types of actuators are ill-suited for the purpose of attitude control because of the limited range and direction of the torques they are capable of producing.

Because CMGs are capable of storing and smoothly manipulating large quantities of momentum over long periods of time, they are often favored in applications requiring precision pointing and the attitude control of large, long duration spacecraft. In contrast to a reaction wheel, the small torque required along the gimbal axes to move the CMG rotor can produce a significantly larger torque on the vehicle. This effect is called torque amplification, and the effective gain is proportional to the angular momentum stored in the CMG rotor. Also, since the CMG rotor spins at a constant rate, it is easier to optimize the hardware for reduced friction and good dynamical characteristics. In contrast to jets, CMGs do not consume fuel which must be periodically resupplied; rather, they use electrical power which is readily available in spacecraft applications. In addition, CMGs can supply a broad range of output torques without the quantization effects associated with jets. Another undesirable characteristic of jets is that their exhaust tends to contaminate the environment in the vicinity of the spacecraft.

1.1 Background

There are two basic mechanical configurations of CMGs; the single-gimbal and the double-gimbal control moment gyroscope. The two types of construction are shown schematically in Figure 1.1. As the name suggests, the single-gimbal CMG has a single-gimbal drive that rotates the flywheel about an axis perpendicular to the spin axis. This constrains the momentum vector to rotate on a circle in $\mathbf{R}^3$. The double-gimbal CMG has an inner and outer gimbal drive which can be used to orient the momentum vector on a sphere in $\mathbf{R}^3$ (in the absence of hardware limits). Single-gimbal CMGs tend to be lighter for a given peak torque rating and are simpler in construction, while control algorithms for double-gimbal CMGs tend to be easier to implement.

In any particular application, control torques are generated by manipulating the momentum stored in a group of CMGs. The net stored momentum is given by a function of the form

$$h = h_{cmg}(\theta) \tag{1.1}$$

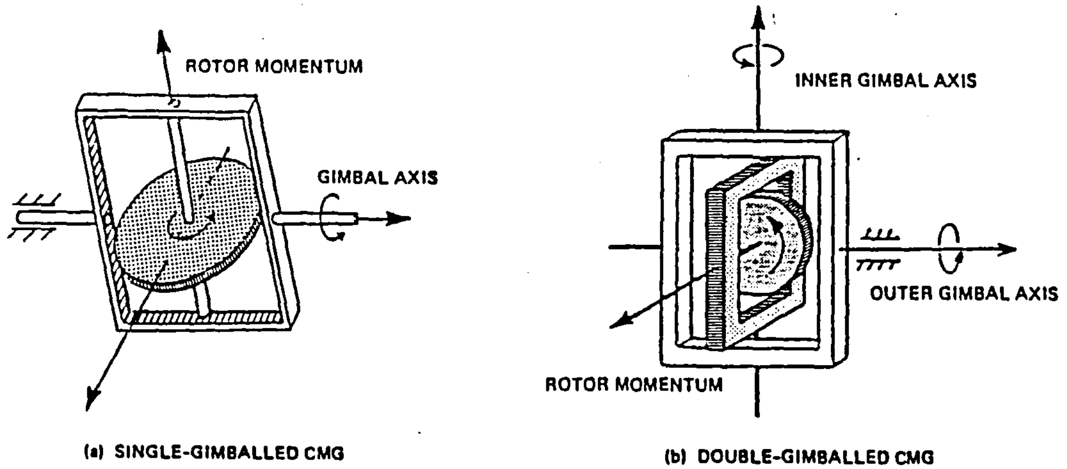


Figure 1.1: Mechanical Construction of CMGs.

where $h \in \mathbf{R}^3$ is the net CMG momentum vector in a spacecraft-fixed reference frame, $\theta \in \mathbf{R}^m$ denotes the set of gimbal displacements, and $h_{cmg} : \mathbf{R}^m \longrightarrow \mathbf{R}^3$ maps the gimbal angles into the momentum space based on the types of CMGs involved and the geometry of the mounting configuration. This equation can be differentiated with respect to time in a spacecraft-fixed reference frame to yield

$$\tau = \dot{h} = \omega \times h + \mathbf{D}[h_{cmg}(\theta)]\dot{\theta} \tag{1.2}$$

where ω is the angular rate of the spacecraft, $\mathbf{D}[h_{cmg}(\theta)]$ denotes the derivative of the mapping h_{cmg}, and $-\tau$ is the torque applied to the spacecraft due to the CMG gimbal motion. The torque is typically specified by an autopilot or a feedback control law. In general, the number of degrees-of-freedom in the CMG system, m, is greater than the number required to provide three-axis control of the spacecraft.

The problem that naturally arises is how to choose the gimbal rates, $\dot{\theta}$, to achieve a specified torque, τ. This so-called *steering problem* is complicated by the inherent nonlinearity in these devices as well as the need to manage the redundancy in a CMG configuration to avoid geometric singularities. Singularities occur when the state of the CMG system is such that the the output torque, τ, cannot be specified arbitrarily due to a drop in rank of the derivative in (1.2). The loss of rank corresponds to a loss of three-axis controllability of the spacecraft. Depending on the nature of the singularity, it may not be possible to rotate the CMGs out of the singular state without producing torques in undesirable directions [2].

The problem of steering single-gimbal CMGs is considerably more difficult than steering double-gimbal CMGs. The reason for this difficulty is that the topology of the momentum envelopes and singular states is more complicated for groups of single-gimbal CMGs [3]. For a given number of CMGs, a group of double-gimbal CMGs has twice the kinematic freedom which can be exploited to avoid singularities. Most papers on CMG steering consider only one of the two types of construction.

To date, most approaches used in solving the CMG steering problem have involved fixing the state (ω, θ) in (1.2) and computing a set of gimbal rate commands that optimize some objective function subject to the constraint that a specified torque is produced. One exception to this method is the approach in [4] where gimbal displacements are computed which produce a specified angular rate change of the vehicle. The objective function used by the various steering algorithms is chosen such that its value reflects the desirability of a given configuration. At some point, most of these algorithms use the Moore-Penrose pseudo-inverse solution of (1.2) to compute a torque producing set of gimbal rates, and then add a *null motion*

[5]. A null motion is a set of gimbal rates which produces no torque, and hence lies in the null space of the matrix $\mathbf{D}[H_{cmg}(\theta)]$. The null-motion is chosen to make the derivative of the objective function as small - in the case of a minimization - as possible without exceeding physical limitations. Such an approach leads to an algorithm that steers the gimbals to configurations which are locally optimal for the given objective function.

In [6], the objective function used by the steering algorithm is the inverse of the *gain* of a double-gimbal CMG configuration. The gain is defined to be the product of the singular values of $\mathbf{D}[H_{cmg}(\theta)]$, and singular configurations correspond to a gain of zero since they are associated with a drop in rank of this matrix. The gain function has been used by other authors, but has not been found to be effective due to the fact that it often cannot predict an impending singularity early enough to avoid it. It has also been demonstrated that trajectories following local optima of the gain can lead directly into singularities [7].

To attain a margin of global effectiveness, other authors have investigated steering laws which either explicitly or implicitly locate singularities in the gimbal space and rotate the gimbal angles to avoid them. For a group of four single-gimbal CMGs, the steering law described in [7] steers to the configuration of global optimal gain for a given net momentum while avoiding tabulated singularities of the CMG configuration. Some disavantages of this approach are a lack of adaptability and the necessity of storing a very large data file. The steering laws proposed in [8], [5] and [4] steer the CMGs away from configurations that are known to be singular. The approach of [8] is to compute the sets of gimbal angles which are singular with respect to the requested torque vector and rotate the gimbals to avoid them. In other words, gimbal angles are found for each CMG such that the CMG cannot instantaneously project momentum in the desired direction. The steering law then chooses rates that steer away from these gimbal states. The steering law in [5] exploits knowledge of a very specific mounting configuration to design an algorithm which implicitly avoids singularities. The method of [4] is probably the most adaptable approach in that changes in hardware and CMG configuration are easily accomodated, and the objective function can be changed to reflect the most urgent of the CMG configuration priorities.

The algorithm proposed in [4] treats (1.2) as a constraint in a linear programming problem, thus avoiding some of the numerical difficulties in solving this equation at or near singularities. In addition, the algorithm has been used to combine the use of CMGs with jets to maintain vehicle control outside the torque envelope of the CMG system alone. Another advantage of this approach is that it explicitly incorporates limits on the maximum rates achievable by the gimbal drive motors. All other approaches that include this limitation generally impose it by scaling after a set of gimbal rates has been computed; thus preserving the direction chosen by the optimization, but violating the requirement that a given torque be produced. Scaling is particularly undesirable near a singular state, when the pseudo-inverse yields very large torque producing gimbal rates. These large rates tend to cancel any attempts at exploiting null-motion to avoid the singular configuration.

In [2], a number of algorithms are investigated for the control of single-gimbal CMGs. One of these methods is the singularity robust inverse method [9], which attempts to solve (1.2) by defining a vector consisting of the error in the torque produced by a choice of gimbal rates concatenated with the set of gimbal rates. A solution is found by minimizing a weighted inner-product of this vector over the set of gimbal rates. Near singular states, this algorithm maintains reasonable gimbal rates by sacrificing torque in the direction associated with the singularity.

In addition to the nonlinear behavior of the CMG system, the dynamical equations of a rotating spacecraft are nonlinear. In particular, there is Euler coupling due to asymmetries in the spacecraft inertia, and coupling of the CMG momentum into these equations. There are also nonlinear attitude dependent torques due to gravity and aerodynamic effects. A number of people have linearized the equations of motion for CMG equipped spacecraft and designed regulators for the linear system which optimize an integral quadratic performance measure [10], [11]. This approach requires assumptions that are almost never satisfied in realistic situations. The usual approach to such a combined spacecraft/CMG control problem is to separate spacecraft control from CMG steering. An attitude control law generates a torque request that is sent to a CMG steering law, and the CMG steering law generates a set of gimbal rate commands for the CMGs. The fact that there is a strong coupling between the spacecraft dynamics and the CMG motion is not exploited in this type of an approach.

As noted before, spacecraft do not operate in an entirely torque-free environment. For example, there are torques due to gravity and to aerodynamic drag in the very thin atmosphere, as well as forces caused by equipment or crew motion. In order to compensate for these disturbances, the CMGs may be required to apply small torques in one direction over a long period of time. Since the CMG rotor spins at a constant rate, each CMG in a configuration stores a fixed magnitude of momentum; hence, there is a maximum amount of momentum that a group of

CMGs can project in any given direction. As a consequence, it is necessary to examine long-term momentum requirements of the spacecraft to insure that the CMG momentum capacity is not saturated. This third problem is called momentum management. The approaches proposed for use with NASA's space station manipulate the attitude dependent gravity gradient torque to control the rate at which momentum is stored in the CMG system. Skylab used a combination of reaction jets and aerodynamic torques to regulate momentum build-up in the CMGs [12]. The solution to the momentum management problem usually employs a model to predict the external torques acting on a spacecraft, and regulates the attitude to keep the resultant torque small or to produce torques which extract momentum built-up during previous orbits. Typically, the gravity torques can be modeled quite well, while aerodynamic torques are usually predicted using data from previous orbits. The problem of computing attitude and momentum histories to be tracked during an orbit has been formulated and solved as a linear quadratic optimization problem in [13], [14]. The linearizing assumptions may be reasonable under the assumption that CMGs are not included in the vehicle model, and that attitude excursions and rates are sufficiently small. The approach in [15] uses data from previous orbits to predict the momentum requirements over the next orbit due to environmental effects and computes a sequence of attitude step commands which result in a desired change in momentum over the orbit. This approach has been shown to be robust to uncertainties in the vehicle and environmental data.

1.2 Thesis Objectives and Summary of Contents

Even though CMGs have been used in a number of applications, they have not found widespread use. Single-gimbal CMGs have been used on balloon born platforms [7], in scissored pairs on early versions of NASA's Manned Maneuvering Unit, and have recently been installed on the Soviet Union's MIR space station [16]. Double-gimbal CMGs were used in NASA's Skylab [17] and are proposed for use on the planned NASA space station [18]. The principle difficulty in using CMGs for spacecraft attitude control is finding reliable solutions to the steering problem. One reason for this difficulty is that the steering problem does not consider the long term behavior of the system, but uses only the current state and torque request to compute gimbal rates. In other words, steering laws are local in nature and do not consider the effects of current decisions on the future. As a result, there is no steering law which can guarantee that it will always keep the CMG configuration out of singular states.

Another difficulty with using separate attitude control and CMG steering algorithms is that it is virtually impossible to predict the system behavior using locally computed CMG trajectories in a realistic environment. The main objective of this thesis is to develop an approach to the control of spacecraft with CMGs in which the spacecraft and the CMG actuators are treated as a combined unit, and to apply the approach to attitude control and momentum management of the proposed NASA space station.

In this thesis, the feedback linearization theory of [19] is used in the formulation and solution of two important problems in spacecraft control with CMGs. The theory identifies a class of nonlinear dynamical systems which can be transformed to exhibit linear input-output behavior. The transformations do not introduce the errors associated with a Taylor series approximation, and existing methodologies for linear systems synthesis can be applied to design closed-loop controllers. A detail theoretical background is presented in Appendix B.

A number of applications using feedback linearization theory have appeared in the literature. In [20], [21], the theory is used to develop control laws for spacecraft and spacecraft with reaction wheels, respectively. Although the attitude control problems considered in both of these papers are quite similar, neither provides a realistic treatment when redundant actuators are included. In addition to the fact that redundancy is an important characteristic of CMG control systems, reaction wheel systems do not exhibit geometric singularities like CMG systems. In [22] and [23], the linearized dynamical systems are used in the design of autopilots for helicopters and aircraft, while [24] studies the control of robots.

There is, however, a difficulty that can occur when designing control laws based on the linearized system dynamics. Because the transformations involve a change of coordinates in the state and control variables, the relationships between the linearized dynamics and the original *control problem* can become complicated or unclear. One usually considers control problems involving a dynamical system along with a set of qualitative and quantitative features that characterize desirable behavior. It is these features may not be easily interpretted in terms of the linearized system. In other words, the minimization of a quadratic performance index for the nonlinear system, is generally not the same as the minimization of a quadratic performance index with a feedback linearized dynamical constraint. In contrast to the developments referenced earlier, feedback linearization techniques are used in this thesis to simplify nonlinear dynamical models, while control and performance objectives are considered in terms of the original variables.

In applying the feedback linearization theory to the spacecraft control problem with CMGs, it was found that the theory is not well suited for use in some problems in which there are redundant actuators. The reason for the difficulty is that the extra degrees-of-freedom are not needed to solve the feedback linearization problem, and the redundancy is effectively decoupled from from the spacecraft dynamics. In Chapter 3, the problem of redundancy is resolved by a method that distributes control effort among a group of redundant actuators. This method is applied to the CMG steering problem to yield a class of steering algorithms in which a desired set of gimbal rates is computed, and then a normed approximation problem is solved to obtain a *best* match to the desired rates. The problem of designing a steering law is then one of determining an algorithm for computing gimbal rates which move the CMG rotors toward desired configurations. This is conceptually simpler than the conventional approach of determining a function whose derivative can be used to compute desired rates. Another advantage is that once a basic algorithm has been implemented, it is possible to add decision rules which can recognize and correct for known failure modes, or to design an algorithm which is entirely rule based. Such an *expert system* type of approach was employed with Skylab where much of the problem resolution was performed manually. In this thesis, a particular algorithm for determining the desired rates is proposed for the parallel mounting configuration to be used on the NASA space station. A number of examples are presented showing the behavior of the system for different formulations of the approximation problem. The formulation of the approximation problem makes use of an affine parameterization of all CMG gimbal rates that produce a given acceleration. Applied to existing algorithms, this parameterization can be used to eliminate the torque constraint associated with the optimization, and also to reduce the number of decision variables by three.

In Chapter 4, the problem of computing trajectories is addressed by formulating and solving an optimal momentum management problem with the CMGs included in the vehicle model. It is assumed that gravity gradient torques are to be balanced against other external torques to control the rate of momentum storage in the system of CMGs. All the external disturbance torques are lumped into a single term that is assumed known prior to solving the optimization problem or can be predicted using information from previous spacecraft orbits [25]. No linearizing assumptions are made with regard to the CMG kinematics or the dynamical equations of the vehicle. The momentum management problem is stated as the minimization of an integral quadratic performance measure with a nonlinear dynamical constraint. This problem is then transformed into an unconstrained nonlinear program by feedback linearizing the dynamical constraint and approximating the solution of the linear system of equations on a reduced basis of functions. The solution of the nonlinear program provides control histories which produce a spacecraft attitude trajectory that does not require the CMGs to store excessive amounts of momentum, and simultaneously yields CMG trajectories that avoid singular configurations. By including the CMGs in the problem, it is possible to obtain a solution in which the CMGs avoid singular states whenever possible. Chapter 4 concludes with a conceptual description of a feedback control system architecture which will track the desired trajectories and reject disturbances. The design of the specific feedback control laws is left for future research.

The momentum management problem is different from the usual problem of rotating a spacecraft from one attitude to another. In the *attitude maneuver* problem, the total momentum of the vehicle is assumed constant, or it is assumed to be impossible to influence the net change in momentum. The method proposed here for momentum management is applicable to the solution of this maneuver problem. The two problems are fundamentally different, however, and the application of the proposed method to the maneuver problem is not studied in this thesis. Since the NASA space station will not be required to perform large angle attitude maneuvers, but will be required to hold a specified attitude for long periods of time, the momentum management problem is the appropriate trajectory planning problem to study. Results presented in the following chapters are derived for general spacecraft/CMG configurations, but the specific examples all use vehicle data and CMG configurations for the planned NASA space station.

Some results in this thesis are applicable to problems other than the control of spacecraft with CMGs. In particular, there is a strong analogy between the steering problem for CMGs and the inverse kinematics problem encountered in robotics. The connection is made in [2] for single-gimbal CMGs and planar, serial-link robots, the relationship between singularities in the two systems is made quite nicely in this thesis. The robotic analogy for a double-gimbal CMG is a robot joint with two degrees-of-freedom. Most steering laws can be interpreted as *resolved motion methods* when applied to inverse kinematic problems involving robots [26]. The attitude maneuver problem discussed in the preceding paragraph corresponds to the problem in robotics of computing optimal inverse kinematic *trajectories* which take a robot end-effector from an initial orientation to a specified final orientation. The method of Chapter 4 is applicable to this problem. In fact, a recent paper describes an approach similar to that taken in Chapter 4 for the robotic problem [27]. In this paper the authors use penalty function methods to satisfy the constraints imposed by the dynamical equations on the nonlinear program. The approach of Chapter 4 could be used to satisfy the constraint exactly and to state the problem in terms of a significantly reduced number of variables.

In general, the approach to optimization outlined in Chapter 4 can be applied to all nonlinear systems which are feedback linearizable as defined in Appendix B. Broad classes of models for aircraft and robotic systems are known to satisfy these conditions. For systems which cannot be linearized, provisional methods are outlined for the solution of the resulting constrained nonlinear programs.

Before developing the feedback linearization theory and applying it to the two general problems considered in the thesis, equations of motion are developed in Chapter 2. Chapter 2 also introduces the methods of describing rotation with quaternions and presents necessary elements of quaternion algebra. Equations of motion are developed which describe the attitude dynamics of a spacecraft free to rotate without the influence of feedback torques. Equations are subsequently developed in which the gravity gradient torque is included. The chapter includes a brief discussion of the feedback linearization problem and a method of solving a simplified version of the problem. The chapter concludes with a derivation of transformations that linearize the two problems. The first *torque-free* system of equations is used in Chapter 3 to study the steering problem. The augmented set is applied in Chapter 4 in which an optimal momentum management problem is stated, and a technique for obtaining solutions is developed.

Chapter 2

Derivation of Equations of Motion

In this chapter, equations of motion are derived for a rigid, rotating spacecraft equipped with CMGs. The attitude equations are written in terms of the quaternion relating a reference frame, fixed to the spacecraft, to an inertial frame. The next section contains an overview of relevant facts regarding quaternions and quaternion algebra. Kinematic models are then derived for single- and double-gimbal CMGs, and appropriate reference frames are defined for expressing the momentum of a CMG in a spacecraft fixed coordinate system. The reference frames depend on the adopted mounting configuration and the type of CMGs employed. In the third section, equations are derived that describe the attitude dynamics of a spacecraft undergoing torque-free rotation and subject to an internal transfer of momentum. The section concludes with the development of a system of first-order differential equations that fully describe the dynamics of the combined spacecraft/CMG system. In the next section, equations of motion are derived that include the effects of gravity gradient torques acting with a general external torque applied to the vehicle. In the final section, a brief introduction to the linearization problem is given, and linearizing transformations are developed for the two spacecraft models that are derived in this chapter.

2.1 Quaternions and Quaternion Algebra

There are a number of ways to characterize finite rotations of rigid bodies - or reference frames - in orthogonal coordinate systems. For example, Euler angles and direction cosines are common choices of coordinates in many applications. It is useful to have a representation for rotation that has an algebraic operation corresponding to successive rotations of a reference frame. Quaternions were invented by Sir William Rowan Hamilton in 1843 to extend ordinary 3-dimensional vector algebra to include operations of multiplication and division. One important use of quaternions is to transform vectors expressed in one reference frame into the basis of another reference frame. The algebra-like structure of quaternion arithmetic makes it easy to perform successive transformations between reference frames. A brief introduction to the relevant aspects of quaternion algebra is presented in this section; see [28] for additional details.

2.1.1 Definition and Elementary Operations

A theorem proven by Euler states that any sequence of rotations of a rigid body - or of a coordinate frame - is equivalent to a single rotation about some axis. Quaternions are essentially mathematical tools that implement this theorem. Let A and B denote a pair of right-handed reference frames, and assume that B has been rotated with respect to A through an angle θ about an axis defined by the unit vector $\hat{u}$ expressed in A. The quaternion of reference frame B with respect to

A is defined to be the ordered quadruple

$$q_A^B = \left\{ \begin{array}{c} q_0 \\ \bar{q} \end{array} \right\} = \left\{ \begin{array}{c} q_0 \\ --- \\ q_1 \\ q_2 \\ q_3 \end{array} \right\} \triangleq \left\{ \begin{array}{c} \cos(\theta/2) \\ ----------- \\ \hat{u}\sin(\theta/2) \end{array} \right\} \tag{2.1}$$

Quaternions are usually regarded as a quadruple composed of a scalar component, q_0, and a vector component, $\bar{q}$, consisting of the last three elements of the quaternion. Definition 2.1 is more restrictive than typically employed in general discussions [28]. This particular definition is useful when quaternions are only being used to transform vectors between reference frames.

Quaternion addition and subtraction are the ordinary operations for ordered n-tuples of numbers. Quaternion multiplication is defined by

$$p\,q \triangleq \begin{bmatrix} p_0\,q_0 - \bar{p} \cdot \bar{q} \\ p_0\,\bar{q} + q_0\,\bar{p} + \bar{p} \times \bar{q} \end{bmatrix} \tag{2.2}$$

where $\cdot$ and $\times$ denote the ordinary vector dot and cross products, respectively. The multiplication operation defined by (2.2) has the following interpretation. If q_A^B denotes the quaternion of frame B with respect to A, and q_B^C denotes the quaternion of C with respect to B, then their product

$$q_A^C = p_B^C\,q_A^B \tag{2.3}$$

is the quaternion that defines the orientation of frame C with respect to frame A. The definition of quaternion multiplication can be expressed in vector-matrix notation to yield

$$p\,q = \begin{bmatrix} p_0 & -p_1 & -p_2 & -p_3 \\ p_1 & p_0 & -p_3 & p_2 \\ p_2 & p_3 & p_0 & -p_1 \\ p_3 & -p_2 & p_1 & p_0 \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix} \tag{2.4}$$

as the definition of quaternion multiplication. The conjugate of a quaternion q is defined to be

$$q^* \triangleq \begin{Bmatrix} q_0 \\ -\bar{q} \end{Bmatrix} \tag{2.5}$$

It can be shown that quaternions which perform transformations between orthogonal reference frames have the property that

$$q_A^B \left(q_A^B\right)^* = \left(q_A^B\right)^* q_A^B = \begin{Bmatrix} 1 \\ \bar{0} \end{Bmatrix} \triangleq i \tag{2.6}$$

Using the definition of quaternion multiplication, it can be shown that the quaternion i is the multiplicative identity for quaternion algebra. The implication of (2.6) is that the conjugate of a quaternion defined by (2.1) is in fact its inverse. This is analogous to a similar result for transformation matrices.

2.1.2 Coordinate Transformations with Quaternions

Quaternions can be used to transform vectors expressed in one reference frame into another coordinate frame. A vector x^A which is coordinatized in a reference frame A can be tranformed into another reference frame B by the sequence of quaternion products

$$x^B \;=\; q_A^{B*}\, x^A\, q_A^B \tag{2.7}$$

where the vectors x^A and x^B are expressed as quaternions whose scalar parts are zero. It is the simplicity of this expression which motivates the use of (2.1) as the working definition of a quaternion. The effect of this transformation is to conically rotate the vector x^A about an axis defined by $\overline{q}$, or equivalently $\widehat{u}$, through an angle θ. Another way of viewing this transformation, is to consider what an observer in reference frame B would see as his reference frame was rotated from a position initially aligned with A. To this observer, it would appear that the vector x^A was being rotated about the axis $-\widehat{u}$. From a computational point of view, the most efficient implementation of the coordinate transformation (2.7) of a vector using quaternions is

$$x^B = (2{q_0}^2 - 1)\, x^A + 2\, (\overline{q} \cdot x^A)\, \overline{q} - 2\, q_0\, (\overline{q} \times x^A) \tag{2.8}$$

Successive transformations of a vector between several reference frames can be computed by repeated application of (2.7)

$$x^C = q_B^{C*}\, (q_A^{B*}\, x^A\, q_A^B)\, q_B^C \tag{2.9}$$

Since it can be shown that $(pq)^* = q^* p^*$, this equation can be rewritten as

$$x^C = (q_A^B\, q_B^C)^*\, x^A\, (q_A^B\, q_B^C) = q_A^{C*}\, x^A\, q_A^C \tag{2.10}$$

This last equation reiterates the connection between multiplication of quaternions and rotations of orthogonal reference frames. The relationship between quaternions and the direction cosine matrix associated with a rigid rotation is given by

$$C(q) \triangleq \begin{bmatrix} 2(q_0^2+q_1^2)-1 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 2(q_0^2+q_2^2)-1 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 2(q_0^2+q_3^2)-1 \end{bmatrix} \tag{2.11}$$

where the fact that $q_0^2+q_1^2+q_2^2+q_3^2=1$ has been used to simplify the diagonal elements. Equivalently, a quaternion can be extracted from a direction cosine matrix, $C = [c_{ij}]$. The diagonal elements of (2.11) yield the system of equations

$$\begin{aligned} q_0^2 &= 1 + \mathrm{Trace}(C) \\ q_1^2 &= 1 + 2c_{11} - \mathrm{Trace}(C) \\ q_2^2 &= 1 + 2c_{22} - \mathrm{Trace}(C) \\ q_3^2 &= 1 + 2c_{33} - \mathrm{Trace}(C) \end{aligned} \tag{2.12}$$

and the off-diagonal elements yield

$$\begin{array}{rclcrcl} p_0p_1 &=& c_{23} - c_{32} & \qquad & p_2p_3 &=& c_{23} + c_{32} \\ p_0p_2 &=& c_{31} - c_{13} & & p_3p_1 &=& c_{31} + c_{13} \\ p_0p_3 &=& c_{12} - c_{21} & & p_1p_2 &=& c_{12} + c_{21} \end{array} \tag{2.13}$$

Choosing a nonzero q_i from (2.12) and taking its positive square root makes is possible to solve three of the equations in (2.13) for the remaining elements of the quaternion.

2.1.3 Derivative of a Quaternion

In the following sections, it will be necessary to compute the time rate of change of a quaternion due to the relative rotation of two reference frames. For example, if the quaternion transforms vectors in inertial coordinates into a spacecraft fixed frame, the quaternion will change with time as the spacecraft rotates. In this section, the derivative of a quaternion, q_A^B, is computed assuming reference frame B is rotating at a relative angular rate ω^B which has been coordinatized in the reference frame B.

In order to determine the derivative of a quaternion, it is necessary to evaluate the limit

$$\dot{q}_A^B(t) \triangleq \lim_{\Delta t \to 0} \frac{\Delta q_A^B}{\Delta t} = \lim_{\Delta t \to 0} \frac{q_A^B(t+\Delta t) - q_A^B(t)}{\Delta t} \tag{2.14}$$

After an interval Δt, the reference frame B will have rotated through an angle $\left|\omega^B\right| \Delta t$ to a new orientation B' about a unit vector $\hat{\omega}^B$ aligned with ω^B. The quaternion which performs this transformation is

$$q_B^{B'} = \left\{ \begin{array}{c} \cos(\left|\omega^B\right| \Delta t/2) \\ \\ \hat{\omega}^B \sin(\left|\omega^B\right| \Delta t/2) \end{array} \right\} \tag{2.15}$$

and the quaternion $q_A^B(t+\Delta t)$ is given by

$$q_A^B(t+\Delta t) = q_B^{B'}\, q_A^B \tag{2.16}$$

The limit (2.14) can now be evaluated from

$$\begin{aligned}
\dot{q} &= \lim_{\Delta t \to 0} \frac{q_B^{B'} q_A^B - q_A^B}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left\{ q_B^{B'} - i \right\} q_A^B \\
&= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left\{ \begin{array}{c} \cos(\left|\omega^B\right| \Delta t/2) - 1 \\ \\ \hat{\omega}^B \sin(\left|\omega^B\right| \Delta t/2) \end{array} \right\} q_A^B \\
&= \frac{1}{2} \left\{ \begin{array}{c} 0 \\ \omega^B \end{array} \right\} q_A^B
\end{aligned} \tag{2.17}$$

where i denotes the unit quaternion, (2.6).

It is also of interest to determine the relationship between the relative accelerations of two reference frames and the second time derivative of the quaternion relating them. Expanding (2.17) using the definition of quaternion multiplication, (2.2), and differentiating yields

$$\begin{aligned}
\ddot{q} &= \frac{1}{2}\frac{d}{dt} \left\{ \begin{array}{c} -\omega^B \cdot \bar{q} \\ \omega^B\, q_0 + \omega^B \times \bar{q} \end{array} \right\} \\
&= \frac{1}{2} \left\{ \begin{array}{c} -\dot{\omega}^B \cdot \bar{q} \\ \dot{\omega}^B\, q_0 + \dot{\omega}^B \times \bar{q} \end{array} \right\} + \frac{1}{2} \left\{ \begin{array}{c} -\omega^B \cdot \dot{\bar{q}} \\ \omega^B\, \dot{q}_0 + \omega^B \times \dot{\bar{q}} \end{array} \right\}
\end{aligned}$$

$$= \frac{1}{2}\left\{\begin{matrix} 0 \\ \dot{\omega}^B \end{matrix}\right\} q + \frac{1}{2}\left\{\begin{matrix} 0 \\ \omega^B \end{matrix}\right\} \dot{q}$$

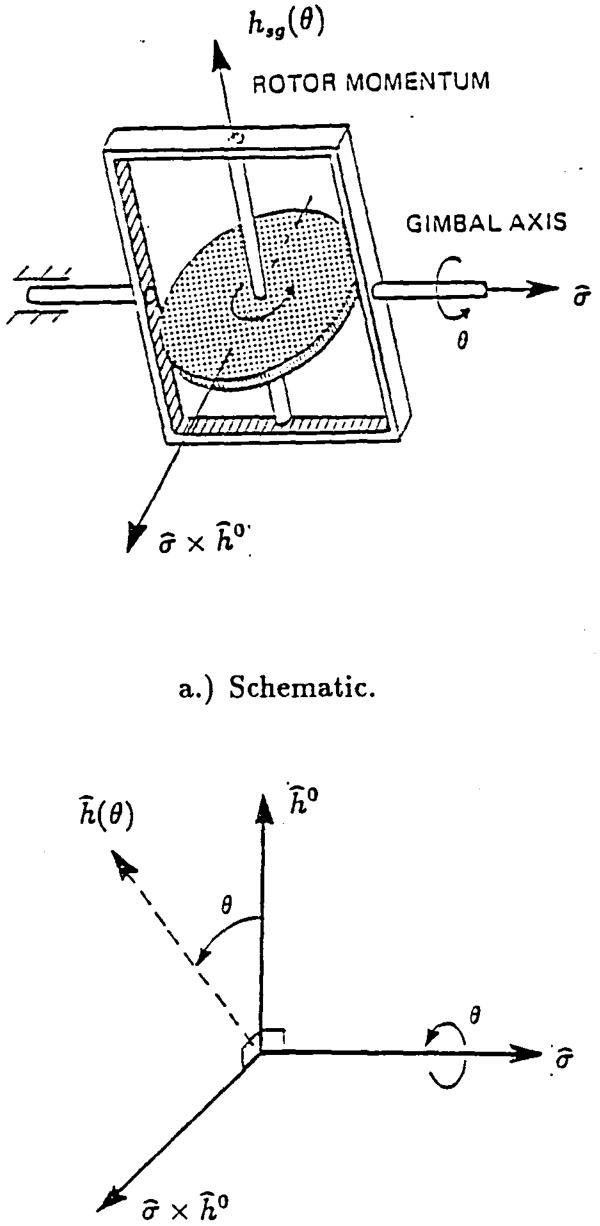
$$= \frac{1}{2}\left[\left\{\begin{matrix} 0 \\ \dot{\omega}^B \end{matrix}\right\} + \left\{\begin{matrix} 0 \\ \omega^B \end{matrix}\right\}^2\right] q \qquad (2.18)$$

where the subscript and superscript on the quaternion have been dropped to simplify the notation.

2.2 CMG Kinematics

In order to derive dynamical equations for a spacecraft equipped with CMGs, it is necessary to develop a kinematical model for CMGs. In this section, the equations relating CMG gimbal angles and CMG momentum are derived. Since the mechanical construction of single- and double-gimbal CMGs are quite different, the relationships are derived separately. In each case, a reference configuration is defined which is used as a coordinate system for the CMG rotor [29]. The kinematical constraints are then used to express the CMG momentum vector in this reference frame as a function of the gimbal angles.

Figure 2.1a shows a schematic drawing of a single-gimbal CMG. The unit vector $\widehat{\sigma}$ lies along the axis about which the CMG is rotated by the gimbal drive, and this vector is fixed with respect to the spacecraft. This axis is called the gimbal axis and the displacement of the rotor about this axis is denoted by θ. The unit vector $\widehat{h}^0$ corresponds to the orientation of of the CMG momentum vector at the reference configuration, $\theta = 0$. Together with the vector $\widehat{\sigma} \times \widehat{h}^0$, these two vectors form a right orthogonal reference frame. The orientation of the CMG rotor in this



a.) Schematic.

b.) Reference frames.

Figure 2.1: Schematic Drawing and Reference Frame for a Single-gimbal CMG

reference frame is

$$\widehat{h}(\theta) = \widehat{h}^0 \cos\theta + (\widehat{\sigma} \times \widehat{h}^0) \sin\theta \tag{2.19}$$

and the momentum of the CMG in the basis $(\widehat{\sigma}, \widehat{h}^0, \widehat{\sigma} \times \widehat{h}^0)$ is given by

$$h_{sg}(\theta) = h_{mag} \left[\widehat{h}^0 \cos\theta + (\widehat{\sigma} \times \widehat{h}^0) \sin\theta\right] \tag{2.20}$$

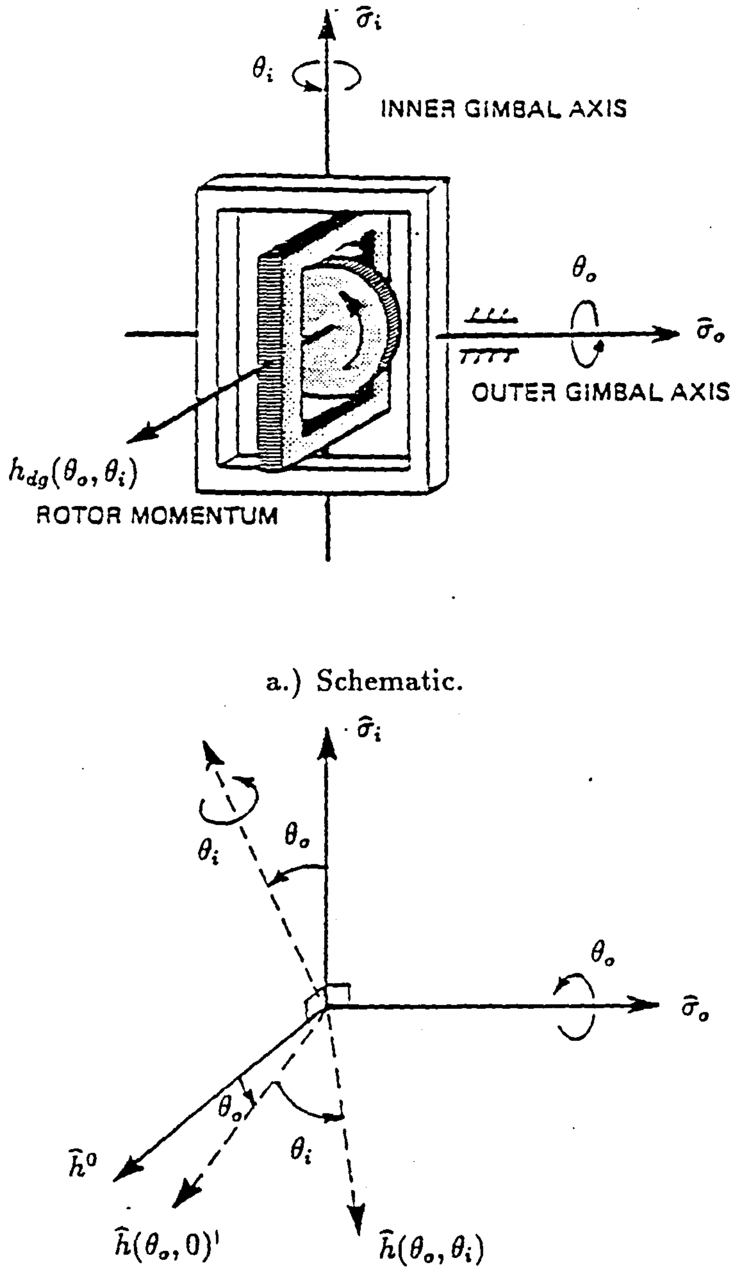
where h_{mag} is the magnitude of the momentum stored in the rotor.

The equations describing the kinematics of a double-gimbal CMG are derived using the same principles as above. The difference is that the rotor of a double-gimbal CMG is rotated about two axes and the orientation of the momentum vector depends on two parameters; the outer gimbal displacement, θ_o and the inner gimbal displacement, θ_i. Figure 2.2a shows a schematic of an Euler mounted double-gimbal CMG. In this figure $\widehat{\sigma}_o$ is a unit vector along the outer gimbal axis. This axis is fixed with respect to the spacecraft. The unit vector $\widehat{\sigma}_i$ lies along the reference inner gimbal axis corresponding to $\theta_o = 0$, and the third vector in the triad is defined to be orthogonal to $\widehat{\sigma}_o$ and $\widehat{\sigma}_o$ (i.e. $\widehat{h}^0 = \widehat{h}(\theta_o = 0, \theta_i = 0) = \widehat{\sigma}_o \times \widehat{\sigma}_i$). The CMG momentum vector can now be expressed in the basis $(\widehat{\sigma}_o, \widehat{\sigma}_i, \widehat{h}^0)$ shown in Figure 2.2b.

Holding the inner gimbal fixed at zero and rotating the CMG about the outer gimbal axis yields the expression

$$\widehat{h}(\theta_o, 0) = \widehat{h}^0 \cos\theta_o - \widehat{\sigma}_i \sin\theta_o \tag{2.21}$$

for the the unit vector along the rotor axis. Rotating the inner gimbal about the



a.) Schematic.

b.) Reference frame.

Figure 2.2: Schematic Drawing and Reference Frame for a Double-gimbal CMG

rotated inner gimbal axis, $\hat{\sigma}_i'$ yields the expression

$$\hat{h}(\theta_o, \theta_i) = \hat{h}(\theta_o, 0)\cos\theta_i + \hat{\sigma}_o \sin\theta_i \tag{2.22}$$

Substituting (2.21) into (2.22) and scaling by the magnitude of the rotor momentum yields

$$h_{dg}(\theta_o, \theta_i) = h_{mag}\left[(\hat{h}^0 \cos\theta_o - \hat{\sigma}_i \sin\theta_o)\cos\theta_i + \hat{\sigma}_o \sin\theta_i\right] \tag{2.23}$$

as the final expression for the momentum of a double-gimbal CMG.

Equations (2.20) and (2.23) express the momentum vector of a single-gimbal and a double-gimbal CMG, respectively. The momentum vectors are expressed in reference frames which are defined by the mounting of the individual CMGs. In general, these reference frames will be coordinatized in a common frame which is fixed with respect to a spaceraft. Given a group of CMGs, the equations derived in this section can be used to express the net momentum of the CMG system in a spacecraft-fixed reference frame as a sum over the set of CMGs:

$$h_{cmg}(\theta) = \sum_{i=1}^{N_{cmg}} h_i(\theta_i) \tag{2.24}$$

where $h_i \in \mathbf{R}^3$ are the momentum vectors of the individual CMGs, $\theta \in \mathbf{R}^m$ is the set of gimbal angles, and θ_i denotes the gimbal angle(s) associated with CMG i.

2.3 Equations of Motion for a Rigid Body: No External Torques

In this section, the equations of motion are derived for an isolated rigid spacecraft rotating in an inertial reference frame. The assumption of isolation implies that there are no external torques acting on the spacecraft and that the total momentum of the spacecraft is constant when expressed in inertial coordinates. Anticipating the need to include CMGs in the dynamical equations, the total momentum of the spacecraft is regarded as the sum of the momentum associated with the angular rate of the spacecraft and the momentum stored in a system of CMGs on board the vehicle. The equations are derived by a straight-forward application of conservation of momentum.

At this point it is useful to make a qualification on the definition of quaternion multiplication. Throughout this thesis, it will be necessary to compute quaternion products of only two types of quaternions: Quaternions associated with transformations between orthogonal reference frames, and quaternions associated with 3-vectors to be transformed. The scalar part of a quaternion representing a 3-vector is always zero. Since there cannot be more than three independent quantities which characterize rotations in a three-dimensional space, the four elements of the quaternion cannot be independent. A consequence of definition (2.1) is that quaternions which perform transformations between orthogonal reference frames have a Euclidean norm of one; given any three elements of such a quaternion, it is possible to compute the fourth. In particular, the expression

$$q_0 = \sqrt{1 - \sum_{i=1}^{3} q_i^2} \tag{2.25}$$

can be used to compute the scalar part of a quaternion from its vector part. From now on, it will be assumed that all quaternions are represented by triples of numbers; hence, the definition of quaternion multiplication (2.2) must be modified to use either zero or (2.25) when the scalar part of the quaternion is needed. With or without subscripts, the symbol q is reserved exclusively for quaternions that perform orthogonal transformations, and quaternions associated with 3-vectors will be denoted by $\left\{ \begin{matrix} 0 \\ x \end{matrix} \right\}$. The notation x will also be used when the interpretation is clear from the context. With these definitions, it is possible to derive a minimal system of first order equations to describe the system.

The attitude of the spacecraft is characterized by a quaternion which defines the relative orientation of a spacecraft-fixed reference frame, denoted B, with respect to an inertial frame, q_I. The total angular momentum of the spacecraft in inertial coordinates, h^I, can be transformed into the basis of the spacecraft-fixed frame using (2.7) to yield

$$h^B = q_I^* \, h^I \, q_I \tag{2.26}$$

Note that while h^I is a constant because there are no external torques, h^B changes in time as the spacecraft rotates due to the evolution of the quaternion, q_I. The momentum, h^B, can be expressed in the spacecraft reference frame as

$$h^B = h_{cmg}(\theta) + h_{spacecraft} = h_{cmg}(\theta) + I\omega^B \tag{2.27}$$

where $h_{cmg}(\theta)$ denotes the net momentum stored in the system of CMGs, ω denotes the spacecraft's angular rate, and I denotes the inertia tensor of the vehicle. Both $h_{cmg}(\theta)$, and ω are expressed in the body fixed coordinate frame B. The angular rate of the spacecraft can be determined from the total momentum and the momentum stored in the CMGs by solving (2.27) for ω

$$\omega^B = I^{-1} \left(h^B - h_{cmg}(\theta)\right) = I^{-1} \left(q_I^* \, h^I \, q_I - h_{cmg}(\theta)\right) \tag{2.28}$$

This expression can be substituted into (2.17) to obtain an expression for the derivative of the attitude quaternion in terms of the constant inertia matrix, the constant inertial angular momentum, and the attitude quaternion of the spacecraft.

$$\dot{q}_I = \frac{1}{2} \left\{ \begin{array}{c} 0 \\ I^{-1} \left(q_I^* \, h^I \, q_I - h_{cmg}(\theta)\right) \end{array} \right\} q_I \tag{2.29}$$

In order to complete the derivation of the equations for rotation in the absence of external torques, it is necessary to consider the control or input variable. CMG steering algorithms usually choose the set of gimbal rates as the control variable. Since the instantaneous torque produced by a group of CMGs is a function of the gimbal rates only, the gimbal rates are the only choice available when the steering law is required to produce specified torques. The gimbal rates are also useful when it is desirable to include physical limits on the individual gimbal drives. In some circumstances, it may be more realistic to consider torque limits on the gimbal drive motors, especially when double-gimbal CMGs are being used. With single-gimbal CMGs, the torque applied to the vehicle is orthogonal to the axis of rotation of the rotor. With double-gimbal CMGs, motion of one gimbal can produce a torque along the axis of the other. The need to back-drive the gimbals can impose restrictions that can not be accounted for without developing a higher-order model. In this discussion, it is assumed that that the gimbal rates - as expressed in the individual

CMG frames - can be commanded as the control variables. This assumption is consistent with conventional assumptions and does not result from a limitation imposed by the feedback linearization theory that will be applied to the problem. It may be desirable in future efforts to examine the effects resulting from including the dynamics of the gimbal drive motors in the model. These phenomena are not significant at the low torque and bandwidth levels encountered in this study and are not incorporated.

Under the stated assumptions, the dynamics of the combined spacecraft/CMG system are described by

$$\begin{aligned} \dot{\theta} &= u \\ \dot{q}_I &= \frac{1}{2} \left\{ \begin{array}{c} 0 \\ I^{-1} \left[q_I^* \, h^I \, q_I - h_{cmg}(\theta) \right] \end{array} \right\} q_I \end{aligned} \tag{2.30}$$

where $\theta, u \in \mathbf{R}^m$. These $m+3$ equations describe the evolution of the state variables for this system of equations: The three elements of the attitude quaternion, q_I, and the m gimbal angles, θ.

2.4 Equations of Motion for a Rigid Body: External Torques

In this section, we drop the assumption that the spacecraft is an isolated rigid body. The external torques that act on a space vehicle are generally small; however, the effects of these torques can be significant over long periods of time. The two principal sources of torques on orbiting spacecraft are due to gravity-gradient and aerodynamic effects. The attitude of the spacecraft will be specified with respect

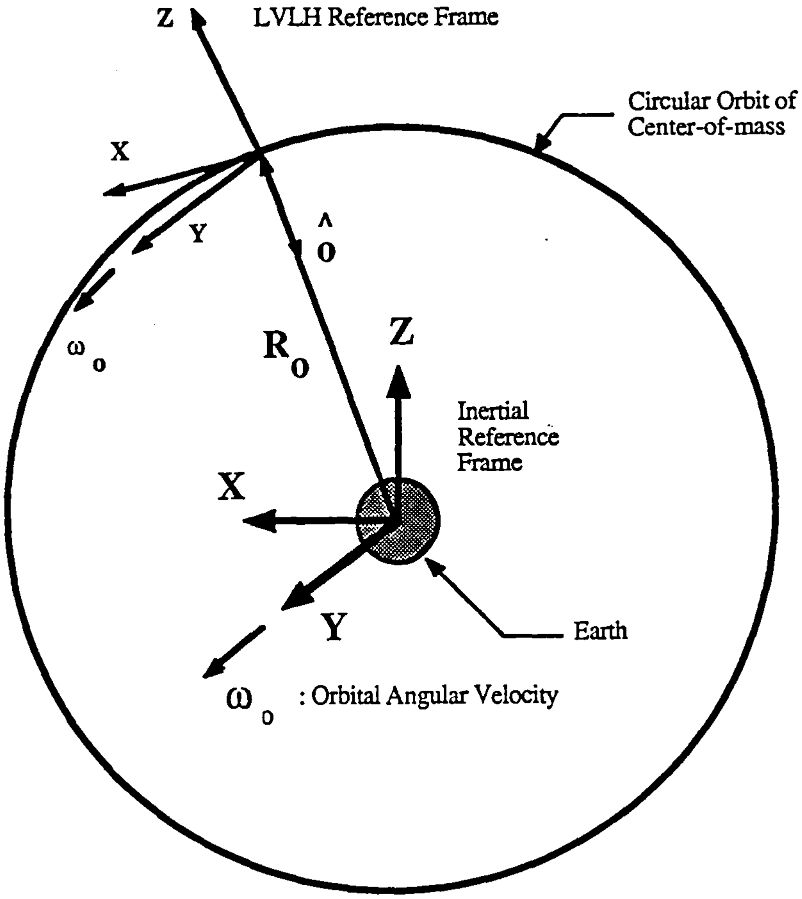


Figure 2.3: Orientation of LVLH Reference Frame

to a *local vertical/local horizontal* (LVLH) reference frame in a circular orbit about the Earth. The origin of this reference frame is fixed to the center-of-mass of the spacecraft. The LVLH reference frame is defined as shown in Figure 2.3. The x-component of the triad is aligned along the velocity vector, the y-component is aligned with the angular velocity vector of the orbit, ω_o, and the z-component is aligned with the radius vector of the orbit. With this definition, the LVLH reference frame defines a plane which is locally *horizontal* with respect to the Earth and a direction which is locally *vertical.* The orientation of the body frame with respect to the LVLH reference frame will be denoted by the quaternion q_L.

In order to compute the derivative of the attitude quaternion, it is necessary to determine the angular velocity of the vehicle relative to the LVLH frame. Assuming that the net angular momentum is known, (2.28) can be used to determine the inertial angular rate of the vehicle in body coordinates, ω^B. The angular rate of the spacecraft with respect to the LVLH frame, ω^L, can be computed in LVLH coordinates by expressing ω^B in the LVLH basis and subtracting the angular rate of the orbit, ω_o:

$$\begin{aligned} \omega^L &= q_L \, \omega^B \, q_L^* - \omega_o \\ &= q_L \left\{ I^{-1} \left(q_L^* \, h^L \, q_L - h_{cmg} \right) \right\} q_L^* \; - \; \omega_o \end{aligned} \tag{2.31}$$

where h^L denotes the net momentum expressed in the LVLH basis.

The gravity-gradient torque arises because the forces on the orbiting vehicle due to gravity and centripetal acceleration vary as different functions of the orbital radius, R_o. In particular, the centripetal force is proportional to R_o^2, while the gravity force is proportional to R_o^{-2}. If a spacecraft is orbiting the Earth in a circular orbit, the radial force due to centripetal acceleration and gravity exactly balance at the center-of-mass. For a small element of mass orbiting at a distance greater then the center-of-mass, the centripetal force is larger than the gravity force and there is a net force in the outward direction. The opposite is true for an element of mass orbiting at a lower altitude. For a long cylindrical object in a circular orbit, the gravity gradient torque will be zero if the longitudinal axis is aligned with the local vertical. The expression for the gravity gradient torque in a body-fixed reference frame in orbit about a spherical primary body (i.e. an ideal Earth) is [30]

$$\tau_{gg} = \frac{3\mu}{R_o^3}\,\hat{o} \times I\hat{o} \tag{2.32}$$

where $\mu = 3.986012 \times 10^{14}\ m/sec.$ is a constant and $\hat{o}$ denotes a unit vector pointing to the center-of-mass of the primary.

Since external torques act on the spacecraft, it is no longer valid to assume that angular momentum is a constant. Newton's Third Law can be applied to obtain an expression for the rate-of-change of momentum caused by the external torques. The gravity gradient torque (2.32) can be expressed in LVLH coordinates by substituting

$$\hat{o}^L = \left\{ \begin{array}{c} 0 \\ 0 \\ -1 \end{array} \right\} \qquad \text{and} \qquad I^L = C^T I C \tag{2.33}$$

where $\hat{o}^L$ denotes the vector to the center of the orbit, I^L denotes the vehicle inertia tensor in the LVLH basis, and C is the transformation matrix (2.11). Using this substitution, the net external torque acting on the vehicle in the LVLH basis is

$$\tau^L = \frac{3\mu}{R_o^3} \begin{bmatrix} -I_{23}^L \\ I_{13}^L \\ 0 \end{bmatrix} + q_L\, \tau_{ext}\, q_L^* \tag{2.34}$$

here τ_{ext} denotes all other external torques on the vehicle (e.g. aerodynamic, crew push-off, etc.) and is expressed in body coordinates.

As defined in the previous section, the gimbal rates will be considered the input variable. A system of first order equations can now be derived using (2.31) with (2.17), and using (2.34) with the fact that the apparent rate of the inertial reference frame with respect to the LVLH frame is $-w_o$, to yield:

$$\begin{aligned}
\dot{\theta} &= u \\
\dot{h}^L &= -\omega_o \times h^L + \frac{3\mu}{R_o^3}\begin{bmatrix} -I_{23}^L \\ I_{13}^L \\ 0 \end{bmatrix} + q_L\,\tau_{ext}\,q_L^* \\
\dot{q}_L &= \frac{1}{2}\left\{ \begin{array}{c} 0 \\ I^{-1}\,\left[q_L^*\,h^L\,q_L - h_{cmg}(\theta)\right] \end{array} \right\} q_L
\end{aligned} \tag{2.35}$$

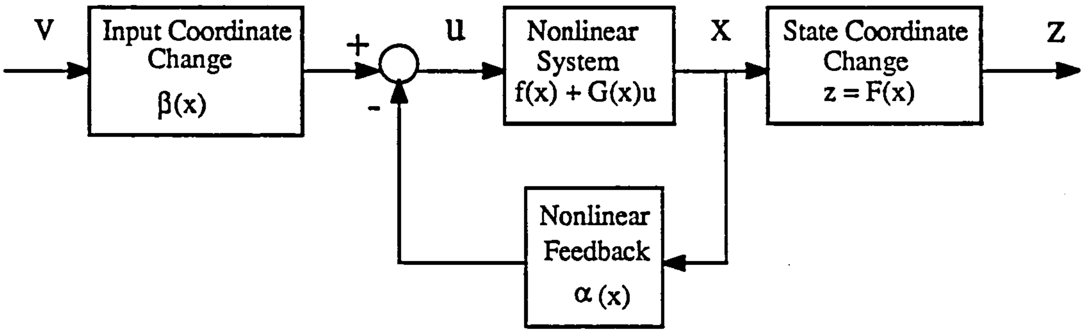
These ten equations describe the time-evolution of the attitude quaternion, q_L, the net momentum in LVLH, h^L, and the gimbal angles, θ.

2.5 Linearization of the Two Spacecraft Models

Beginning with the work of Krener [31], many researchers have investigated the conditions under which nonlinear dynamical systems can be transformed locally into linear, controllable systems. The transformations are local in the sense that they exist at all states sufficiently close to a nominal state. Depending on the specific types of transformations being allowed, different classes of systems can be linearized. Unlike Taylor series approximations, the linearized model is an exact representation of the nonlinear system.

Krener introduced and solved the linearization problem when the allowed transformation was only a local change of coordinates in the state space [31]. This *state-space linearization* problem was generalized to allow a linear change of input coordinates in [32]. The authors of [19] found a class of systems which can be transformed to a linear system using state dependent feedback, linear coordinate changes in the input, and nonlinear state coordinate changes. Figure 2.4 illustrates how the feedback and coordinate changes define a transformation which linearizes the nonlinear dynamics of a system. The figure shows that the transformation is applied externally to a nonlinear system; in theory, these transformations could be implemented directly in hardware and/or software for application to a physical system.

Since the foundations of this problem were established, a number of authors have extended it to include a wide variety of generalizations. The problem of transforming nonlinear systems with specified outputs using state coordinate changes is solved in [33], and a class of systems which are partially linearizable is studied in [34]. The problem of globally linearizing scalar input systems is solved in [35]. These ideas



$$\dot{x} = f(x) + G(x)\,u \xleftrightarrow[z = F(x)]{u = \alpha(x) + \beta(x)\,v} \dot{z} = A\,z + B\,v$$

Figure 2.4: Implementation of a Linearizing Feedback Transformation.

have also been extended to the dual problem of transforming nonlinear systems with output measurement into systems for which linear, spectrally assignable state estimators can be built [36].

Rather than treat the general feedback linearization problem here, a simpler version of the problem will be defined and solved. The general problem along with a detailed background discussion can be found in Appendix B. The problem discussed here is the feedback linearization problem *with output specified.* In this formulation of the linearization problem, all of the state variables of the linearized system are effectively specified at the outset. If the system is linearizable in terms of these coordinates, then the constructive procedure for obtaining the linearizing transformation is quite simple. The question of *existence* of such a set of state variables is far more delicate, and is considered in detail in Appendix B. The procedure for solving the problem discussed here is then applied to linearize the two dynamical spacecraft models that have been derived in this chapter.

2.5.1 Feedback Linearization with Output Specified

In this section, linearizing transformations are obtained for nonlinear systems of the form

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x)u_i = f(x) + G(x)\,u && (2.36) \\ y &= h(x) && (2.37) \end{aligned}$$

where the nonlinear, vector valued output equation $y = h(x)$ has been added. It is desired to transform this equation so that it exhibits the linear behavior of the system shown in Figure 2.5. The system of decoupled strings of integrators shown

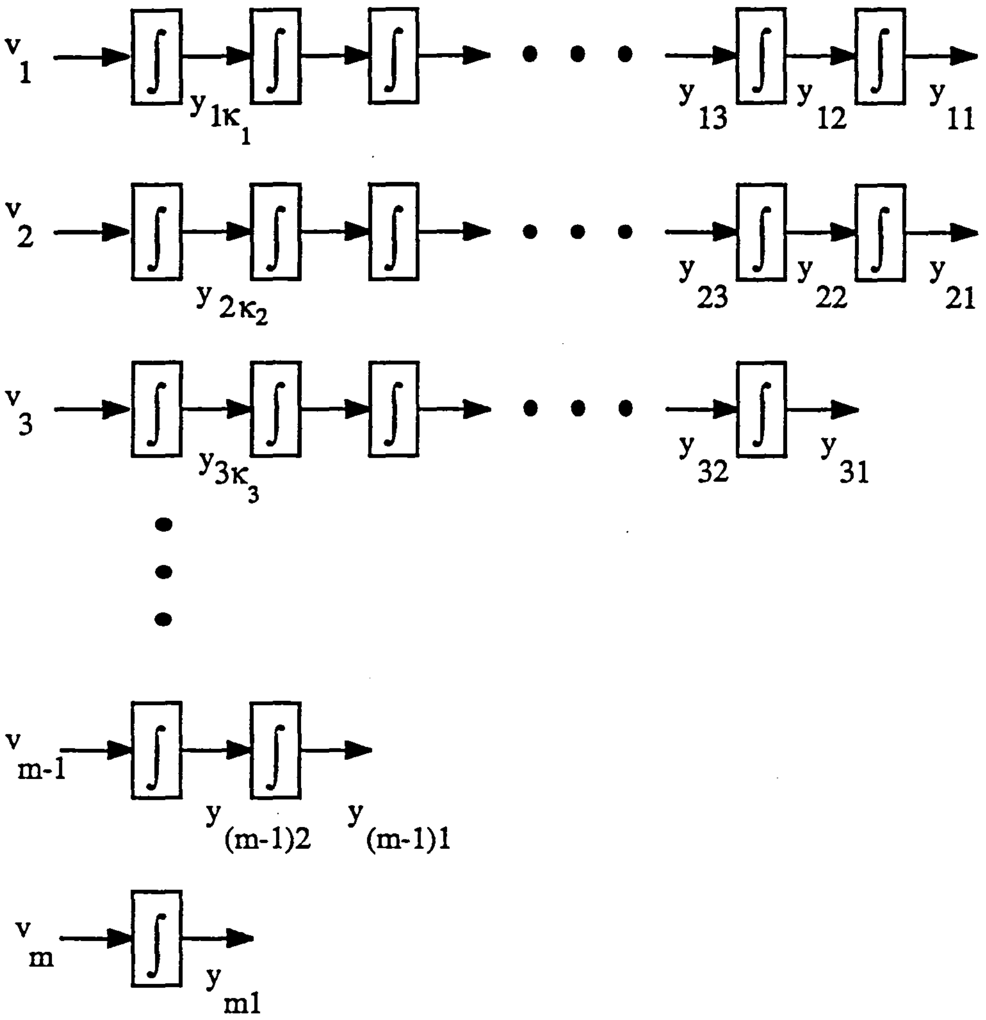


Figure 2.5: Brunovsy Form for the Linearization with Output.

in the figure is known as a *Brunovsky cononical form.* Suffice it to say that all controllable linear systems can be brought to this form, and if the linearizaton problem is solvable at all, the nonlinear system can be made to behave like the system in Figure 2.5 (refer to the appendix for details). The indices κ_i are the *Kronecker indices* of the linear system [37], and characterize the number of integrators in each row of the canonical form.

It is assumed that the output functions $y_i = h_i(x)$ in (2.37) are to correspond the outputs of the strings of decoupled integrators in the Brunovsky form. If there is a transformation that maps the dynamics of the nonlinear system to the system of Figure 2.5, then the functions that map x to the state variables of the linear system are in a sense time derivatives of the functions $h_i(x)$. Assuming that the linearization problem is solvable with these outputs, then the coordinate transformation and the feedback that linearize the system can be obtained by differentiation. In the following paragraphs, the coordinate tranformation $z = F(x)$, and the feedback $u = \alpha(x) + \beta(x)v$ that transform the nonlinear equations (2.36) and (2.37) to the canonical form shown in Figure 2.5 are derived.

In order to simplify some of the following equations, the notation $\mathbf{L}_w\phi(x)$ is used to denote the *Lie derivative* of the scalar valued function $\phi : \mathbf{R}^n \longrightarrow \mathbf{R}$ with respect to the vector valued function $w(x)$. The Lie derivative is computed according to the rule

$$\mathbf{L}_w\phi(x) \triangleq \mathbf{D}\phi(x)\,w(x) = \sum_{i=1}^{n} \frac{\partial \phi_i(x)}{\partial x} w_i(x) \tag{2.38}$$

and is simply the directional derivative of the function $\phi(x)$, in the direction $w(x)$, at the point x.

Suppose the component function $h_1(x)$ in (2.37) corresponds to the state variable

y_{11} in Figure 2.5. One component of the coordinate transformation $F(x)$ is clearly the function $h_1(x)$. Since y_{12} is the time derivative of y_{11}, the function that maps to y_{12} in Figure 2.5 must be given by

$$y_{12} \;=\; \dot{y}_{11} \;=\; \mathbf{D}h_1(x)\,\dot{x} \;=\; \mathbf{L}_{\dot{x}}\,h_1(x) \tag{2.39}$$

Substituting the expression for $\dot{x}$ given by (2.36) yields

$$\mathbf{L}_{\dot{x}}h_1(x) \;=\; \mathbf{L}_{f+Gu}h_1(x) \;=\; \mathbf{L}_f h_1(x) \;+\; \mathbf{L}_{Gu}h_1(x) \tag{2.40}$$

It should be clear from the figure that y_{12} is functionally independent of the input to the linear system, v. Since u and v are related algebraically, it must also be true that y_{12} is independent of u, implying $\mathbf{L}_{Gu}h_1(x) = 0$ for all choices of u. This leads to the conclusion that $y_{12} = \mathbf{L}_f h_1(x)$, By succesive application of the Lie derivative operator $\mathbf{L}_f$ to the function $h_1(x)$, the value of each of the state variables in the first set of integrators can be obtained according to the rules

$$\begin{array}{rcll} y_{11}(x) & = & h_1(x) & \\ y_{12}(x) & \triangleq & \dot{y}_{11} & = \mathbf{L}_f h_1(x) \\ y_{13}(x) & \triangleq & \dot{y}_{12} & = \mathbf{L}_f \mathbf{L}_f h_1(x) \\ & \vdots & & \\ y_{1\kappa_i}(x) & \triangleq & \dot{y}_{i(\kappa_i-1)} & = \mathbf{L}_f^{\kappa_i-1} h_1(x) \end{array} \tag{2.41}$$

In a similar fashion, the functions that map the variable x to each of the y_{ij} in Figure 2.5 can be obtained. By ordering these functions in a vector of the form $z \triangleq [y_{11}, \ldots, y_{m,\kappa_m}]^T$, the mapping $z = F(x)$ can be constructed.

The state transformation was constructed by evaluating Lie derivatives with respect to $\dot{x}$ of the functions $h_i(x)$ up to order $\kappa_i - 1$. Inspection of Figure 2.5 indicates that evaluating the Lie derivative of the left-most state variable, $y_{i\kappa_i}$ of the i^{th} string of integrators will yield an expression for the i^{th} input to the linear system, $v_i = \mathbf{L}_{\dot{x}} y_{i\kappa_i}$. The manipulations yield a set of functions which are not independent of u

$$\begin{aligned} v_i = \mathbf{L}_{\dot{x}} y_{i\kappa_i}(x) &= \mathbf{L}_{f+Gu} \mathbf{L}_f^{\kappa_i - 1} h_i(x) \\ &= \mathbf{L}_f \mathbf{L}_f^{\kappa_i - 1} h_i(x) + \mathbf{L}_{Gu} \mathbf{L}_f^{\kappa_i - 1} h_i(x) \\ &= \mathbf{L}_f^{\kappa_i} h_i(x) + \textstyle\sum_{j=1}^{m} \mathbf{L}_{g_i u_i} \mathbf{L}_f^{\kappa_i - 1} h_i(x) \\ &= \mathbf{L}_f^{\kappa_i} h_i(x) + \sum_{j=1}^{m} \left[\sum_{k=1}^{m} \frac{\partial y_{i\kappa_i}}{\partial x_k} (g_{ki}) \right] u_i \end{aligned} \tag{2.42}$$

where g_{ki} denotes the k^{th} element of the column vector $g_i(x)$. The m expressions for the individual components v_i can be combined to obtain a single vector-matrix equation for the control vector v

$$v \triangleq w(x) + A(x)\, u \tag{2.43}$$

where

$$w(x) \triangleq \begin{bmatrix} \mathbf{L}_f^{\kappa_1} h_1(x) \\ \vdots \\ \mathbf{L}_f^{\kappa_m} y_{m1}(x) \end{bmatrix} \quad \text{and} \quad A(x) = [a_{ij}(x)] \triangleq \left[\sum_{k=1}^{m} \frac{\partial y_{i\kappa_i}}{\partial x_k}\, g_{kj}(x) \right] u \tag{2.44}$$

If the linearization problem is indeed solvable, then the matrix $A(x)$ is nonsingular and (2.43) can be solved for the u which produces a desired v. Defining

$$u = \alpha(x) + \beta(x)v \quad where \quad \begin{aligned} \alpha(x) &= -A^{-1}(x)\begin{bmatrix} \mathbf{L}_f^{\kappa_1} y_{11}(x) \\ \vdots \\ \mathbf{L}_f^{\kappa_m} y_{m1}(x) \end{bmatrix} \\ \beta(x) &= A^{-1}(x) \end{aligned} \tag{2.45}$$

the feedback linearization problem is solved by the state coordinate change $F(x)$, the state feedback $\alpha(x)$, and the linear change of coordinate in the input space $\beta(x)$.

From the preceeding discussion, it is clear that a solution to the feedback linearization problem can be obtained if the structure of the Brunovsky form in Figure 2.5 is known (characterized by its Kronecker indices), and a set of m functions y_{i1} are known which are the outputs of the strings of integrators in the figure. In the following two subsections, linearizing transformations are derived for the two models developed in the previous chapter. In both subsections, the Kronecker indices are given along with a set of functions which solve the linearization problem.

2.5.2 Linearization of the Spacecraft Model without External Torques

The spacecraft model (2.30) can be put in the standard form for the linearization problem by letting

$$f(x) \;=\; \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{2}\left\{ \begin{array}{c} 0 \\ I^{-1}\left[q_I^* h^I q_I - h_{cmg}(\theta)\right] \end{array} \right\} q_I \end{pmatrix} \quad \text{and} \quad g_i(x) \;=\; \begin{pmatrix} \delta_{1i} \\ \vdots \\ \delta_{mi} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.46}$$

with the nonlinear system to be linearized described by the vector fields $f, g_1, \ldots, g_m$. The first m elements of the state vector x are the gimbal angles, and the last three elements of x are the vector components of the attitude quaternion. Direct computation shows that the Kronecker indices of the spacecraft model (2.30) are

$$\kappa_i \;=\; \begin{cases} 2 & \text{for} \quad i = 1,2,3 \\ 1 & \text{for} \quad i = 4,\ldots,m \end{cases} \tag{2.47}$$

The Brunovsky form is a set of 3 double-integrators and $m-3$ single-integrators. The double-integrators correspond to the rotational dynamics of the spacecraft, and the single integrators are related to the rotation of gimbals. While the three double-integrators should have been expected from the physics, the number of single integrators may seem unusual. The reason for this occurence was alluded to in the introductory paragraphs. Once an attitude variable is chosen, it is necessary to include its time-derivative as a state variable. The derivative is algebraically related to the angular rate of the vehicle, and therefore, to its momentum. The fact that the total inertial momentum is fixed necessarily fixes the net momentum of the system of CMGs. Constraining the momentum of the CMG system implies a loss of 3 degrees-of-freedom; hence, the number of *gimbal-like* control variables in the linearized system is $m - 3$.

One choice of functions $y_{i1}(x)$ that solves the linearization problem is

$$\begin{aligned} y_{11}(x) &= q_1 \\ y_{21}(x) &= q_2 \\ y_{31}(x) &= q_3 \\ y_{i1}(x) &= \theta_i \text{ for } i \in I^* \end{aligned} \tag{2.48}$$

where I^* denotes an index set consisting of $m - 3$ integers from 1 through m. The three indices which are dropped must correspond to three gimbals which together provide three axis torque control of the spacecraft. The remaining state variables are computed by Lie differentiation

$$\begin{aligned} y_{i2}(x) &= \dot{y}_{i1}(x) \\ &= \mathbf{L}_f y_{i1}(x) \\ &= \sum_{j=m+1}^{n} f_j(x) \frac{\partial y_{i1}}{\partial x_j} \end{aligned} \tag{2.49}$$

for $i = 1, 2, 3$. The linear change of coordinates in the input space is computed from

(2.43) by defining

$$A(x) = \begin{bmatrix} \frac{\partial y_{12}}{\partial \theta_1} & \cdots & \frac{\partial y_{12}}{\partial \theta_m} \\ \frac{\partial y_{22}}{\partial \theta_1} & \cdots & \frac{\partial y_{22}}{\partial \theta_m} \\ \frac{\partial y_{32}}{\partial \theta_1} & \cdots & \frac{\partial y_{32}}{\partial \theta_m} \\ \delta_{i_1^* 1} & \cdots & \delta i_1^* m \\ \vdots & \vdots & \vdots \\ \delta_{i_{m-3}^* 1} & \cdots & \delta i_{m-3}^* m \end{bmatrix} \qquad (2.50)$$

where the notation i_j^* denotes the j^{th} element of the index set I^*. A second set of Lie derivatives is computed according to

$$\begin{aligned} w_i(x) &= \mathbf{L}_f^2 y_{i1}(x) \\ &= \mathbf{L}_f y_{i2}(x) \\ &= \sum_{j=m+1}^{n} f_j \frac{\partial \mathbf{L}_f y_{i2}}{\partial x_j} \end{aligned} \qquad (2.51)$$

for $i = 1, 2, 3$. The linearizing feedback and input coordinate change are given by

$$\alpha(x) = -A^{-1}(x) w(x) \quad \text{and} \quad \beta(x) = A^{-1}(x) \qquad (2.52)$$

The state variables of the linearized system are the vector component of the attitude quaternion, the derivative of the attitude quaternion, and a subset of the available gimbal angles. The control variables associated with the linear system are the second derivative of the attitude quaternion and a subset of the available gimbal rates. In terms of the notation of the previous section, (z, v) are given by

$$z \triangleq \begin{pmatrix} q \\ \dot{q} \\ \theta^* \end{pmatrix} \quad \text{and} \quad v \triangleq \begin{pmatrix} \ddot{q} \\ \dot{\theta}^* \end{pmatrix} \tag{2.53}$$

This choice of coordinates is by no means unique; any independent set of attitude variables would work. In most attitude control problems, it is customary to think of applying torques to a vehicle to cause rotation. The fact that $\tau = I\dot{\omega}^B$ together with (2.18) provide the algebraic connection between commanding torques or accelerations, and commanding a quaternion second derivative for the purpose of control.

2.5.3 Linearization of the Spacecraft Model with External Torques

The spacecraft model that includes gravity gradient torques can be put into the standard form by letting

$$f(x) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\omega_o h_3^L - \dfrac{3\mu}{R_0^3} I_{23}^L \\ \dfrac{3\mu}{R_0^3} I_{13}^L \\ \omega_o h_1^L \\ \dfrac{1}{2} \left\{ \begin{matrix} 0 \\ I^{-1} \left[q_L^* h^L q_L - h_{cmg}(\theta) \right] \end{matrix} \right\} q_I \end{pmatrix} \quad \text{and} \quad g_i(x) = \begin{pmatrix} \delta_{1i} \\ \vdots \\ \delta_{mi} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.54}$$

with the nonlinear system to be linearized described by the vector fields $f, g_1, \ldots, g_m$. The first m elements of the state vector x are the gimbal angles, the next three elements are the components of the total momentum in LVLH, and the last three elements of x are the vector components of the attitude quaternion. The term associated with the external torque, $q_L \tau_{ext} q_L^*$, has not been included in (2.54) because the linearization problem is solved for a set of coordinates in which this term appears linearly. In this way, the external torque appears in the linearized system as a disturbance, rather than being cancelled automatically by the feedback associated with the linearization.

The Kronecker indices for the spacecraft described by (2.54) are

$$\kappa_i = \begin{cases} 4 & \text{for} \quad i = 1 \\ 3 & \text{for} \quad i = 2 \\ 2 & \text{for} \quad i = 3 \\ 1 & \text{for} \quad i = 4, \ldots, m \end{cases} \tag{2.55}$$

The Kronecker indices of this system clearly indicate that the relationships between the state variables are more complicated for the case in which gravity gradient torques are included. As in the previous case, three gimbals must be used to provide attitude control of the spacecraft; therefore, only $m-3$ of the available gimbal angles can be associated with state variables in the linearized system. The triple-integrator and the first three integrators associated with κ_1 appear because the rate of change of momentum is related to the attitude. In other words, the momentum is given by an integral of a function of the attitude. There is a double-integrator in the linearized system because no attitude-dependent gravity gradient torque acts about the z direction in LVLH axis.

Before proceding with the linearization, it is usefull to exploit the linearity of the equation for the z-component of momentum in LVLH coordinates, namely

$$\dot{h}_3^L = \omega_o h_1^L \tag{2.56}$$

Dropping this equation from (2.54), one obtains a reduced system of equation with Kronecker indices $\kappa_i = \{3, 3, 2, 1, \ldots, 1\}$ that is linearizable in conjunction with (2.56). The reduced system of equations is linearizable with the set of functions

$$\begin{aligned} y_{11}(x) &= h_1^L \\ y_{21}(x) &= h_2^L \\ y_{31}(x) &= q_3 \\ y_{i1}(x) &= \theta_i \text{ for } i \in I^* \end{aligned} \tag{2.57}$$

The remaining state variables are computed by Lie differentiation to obtain the k^{th} time derivative

$$y_{ik}(x) = \mathbf{L}_f y_{i(k-1)}(x) = \mathbf{L}_f^{k-1} y_{i1}(x) \tag{2.58}$$

for $i = 1, 2, 3$ and $k \leq \kappa_i$. The linear change of coordinates in the input space is computed from (2.43) by defining

$$A(x) = \begin{bmatrix} \frac{\partial y_{13}}{\partial \theta_1} & \cdots & \frac{\partial y_{13}}{\partial \theta_m} \\ \frac{\partial y_{23}}{\partial \theta_1} & \cdots & \frac{\partial y_{23}}{\partial \theta_m} \\ \frac{\partial y_{32}}{\partial \theta_1} & \cdots & \frac{\partial y_{32}}{\partial \theta_m} \\ \delta_{i_1^* 1} & \cdots & \delta i_1^* m \\ \vdots & \vdots & \vdots \\ \delta_{i_{m-3}^* 1} & \cdots & \delta i_m^* m \end{bmatrix} \tag{2.59}$$

where the notation i_j^* denotes the j^{th} element of the index set I^*. A second set of Lie derivatives is computed according to

$$w_i(x) = \mathrm{L}_f^{\kappa_i} y_{i1}(x) \tag{2.60}$$

for $i = 1,2,3$ and $w_i(x) = 0$ for $4 \leq i \leq m$. The linearizing feedback and input coordinate change are given by

$$\alpha(x) = -A^{-1}(x)w(x) \quad \text{and} \quad \beta(x) = A^{-1}(x) \tag{2.61}$$

The state variables of the linearized system consist of the components of momentum in the x and y directions in LVLH coordinates, the first and second time derivatives of these momentum components, the momentum in the z direction in LVLH, the third component of the attitude quaternion, q_3, its time derivative, $\dot{q}_3$, and a subset of the available gimbal angles. The control variables are the third time derivatives of the momentum components h_1^L and h_2^L, the second time derivative of q_3, and a subset of the original gimbal rates. In terms of the notation from the previous section, (z, v) are given by

$$z \triangleq \begin{pmatrix} h_1^L \\ \dot{h}_1^L \\ \ddot{h}_1^L \\ h_2^L \\ \dot{h}_2^L \\ \ddot{h}_2^L \\ q_3 \\ \dot{q}_3 \\ \theta^* \end{pmatrix} \quad \text{and} \quad v \triangleq \begin{pmatrix} \dddot{h}_1^L \\ \dddot{h}_2^L \\ \ddot{q}_3 \\ \dot{\theta}^* \end{pmatrix} \tag{2.62}$$

Chapter 3

Application of Feedback Linearization to the Steering Problem

In the introduction to this thesis, CMGs were described as actuators capable of applying torques to a spacecraft for the purpose of attitude control. From the derivation of the equations of motion in Chapter 2, it can be seen that a system of CMGs is effectively a momentum *storage system.* By manipulating the rotors of individual gyroscopes - through the gimbal drives - it is possible to change the rotation rate of a spacecraft by changing the net momentum of its CMG system. Although other approaches exist [38] [4], spacecraft autopilots and control systems usually command their actuators to apply a desired torque, τ_{des}, to the vehicle to regulate its attitude. Using CMGs to generate the torques requires solving (1.2) for a set of gimbal rates which satisfies

$$\tau_c = \tau_{des} - \left(\omega^B \times h_{cmg}(\theta)\right) = \mathbf{D}[h_{cmg}(\theta)]\dot{\theta} \tag{3.1}$$

The problem of finding a set of gimbal rates is complicated by the fact that the number of available gimbals, m, is greater than the number required to provide three-axis control of the spacecraft. Also, there are *singular configurations* of the rotors for which the rank of the matrix $\mathbf{D}[h_{cmg}]$ is less than three, implying that it is not possible to apply torques to the vehicle in all directions. The problem of finding a set of gimbal rates $\dot{\theta}$ which satisfies (3.1) and moves the rotors away from singular configurations is called the CMG *steering problem.*

The CMG steering problem is related to the *inverse kinematic* problem encountered in robotics. The inverse kinematic problem for a robot involves finding trajectories that take an end-effector from an initial position and orientation to another specified configuration. The related CMG steering problem is to move individual rotors to obtain a change in total stored momentum. The relationships between the momentum and gimbal angles of a CMG system, (h, θ), and the end-effector position and joint angles of a robot, (x, q), are shown in Table 3.1 [2]. The robotics literature in particular contains a variety of approaches to this problem. Most approaches used in solving the CMG steering problem use the Moore-Penrose pseudoinverse solution of (3.1) to compute a set of gimbal rates that produces the desired torque, and then add a set of *null rates.* The null rates are chosen to redistribute the CMG rotors toward more desirable configurations while producing no torque. This approach is related to the *resolved motion method* encounted in the robotics literature [26]. In both the CMG and robot problems, a performance objective or optimality criterion is used to characterize the desirability of a given state or control. In the CMG problem, a set of gimbal rates is chosen by optimizing the projection of the rates onto the gradient of the objective function subject to the constraint that a specified torque is produced. This process satisfies the torque requirements while acting to avoid singular configurations.

In this chapter, an approach to CMG steering is developed that employs the linearizing transformation developed in Chapter 2 for the spacecraft without external torques. Although the system to be controlled has been made linear by the

MANIPULATOR		⇔	SG CMG SYSTEM	
Position	$\underline{x} = \underline{x}(\underline{q})$		Momentum	$\underline{h} = \underline{h}(\underline{\theta})$
Velocity	$\dot{\underline{x}} = J(\underline{q})\, \dot{\underline{q}}$		Torque	$\dot{\underline{h}} = J(\underline{\theta})\, \dot{\underline{\theta}}$
Acceleration	$\ddot{\underline{x}} = J(\underline{q})\, \ddot{\underline{q}} + \dot{J}(\underline{q})\, \dot{\underline{q}}$		Tor$\dot{\text{q}}$ue	$\ddot{\underline{h}} = J(\underline{\theta})\, \ddot{\underline{\theta}} + \dot{J}(\underline{\theta})\, \dot{\underline{\theta}}$
		Singularity		
No motion possible in a certain direction			No torque possible in a certain direction	

Table 3.1: Kinematical Quantities in CMG and Robotic Problems.

transformation (2.52), the steering problem has not been solved. The first difficulty in implementing a steering law is due to a property of the transformation itself: The attitude dynamics of the spacecraft are made linear by using three CMG gimbals to apply torques as needed to eliminate cross-axis coupling of the attitude dynamics and to decouple the spacecraft from the remaining gimbal drives. In effect, the redundant actuators have been decoupled from the spacecraft. The second difficulty is that there are no provisions for steering the CMGs away from undesirable configurations. The linearizing transformation solves a kinematical problem; it does not help with the qualitative requirements of the CMG steering problem. If it were not for the kinematic redundancy in the CMG system, the solution to the linearization problem would be adequate and unique; no steering problem would exist. This feedback approach has already been applied to a number of robotic problems which are not kinematically redundant [39],[24].

Note that the feedback part of the linearizing transformation (2.52) yields a parameterization of all sets of gimbal rates that produce a specified attitude acceleration. Once the first three elements of the control vector of the linear system have been fixed, the remaining elements can be selected arbitrarily without changing the commanded acceleration. The remaining elements are used to determine null motions. The parameterization effectively eliminates the constraint associated with the steering problem and reduces the number of variables that need to be considered in an optimization three. This parameterization can be applied to existing steering algorithms to simplify the numerical optimization problem and to reduce the number of required calculations.

A number of algorithms for CMG steering and robot control select a solution by using a generalized inverse of the form

$$\dot{\theta} = C^{\dagger}\tau_c + (I_m - C^{\dagger}C)w \tag{3.2}$$

where $C \triangleq \mathbf{D}[h_{cmg}]$ and $C^{\dagger} \triangleq C^T(CC^T)^{-1}$ is the Moore-Penrose pseudoinverse of C. This expression is similar to the parameterization obtained from the linearizing feedback in that the second term in (3.2) projects an arbitrary vector $w \in \mathbf{R}^m$ into the null space of C. Because w is an m-vector, the expression (3.2) does not give a minimal characterization of the null motions; a fixed w does not map to a unique set of gimbal rates. In addition, the number of arithmetic computations required to evaluate (3.2) is greater than that required to evaluate the linearizing transformation. The number of computations involved in calculating the matrix A in the linearizing transformation (2.52) and the matrix C are essentially the same. Inversion of the matrix A can can be reduced by column operations to inversion of a 3×3 matrix and multiplication of the inverse by a $3 \times (m-3)$ matrix; more computations are required just in the computation of $C^{\dagger}$.

In the next section, an approach to distributing control effort among a group of redundant actuators is presented. This approach uses the linearizing feedback to state the distribution problem as a normed approximation problem. The following section then develops this approach for solving the CMG steering problem. This strategy has a conceptual advantage over conventional approaches to CMG steering in that developing a steering law requires developing an algorithm for computing desired gimbal rates, rather than the standard method of developing a function whose gradient yields the desired rates. In addition to being more direct, such an algorithm can be programmed to recognize problem situations and take specific corrective measures. Finally, several examples are presented which show the performance of a steering law developed for the parallel mounted CMG configuration proposed for the NASA space station.

3.1 Formulation of the Normed Approximation Problem

In the proof of the feedback linearization theorem presented in Chapter 3, it was assumed that the dimension of the distribution $\mathcal{G}_0$ was equal to m, the number of inputs in (B.52). If there are redundant inputs, it is possible to find an invertible matrix function $H(x)$ such that

$$G(x)\,H(x) \;=\; \begin{bmatrix} \tilde{g}_1(x) & \cdots & \tilde{g}_{\tilde{m}}(x) & 0 & \cdots & 0 \end{bmatrix} \tag{3.3}$$

Under this coordinate change, it is only necessary to find a transformation that linearizes the system with inputs $\tilde{g}_i$. If the input transformation $(\tilde{\alpha},\tilde{\beta})$ solves the reduced problem, then the feedbacks given by

$$\alpha \;=\; H(x)\begin{bmatrix} \tilde{\alpha} \\ 0 \end{bmatrix} \quad \text{and} \quad \beta \;=\; H(x)\begin{bmatrix} \tilde{\beta} & 0 \\ 0 & I_{m-\tilde{m}} \end{bmatrix} \tag{3.4}$$

solve the original linearization problem.

Clearly, the columns of the matrix $H(x)$ are related to the right eigenvectors of the matrix $G(x)$. The solution to the linearization problem given by (3.3) may be of little practical value because the redundancy has been eliminated by the transformation H. The last $m - \tilde{m}$ control variables have their influence in the null space of $G(x)$ - they do not influence the nonlinear system being controlled.

It is quite possible that the linearization is such that certain actuators are used excessively for control purposes, while other actuators are not used at all. In effect, the burden of control is placed on only a few of the available actuators, and these actuators may not have the physical authority to obtain the desired results. As a result, linear control laws based solely on the linear system may not yield reasonable performance. In fact, a similar situation occurs with the two spacecraft models that have been linearized.

Redundancy appears in the spacecraft problems in a slightly different manner than has just been described. Even though only three CMG gimbals are required for control of the spacecraft, they appear in the linearized coordinates as independent inputs to the nonlinear system. In other words, the distribution $\mathcal{G}_0$ has maximal rank. The reason that the actuators are independent is that there are dynamics associated with the gimbal drives; however, the difficulties that have been described for systems with redundant inputs still occur. The feedback transformation uses three of the gimbals to control the spacecraft, and the remaining gimbal rates can be commanded arbitrarily. Generally, three of the gimbal drives will not have sufficient authority to perform attitude control functions alone. This is especially true if a number of other CMGs are being freely gimballed, or if the three gimbals are in a poor configuration for producing acceleration in the desired direction. Before control laws based on the linearizing transformations can be effectively applied, it is necessary to develop an approach that distributes the control effort required to produce a specified attitude acceleration among the available actuators.

Redundancy arises in the spacecraft problem because there is a subsystem in the model equations that is to be controlled. Here it is if interest to control the spacecraft; the fact that there are dynamics associated with the actuators is incidental in this case. For the purpose of control, it is desirable to have the spacecraft attitude dynamics appear independently of other system dynamics. More specifically, the linearization problem could have been stated with the additional requirement that $y_{i1}(x) = q_i$ for $i = 1, 2, 3$ in (B.89). This requirement necessarily implies that $v_i = \ddot{q}_i$ for $i = 1, 2, 3$. The linearization problem with this type of restriction on some of the state variables has been solved in [24]. The linearizing feedback for this class of problems has the form

$$u = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} + \begin{bmatrix} \beta_1 & \beta_2 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} \tag{3.5}$$

where the set of control variables (v_1, β_1) influence the subsystem whose dynamics are of particular interest to control. The remaining pair (v_2, β_2) can be associated with other subsystems, or with the dynamics of actuators that are not required for controlling the important subsystem.

A natural way to resolve the redundancy issue would be to choose a set of inputs that requires a minimal effort to produce a desired influence on the controlled subsystem. This would be adequate in the case where there are no dynamics associated with the actuators. In more general situations, however, it may be desirable to drive the state of the actuators to a particular value. Rather than simply minimizing the control effort, there may be a desired value u_d that the input should take. The problem of computing an input variable u that is close to the desired value can be stated as a normed approximation problem

$$\min_{u} \ \|u_d - u\| \tag{3.6}$$

subject to the constraint that the controlled subsystem is influenced in a specified way. Note that (3.5) provides a way to eliminate the constraint in the above problem. The variables v_1 were chosen because they are time-derivatives of system parameters which are to be controlled. Choosing v_1 appropriately yields a set of inputs to the nonlinear system that produces the desired result.

Letting $a \triangleq u_d - \alpha - \beta_1 v_1$, the contrained optimization problem can be restated as an unconstrained optimization in terms of the variable v_2

$$\min_{v_2} \ \|a - \beta_2 v_2\| \tag{3.7}$$

From a practical point of view, it is useful to impose limits on the individual inputs of the form

$$-u^i_{peak} \ \leq \ u_i \ \leq \ u^i_{peak} \tag{3.8}$$

where u^i_{peak} is a physical limitation associated with device i. Fast numerical methods exist for solving this problem in three important norms: The 1-norm defined by

$$\|x\|_1 \ \triangleq \ \sum_{i=1}^{n} |x_i| \tag{3.9}$$

the 2-norm is the usual Euclidean norm defined by

$$\|x\|_2 \ \triangleq \ \left\{ \sum_{i=1}^{n} |x_i|^2 \right\}^{\frac{1}{2}} \tag{3.10}$$

and the ∞-norm is defined to be

$$\|x\|_\infty \triangleq \max_{x_i} |x_i| \tag{3.11}$$

In the following three subsections, the 1-norm and ∞-norm problems are formulated as linear programming problems, and the 2-norm problem is stated as a quadratic programming problem (numerical methods are discussed later).

3.1.1 Solution of the 1-norm Approximation Problem

To formulate the $\|\cdot\|_1$ optimization problem, a new set of variables are introduced by defining the vector y as

$$y \triangleq u_d - \alpha - \beta_1 v_1 - \beta_2 v_2 \tag{3.12}$$

The new variables y_i represent the error in approximating the desired rates and are written in the form

$$y_i = y_i^+ - y_i^- \tag{3.13}$$

with $y_i^+, y_i^- \geq 0$ for $1 \leq i \leq m$. The linear program that solves the optimization problem is

$$\min_{v_2} \|u_d - u\|_1 = \min_{v_2, y^+, y^-} \sum_{i=1}^{m} y_i^+ + y_i^- \tag{3.14}$$

subject to the three sets of m constraints

$$\begin{aligned} \beta_2 v_2 \; + \; y^+ - y^- &= a \\ \alpha + \beta_1 v_1 + \beta_2 v_2 &\leq u_{peak} \\ -\alpha - \beta_1 v_1 - \beta_2 v_2 &\leq u_{peak} \end{aligned} \qquad (3.15)$$

In this formulation of the problem, the decision variables y_i^+, y_i^- are non-negative, and the $(v_2)_i$ are free.

3.1.2 Solution of the 2-norm Approximation Problem

The minimization in $\|\cdot\|_2$ can be formulated as a quadratic program in a straight forward manner by defining

$$\begin{aligned} \min_{v_2} \; \|u_d - u\|_2^2 &= \min_{v_2} \; (u_d - u)^T(u_d - u) \\ &= \min_{v_2} \; \left(a^T a - 2a^T \beta_2 v_2 + v_2^T \beta_2^T \beta_2 v_2\right) \end{aligned} \qquad (3.16)$$

subject to the two sets of m constraints

$$\begin{aligned} \alpha + \beta_1 v_1 + \beta_2 v_2 &\leq u_{peak} \\ -\alpha - \beta_1 v_1 - \beta_2 v_2 &\leq u_{peak} \end{aligned} \qquad (3.17)$$

In this formulation, the decision variables can be positive or negative, and the constraints impose limits on the allowed values.

3.1.3 Solution of the ∞-norm Approximation Problem

The $\|\cdot\|_\infty$ optimization problem is formulated by first introducing a new variable, λ, such that

$$-\lambda \leq a_i - (\beta_2 v_2)_i \leq \lambda \tag{3.18}$$

for $1 \leq i \leq m$. This is equivalent to

$$|a_i - (\beta_2 v_2)_i| \ \leq \ \lambda \quad \forall i \tag{3.19}$$

and λ is a bound on $\|\cdot\|_\infty$. The optimization problem is to minimize λ

$$\min_{v_2} \ \|u_d - u\|_\infty \ = \ \min_{v_2} \ \lambda \tag{3.20}$$

subject to the four sets of m constraints

$$\begin{array}{rcl} \lambda \ + \ \beta_2 v_2 & \geq & a \\ \lambda \ - \ \beta_2 v_2 & \geq & -a \\ -\beta_2 v_2 & \geq & -u_{peak} + \alpha + \beta_1 v_1 \\ \beta_2 v_2 & \geq & -u_{peak} - \alpha - \beta_1 v_1 \end{array} \tag{3.21}$$

Since revised simplex algorithms require more computational overhead to deal with a given number of constraints than a given number of decision variables, it is more efficient to solve the dual problem associated with (3.20,3.21). The dual linear program is stated in terms of the variable $w = [w_1, \ldots, w_{4m}]^T$ as

$$\max_{w} \sum_{i=1}^{m} a_i w_i \ - \ a_i w_{2i} \ + \ (-u_{peak} + \alpha + \beta_1 v_1)_i w_{3i} \ - \ (u_{peak} + \alpha + \beta_1 v_1)_i w_{4i} \tag{3.22}$$

subject to the $m+1$ constraints

$$\begin{aligned} \sum_{i=1}^{2m} w_i &\leq 1 \\ \sum_{j=1}^{m} [\beta_2]_{j,i} (w_j - w_{2j} + w_{3j} - w_{4j}) &= 0 \text{ for } 1 \leq i \leq m \end{aligned} \tag{3.23}$$

The problem is formulated such that all decision variables are non-negative, i.e. $w_i \geq 0 \; \forall \, i$.

3.2 A Steering Algorithm Based on Normed Approximation

In this section, an algorithm for computing a desired set of gimbal rates for a group of double-gimballed CMGs is proposed. The algorithm is based on computing a desired rotation vector for each CMG. The direction and magnitude of the rotation vector of a CMG rotor correspond to the axis and rate of rotation, respecively, of the rotor due to gimbal motion. The rotation vectors are chosen to move the CMG rotors to maximize the amount of momentum which can be projected instantaneously in any direction, to avoid line-up with the commanded torque vector, and to keep inner-gimbal angles small. The desired rotation vector for a CMG is computed as the sum of a set of rotation vectors each of which individually attempts to achieve one of these objectives. From the desired rotation vector, desired gimbal rates are derived for each CMG. This section concludes with a brief explanation of how the proposed steering law is incorporated into an attitude control system.

3.2.1 Computation of a Desired Set of Gimbal Rates

One method for maintaining good three-axis controllability with double-gimballed CMGs is to keep the angles between rotors as large as possible. Achieving this goal corresponds to maximizing the minimum torque that can be projected in any direction and corresponds to keeping three-axis controllability of the spacecraft high in all directions (this is not equivalent to *gain* maximization [7]). Since singular states of double-gimballed CMG systems are always associated with a mutual alignment of rotors, this is also equivalent to avoiding singularities. For each CMG, the term which achieves this in the rotation vector computation is

$$\sigma_1^i = \sum_{j \neq i} \frac{u_p}{\pi} \left[\cos^{-1}(\widehat{h}^i \cdot \widehat{h}^j) - \pi\right] \, unit(\widehat{h}^i \times \widehat{h}^j) \tag{3.24}$$

where u_p denotes the peak gimbal rate, $\widehat{h}^i$ denotes a unit vector along rotor axis i, and the cross product is normalized to have unit norm. Rotating each CMG in this way will move the rotors to a configuration in which the rotors are distributed symmetrically about the total CMG momentum vector. The desirability of such configurations has been pointed out in [4].

If a rotor is allowed to align with the commanded torque, then that rotor cannot contribute to the torque. The term which steers rotors away from such configurations is

$$\sigma_2^i = \begin{cases} 0 & if \ \cos^{-1}(\widehat{h}^i \cdot \widehat{\tau}) \geq k^\tau \\ \frac{u_p}{k^\tau} \left[\cos^{-1}(\widehat{h}^i \cdot \widehat{\tau}) - k^\tau\right] unit(\widehat{h}^i \times \widehat{\tau}) & if \ \cos^{-1}(\widehat{h}^i \cdot \widehat{\tau}) < k^\tau \end{cases} \tag{3.25}$$

where $\hat{\tau}$ denotes a unit vector in the direction of the requested torque, k^{τ} is a constant, and the cross product is normalized as before. Some investigators have suggested steering toward the negative of the desired torque to avoid singularities [8]. To avoid numerical difficulties, an arbitrary gimbal axis is used in the neighborhood of an exact line-up of vectors in (3.24) and (3.25).

To simplify notation in defining the next two contributors to the rotation vector, the function Φ is defined by

$$\Phi(a,b,c,d) \triangleq \begin{cases} 0 & \text{if } \left|\frac{a}{b}\right| \le c \\ \frac{a}{\|a\|}\frac{|a/b|-c}{d-c} & \text{if } c \le \left|\frac{a}{b}\right| \le d \\ 1 & \text{if } \left|\frac{a}{b}\right| \ge d \end{cases} \tag{3.26}$$

where (a,b,c,d) are constants with $|a| \le b$, and $0 \le c \le d \le b$.

It is desirable to keep the inner gimbal angles small, both to avoid potential hardware stops, and because the torque due to the outer gimbal rate is proportional to the *cosine* of the inner gimbal angle measured from the reference angle. A rotation vector which reduces the inner gimbal displacements is

$$\sigma_3^i = -u_{peak}\,\Phi(\theta_i, \theta_{stop}, k_1^{inner}, k_2^{inner})\,\hat{\sigma}_{inner} \tag{3.27}$$

where θ_i denotes the displacement of gimbal i, θ_{stop} is the displacement to a gimbal stop, k_1^{inner}, k_2^{inner} are constants, and $\hat{\sigma}_{inner}$ is a unit vector along the inner gimbal axis.

In studying several maneuvers, it was found necessary to include a term in the calculation of the desired rotation vector which would rotate the outer gimbal angle if the inner gimbal was being forced against a stop. Associated with the inner

3.2.2 Implementation of the Proposed Algorithm

In this section, an algorithm is described which implements a CMG steering law based on the normed approximation approach to redundancy management. The program has been implemented in FORTRAN on a VAX computer along with a rigid body simulation of a spacecraft. A flow chart of the steering algorithm is shown in Figure 3.1. The inputs are the current CMG gimbal angles, the spacecraft attitude quaternion, the derivative of the quaternion, and the second derivative of the quaternion, which is determined by an autopilot or control law. Since quaternions are algebraically related to any set of attitude variables, the proposed implementation is quite general.

The steering algorithm first tests the CMG gimbal angles to see if the CMG configuration is near a singular state. If the gain function is above a certain threshold, a routine that monitors the transformation is called. The routine MONITOR checks that the three gimbal rates that do not appear in the linearized control vector are capable of producing torques in all directions (these three gimbal rates are used to implement the linearizing feedback). Their ability to control the spacecraft is determined by the determinant of the matrix formed by the three columns of the matrix $\mathbf{D}[h_{cmg}]$ corresponding to these gimbal rates. When this determinant is small, the matrix that must be inverted during the computation of the linearizing transformation is nearly singular. If the determinant is below a certain threshold, a new combination of three gimbal angles is found which produces the largest determinant, and these three gimbal rates will not appear in the transformed control.

If the CMG rotors are near a singular configuration, then there will be no combination of three CMGs that can produce torques adequately in all directions. Most gimbal vector of each CMG is the rotation rate

$$\sigma_4^i \ = \ -u_{peak}\, \Phi(\theta_i, \theta_{stop}, k_1^{outer}, k_2^{outer})\, \widehat{\sigma}_{outer} \tag{3.28}$$

where θ_i and θ_{stop} are as before, k_1^{outer}, k_2^{outer} are constants, and $\widehat{\sigma}_{outer}$ is a unit vector along the outer gimbal axis. The initial sign of this rotation is chosen to move the rotor away from the inner gimbal rotation plane, and is not changed until σ_4^i is set to zero. The need for such a term will be explained in connection with one of the examples.

For each CMG, the desired rotation vector is found by taking a weighted combination of the four rotation vectors defined above.

$$\sigma_d^i = \sum_{j=1}^{4} k_j\, \sigma_j^i \tag{3.29}$$

From this composite rotation vector, it is possible to compute a unique set of gimbal rate commands that produce the desired rotor motion for the given CMG by solving the linear equation

$$\sigma_d^i \times \widehat{h}^i \ = \ u_d^{inner} \left(\widehat{\sigma}_{inner} \times \widehat{h}^i\right) \ + \ u_d^{outer} \left(\widehat{\sigma}_{outer} \times \widehat{h}^i\right) \tag{3.30}$$

for the desired inner and outer gimbal rates. Proceeding in this way for each CMG, a set of desired gimbal rates, u_d, is computed, and the normed approximation strategies of Section 4.1 can be used to calculate actual gimbal rates based on these desired rates. In the following subsection, an implementation of a steering law based on this desired rate computation and the approximation algorithms is described.

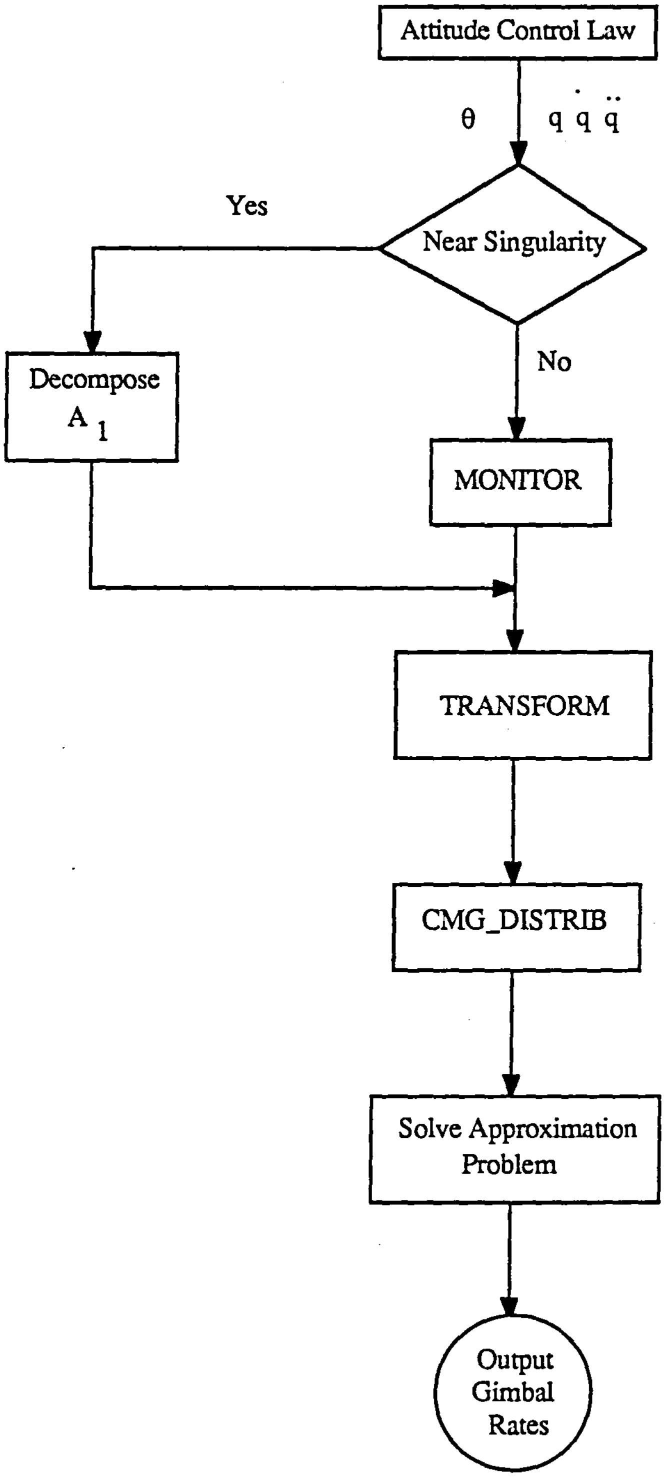


Figure 3.1: Flow Chart of Attitude Control Algorithm

steering algorithms do not consider this situation at all; even though no steering algorithm can really guarantee absolute singularity avoidance. The most common symptom of failure is termed *gimbal lock*, because the rates required to produce a desired torque become so large that no null motion is achieved. For CMG systems consisting of single-gimballed devices, there are singular states for which there is no possible escape through null motion, and the CMG system will lock. It is necessary that a steering algorithm continue to operate in a sensible manner under these circumstances.

Near singular states, the linearization problem is not solvable as stated, and the problem must be modified so that the algorithm can proceed. The numerical difficulty that arises in computing the transformations is due to the fact that the matrix $A(x)$ in (2.52) is nearly singular. By exchanging columns, this matrix can be brought into the form

$$A(x) = \begin{bmatrix} A_1(x) & A_2(x) \\ 0 & I_{m-3} \end{bmatrix} \tag{3.31}$$

where A_1 is a 3×3 matrix. The MONITOR routine attempts to pick a set of three gimbal angles that guarantee numerical invertibility of $A_1(x)$; however, near a singular state, no such set of gimbals exists. The matrix A_1 can be decomposed into

$$A_1 = V^T \Lambda V \tag{3.32}$$

where Λ is the diagonal matrix of eigenvalues of A_1, and V is the matrix whose columns are the orthonormalized eigenvectors. The problem of computing the linearizing feedback, (2.52) can be transformed to the form

$$\begin{bmatrix} V & 0 \\ 0 & I_{m-3} \end{bmatrix} v = \begin{Bmatrix} V\,\ddot{q} \\ \dot{\theta}^{*} \end{Bmatrix} = \begin{bmatrix} \Lambda V & V\,A_2 \\ 0 & I_{m-3} \end{bmatrix} \{w(x) + u\} \tag{3.33}$$

where the third row in the matrix on the right is nearly zero - assuming the eigenvalues decrease along the diagonal of Λ. Eliminating this row from the equation and increasing the dimension of the identity matrix by one yields the feedback transformation

$$u = \alpha(x) + \begin{bmatrix} \begin{bmatrix} -\lambda_1 v_1- \\ -\lambda_2 v_2- \end{bmatrix} & \begin{bmatrix} -v_1- \\ -v_2- \end{bmatrix} A_2 \\ 0 & I_{m-2} \end{bmatrix} \begin{Bmatrix} \begin{bmatrix} -v_1- \\ -v_2- \end{bmatrix} \ddot{q} \\ \dot{\theta}^{*} \end{Bmatrix} \tag{3.34}$$

Near a singular state, there is a direction about which the CMGs can produce no acceleration. This modification to the algorithm projects the desired acceleration onto the space orthogonal to the singular direction, and control of the spacecraft about the singular direction is replaced by an extra gimbal rate in the transformation. Indeed, the spacecraft will not be controlled in the singular direction; however, control will be maintained along the other axes. Controllability will be recovered in the singular direction after the singular orientation has been eliminated. Depending on the relative orientations of the singular direction and the torque direction, singularity may be a problem for only a very short time. This modification guarantees that the linearization problem is always solvable.

After the program resolves potential computational difficulties, the linearizing state and feedback transformations are evaluated in the routine TRANSFORMS.

The routine CMG_DISTRIB is then called to compute the desired set of gimbal rates based on the current state of the CMG system. The routine APPROX calls the appropriate subroutines for formulating and solving the desired normed approximation problem. The 1-norm and 2-norm approximation problems are solved using the subroutines DLPRS and QPROG, respectively, from release 10.0 of the IMSL subroutine library. The ∞-norm approximation problem is solved using subroutine ZX3LP from release 9.2 of the IMSL library. The output of the steering program is a set of gimbal rates, u which produce the commanded $\ddot{q}$.

This steering algorithm has been implemented as part of a simple simulation of a rigid spacecraft. The equations of motion are integrated using a fourth-order Runge-Kutta integration scheme. Before computing an integration step, a subroutine is called that can implement a variety of attitude control functions, and then the CMG steering algorithm is called. The chosen gimbal rates are then passed to a subroutine that executes a single integration step. The following section presents results from simulations using the program described in this subsection.

3.3 Simulation Examples

In this section, several simulations are presented to demonstrate the behavior of the proposed steering algorithm performing a variety of attitude control functions. The first three examples illustrate the characteristics of the gimbal angle trajectories generated by the three normed approximation algorithms. It will be shown that the 1-norm and ∞-norm algorithms occasionally exhibit undesirable oscillations in the present application. The 2-norm algorithm generally yields well behaved solutions and is used in the majority of the examples. The fourth example demonstrates the need for the fourth term in the rotation axis computation, (3.28). This is an example of a situation in which the steering algorithm must deal explicitly with a problem that conventional gradient based methods would fail to recognize. The next two examples show how the algorithm redistributes the CMG rotors when they are initialized in poor configurations. In one of the examples, the algorithm is initialized in a singular configuration. The final example demonstrates the response of the vehicle/CMG system when a large onboard disturbance is applied. The disturbance is large enough to rate saturate several of the CMGs, and the example demonstrates the advantage of including rate limits on the individual gimbals.

The inertia properties of the vehicle are listed in Table 3.2 and represent the rephased dual-keel space station, after assembly sequence flight 7 [40]. Attitude control torques are provided by four parallel-mounted double-gimballed CMGs. In the parallel mount configuration, the outer gimbals are aligned with the vehicle pitch axis and the inner gimbals are aligned along roll. Each CMG rotor possesses *5000 ft-lb-sec* of angular momentum and can be gimballed about inner and outer mounts at peak rates of *5 deg/sec*. Inner gimbal displacements are limited to less than *90 deg* and outer gimbals may rotate continuously. Figure 3.2 shows a schematic diagram of the vehicle along with the body reference frame, and the CMG mounting configuration is shown in Figure 3.3. The set of constants used by the steering law

Inertia Matrix *slug-ft*^{2}		
2.425×10^7	-1.917×10^5	3.060×10^5
-1.917×10^5	3.556×10^6	2.333×10^5
3.060×10^5	2.333×10^5	2.515×10^7

Table 3.2: Inertia Properties for Re-phased Dual-keel Space Station.

Relative Weights	Rotation Vector Constants
$k_1 = 0.2165$	
$k_2 = 1.0$	$k_\tau = 1.57$
$k_3 = 1.0$	$k_1^{inner} = 0.0$ $k_2^{inner} = 0.6$
$k_4 = 1.0$	$k_1^{outer} = 0.6$ $k_2^{outer} = 0.7$

Table 3.3: Constants Used in Rotation Vector Computation.

to compute the desired gimal rotation vectors are shown in Table 3.3. Although these simulations employ parallel CMG mounting, the structure of the steering law by no means requires this, and is ammenable to any mounting configuration.

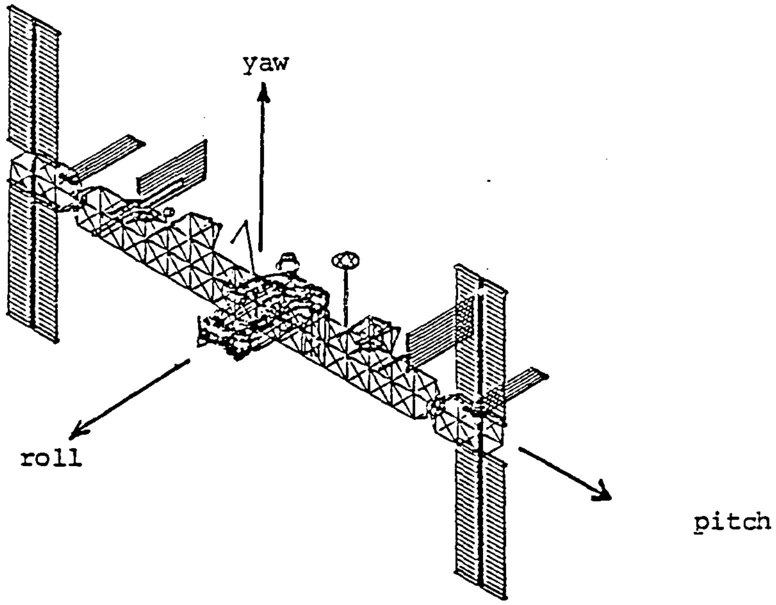


Figure 3.2: Schematic of Rephased Dual-Keel Space Station; Flight 7.

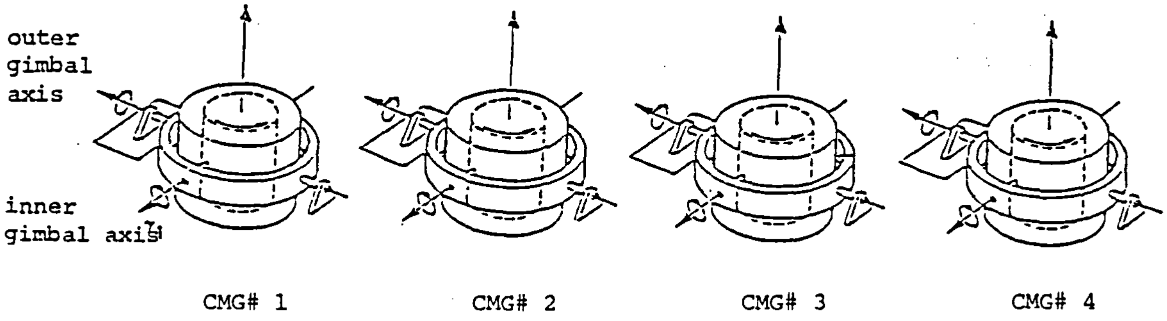


Figure 3.3: Reference Configuration for Four Parallel Mounted CMGs.

3.3.1 CMG Saturation Along Pitch

The first set of simulations illustrate the characteristics of the gimbal angle trajectories generated by the three normed approximation algorithms. The CMG system is initialized in a zero momentum state with gimbal angles (in degrees):

$$\begin{array}{rcl} \theta^1_{inner} = 5.73 & & \theta^1_{outer} = 0.0 \\ \theta^2_{inner} = -5.73 & \text{and} & \theta^2_{outer} = -90 \\ \theta^3_{inner} = -5.73 & & \theta^3_{outer} = 90 \\ \theta^4_{inner} = 5.73 & & \theta^4_{outer} = 180 \end{array} \qquad (3.35)$$

(Note that *0.1 rad.* = *5.73 deg.*). In each of the three cases in this subsection, the CMGs are commanded to produce a torque along the negative pitch axis of the vehicle. This causes the net CMG momentum to saturate along the positive pitch axis. Figure 3.4 shows the rotor trajectories projected onto a unit sphere for the case in which the 2-norm approximation is used to compute the gimbal rates, and the torque produced is a constant *-195 ft-lb.* In the starting configuration, the four momentum vectors (CMG rotors) nearly align with the positive and negative roll and yaw axes. The requested torque causes the rotors to project momentum along the positive pitch axis and the rotors align along this direction. The gimbal angles and gimbal rates are shown in Figure 3.5 and Figure 3.6, respectively. For most of the simulation, the desired inner gimbal rates are positive since saturating along the pitch axis requires pushing the inner gimbals against their (negative) stops. Since inner gimbals 2 and 3 have a negative dispacement initially, their commanded rates are smaller until all inner gimbal angles are approximately the same. After about *85 sec.*, the outer gimbals are forced to rotate by the fourth term in the rotation axis computation (3.28). This causes the rotors to saturate unevenly during the last *10 sec.*

In Figure 3.7 and Figure 3.8 the gimbal angles and rates are shown for the simulation in which the 1-norm approximation algorithm is used. Again, the commanded torque is *-195 ft-lb.* At the beginning of the simulation, the inner gimbal angles converge linearly until the angles are nearly the same. The inner gimbal angles then oscillate about a nominal trajectory. For a time, Figure 3.8 indicates that these oscillations are caused by the inner gimbal rates being switched between the positive and negative peak rates. This undesirable behavior occurs when the gimbal angles are near an optimal configuration, and is due to the fact that the approximation algorithm is computing an instantaneously optimal solution, not a continuous solution. A CMG configuration is referred to as optimal when the desired gimbal rates have almost no null component. In other words, redistribution at constant momentum can not improve the orientation, hence the only component in the calculation of the desired rates reflects a desire to transform momentum in the opposite direction required by the commanded torque. When the oscillations begin, one pair of gimbals is displaced slightly more toward the inner gimbal stop than the others, and hence, has a larger positive desired gimbal rate reflecting a desired to reduce the inner gimbal angle. The optimal solution attempts to approximate these rates accurately and allow the rates of the other pair of gimbals to be large and negative. During the next computational cycle, the situation is reversed, and the oscillations result.

A number of attempts were made to eliminate the oscillations that occur when the 1-norm approximation is used. The most successful used a first-order linear filter to to average the gimbal rates computed by the approximation algorithm to produce a smoother response. The filter suppresses the oscillations, but imposes an unacceptably low bandwidth when torque levels are high.

In Figure 3.9 and Figure 3.10, the gimbal angles and rates are shown for the simulation in which the ∞-norm algorithm is used. In order to avoid causing a gimbal to be pushed against a stop before the end of the simulation, the requested torque is *-190 ft-lb.* Like the previous two cases, the inner gimbals converge on the same value as they begin to saturate. The ∞-norm alogorithm allows significant motion of the outer gimbals, even though this motion cannot contribute to the torque applied to the spacecraft. The excessive outer gimbal motion occurs because the optimization is attempting to minimize the largest absolute error in approximating the desired gimbal rates. Nominally, the desired rates for the outer gimbals are zero, while the desired inner gimbal rates increase as the simulation proceeds. Since the error in approximating the desired inner gimbal rates becomes large, it is clear from the definition of $\|\cdot\|_\infty$ that the outer gimbals can be commanded to rotate without influencing the optimal value of the approximation. In fact, if it is possible to reduce the peak error by any amount using the outer gimbals, perhaps due to numerical round-off, they will be employed at the largest possible rate consistent with the norm of the optimal solution. Evidence of this can be seen by noting that the peak values of the outer gimbal rates approximately correspond to the value of the desired rate for the inner gimbal angles shown in Figure 3.11.

Both the 1-norm and ∞-norm approximation algorithms can exhibit undesirable oscillations under certain circumstances, and it is not desirable to command CMG gimbals to follow these kinds of rates. The chattering generally occurs around a near-optimal configuration, this undesirable behavior does not occur when there is an unambiguous policy to follow in redistributing the rotors. For example, Figure 3.7 shows the initial convergence to an optimal configuration is quite fast using the 1-norm approximation, indicating that both the 1-norm and ∞-norm approximations may have useful applications in certain circumstances. For example, the 1-norm is particularly relevant when the control is related to a displacement, and the ∞-norm solution is useful when it is desired to minimize the maximum effort required from each individual actuator. Because of its intrinsic smoothness, the remaining simulations were all performed using only the $\|\cdot\|_2$ optimal approximation.

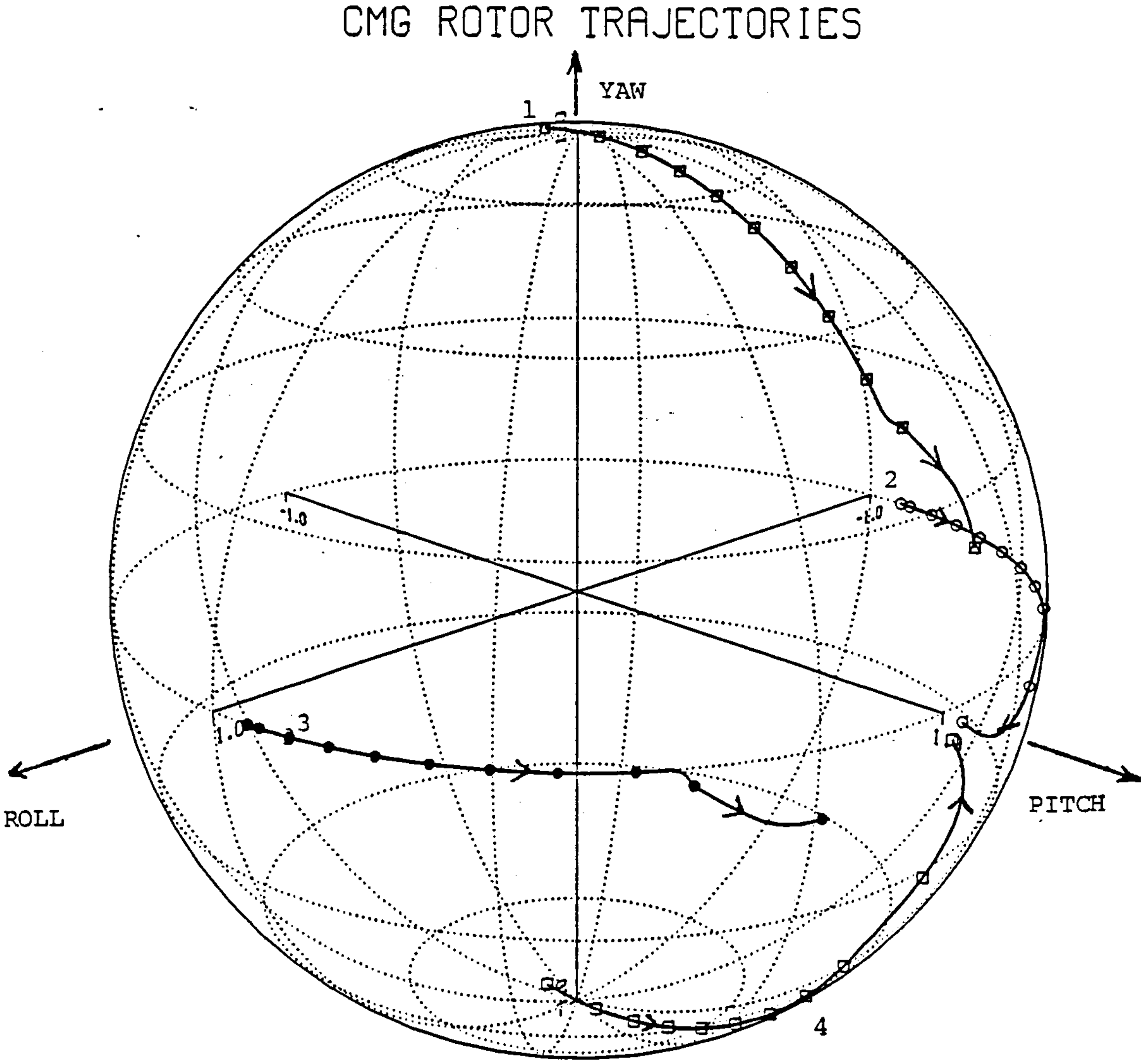
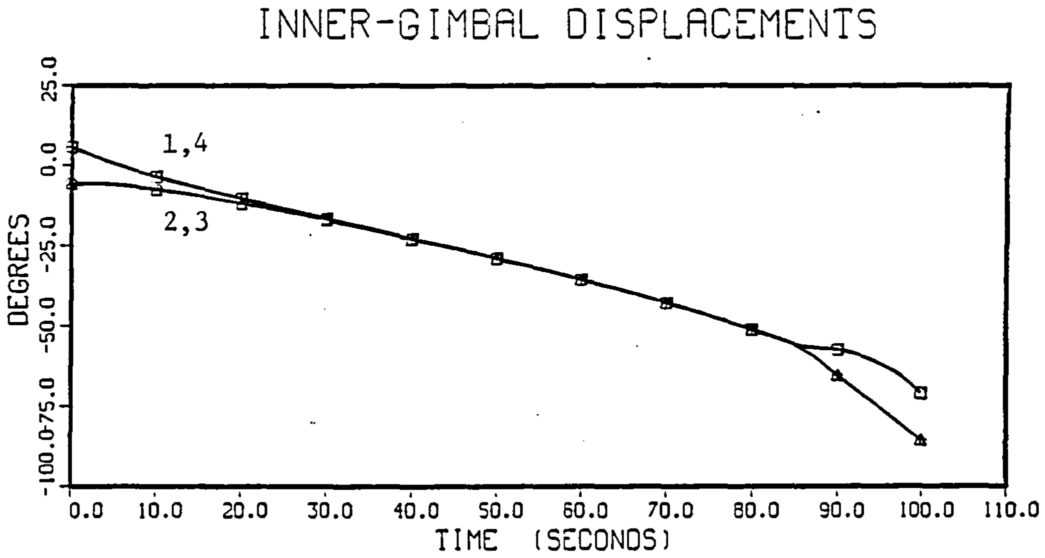


Figure 3.4: Saturation of CMG Rotors along Pitch Axis (668).



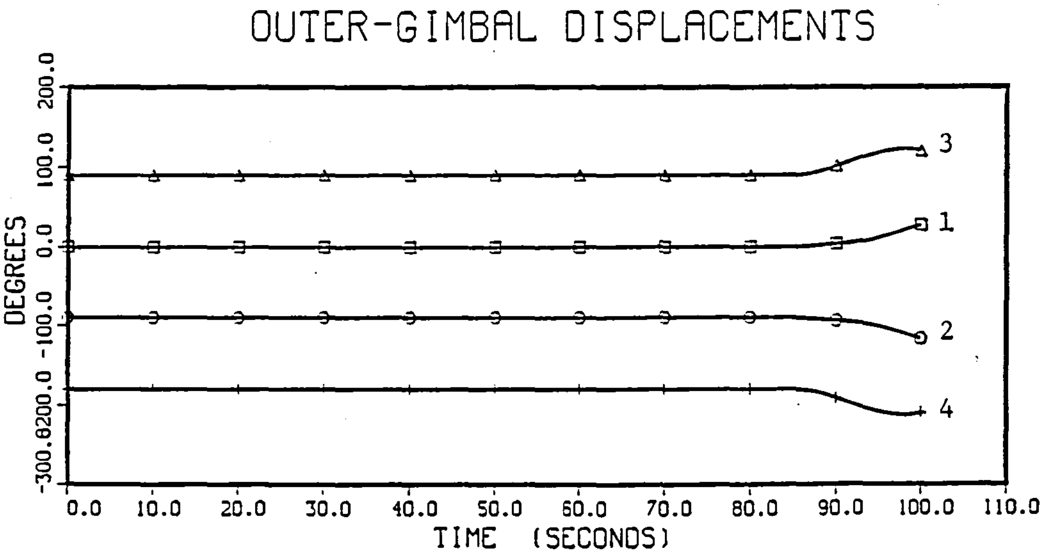


Figure 3.5: Gimbal Angles; Momentum Saturation along Pitch, 2-norm (668).

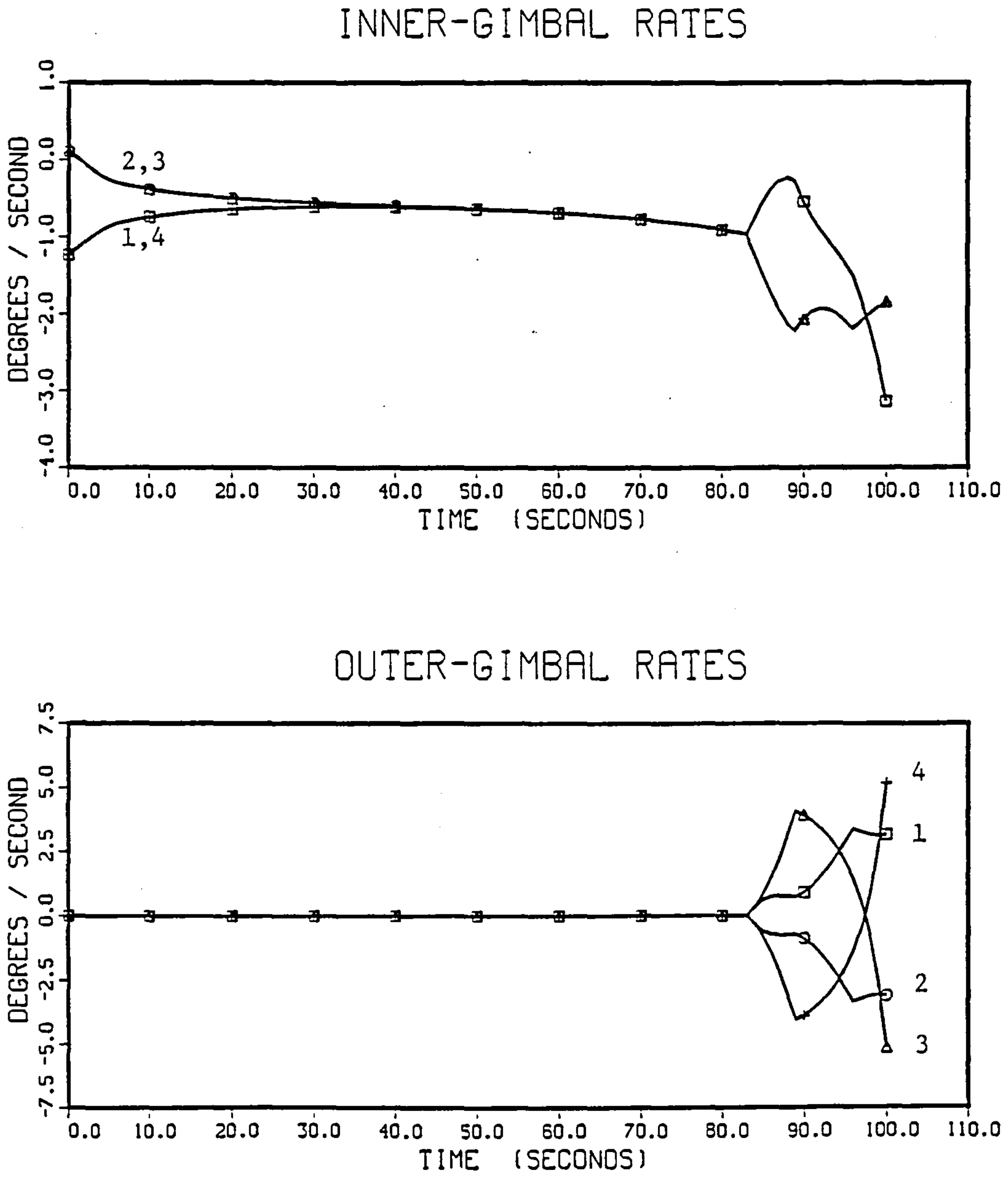


Figure 3.6: Gimbal Rates; Momentum Saturation along Pitch, 2-norm (668).

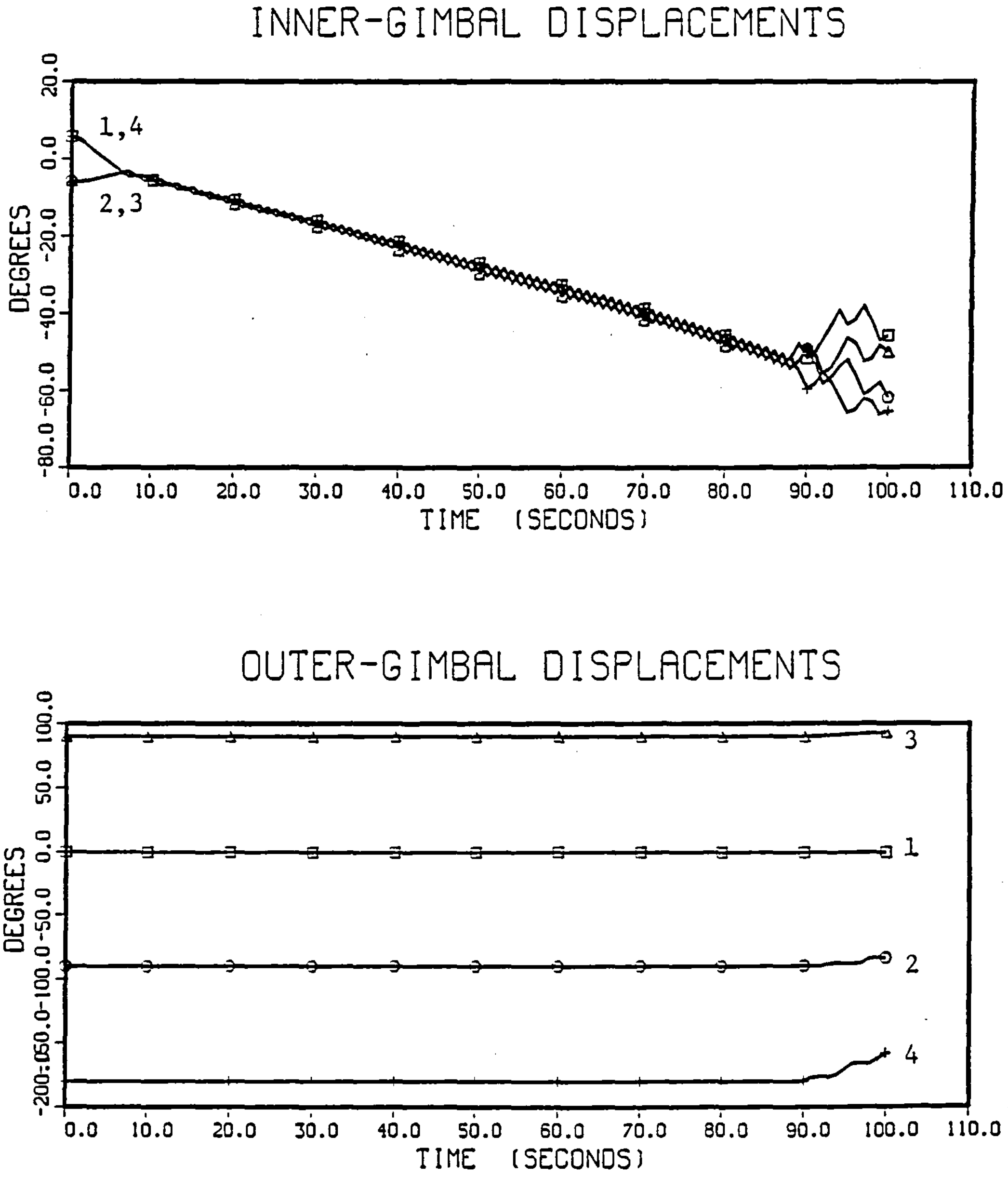


Figure 3.7: Gimbal Angles; Momentum Saturation along Pitch, 1-norm (669).

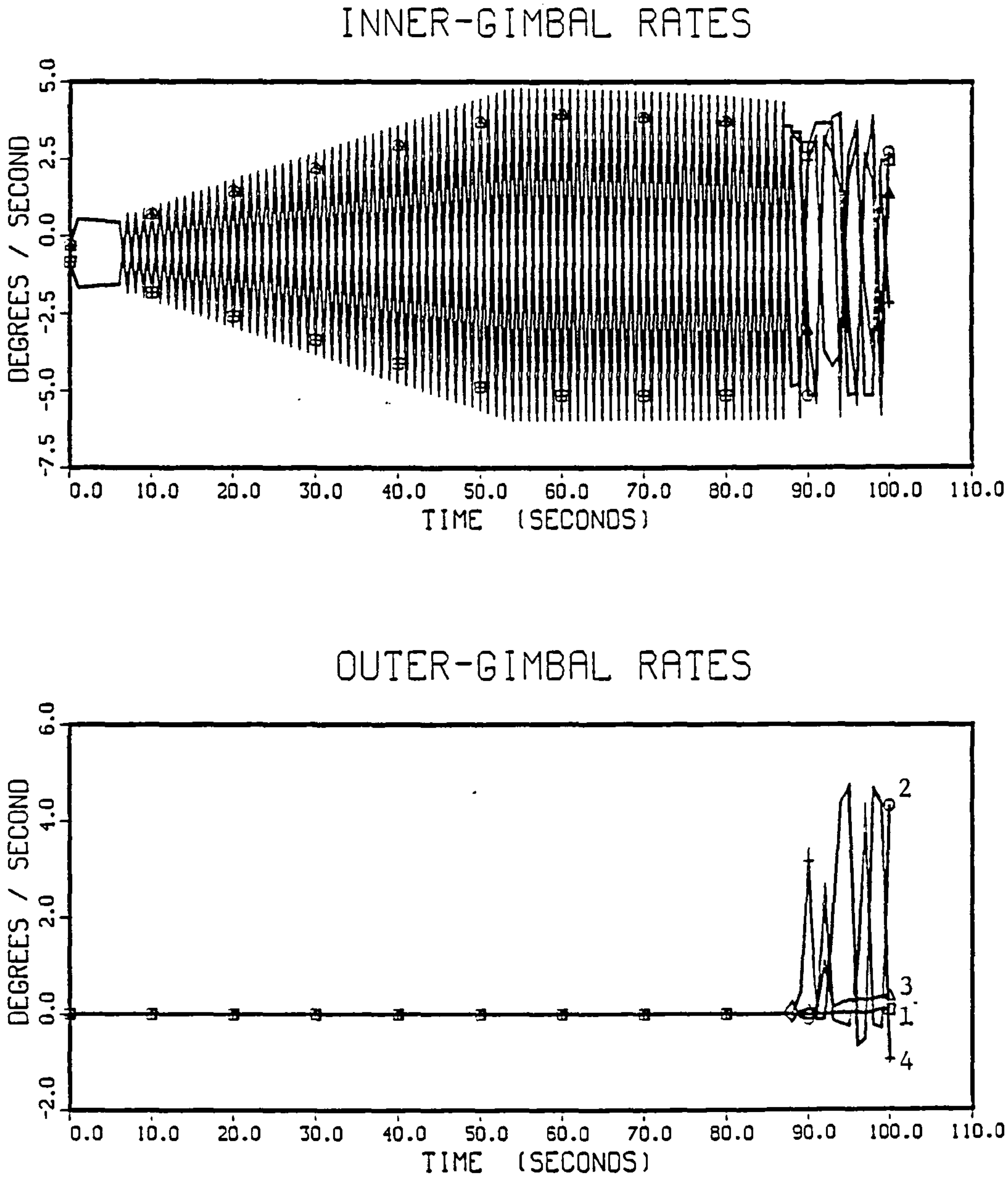


Figure 3.8: Gimbal Rates; Momentum Saturation along Pitch, 1-norm (669).

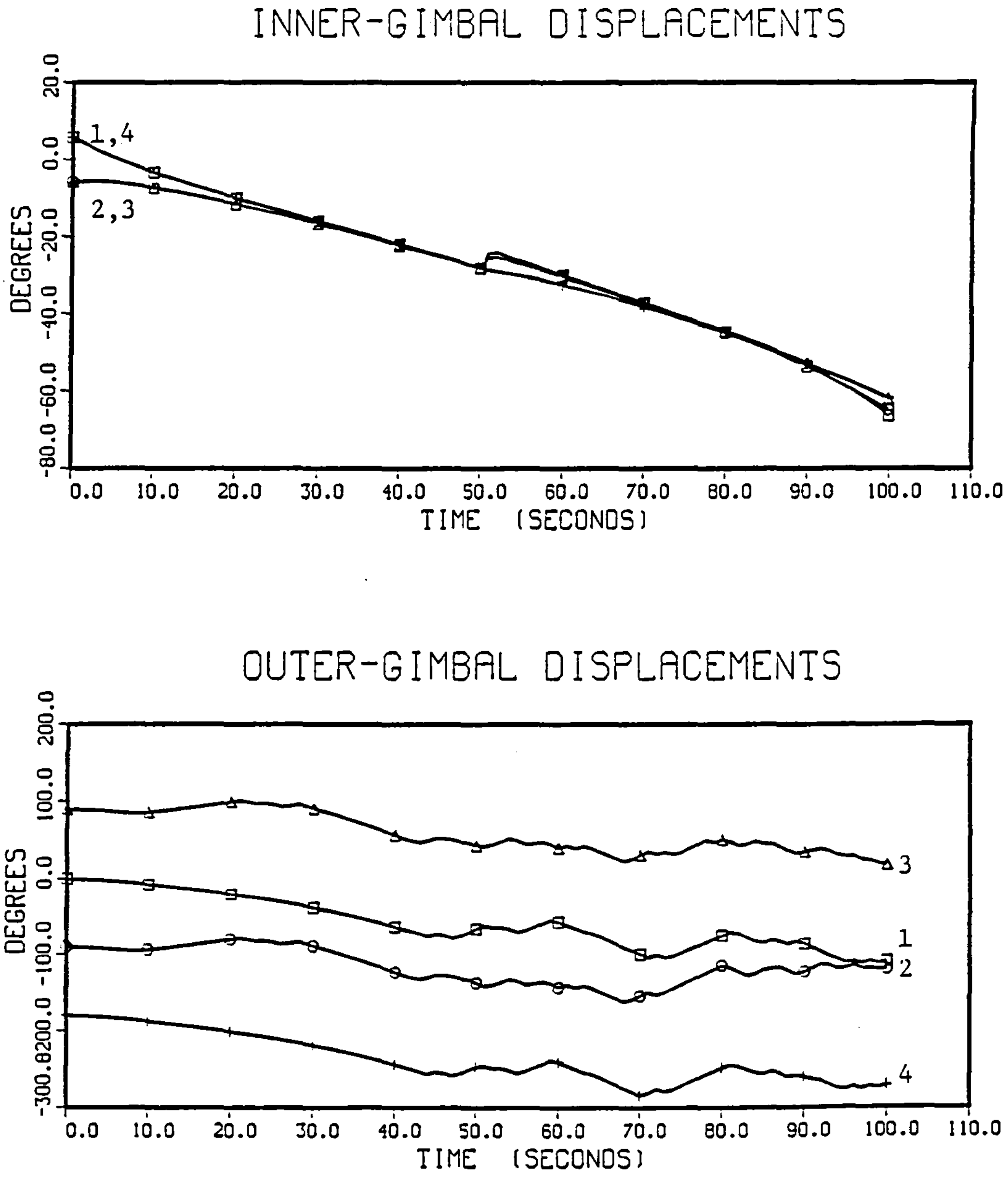
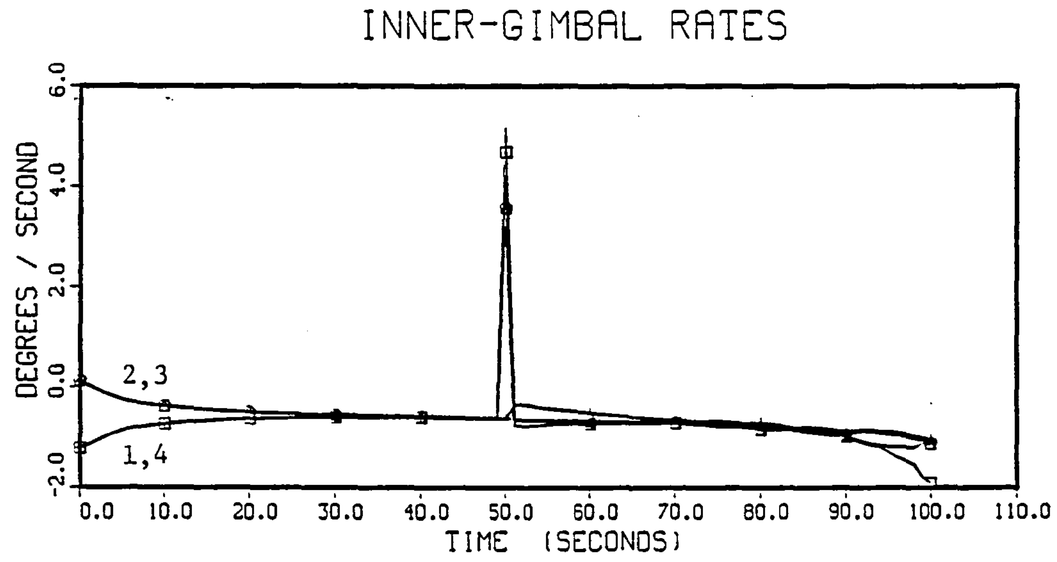


Figure 3.9: Gimbal Angles; Momentum Saturation along Pitch, ∞-norm (731).



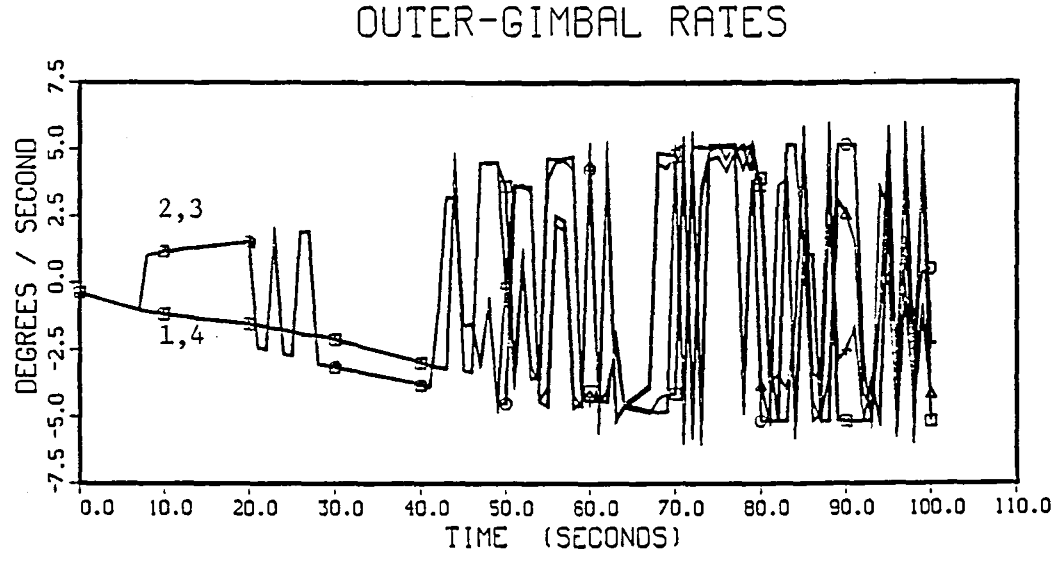


Figure 3.10: Gimbal Rates; Momentum Saturation along Pitch, ∞-norm (731).

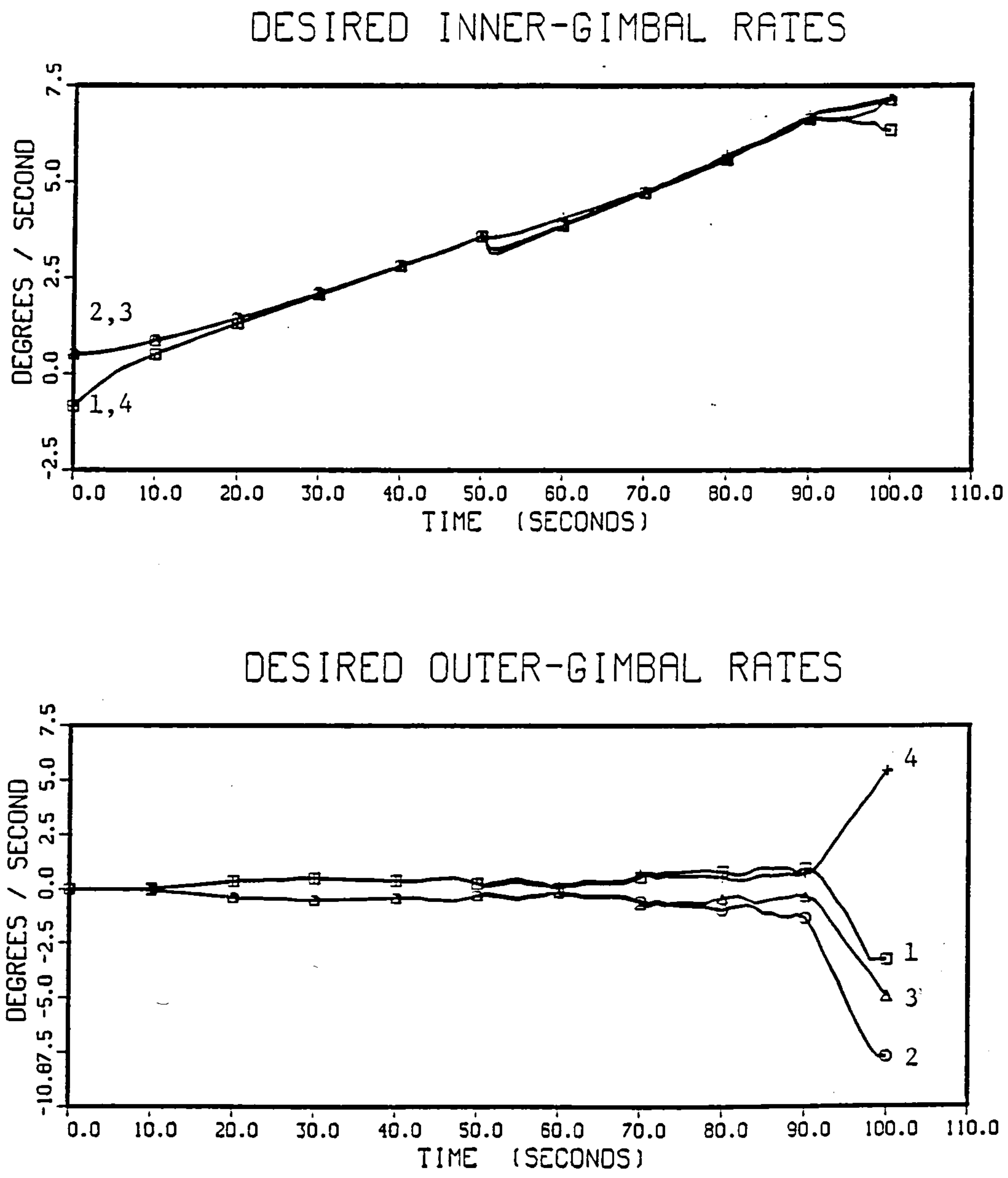


Figure 3.11: Desired Gimbal Rates; Momentum Saturation along Pitch, ∞-norm (731).

3.3.2 CMG Saturation Along Roll

In the second example, a torque of *-100 ft-lb* is commanded along the negative roll axis of the vehicle, and the initial configuration is the same as in the previous case. This torque will cause the CMG rotors to align along the positive roll axis. Because of the particular symmetry of the situation, the desired rotation vector of rotor 2 consistantly lies along its inner gimbal axis. As a consequence, this gimbal will eventually be rotated against its stop. The outer gimbal rotation term (3.28) was added to the rotation vector computation to prevent this from happening.

Figure 3.12 and Figure 3.13 show the gimbal displacements and gimbal rates from the simulation. At about *125 sec.* into the simulation, the outer gimbal of rotor 2 begins to rotate at its peak rate. Initially, the inner gimbal axis is aligned with the negative yaw axis of the vehicle. As the outer gimbal angle rotates, the inner gimbal axis eventually has a component along the positive yaw axis and the gimbal rate must change sign in order to continue projecting more monentum in the positive roll direction. This causes the sudden change of sign in the inner gimbal rate. Figure 3.14 shows that rotating the outer gimbal causes a drop in the gain function shortly before *t=125 secs.* Steering laws based on gain optimization would never choose to rotate the outer gimbal in this situation, and the inner gimbal would be forced against a stop. A similar statement applies to steering laws that attempt to avoid rotor line-up. This is an example of a situation in which the steering algorithm must deal explicitly with a problem situation. Note also that the approximation algorithm is bounding the rate of outer gimbal 3 at its peak value.

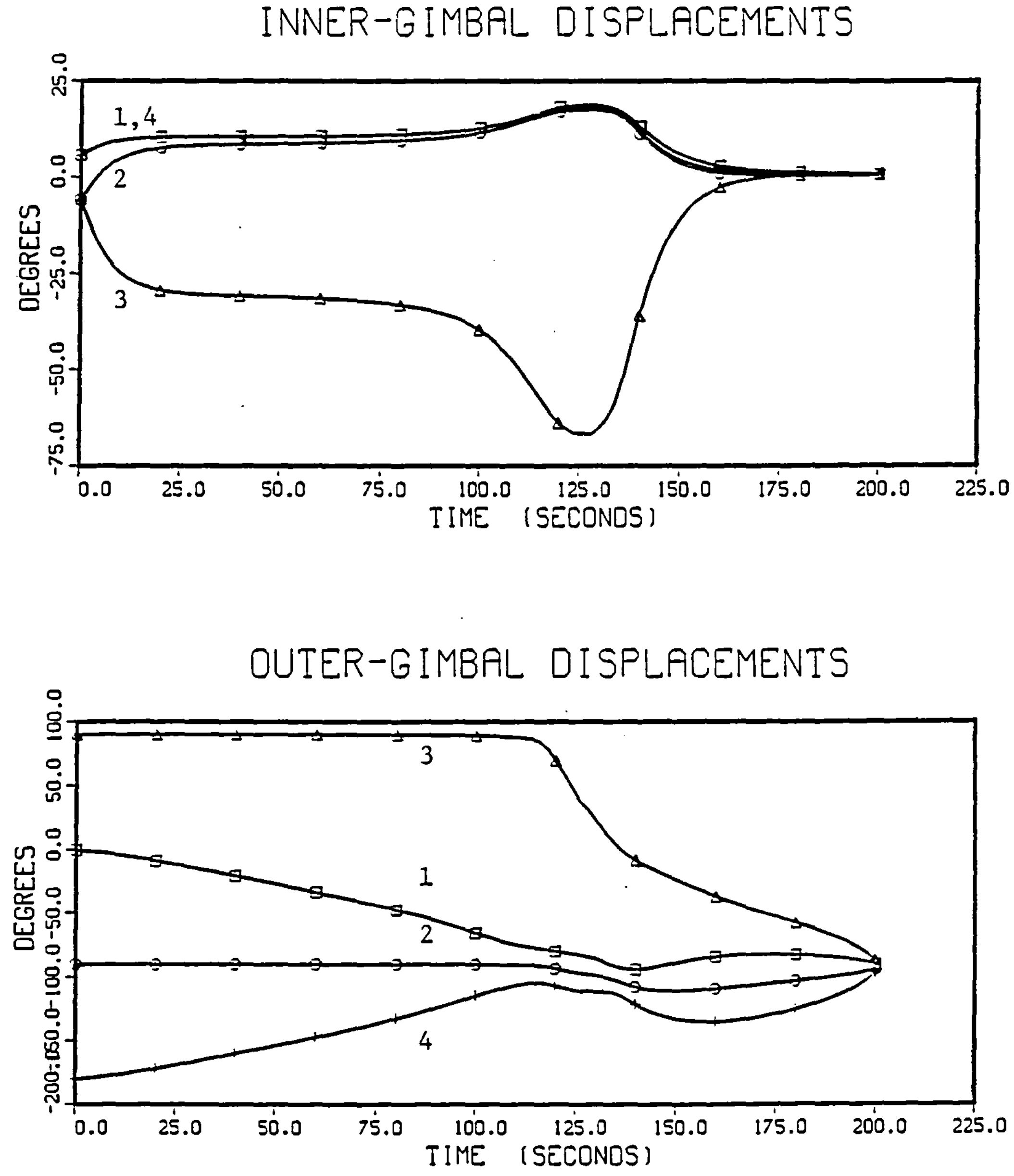


Figure 3.12: Gimbal Angles; Momentum Saturation along Pitch, 2-norm (674).

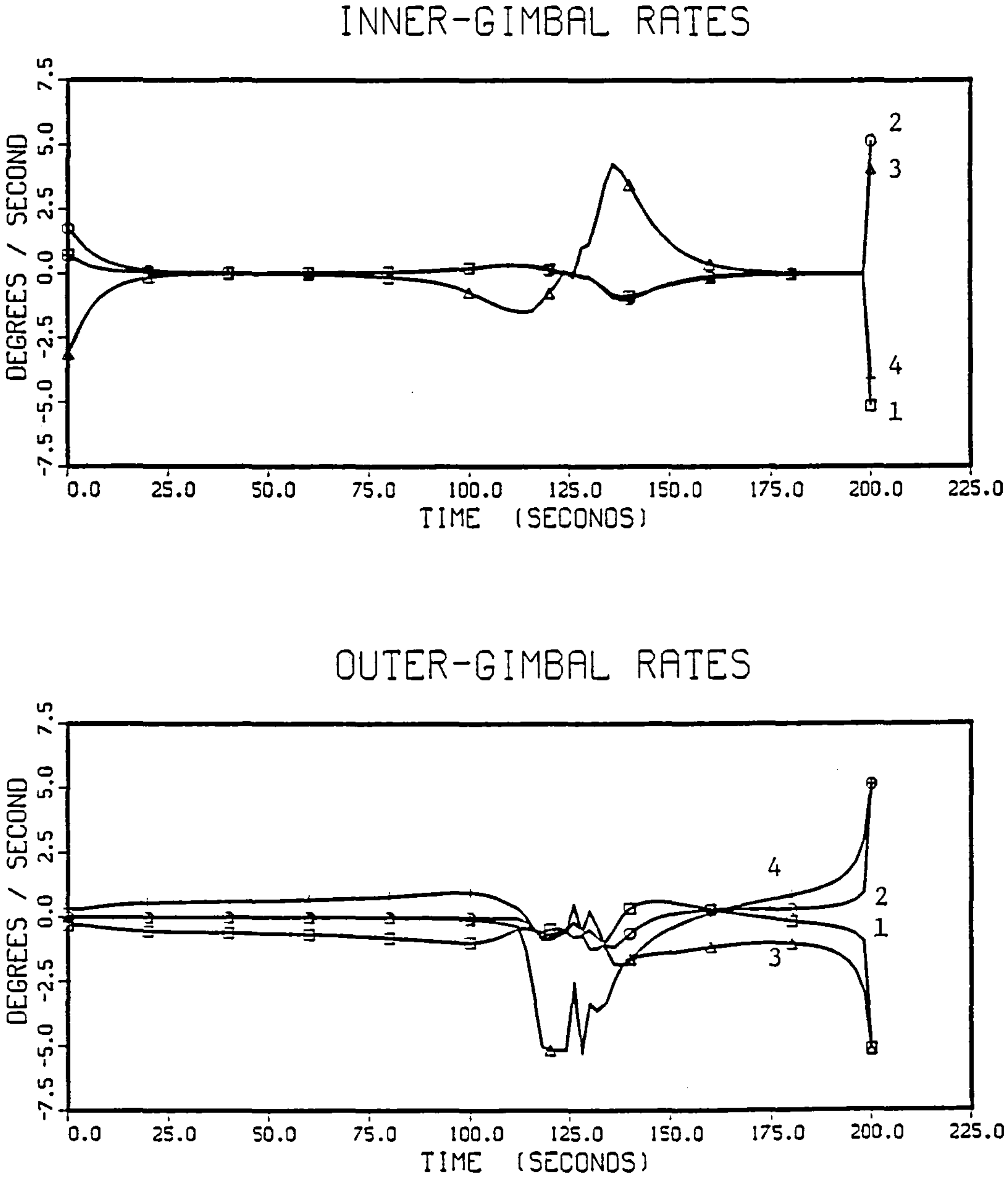


Figure 3.13: Gimbal Rates; Momentum Saturation along Pitch, 2-norm (674).

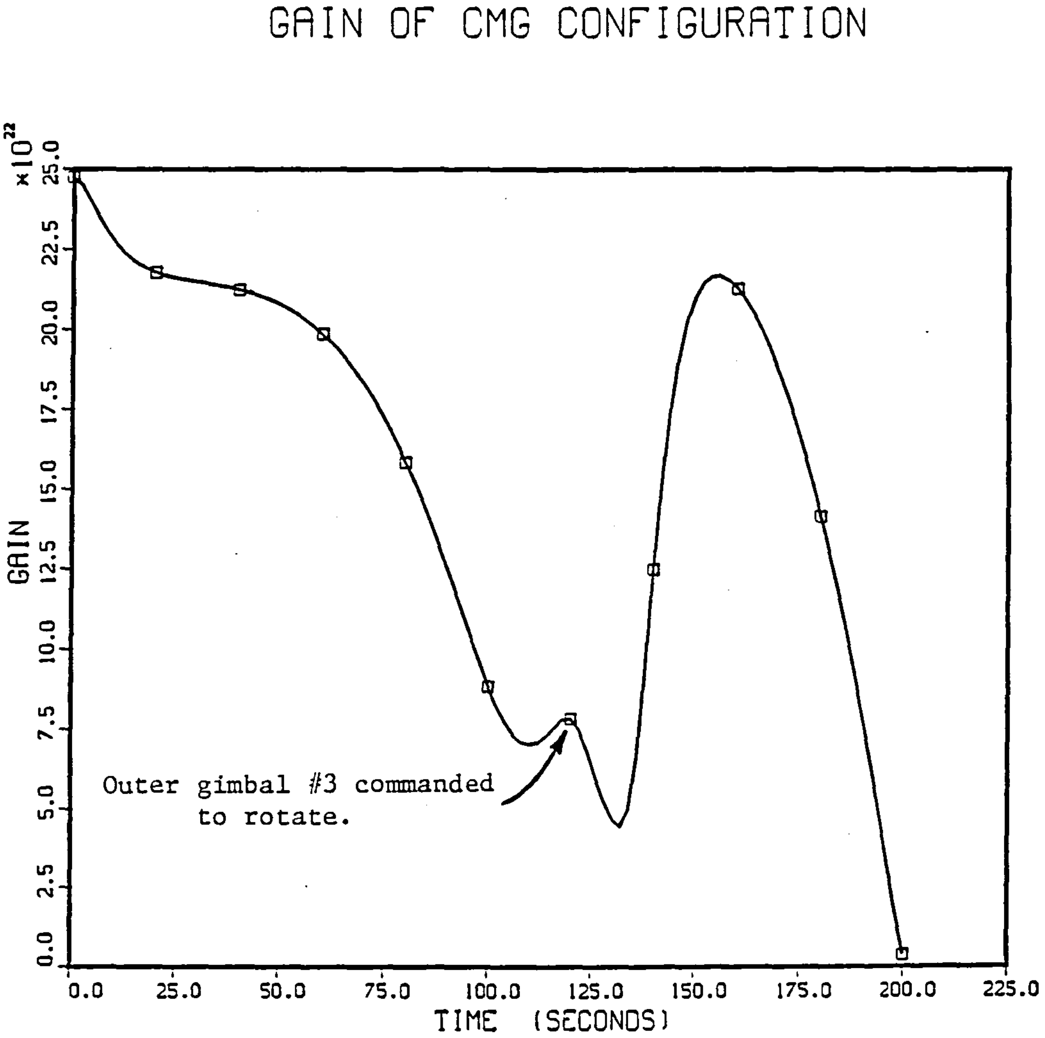


Figure 3.14: Gain Function during Momentum Saturation along Pitch, 2-norm (674).

3.3.3 CMG Redistribution from Poor Configurations

In the next two simulations, an angular rate change is commanded with the CMGs beginning in an initially poor configuration. The attitude control law commands an angular acceleration proportional to the error in the angular rate. The vehicle rate response to a step change in rate of *-0.23 deg/sec* in pitch is shown in Figure 3.15. The rotors are initialized in a zero momentum state with rotors 1 and 2 aligned with the positive roll axis, and rotors 3 and 4 nearly aligned with the negative roll axis; this configuration is nearly singular. The gimbal angle responses are shown in Figure 3.16. The inner gimbals must project momentum along the positive pitch axis to achieve the rate change, but the outer gimbals move until they are evenly distributed about the net CMG momentum vector in order to alleviate the singular condition and attain a best orientation. In Figure 3.17 the gain function is shown. The gain is initially almost zero and increases rapidly to about 80% of its maximum value (recall that gain is not being optimized). It then decreases as the CMGs must project momentum in the pitch direction to maintain the rate change. This example illustrates that the proposed algorithm functions in the vicinity of a singular state. Even if the requested torque had been along the singular direction, null motions would drive the rotors to a configuration which could produce the desired torque.

The same rate response was simulated with the CMGs beginning in another poor configuration with net momentum projected in the roll direction. In this case the inner gimbal angles are initially zero and the outer gimbal angles are $\theta_o = [-\frac{\pi}{3}, -\frac{\pi}{3}, -\frac{2\pi}{3}, -\frac{2\pi}{3}]$ *rad* (rotors 1,2 and 3,4 are initially aligned). The gimbal angle histories are shown in Figure 3.18. Once again, the inner gimbals displace negatively to project momentum in the pitch direction, while the outer gimbals simultaneously displace to avoid line-up. In this case, the final gimbal rate is not zero, since the gimbals must rotate at the same rate and in an opposite sense to the spacecraft. This insures that torques are not applied and that the net CMG momentum remains inertially constant once the vehicle has achieved the desired rate. The signs of the gimbal and spacecraft rates are the same because the outer gimbal axis points along the negative vehicle pitch axis.

A difficulty encountered with algorithms that seek local optima of functions is that the algorithm can get stuck in a configuration that is locally optimal. For the parallel mount configuration of four CMGs, the technique demonstrated here a unique, stable, optimal configuration of the rotors provided that the CMGs project momentum in either the pitch or the yaw direction (the placement of the rotors is unique, not the location of the rotors by index). For each value of net momentum, there is also a stationary point of the steering algorithm that is unstable. If the CMG project momentum only in the pitch direction, any optimal configuration can be rotated as a whole about the outer gimbal axis and still be optimal.

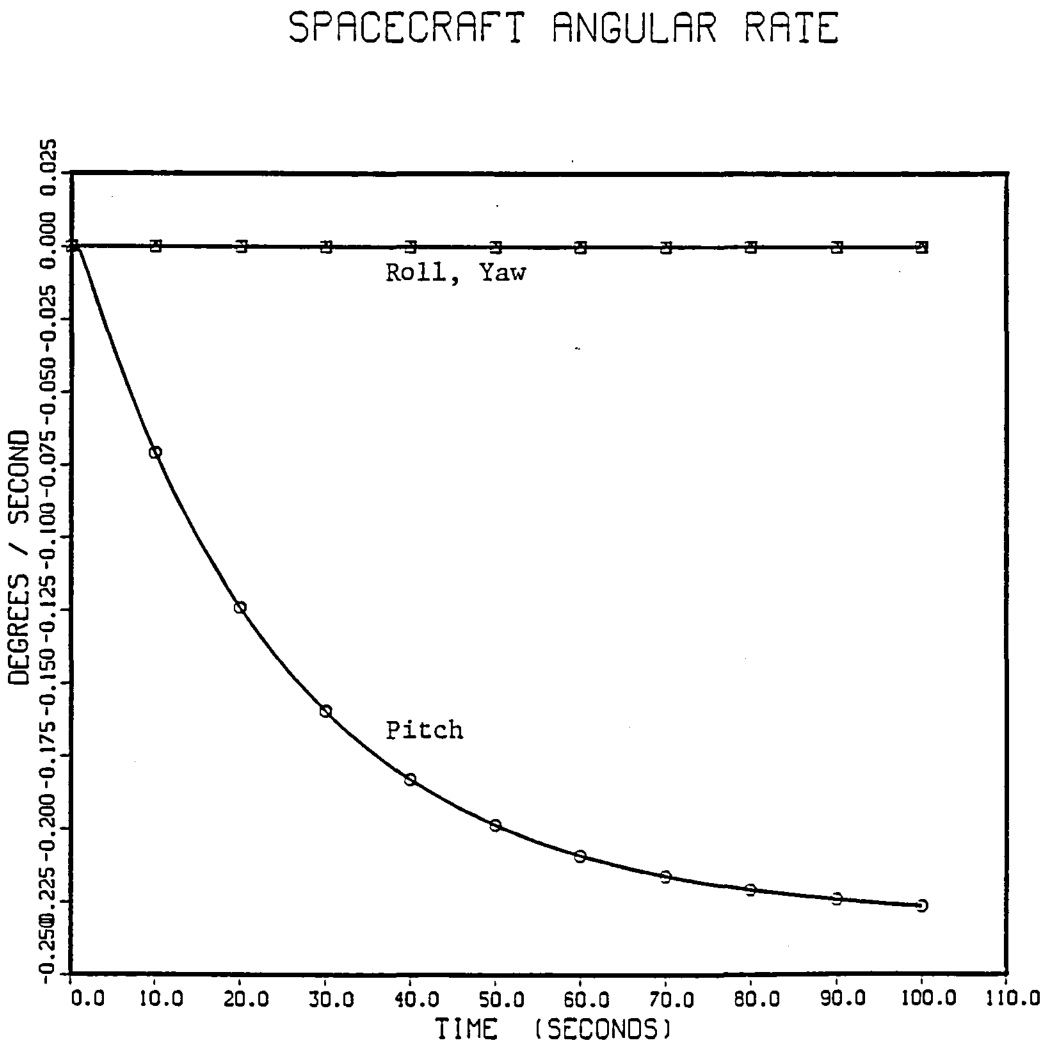


Figure 3.15: Attitude Rate Response to Step Change in Reference Rate, 2-norm (686).

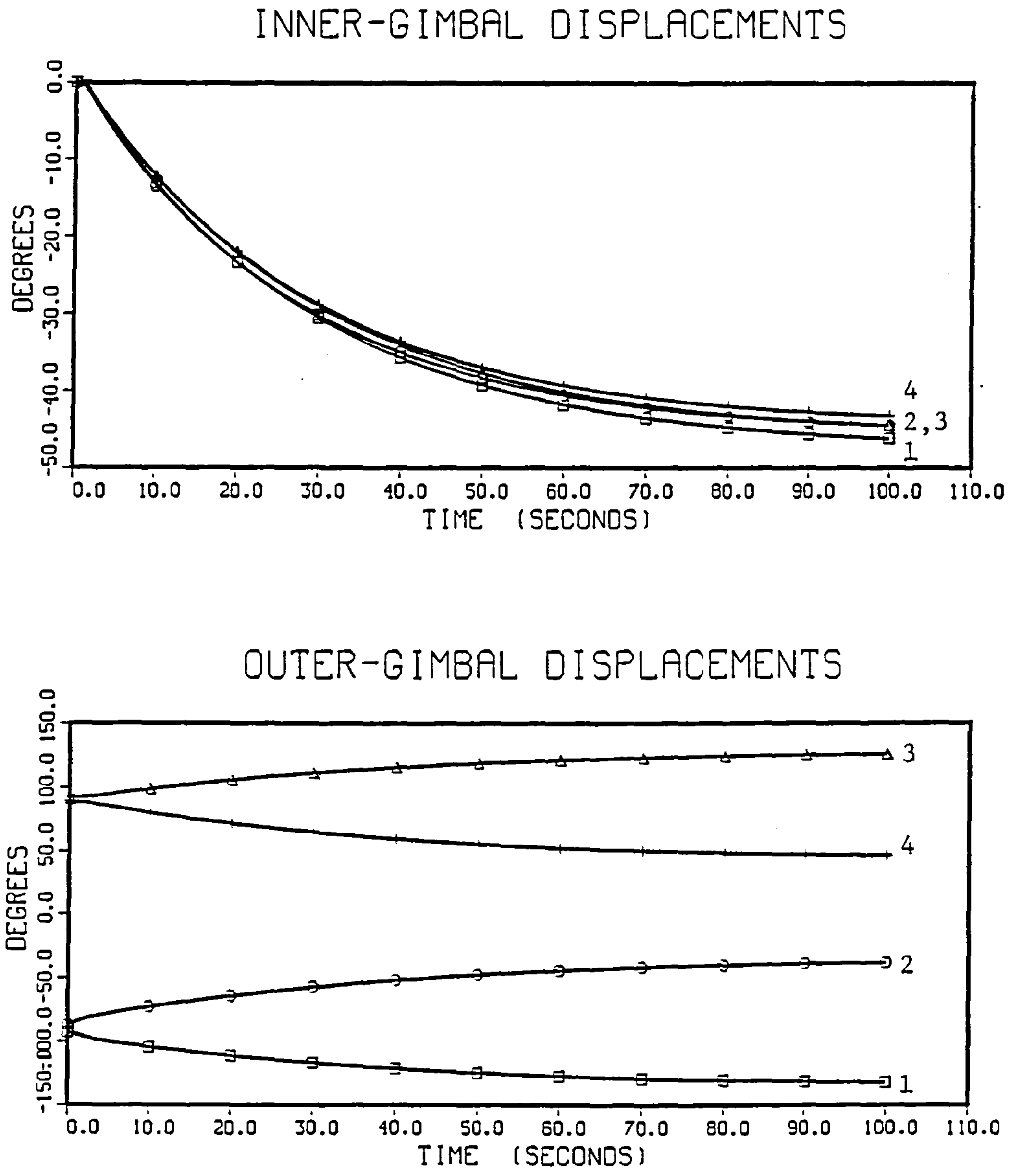


Figure 3.16: Gimbal Angles; Rate Change along Pitch, 2-norm (686).

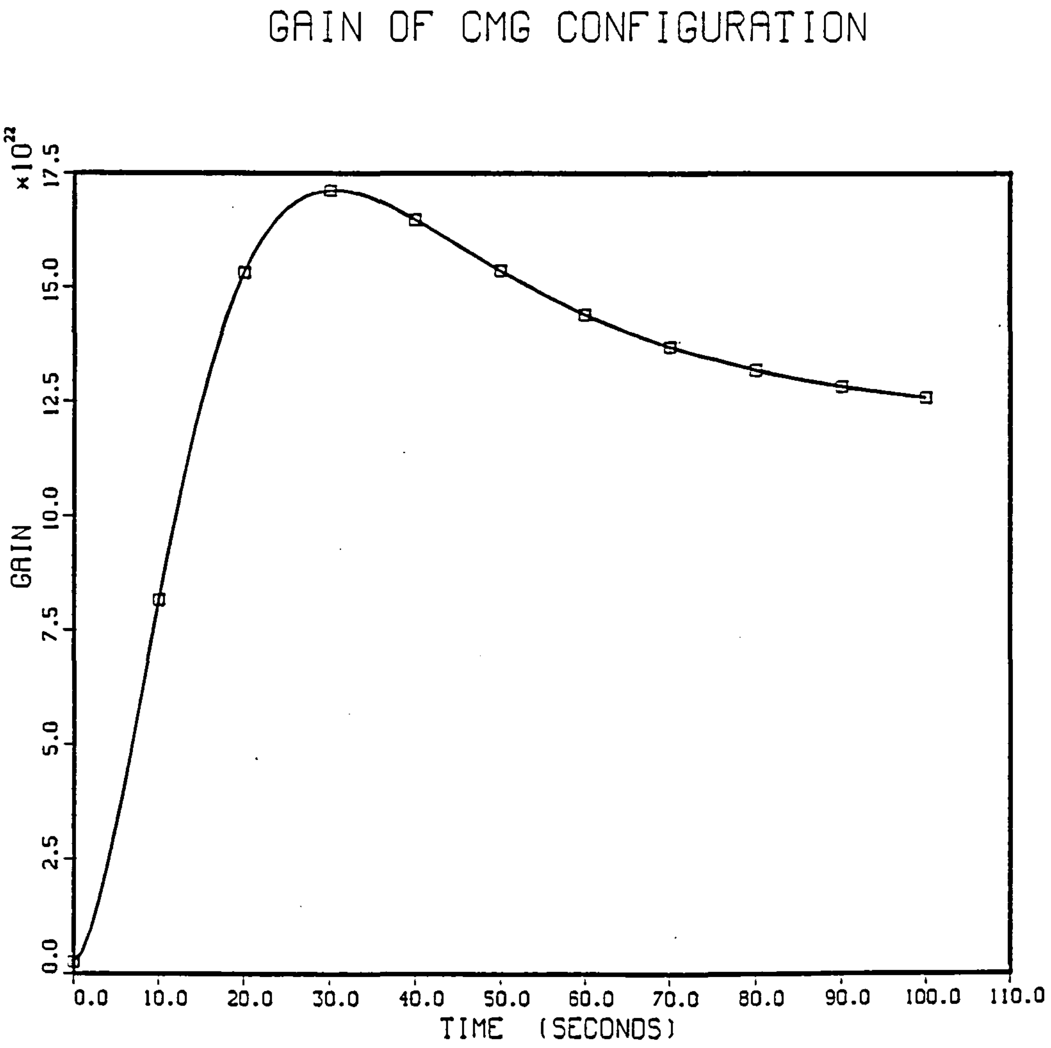


Figure 3.17: Gain Function during Momentum Saturation along Pitch, 2-norm (686).

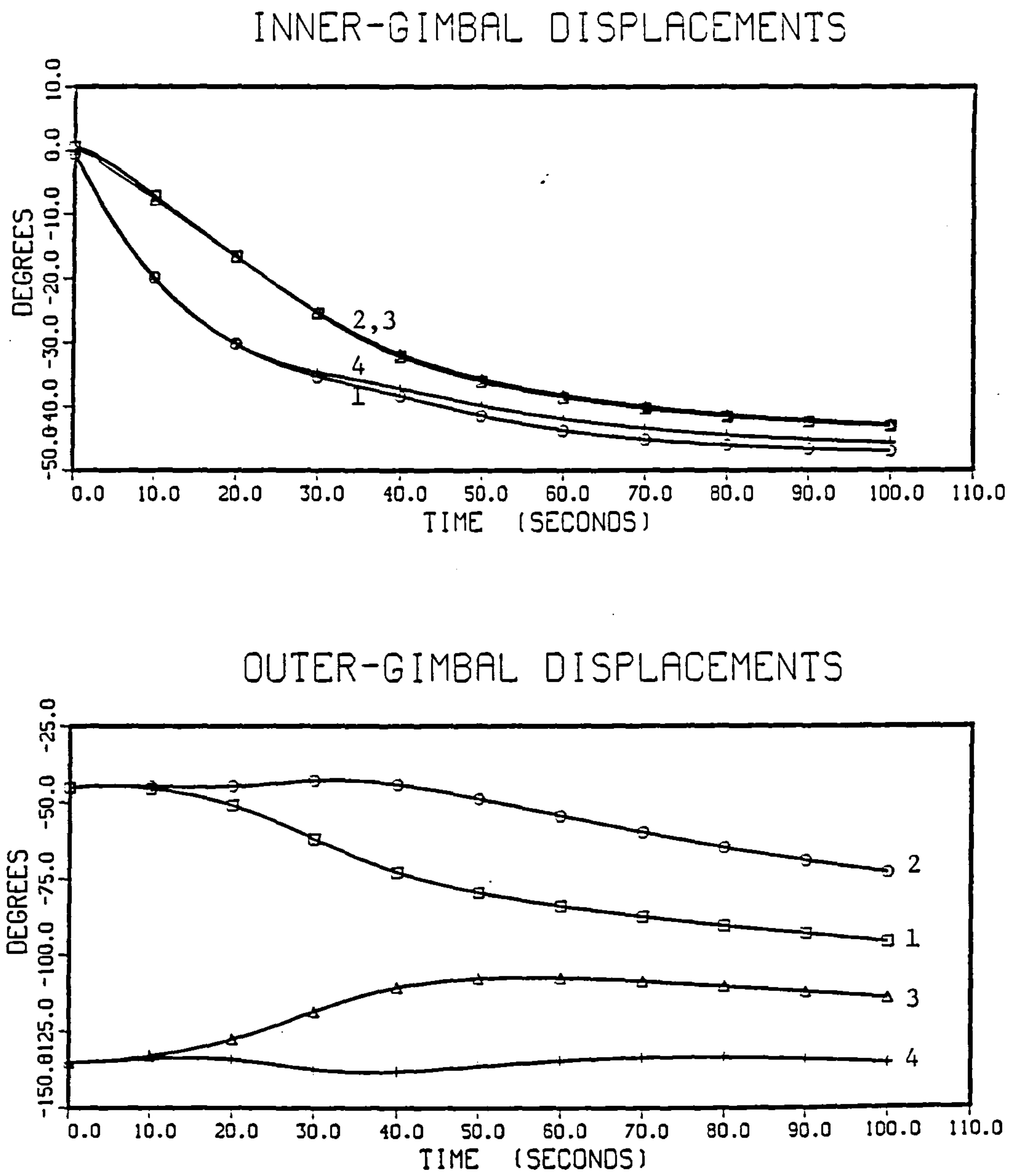


Figure 3.18: Gimbal Angles; Rate Change along Pitch, 2-norm (688).

3.3.4 Attitude Response to Torque Pulses

In this simulation, a feedback control law is used to regulate the spacecraft at a reference attitude, $q = (0,0,0)$. Quaternion and quaternion rate feedback are chosen so that the quaternion dynamics are characterized by decoupled, second order systems with natural frequencies of *0.1 rad/sec* and damping ratios of *0.7*. At the beginning of the simulation, the torque

$$\tau = \begin{Bmatrix} -700 \\ -1100 \\ -700 \end{Bmatrix} ft - lb \tag{3.36}$$

is applied to the vehicle for *7.5 sec.*. Twenty seconds later, the torque is applied in the opposite direction. Momentum conserving impulses of this type are representative of onboard disturbances such as crew motion. The attitude response to these torque disturbances is shown in Figure 3.19 and the gimbal displacements and rates are shown in Figure 3.20 and Figure 3.21, respectively. During the interval between the two torque pulses, the gimbals must remain displaced form the starting configuration, since the initial disturbance causes a change in the total inertial momentum of the vehicle. By the end of the simulation, the rotors return to an orthogonal configuration with the outer gimbal angles slightly rotated with respect to the starting configuration. Note that compensation of the disturbance torques saturated some of the gimbal rates. The approximation algorithm recognizes the rate constraints, and uses other gimbals accordingly to prevent their violation. The only other steering algorithm to consider rate constraints explicitly is [4].

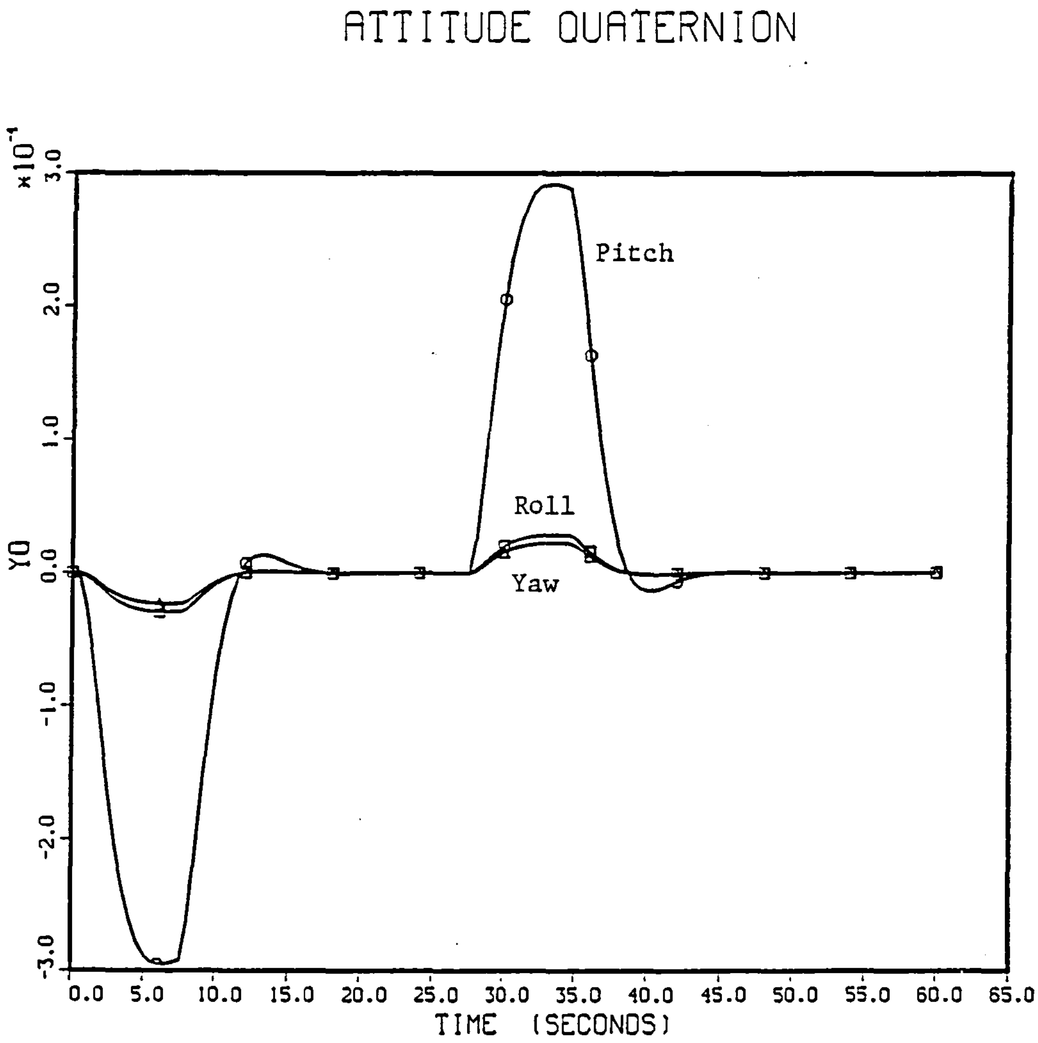


Figure 3.19: Attitude Response to Opposite Torque Pulses (721).

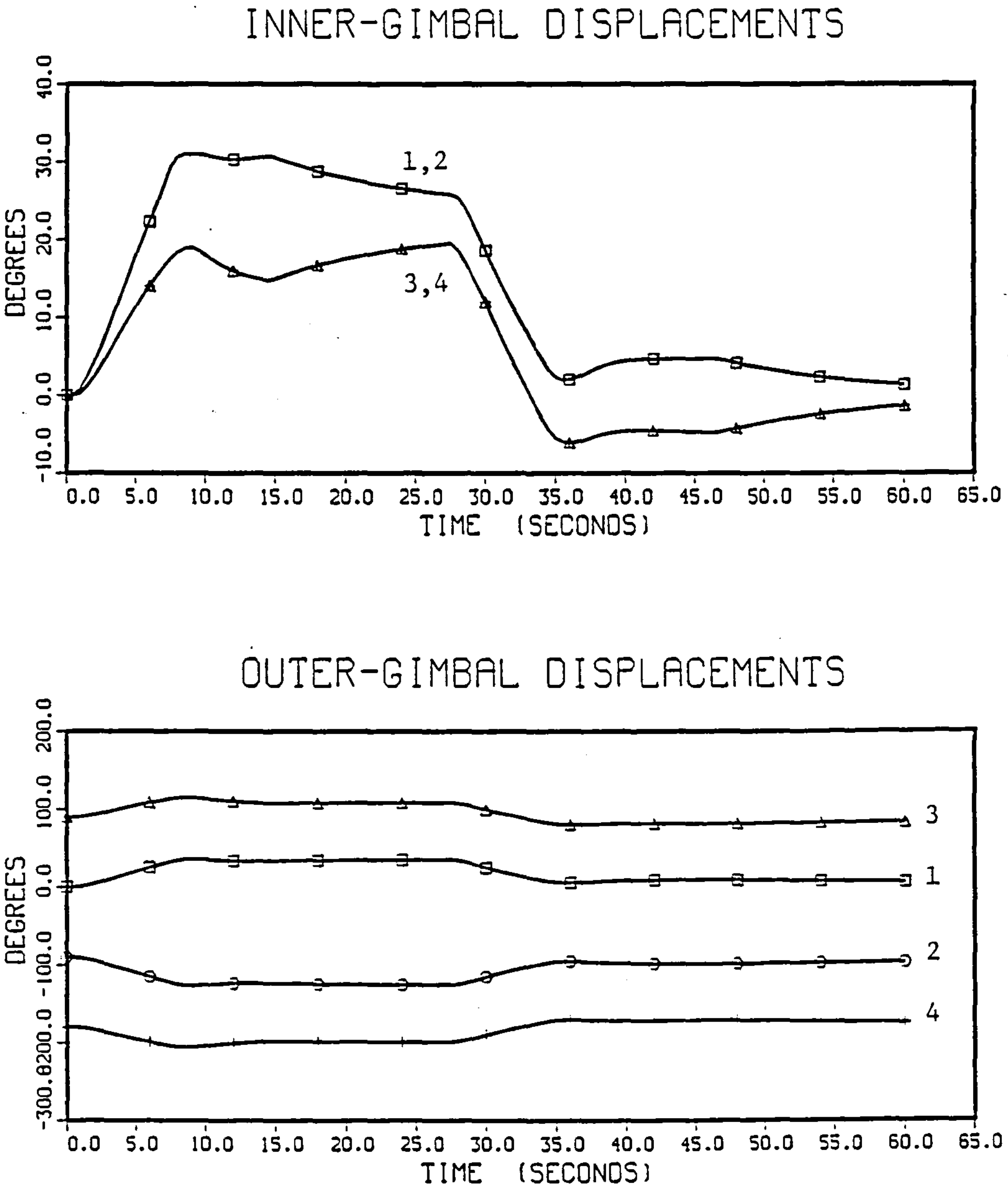


Figure 3.20: Gimbal Angles; Opposite Torque Pulses, 2-norm (721).

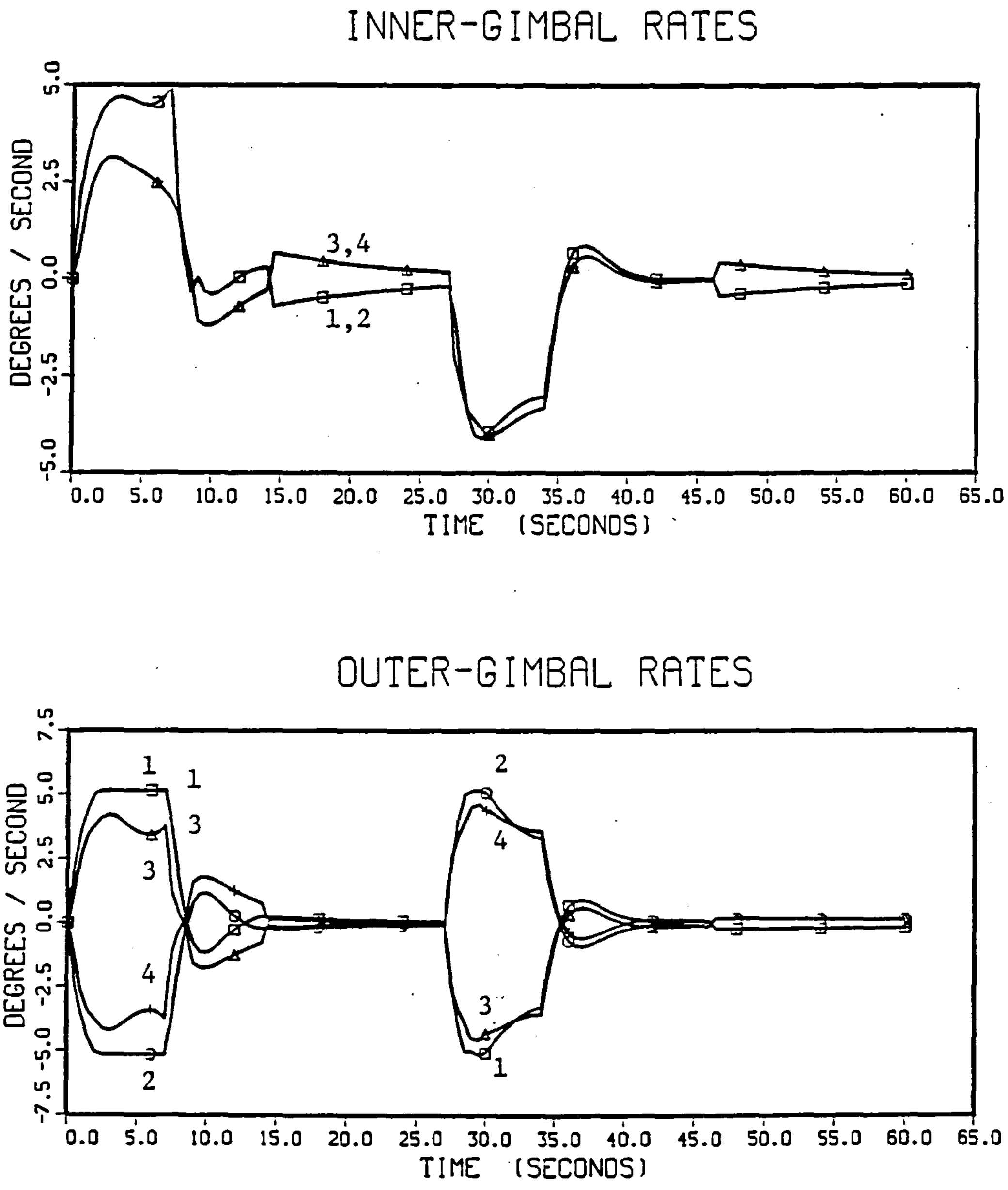


Figure 3.21: Gimbal Rates; Opposite Torque Pulses, 2-norm (721).

Chapter 4

Application of Feedback Linearization to the Momentum Management Problem

A spacecraft orbiting the Earth is subject to a variety of disturbance torques that influence its attitude. In general, disturbances due to motion occuring onboard such spacecraft are momentum conserving. For example, torques caused by crew motion and rotating machinery do not result in a net change in the total momentum of the vehicle. Even though there will be a short term fluctuation in spacecraft momentum during the interval between push-off and impact of a crew member, the total system momentum remains constant. There are, however, a number of environmental torques that do cause a net change in momentum. Small torques due to gravity gradient and aerodynamic effects can lead to substantial changes in the total momentum of the spacecraft over a long period of time. In certain on-orbit applications, large transfers of momentum may occur due to disturbances such as docking with other spacecraft or venting of waste gasses. It is therefore necessary to examine the long-term momentum requirements of CMG equipped spacecraft to insure that the momentum capacity of the CMG system is not exceeded.

Most approaches to momentum management involve manipulating the attitude dependent gravity gradient torque to produce a desired transfer of momentum [14],[15]. Some authors have studied the use of appendages to manipulate aerodynamic torques [41],[1]. In general, the dynamics of the spacecraft are linearized or approximated in some way, and a dynamical representation of the CMGs is usually not included in the vehicle model. In the following section, the momentum management problem is formulated as an optimization problem with an integral quadratic perfomance index. The performance index is to be optimized subject to the constraint that the nonlinear equations describing the spacecraft attitude dynamics (2.35) are satisfied. In (2.35) the control variables are the individual gimbal rates, and the momentum management problem formulated here includes a dynamical model of the CMG actuating system. In addition, the objective function is written to include a term insuring that CMG trajectories avoid singular configurations.

The approach to momentum management studied in this chapter generalizes existing results in two ways: First, the spacecraft model (2.35) is used directly without the assumptions that accompany a Taylor series linearization, and second, the CMGs and singularity avoidance are treated as an integral part of the momentum management scheme. Note that the optimal control problem formulated and solved in this chapter can also be regarded as an optimal maneuver problem; therefore, this solution procedure may be applicable to other areas of spacecraft control.

The optimal momentum management problem is formulated in Section 5.1. In Section 5.2, a suboptimal approach is described for solving nonlinearly constrained optimization problems of the type posed in Section 5.1. The approach is applicable when the equations of motion describing the plant satisfy the external linearizability conditions of Chapter 3. The original optimization problem is reduced to an unconstrained nonlinear program in two steps. First, a linearizing tranformation is applied to the system and the objective function. In this way, a new optimization problem is obtained in which the dynamical constraint is a linear system. Then, the system dynamics are approximated by a finite sum of independent functions of time, and a solution for the linear equations is obtained using collocation.

In Section 5.3, the techniques of the previous section are applied to develop an algorithm for solving the momentum management problem stated in Section 5.1. This algorithm is exercised in a number of examples presented in Section 5.4.

4.1 Formulation of the Optimal Momentum Management Problem

In this section, momentum management is formulated as a quadratic optimization problem. First, the characteristics of a *desired* state trajectory are discussed for the model (2.35). Basically, it is desired to keep CMG momentum near some nominal value while keeping vehicle attitude excursions small and avoiding singular CMG configurations. In situations where the spacecraft initially has a significant amount of stored momentum, the objective of momentum management is to transfer that momentum out of the vehicle/CMG system. The approach to momentum management solved here is similar to [14], except that a nonlinear vehicle model is used. The optimal control problem posed at the end of the section is to minimize the deviation from the desired trajectory without excessive use of control effort.

It has already been mentioned that it is undesirable to allow the CMG system to accumulate stored momentum along any particular axis. A large amount of CMG momentum projected in a particular direction implies that the ability of the CMG system to control the spacecraft about that axis is degraded. An exception may occur if a large momentum impulse is anticipated. Under such circumstances, it may be desirable to transfer momentum in a particular direction to increase the magnitude of the impulse that can later be absorbed. These considerations can be used to characterize a desired trajectory for the net CMG momentum to track, $h^d_{CMG}(t)$. The desired CMG momentum may be a constant value or a ramp to a desired value at some point in time. Similarly, there may be a desired attitude rate that the vehicle should maintain or track, $\omega_d(t)$. The desired net momentum can be expressed as

$$h^L_d(t) \;=\; h^d_{CMG}(t) \;+\; I^{TEA}\,\omega_d(t) \qquad (4.1)$$

where all quantities are expressed in LVLH coordinates, and I^{TEA} denotes the inertia matrix in LVLH of the vehicle at the torque equilibrium attitude. For applications such as the Space Station, the desired vehicle rate will be the orbital rate of the LVLH reference frame, $\omega_d = \omega_o$.

Certain applications may require a spacecraft to track an inertial or LVLH attitude for long periods of time. Such requirements are usually specified to within a certain error tolerance. In other applications, characteristics of the spacecraft may place limits on the attainable attitudes. For example, gravity gradient torques on spacecraft with large asymmetries in the principal moments of inertia may make it difficult to maintain an attitude far from equiliburium. The requirements for attitude control define a desired attitude history, $q_d(t)$, that the vehicle should maintain.

In formulating a performance index for the CMG configuration, initial consideration was given to a measure such as the gain function [7], or a measure of rotor orthogonality such as (3.24). The objective function to be formulated here was originally written in the latter fashion with a term of the form

$$\sum_{i \neq j}^{N_{cmg}} \left[\left\langle h^i_{CMG}, h^j_{CMG} \right\rangle \right]^2 \tag{4.2}$$

that was integrated as part of the objective function. In developing the numerical algorithm for application to space station momentum management, it was found more efficient to compute desired gimbal angle histories, $\theta_d(t)$, based on the desired CMG momentum and knowledge of the optimal configuration determined by (4.2), rather than deal with (4.2) directly. Even though this particular term has not been included in the development presented here, it is not difficult to extend the following formulation to include it.

The preceeding discussion illustrated that there is a natural way to define a desired state trajectory for an orbiting spacecraft equipped with momentum exhcange devices. Denoted $x_d(t)$, the desired trajectory can be used to define a state error vector

$$e(t) \triangleq x(t) - x_d(t) = \begin{Bmatrix} h^L(t) \\ q^L(t) \\ \theta(t) \end{Bmatrix} - \begin{Bmatrix} h^L_d(t) \\ q^L_d(t) \\ \theta_d(t) \end{Bmatrix} \tag{4.3}$$

Assuming that there is also a penalty associated with the use of control effort, the *cost* of a particular control/state trajectory can be quantified using an integral quadratic performance index

$$J(x,u) \triangleq \int_0^T \left[e^T(t)\, Q(t)\, e(t) + u^T(t)\, R(t)\, u(t) \right] dt + e^T(T)\, P(T)\, e(T) \tag{4.4}$$

where $Q(t)$ and $R(t)$ denote time-dependent weighting matrices, $u = \dot{\theta}$ is the control variable in (2.35), and T denotes a fixed final time. The matrix $P(T)$ is an additional penalty on the error at the terminal time that can be used to drive the system to desired end states. The *optimal momentum management* problem is to minimize the performance index (4.4) subject to the constraint that the control/state pair satisfy the differential equation (2.35) with initial condition $x(t = 0) = x_0$. In the next section, a general procedure is discussed for solving a broad class of optimization problems of the type posed here.

4.2 Formulation of a Class of Optimization Problems as Nonlinear Programs

In this section, an approach is developed for obtaining approximate solutions to a class of optimization problems which includes the optimal momentum management derived in the previous section. The classical approach to this type of optimization is to employ variational methods to obtain a set of necessary conditions that must be satisfied by an optimal control and its associated state and costate trajectories [42],[43]. The necessary conditions are in the form of a set of coupled, generally nonlinear, ordinary differential equations with boundary data specified at both the initial and final times. This *two-point boundary-value problem* must be solved to obtain an explicit solution for the optimal control.

There are a number of difficulties that arise when one attempts to solve the boundary value problem for the optimal control. Because boundary data for the state and costate equations are specified at different times, the equations cannot be solved directly. In addition, different types of optimality criteria and terminal constraints lead to different types of boundary value problems. In general, a method that works well for a particular class of equations may not be applicable to similar equations with different boundary data.

A variety of gradient based methods have been developed to solve the two-point boundary value problems arising in optimal control theory. The ultimate convergence of these algorithms is often strongly dependent on a starting solution supplied by the programmer. Another difficulty is related to the fact that gradient-based methods can be sensitive to computational errors, and they become increasingly ill-behaved as the order of the problem increases. Because of these limitations, such techniques have not been extensively employed in autonomous control systems, and applications have been primarily limited to research investigations.

Dynamic programming is another method that has been proposed for solving optimization problems. The difficulty with this method is that the computational burden increases much more rapidly than the order of the model [44]. Even though a number of algorithms have been proposed for resolving many of the difficult issues, none has proved generally reliable, and the search for methods for solving optimal control problems for high-order, nonlinear systems is still an active research area.

In this section, a method is proposed for transforming a class of optimization problems of the form

$$\left.\begin{array}{ll} \min\limits_{u} & S(x(t_f),t_f) \;+\; \displaystyle\int_{t_0}^{t_f} L(x,u,t)\,dt \\ \text{subject to} & \\ & \dot{x} \;=\; f(x) \;+\; \displaystyle\sum_{i=1}^{m} g_i(x)u_i \\ & C(x(t_f),t_f) \;=\; 0 \\ & x(t_0) \;=\; x_0 \end{array}\right\} \tag{4.5}$$

where $S(x_f, t_f)$ is the penalty on the terminal state of the system, and $L(x, u, t)$ is the penalty for a particular state/control pair. The first constraint requires that the state and control histories satisfy the equations of motion of the system to be controlled, and the constraints $C(x(t_f), t_f)$ denote restrictions on the terminal state of the system. Limits on the magnitudes of certain state variables or controls can be included implicity in the objective $L(x, u, t)$ using penalty functions. The specific class of problems to which the proposed method is applicable are those for which the vector fields $f, g_1, \ldots, g_m$ are feedback linearizable along the optimal trajectory.

The idea behind the development which follows is to transform (4.5) into an unconstrained nonlinear program. The advantage of the nonlinear programming formulation is that there exist robust algorithms for these problems that have guaranteed convergence properties. The transformation is achieved in two steps. The first step uses the feedback linearization theory to transform the stated problem into a related problem in which the dynamical constraint is a linear system. The second step is to use collocation on a finite basis of functions to obtain an approximate solution to the linear system of equations. In applying the transformation to solve the optimization, care must be taken that optimal solutions remain optimal under the linearizing transformations, and that the collocation procedure yields a solution that converges to the optimal solution as the order of the approximation increases. These technical considerations are examined in the following two subsections.

4.2.1 Equivalence of Optimal Control Problems

Previously, feedback linearization has been presented as a method of transforming nonlinear systems into systems which exhibit linear input-output behavior. There is a deeper interpretation to this problem that has not been discussed. The interpretation comes from the observation that the set of transfomations (F, α, β) defines an ***equivalence relation*** on systems of differential equations of the form

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)\, v \tag{4.6}$$

Two systems of equations are in the same equivalence class if they are related by an admissible change of coordinates. Put quite simply, any system in a particular class can be made to behave like any other system in that class by using the appropriate transformation (F, α, β). The conditions that guarantee the existence of a solution to the feedback linearization problem guarantee that the equivalence class associated with the nonlinear system (B.52) contains one particular linear system - the Brunovsky canonical form (if the equivalence class contains one linear system it contains many). Naturally, there are different definitions of equivalence associated with different classes of admissible transformations [31],[32],[19], and there is also no reason to limit this discussion to systems which can be linearized [45].

Clearly, the class of transformations that have been studied for nonlinear systems can be used to transform (4.5) into a related problem in which the model for the plant is linear. Direct substitution of the functions

$$x = F^{-1}(z) \quad \text{and} \quad u = \alpha(F^{-1}(z)) + \beta(F^{-1}(z))\, v \tag{4.7}$$

into the equations (4.5) yields the optimization problem

$$\left.\begin{aligned} \min_{v} \quad & \bar{S}(z(t_f)) + \int_{t_f}^{0} \bar{L}(z,v,t)\,dt \\ \text{subject to} \quad & \\ & \dot{z} = Az + Bv \\ & \bar{C}(z(t_f),t_f) = 0 \\ & z(t_0) = z_0 \end{aligned}\right\} \tag{4.8}$$

Formally, there is no problem with this substitution; however, one would like to be certain that an optimal solution of (4.8) is an optimal solution of (4.5) under the mapping (4.7).

The Hamiltonian function associated with the original optimization (4.5) is defined by

$$\mathcal{H} \triangleq L(x,u,t) + \lambda^T(f(x) + G(x)u) \tag{4.9}$$

where the n additional variables λ are the costate or adjoint variables. It is well known that an optimal solution necessarily satisfies the conditions listed below [43].

$$\dot{x} = f(x) + \sum_{i=1}^{m} g_i(x)u_i \tag{4.10}$$

$$\dot{\lambda} = -\left[\frac{\partial f}{\partial x} + \sum_{i=1}^{m} \frac{\partial g_i}{\partial x} u_i\right]^T \lambda - \left(\frac{\partial L}{\partial x}\right)^T \tag{4.11}$$

$$\frac{\partial \mathcal{H}}{\partial u} = 0 \tag{4.12}$$

$$x(t_0) = x_0 \tag{4.13}$$

$$\lambda(t_f) = \left[\frac{\partial S}{\partial x} + \nu^T \frac{\partial C}{\partial x}\right]_{t_f}^T \tag{4.14}$$

$$C(x(t_f),t_f) = 0 \tag{4.15}$$

The set of algebraic equations (4.12) are the stationarity conditions that determine the optimal control in terms of the state and costate trajectories. Equations (4.10) and (4.11) together with the the boundary conditions (4.13) and (4.14) form a two point boundary value problem that determines the optimal state and costate trajectories. The constant vector ν denotes a set of parameters that are chosen to satisfy the side conditions (4.15).

A similar set of necessary conditions exists for the problem with feedback linearized dynamics (4.8). These necessary conditions are derived from the Hamiltonian

$$\begin{aligned} \bar{\mathcal{H}} &= \bar{L}(z,v,t) \;+\; p^T(Az+Bv) \\ &\triangleq L[F^{-1}(z),\ \alpha(F^{-1}(z))+\beta(F^{-1}(z))v,\ t] \;+\; p^T(Az+Bv) \end{aligned} \tag{4.16}$$

and are given by

$$\dot{z} = Az + Bv \tag{4.17}$$

$$\dot{p} = -A^Tp - \left(\frac{\partial \bar{L}}{\partial z}\right)^T \tag{4.18}$$

$$\frac{\partial \bar{\mathcal{H}}}{\partial v} = 0 \tag{4.19}$$

$$z(t_0) = z_0 = F(x_0) \tag{4.20}$$

$$p(t_f) = \left[\frac{\partial \bar{S}}{\partial z} + \nu^T\frac{\partial \bar{C}}{\partial z}\right]^T_{t_f} \tag{4.21}$$

$$\bar{C}(z(t_f),t_f) = 0 \tag{4.22}$$

In the following discussion, it is shown that if (z,v) satisfies the above conditions and the two systems are equivalent along $z(t)$, then the state-control pair (x,u) derived from (4.7) satisfies the necessary conditions (4.10) through (4.15). Note that because the relations (4.7) are invertible, the converse must also hold. The necessary conditions are transformed explicitly in order to highlight the relationships between the two problems. A fairly obvious sufficient condition is also presented that only relies on the continuity of the linearizing transformation. This sufficient condition justifies solving the original optimization problem by solving the problem with linear state dynamics, and then transforming the solution into the original coordinates.

The fact that the state history of the nonlinear system is related to the solution of the linear system should be obvious from the definition of the linearizing transformation. Direct substitution of the feedback transformation into (4.17) yields

$$\begin{aligned} [\mathbf{D}F(x)]\dot{x} &= [\mathbf{D}F](f + G\alpha)(x) + [\mathbf{D}F]G\beta(x)v \\ \dot{x} &= f(x) + G(x)[\alpha(x) + \beta(x)v] \\ &= f(x) + G(x)u \end{aligned} \tag{4.23}$$

The real matrix $\mathbf{D}F$ denotes the derivative of the vector valued function F. Later results will make use of the fact that $\mathbf{D}F$ is nonsingular and $\frac{\partial F^{-1}}{\partial z} = \left[\frac{\partial F}{\partial x}\right]^{-1}$ (see the appendix on feedback linearization for more detail). Since $x(t_0) = F^{-1}(z(t_0))$ and $u(t) = \alpha + \beta v(t)$, the trajectories must be related by $x = F^{-1}(z)$ on the interval $[t_0, t_f]$.

The equation for the costate p is obtained from the derivative of the Hamiltonian (4.16) with respect to the state variable z. This derivative can be expanded to yield

$$\begin{aligned} \dot{p}^T &= -p^T \frac{\partial}{\partial z}\{Az + Bv\} - \frac{\partial \bar{L}}{\partial z} \\ &= -p^T \frac{\partial}{\partial z}\{[\mathbf{D}F](f + G\alpha) + [\mathbf{D}F]G\beta v\} \circ F^{-1}(z) - \frac{\partial L}{\partial x}[\mathbf{D}F^{-1}] \circ F^{-1}(z) \\ &\quad - [\frac{\partial L}{\partial u}][\frac{\partial \alpha}{\partial x} + \sum_{i=1}^{m} v_i \frac{\partial \beta_i}{\partial x}][\mathbf{D}F^{-1}] \circ F^{-1}(z) \end{aligned}$$

$$\begin{aligned}
&= -p^T\frac{\partial}{\partial x}\{[\mathbf{D}F](f+G\alpha)+[\mathbf{D}F]G\beta v\}[\mathbf{D}F]^{-1} \;-\; \frac{\partial L}{\partial x}[\mathbf{D}F]^{-1} \\
&\quad -[\frac{\partial L}{\partial u}][\frac{\partial \alpha}{\partial x}+\sum_{i=1}^{m} v_i\frac{\partial \beta_i}{\partial x}][\mathbf{D}F]^{-1} \\
\dot{p}^T[\mathbf{D}F] &= -p^T\frac{\partial}{\partial x}\{[\mathbf{D}F](f+G\alpha)+[\mathbf{D}F]G\beta v\} \;-\; \frac{\partial L}{\partial x} \\
&\quad + p^T[\mathbf{D}F]G\frac{\partial}{\partial x}\{\alpha+\beta v\} \qquad (4.24)
\end{aligned}$$

The last equality follows since (4.19) and 4.16 imply

$$\frac{\partial L}{\partial u}\beta+p^TB \;=\; \frac{\partial L}{\partial u}\beta+p^T[\mathbf{D}F]G\beta \;=\; 0 \qquad (4.25)$$

Extending the usage of the Liebnitz notation to include

$$\begin{aligned}
\frac{\partial}{\partial x}[\mathbf{D}F\cdot G\alpha] &= \frac{\partial}{\partial x}[\mathbf{D}F\cdot G]\alpha \;+\; [\mathbf{D}F\cdot G]\frac{\partial}{\partial x}[\alpha] \\
&\triangleq \sum_{i=1}^{m}[\mathbf{D}F\cdot G]_i\alpha_i \;+\; [\mathbf{D}F\cdot G][\mathbf{D}\alpha] \qquad (4.26)
\end{aligned}$$

(4.24) can be further simplified

$$\begin{aligned}
\dot{p}^T[\mathbf{D}F] &= -p^T\frac{\partial}{\partial x}\{\mathbf{D}F\cdot f \;+\; \mathbf{D}F\cdot G\alpha \;+\; \mathbf{D}F\cdot G\beta v\} \\
&\quad +p^T\left\{\frac{\partial}{\partial x}[\mathbf{D}F\cdot G\alpha]-\frac{\partial}{\partial x}[\mathbf{D}F\cdot G]\alpha\right\} \\
&\quad +p^T\left\{\frac{\partial}{\partial x}[\mathbf{D}F\cdot G\beta v]-\frac{\partial}{\partial x}[\mathbf{D}F\cdot G]\beta v\right\} \;-\; \frac{\partial L}{\partial x} \\
&= -p^T\frac{\partial}{\partial x}\left\{\,\mathbf{D}F\cdot f \;+\; \mathbf{D}F\cdot Gu\right\} \;-\; \frac{\partial L}{\partial x} \\
&= -p^T\left\{\left[\frac{\partial}{\partial x}[\mathbf{D}F]\right]\dot{x} \;-\; [\mathbf{D}F]\frac{\partial}{\partial x}(f \;+\; Gu)\right\} \;-\; \frac{\partial L}{\partial x} \qquad (4.27)
\end{aligned}$$

The costate equation (4.11) is recovered exactly once the substitution

$$\lambda^T = p^T \mathbf{D}F \tag{4.28}$$

is made in the last equation. The terminal condition on λ can be obtained from (4.21) by applying the transformation $z = F^{-1}(x)$ and substitution of (4.28) for $p^T\mathbf{D}F$.

The requirement (4.12) follows from (4.19) by

$$0 = \frac{\partial \bar{\mathcal{H}}}{\partial v} = \frac{\partial \mathcal{H}}{\partial u}\frac{\partial u}{\partial v} = \frac{\partial \mathcal{H}}{\partial u}\beta \tag{4.29}$$

Since the matrix β is nonsingular, multiplication on the right by its inverse yields the desired result, $\frac{\partial \mathcal{H}}{\partial u} = 0$.

The preceeding argument shows that an optimal solution of the linearized problem maps to a solution of the nonlinear problem that satisfies the required necessary conditions. The additional conditions which guarantee that a solution is a minimum - rather than a saddle point or maximum - do not transform directly to similar conditions on the equivalent problem. These conditions generally involve statements about second or higher-order variations of the objective function, which in turn depend on higher-order derivatives of the transformations. The only assumption that has been made regarding the transfomations is that both maps have nonsingular first derivatives.

The difficulty with obtaining equivalent sufficient conditions is due in part to the fact that convexity is not preserved by the linearizing transformation. Consider, for example, a convex function on $\mathbf{R}$ that behaves like x near $x = 0$, and the local change of coordinates $x = \sin z$. Clearly, the composition of the convex function with the coordinate change cannot be convex at zero because the second derivative of $\sin z$ changes sign.

Fortunately, it is only necessary to show that the transformations cannot turn minima into maxima or saddle-point solutions. Let $J(z,v)$ denote the value of the objective function associated with the optimal solution (z,v). If this solution is a local minimum, then there is an open neighborhood of this solution on which $J(z,v) \leq J(z+\delta z, v+\delta v)$. Because the transformation $(z,v) \mapsto (x,u)$ is continuous, there is an open neighborhood of (x,u) on which $J(x,u) \leq J(x+\delta x, u+\delta u)$ and the solution is a local minimum. Note that unless a transfomation is globally linearizing, uniqueness of an optimal solution is not preserved by the class of transformations that are being considered.

4.2.2 Solution of Initial Value Problems by Collocation

Functional approximation techniques have long been used in engineering and science to obtain solutions for boundary value problems of the form

$$\mathcal{A}(x,t) = f(x,t) \quad \forall\, t \in D \tag{4.30}$$

$$\mathcal{B}(x) = g(t) \quad \forall\, t \in \partial D \tag{4.31}$$

In these equations, $\mathcal{A}$ denotes a differential operator on a domain D, and $\mathcal{B}$ is a constraint on the values of x imposed on the boundary ∂D. Such equations occur commonly as models for processes in heat transfer, fluid dynamics, and mechanics. For example, x might denote temperature, flow velocity, or strain, while t denotes a position in a medium. Approximate solutions to this system of equations can be obtained by expressing the solution as a weighted sum of functions in the independent variable

$$x^p(t) = \sum_{i=0}^{p} x_i \phi_i(t) \tag{4.32}$$

The functions $\phi_i(t)$ are usually chosen to have some desired properties. Often they are required to satisfy the boundary conditions of the problem for all choices of the coefficients x_i. A solution for the boundary value problem is obtained by substitution of (4.32) into (4.30) and choosing the x_i to minimize

$$\|\varepsilon(x_i)\| \triangleq \|\mathcal{A}(x^p, t) - f(x^p, t)\| \tag{4.33}$$

This is a straightforward problem in functional approximation [46]. If the operators $\mathcal{A}$ and f are linear, a solution can be obtained by simply matching coefficients of the function inside the norm symbols. If it is required that the error residual ε be orthogonal to the approximating functions, then the solution procedure the well-known *Galerkin method* [46]. Another technique is to require that the error function ε be identically zero on a selected set of points. This method is called *collocation* or *interpolation* [46]. Note that each of these techniques can be classified in some sense as a particular example of the *Method of Weighted Residuals* [47].

Note that the initial value problem associated with the optimization (4.5) is a special case of (4.30) and (4.31), and the methods described above can be applied to obtain a solution. These methods are not commonly applied to initial value problems since there are many simple forward differencing algorithms that yield accurate results. Also, approximation methods require many terms of the approximating series for accurate results when the exact solution is not smooth. Experience gained in this research, as well as comments by a number of authors, [48] indicate that when the functions involved are smooth, the approximation methods very often give more accuracy for a given level of computational effort than differencing methods.

Collocation methods are particularly simple to apply to the solution of initial value problems of the form [49],[50].

$$\dot{x} \; = \; f(x,u) \, , \quad x(t_0) = x_0 \tag{4.34}$$

Approximating $x(t)$ and $u(t)$ using (4.32), and substitution into the previous expression yields

$$\sum_{i=0}^{p} x_i \, \dot{\phi}_i(t) \;\approx\; f(x_1,\ldots,x_p,u_1,\ldots,u_p,t) \tag{4.35}$$

$$\sum_{i=0}^{p} x_i \, \phi_i(t_0) \;=\; x_0 \tag{4.36}$$

Though it is not necessary to satisfy the the initial condition exactly, it seems reasonable to do so. The collocation procedure is to choose a set of times, $t_1,\ldots,t_p$, at which the trial functions, ϕ_i, are independent, and then chose the coefficients x_i to satisfy (4.35) exactly at those times, while picking x_0 such that the initial condition (4.36) is satisfied. If the trial functions are complete in an appropriate norm, then the truncated solution converges to the exact solution of the equation as p increases. Using this procedure, differential equations are transformed into systems of algebraic equations

$$0 \;=\; \sum_{i=0}^{p} x_i \, \dot{\phi}_i(t_1) \;-\; f(x_1,\ldots,x_p,u_1,\ldots,u_p,t_1)$$

$$\begin{aligned} &\vdots && (4.37) \\ 0 &= \sum_{i=0}^{p} x_i\,\dot{\phi}_i(t_p) - f(x_1,\ldots,x_p,u_1,\ldots,u_p,t_p) \\ 0 &= \sum_{i=0}^{p} x_i\,\phi_i(t_0) - x_0 && (4.38) \end{aligned}$$

that can be solved using readily available numerical procedures. If x is n-dimensional, then (4.37) with the constriant on the initial value (4.38) will consist of $n(p+1)$ equations in $n(p+1)$ unknowns.

When the initial value problem is linear and $x \in \mathbf{R}^n$, it is convenient to write the approximation (4.32) in the form

$$x(t) \approx \Phi_x(t)X \triangleq \left[\phi_0(t)I_n \middle| \cdots \middle| \phi_p(t)I_n\right] \begin{bmatrix} x_0 \\ \vdots \\ x_p \end{bmatrix} \qquad (4.39)$$

Approximating the inputs in a similar fashion, $u(t) \approx \Phi_u(t)U$, the differential equation is satified exactly at t_j if

$$\left[\dot{\Phi}_x(t_j) - A\Phi_x(t_j)\right] X = B\Phi_u(t_j)U \qquad (4.40)$$

If the trial functions $\phi_i(t)$ are independent at the initial time and at the collocation points $t_1,\ldots,t_p$, then the requirement that the initial condition be satisfied, together with the system of linear equations

$$\begin{bmatrix} \left[\dot{\Phi}_x - A\Phi_x\right](t_1) \\ \vdots \\ \left[\dot{\Phi}_x - A\Phi_x\right](t_p) \end{bmatrix} X = \begin{bmatrix} B\Phi_u(t_1) & & 0 \\ & \ddots & \\ 0 & & B\Phi_u(t_p) \end{bmatrix} U \qquad (4.41)$$

has a unique solution X that can be found by inverting the square matrix on the left.

4.2.3 Transformation of the Optimization Problem to an Unconstrained Nonlinear Program

In this section, linearizing transformations and collocation are exploited to obtain suboptimal solutions to the optimization problem (4.5). The basic idea is to use feedback linearization to transform the optimization (4.5) into (4.8), and then to sustitute an approximation of the form (4.32) into (4.8). The constraints become algebraic equations, and the integral performance measure can be evaluated using a suitable quadrature formula. In this way, the optimization problem is transformed into a nonlinear programming problem. In the absence of terminal constraints, the linear equations for the state approximation, Z, can be solved in terms of the approximation to the control V. If there are nonlinear terminal constraints, they can be incorporated into the objective function using Lagrange multipliers. In either case, the resulting programming problem is unconstrained.

A number of variations to this basic approach have already appeared in the literature. The methods differ primarily in the way in which the original problem is approximated, and in the numerical method used to solve the problem. Solutions have been approximated using orthogonal polynomials, splines, and Fourier series. In some cases, both the control and the state variables are parameterized, and a solution is obtained by solving the resulting constrained nonlinear program. In [51], the powerful method of successive quadratic programming [52] is applied directly to the constrained nonlinear program. Because the authors use cubic splines to approximate the time functions on intervals, it is necessary to satisfy rather cumbersome compatibility conditions at the end points. These conditions require that four equations be solved for each collocation point. Another procedure incorporates the constraint into the objective function using Lagrange multipliers [48]. The multipliers effectively double the size of the programming problem, which may already be quite large ($\sim np$ variables requiring $\sim (np)^2$ storage).

Other techniques parameterize only the state variables or only the control variables and solve the resulting algebraic equations for the control or state history, respectively. In [27], truncated *Fourier* series are used to parameterize the state trajectory for a class of systems of the form

$$F(x, \dot{x}, \ddot{x}) = u \tag{4.42}$$

The simple form of the system equations makes it possible to eliminate the control from the optimization problem. Two difficulties with this approach are that completeness of the approximating series becomes a delicate issue when parameterizing the state history, and a number of technical difficulties arise when the equations are over- or under-determined in the control variable (especially under-determined).

Note that the choice of approximating polynomials is not critical to the solution of the initial value problem by the collocation method. Any set of functions which are independent and complete in an appropriate sense will work. In the published literature, *Chebyshev* polynomials are used quite extensively [49],[48][1] (see [53] for a detailed discussion of the approach using these functions). In certain circumstances, the particular problem at hand may suggest a choice of functions that should yield accurate results [54](boundary value problems), or perhaps simplifies the computations. The polynomials that have been chosen for use here are the interpolating polynomials given by

$$\ell_i(t) = \prod_{\substack{j=0 \\ i \neq j}}^{p} \frac{t - t_j}{t_i - t_j} \quad i = 0, \ldots, p \tag{4.43}$$

and the polynomial approximation to the state is given by

$$x(t) \approx \sum_{k=0}^{p} x_k \, \ell_k(t) \tag{4.44}$$

It is assumed that t_0 is the initial time and the coefficient x_0 is chosen to satisfy the constraint at the initial time, $x_0 = x(t_0)$. The x_i for $i \geq 1$ are to be chosen by collocation at the times t_j, for $1 \leq j \leq p$, which are chosen to be the zeros of a p^{th}-order *Legendre polynomial* defined on the interval $[t_0, t_f]$. This choice of collocation times was made because of the fact that the product defining the interpolating polynomials ranged from 1 to p, and these zeros were used for the times t_j, then the resulting set of functions would be orthogonal on the interval $[t_0, t_f]$. In addition, because the zeros are more closely spaced at the end points of the interval, the approximation is better behaved near the end points of the interval (see [46] and the discussion of the Runge phenomenon).

[1] Note that the dates on these references are 1956 and 1988, respectively.

Note that it is also possible to include the final time, t_f, as one of the interpolation times. In some cases, this was found to improve the *appearance* of the plotted solutions by making the last segment of the time histories better behaved. The effect on the value of the objective function was negligible, provided the choice of p was sufficiently large. Including the final time as a collocation point was found to be undesirable when terminal costs were included in the cost function. This tended to make the gradient of the objective function very steep in a particular direction, and adversely affected convergence of the algorithm.

The advantage of the interpolating polynomials derives from the fact that $\ell_i(t_j) = 0$ unless $i = j$, which implies that the value of the approximation at a collocation point t_i depends only on x_i. The *collocated approximations* to state dependent functions, such as the cost functional and the linearizing transformations, can be evaluated by considering only one finite set of data, (z_i, v_i), at a time. The linearizing transformation as well as the approximate solution of the nonlinear dynamics can be evaluated using

$$\begin{aligned}
x_i &\triangleq F^{-1}(z_i) && (4.45)\\
\alpha(x(t)) &\approx \sum_{i=0}^{p} \alpha(x_i)\,\ell_i(t) && (4.46)\\
\beta(x(t)) &\approx \sum_{i=0}^{p} \beta(x_i)\,\ell_i(t) && (4.47)\\
x(t) &\approx \sum_{i=0}^{p} F^{-1}(z_i)\,\ell_i(t) && (4.48)\\
u(t) &\approx \sum_{i=0}^{p} (\alpha(x_i) + \beta(x_i)v_i)\,\ell_i(t) && (4.49)\\
& && (4.50)
\end{aligned}$$

A similar set of expressions exist to invert these transformations. Computing the approximation of the transformation $(z,v) \mapsto (x,u)$ reduces to solving p independent sets of n equations in n unknowns, as opposed to one system of np equations that would result if another set of polynomials had been chosen[2].

It is now simple to state the nonlinear program which yields the approximate solution to (4.5). Letting $L_i \triangleq L(x_i, u_i, t_i)$, the integral performance index is approximated by

$$\begin{aligned}
\int_{t_0}^{t_f} L(x,u,t)dt &\approx \sum_{i=0}^{p} L_i \left(\int_{t_0}^{t_f} \ell_i(t)dt\right) \\
&\triangleq \sum_{i=0}^{p} \bar{L}(z_i, v_i, t_i) \left(\int_{t_0}^{t_f} \ell_i(t)dt\right) \\
&\triangleq \sum_{i=0}^{p} \bar{L}(z_i, v_i, t_i)\, \bar{\ell}_i \qquad (4.51) \\
& \qquad (4.52)
\end{aligned}$$

The integral of $\ell_i(t)$ has been evaluated numerically for $p < 20$ and is positive for all i. A formal proof that this is true for all p is not possible since a closed-form solution for the interpolation times is not available; however, inspection of the functions involved suggests that this is true for larger p. The terminal cost can be evaluated as $\bar{S}(z(t_f), t_f)$, and the terminal time must satisfy the constraints $\bar{C}(z(t_f), t_f)$. Since the z_i are derived from a linear matrix equation of the form $Z = Z_0 + D\, V$, the approximate cost and the terminal constraints can be expressed entirely in terms of the mp-dimensional vector V. Letting λ denote a vector of Lagrange multipliers, the nonlinear programming problem to be solved for a suboptimal solution to (4.5)

[2] The initial time is not counted with the others

is

$$\min_{\lambda,\, V} \sum_{i=0}^{p} \bar{L}(z_i, v_i, t_i)\, \bar{\ell}_i \; + \; \bar{S}(z(t_f), t_f) \; + \; \lambda^T \bar{C}(z(t_f), t_f) \tag{4.53}$$

In the following section, the optimal momentum management problem is expressed in this form, and a computational algorithm is outlined to solve the nonlinear program.

4.3 Solution of the Optimal Momentum Management Problem

In this section, the technique of the previous section is applied to obtain suboptimal solutions to the optimal momentum management problem (4.4). The integral quadratic performance measure can be converted to an algebraic function using the polynomial approximation. Letting $e(t) \approx \sum_{i=0}^{p} e_i \ell(t) \stackrel{\triangle}{=} \mathcal{L}(t) E$, and considering only one of the terms in the performance index (4.4), the criteria can be written as

$$\begin{aligned}
\int_0^T e(t)^T Q\, e(t)\, dt \;&\approx\; \int_0^T E^T \mathcal{L}^T(t)\, Q\, \mathcal{L}(t)\, E\, dt \\
&= E^T \left[\int_0^T \mathcal{L}^T(t)\, Q^{\frac{1}{2}}\, Q^{\frac{1}{2}} \mathcal{L}(t)\, dt \right] E
\end{aligned} \tag{4.54}$$

$$\begin{aligned}
&= E^T \begin{bmatrix} Q^{\frac{1}{2}} & & 0 \\ & \ddots & \\ 0 & & Q^{\frac{1}{2}} \end{bmatrix} \\
&\quad \left\{ \int_0^T \begin{bmatrix} \ell_0(t)\, I_n \\ \vdots \\ \ell_p(t)\, I_n \end{bmatrix} \left[\; \ell_0(t)\, I_n \quad \cdots \quad \ell_p(t)\, I_n \;\right] dt \right\} \\
&\quad \begin{bmatrix} Q^{\frac{1}{2}} & & 0 \\ & \ddots & \\ 0 & & Q^{\frac{1}{2}} \end{bmatrix} E
\end{aligned} \tag{4.55}$$

$$\triangleq \; E^T \bar{Q}\, \bar{Q}\, E \tag{4.56}$$

$$\triangleq \; \Psi_e^T(Z)\, \bar{Q}\, \bar{Q}\, \Psi_e(Z) \tag{4.57}$$

Because E is a vector of constants, it can be taken outside the integral in (4.54). If Q is a function of t, then this equation can be integrated during the initialization of the optimization algorithm to obtain a constant matrix. With Q assumed to be constant, it can also be extracted from under the integral by rewriting the matrix product. In (4.55), the integral depends only on products of the approximating polynomials, and a closed-form solution is easily obtained for each fixed value of p. The fact that only one of the interpolating polynomials is non-zero at each of the collocation points can be used to express $\Psi_e(Z)$ in the form

$$X \; - \; X_d \; \triangleq \; \Psi_e(Z) \; = \; \left\{ \begin{array}{c} F^{-1}(z_0) \\ \vdots \\ F^{-1}(z_p) \end{array} \right\} \tag{4.58}$$

An expression similar to (4.57) can be derived for the integral of the term quadratic in the control in (4.4). This expression has the form

$$\int_0^T \; u^T(t)\, R\, u(t)\, dt \; \approx \; \Psi_u^T(V)\, \bar{R}\, \bar{R} \Psi_u(V) \tag{4.59}$$

where

$$\Psi_u(V) \; \triangleq \; \left\{ \begin{array}{c} \alpha(F^{-1}(z_0)) \; + \; \beta(F^{-1}(z_0))\, v_0 \\ \vdots \\ \alpha(F^{-1}(z_p)) \; + \; \beta(F^{-1}(z_p))\, v_{p+1} \end{array} \right\} \tag{4.60}$$

Finally, the cost for the terminal error is simply expressed as

$$e^T(T)\, P\, e(T) \;=\; E^T\, \mathcal{L}^T(T)\, P^{\frac{1}{2}}\, P^{\frac{1}{2}}\, E\, \mathcal{L}(T) \tag{4.61}$$

Note that each of the terms in the cost functional has been reduced to a nonlinear quadratic form which can be expressed as

$$\min_{V}\; F(V) \;\triangleq\; \min_{V} \sum_{i=1}^{n(p+2)+m(p+1)} f_i \cdot f_i \tag{4.62}$$

where the f_i are the components of a vector valued function f.

Figure 4.1 shows a flow chart of the computational algorithm that has been implemented to solve the optimal momentum management problem. The initialization routine reads data from an external file and computes an initial guess for the optimal solution. Along with initial state data, a data file is read that includes the integrated polynomial products $\ell_i(t)\, \ell_j(t)$, and the matrix W that solves the collocation problem for the state history $Z = W(Z_0 + V)$. The initial guess is obtained by initializing the state trajectory of the spacecraft so that the vehicle follows a torque equilibrium attitude, the momentum remains fixed in inertial coordinates, and the CMG gimbals rotate to hold the rotors fixed in inertial space. This solution is transformed to an equivalent state history for the linearized system, and the control producing the best approximation to this solution in a least-squares sense is the starting solution.

The main sequencing routine then calls a subroutine DUNLSJ[3] that solves the nonlinear quadratic programming problem using the Lavenberg-Marquardt algo-

[3] IMSL subroutine library, version 10.0

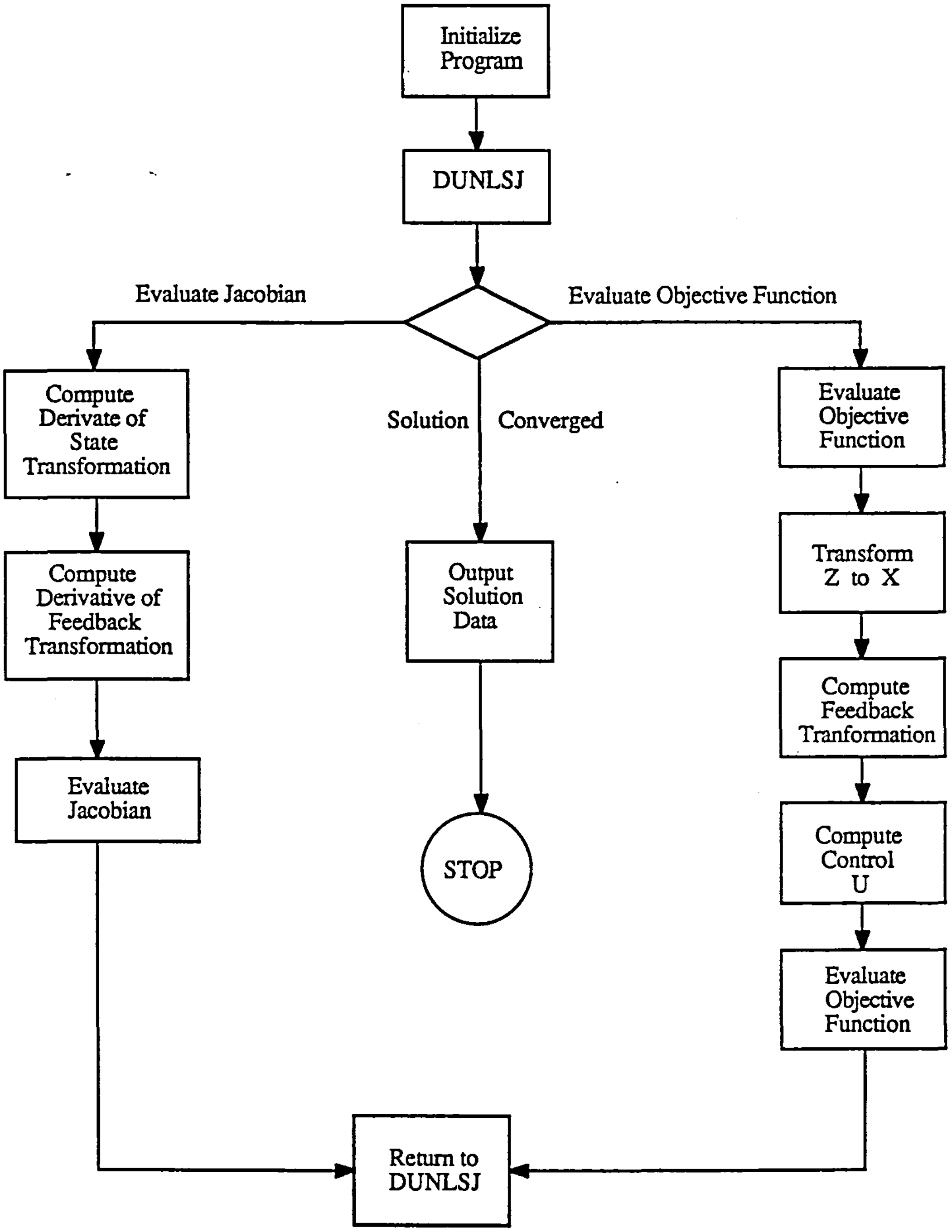

Initialize Program
DUNLSJ
Evaluate Jacobian
Evaluate Objective Function
Compute Derivate of State Transformation
Solution
Converged
Evaluate Objective Function
Compute Derivative of Feedback Transformation
Output Solution Data
Transform Z to X
Compute Feedback Tranformation
Evaluate Jacobian
STOP
Compute Control U
Evaluate Objective Function
Return to DUNLSJ

Figure 4.1: Flow Chart of Optimal Momentum Mangagement Algorithm.

rithm. This routine calls two subroutines that compute the components of the function f in (4.62) as well as the Jacobian of f. The function is evaluated in several steps. First, a solution to the linear system dynamics is obtained, and this solution is transformed to obtain the solution in terms of the approximation to the nonlinear system dynamics X. Then the feedback transformation is computed, and the vector U is evaluated. The components of the function f are evaluated and execution is returned to the optimization subroutine. As mentioned above, the routine DUNLSJ also calls a subroutine that evaluates the Jacobian of the function f. The solution is assumed to have converged when one of the following three conditions holds: i) The value of the function F(V) is below a certain threshold, ii) the gradient of the function is below a certain threshold, and iii) three successive iterations have failed to improve the objective function by more than one percent. In the following sections, results are presented that have been generated using this program.

4.4 Sample Results of Optimal Momentum Management

In this section, a number of examples are presented to illustrate the benefits obtained by solving the optimal momentum management problem. The primary advantage of the proposed approach is that issues relating to attitude control and momentum management are treated in a combined framework which includes a nonlinear vehicle model.

A number of continuous time, linear optimal momentum management schemes have appeared in the literature [13],[14],[55]. These schemes assume that both attitude and momentum excursions will remain small. Assuming that the spacecraft is to track a torque equilibrium attitude, the former assumption is probably valid. The latter assumption is generally not justified; results presented in this section as well as [13],[14] themselves indicate variations in net vehicle momentum of the same order of magnitude as the nominal values.

Another approach to momentum management involves computing a specified amount of momentum that should be transferred during an orbit. The desired momentum transfer is achieved by manipulating the attitude dependent gravity gradient torque [15],[55]. The momentum to be transferred is computed by adding the amount of momentum currently stored in the CMGs to the momentum that is expected to accumulate over the next orbit. These methods consider neither the constraints imposed by vehicle dynamics nor advantageous ways to exploit vehicle dynamics. Both cited references neglect the fact that the inertial orientation of the gravity gradient torque is changing continuously. Since these methods are also based on a linearized model, they are not applicable when there are large attitude excursions.

One of the most important features of the optimal momentum management algorithm is the way in which the solutions exploit the total vehicle momentum iself for regulating both momentum and attitude. This feature is highlighted in the first set of examples in which optimal solutions are obtained with the spacecraft subjected to various environmental torques. In the first example, only the gravity gradient torque is considered. The next example includes a constant bias torque about the yaw axis, while in the third example, the torque was chosen to be representative of aerodynamic disturbances the space station is likely to encounter on-orbit. The final two examples show solutions in which the CMGs are initialized with net momentum storage in the pitch and yaw directions.

The spacecraft model used in the examples presented in this section is the same as that employed in the steering examples of the previous chapter. The vehicle inertia properties are listed in Table 3.2 and represent data for the rephased dual-keel space station, after assembly sequence fight 7 [40]. This vehicle is illustrated schematically in Figure 3.2, and the CMG mounting configuration is shown in Figure 3.3.

In each of the examples to be discussed, the vehicle is in a circular orbit at an altitude of *300 km*, and the orbital period is *90 minutes*. The momentum management problem is solved for one orbital period. The desired solution is computed by assuming that the vehicle is to track the torque equilibrium attitude, and that the CMGs are to track a zero momentum state. For the vehicle data used in this example, the gravity gradient torque required the equilibrium attitude to be pitched almost *-17 degrees* from the local vertical. The desired yaw attitude is assumed to be zero. In order to track an orientation that is fixed in the LVLH reference frame, the vehicle must rotate with respect to an inertial frame at a rate equal to the LVLH or orbital rate. As a consequence, the momentum of the spacecraft will not be zero, and because of assymetries in the mass distribution, the vehicle's angular momentum need not align with the LVLH rate. The desired momentum at the initial time is computed by determining the momentum of the vehicle as if it were fixed at the desired attitude, and rotating with the LVLH frame. The desired momentum trajectory is then computed such that the momentum at the initial time remains constant in inertial coordinates as the LVLH frame rotates. The desired gimbal angle history is computed by assuming that the CMGs are to track the zero momentum configuration initialized with gimbal angles (in radians)

$$\begin{array}{rcl} \theta^1_{inner} & = & 0.1 \\ \theta^2_{inner} & = & -0.1 \\ \theta^3_{inner} & = & 0.1 \\ \theta^4_{inner} & = & -0.1 \end{array} \quad \text{and} \quad \begin{array}{rcl} \theta^1_{outer} & = & 0 \\ \theta^2_{outer} & = & \frac{\pi}{2} \\ \theta^3_{outer} & = & \frac{3\pi}{2} \\ \theta^4_{outer} & = & \pi \end{array} \tag{4.63}$$

The outer gimbal angles are assumed to rotate at the orbital rate. Because the CMGs are mounted with outer gimbal axes along the negative vehicle pitch axis (which aligns with the orbital rate), the desired trajectory causes the individual rotors to remain fixed in inertial coordinates.

It should be remarked that a number of the optimal solutions exhibit trends near the terminal time that seem to be inconsistent with *optimal* behavior. In particular, there can be large excursions in the state errors and controls near the final time. These appear to be due to properties of the polynomials being employed, and are probably related to the Runge phenomenon that occurs in polynomial approximation [46]. For a given optimization problem, the precise behavior at the final time was seen to change noticably as the order of the approximation increased. This occured even though the magnitude of the objective function appeared to have converged quite close to the optimal value, and differences between solutions were hardly discernable, except near the endpoint. Including an interpolation point at the final time improved the behavior there, but slowed convergence of the objective function. There are two potential remedies for this difficulty; the first is to include a penalty on the final error, and the second is to extend the optimization beyond the orbital period to obtain useful solutions for the time interval of interest.

4.4.1 Baseline Solutions Illustrating Required Computational Time

Before considering specific examples, a generic set of runs are investigated to illustrate how the amount of processor time required to obtain solutions to the nonlinear program varies as the order of the approximation, p, was increased. In these cases, it is assumed that only gravity gradient torques act on the vehicle, and that the initial attitude, as well as the desired attitude, correspond to the equilibrium orientation with zero yaw. The desired momentum trajectory is computed by requiring that the momentum should remain constant in inertial coordinates. The desired gimbal trajectory was generated by rotating the CMGs about their outer gimbal axes at the orbital rate. The penalty on the terminal error, P, is set to zero, the weighting on the control, R, is a diagonal matrix of 10^{+4}, and the error weighting, Q, is a diagonal matrix of 10^{-4} for the momentum errors, 10^{+4} for the attitude errors, and 10^{+2} for the gimbal angle errors. These weightings were chosen so that the contribution to the total cost associated with the attitude error, momentum error, gimbal angle error, and control effort are of approximately the same order of magnitude in the absence of disturbance torques.

Table 4.1 shows the variation of the required computation time and the value of the objective function as the order of the approximation is increased. The CPU times are for a Digital Equipment Corporation VAX 8600[4]. Note that the solutions for $p = 5, 6$ have the same value of the objective function, even though the value should decrease with increasing p. This is probably due to the fact that for $p < 6$, the algorithm terminates when the gradient of the objective crosses a threshold. For larger values, it terminates when three iterations fail to reduce the objective by more than one percent. The CPU time required for the case with $p = 7$ to converge is clearly inconsistent with what should have been expected; although this was observed in another situation, no satisfactory explanation has been found.

p	CPU time	Objective	% Improvement
3	9.72	3174	-
4	17.94	2781	12.5
5	21.74	1603	42.4
6	27.79	1603	0.0
7	90.71	1588	0.9
8	80.61	1586	0.1
9	124.63	1584	0.1

Table 4.1: Nominal CPU times require to solve optimal momentum management problem.

The approximation using $p = 8$ has been used most extensively, and required computation times were typically less than the *80 sec.* indicated in Table 4.1. It was generally more common to see times in the range *40 - 60 sec.* for $p = 8$, and

[4] Conversion factor to equivalent VAX 11/785 CPU seconds is ≈ 3.5

the numbers shown in the table might be regarded as approximate upper bounds. It is interesting to note that the required convergence time was not very dependent on how near the initial guess was to the optimal solution. The number of iterations tended to increase with p, but was fairly consistent for a fixed p, and almost never exceeded 10. Because of a number of large matrix computations that must be performed, the percent increase in CPU time is larger than the percent increase in p.

Extensive experience using the computer program developed to solve the optimal momentum management program has shown it to be very robust. In contrast to methods that solve the two-point boundary value problems arising in otpimal control, the algorithm developed here was found to be quite insesitive to the choice of starting conditions. In all the results presented here, the same initial guess is used to start the omptimization. When the algorithm did fail to converge, it was generally necessary to vary only a parameter that limited the amount the solution could be changed for each iteration. While it was desired to keep this limit as large as possible, situations occured where the algorithm oscillated about the optimum or an initial step was so large that the linearizing transformations could not be computed.

4.4.2 Momentum Management with Gravity Gradient Torque Only

In this example, the momentum management problem is solved assuming that gravity gradient torques are the only external torques acting on the vehicle. The results in this example show how the optimal solutions use CMG momentum transfer and gravity gradient torques to offset the Euler torque acting on the vehicle. The example also provides a baseline for comparison with later results.

Because state excursions are small, it is reasonable to assume that a linearized model would yield approximately the same results shown here. Since the CMGs are initialized in a zero momentum state, and there are no external torques, a momentum manager like [15] based on a momentum dumping scheme would choose to follow the gravity gradient equilibrium attitude. Since the attitude control law of a vehicle using this type of momentum manager should compensate for Euler coupling, these results will also be similar. Note that both of these schemes assume that the steering law is causing the outer gimbals to rotate at the orbital rate. While a well designed steering law will have this feature, not all steering laws can guarantee this behavior, and momentum may accumulate in the CMG system.

For this example, the vehicle is initialized at the torque equilibrium attitude with the CMGs in the configuration (4.63). The total LVLH momentum components are plotted in Figure 4.2. This figure shows that the momentum rotates in the LVLH frame so that it remains fixed in inertial coordinates. Because the orbital rate aligns with the pitch component of momentum, this component remains constant. Since the initial momentum had a small component along the negative roll direction in the LVLH frame, the roll and yaw components appear as out-of-phase sine-waves due to the rotation of the LVLH reference frame along the orbit.

A plot of CMG momentum in LVLH coordinates is shown in Figure 4.3. Initially, the Euler torque due to rotation of the LVLH frame ($\omega \times I\omega$) is on the order of *-0.2 ft-lb* in the LVLH yaw direction, with a very small component in the positive roll direction. The Euler torque is caused by the rotation of the spacecraft's angular momentum in inertial coordinates. The spacecraft momentum is initially *287 ft-lb* in the negative roll direction in LVLH. To remain fixed in inertial coordinates, the momentum must rotate with respect to LVLH at a rate equal to the orbit rate. Since Euler coupling is not considered in computing the initial or desired solutions, the CMGs must dynamically compensate for its influence during the orbit. Figure 4.3 indicates that the CMGs transfer momentum at a rate approximately equal to the rate of change of the vector $I\omega$ in LVLH coordinates. This produces a torque on the spacecraft approximately opposite to the Euler torque.

The plot of attitude error, Figure 4.4, indicates that the optimal solution favors a positive displacement about yaw and roll. The displacement about roll causes a small positive bias in the gravity gradient torque about that axis. Because the desired attitude is gravity gradient unstable in both pitch and roll, positive deviations from the equilibrium result in positive gravity gradient torques. The effect of the gravity gradient torque together with the momentum associated with the spacecraft rotation causes the errors in total momentum seen in Figure 4.5. Because the gravity gradient torque is nearly constant in LVLH coordinates, the rotation of the LVLH frame yields variations in the yaw and roll components of total momentum that appear approximately as out-of-phase sine-waves. The momentum errors that appear in this figure cause the net momentum to become more aligned with the angular rate vector. This produces a reduction in the Euler coupling and a corresponding reduction in CMG control effort.

Note that some of the time histories in the figures appear to diverge near as the solutions approach the final time. This behavior is due to the nature of the approximating functions, and was discussed in some detail in the introduction to the examples. The behavior is also evident in the other examples presented in this chapter.

When a constant bias is applied in the roll or pitch directions, the optimal solutions are virtually identical to the solutions shown in this example. The reason for the similarity is that the optimal momentum manager adjusts the attitude about these axes so that the gravity gradient torque opposes the external torque on the vehicle. Since the gravity gradient torque is orthogonal to the yaw direction in LVLH, constant torques in this direction yield solutions that are noticably different as seen in the next example.

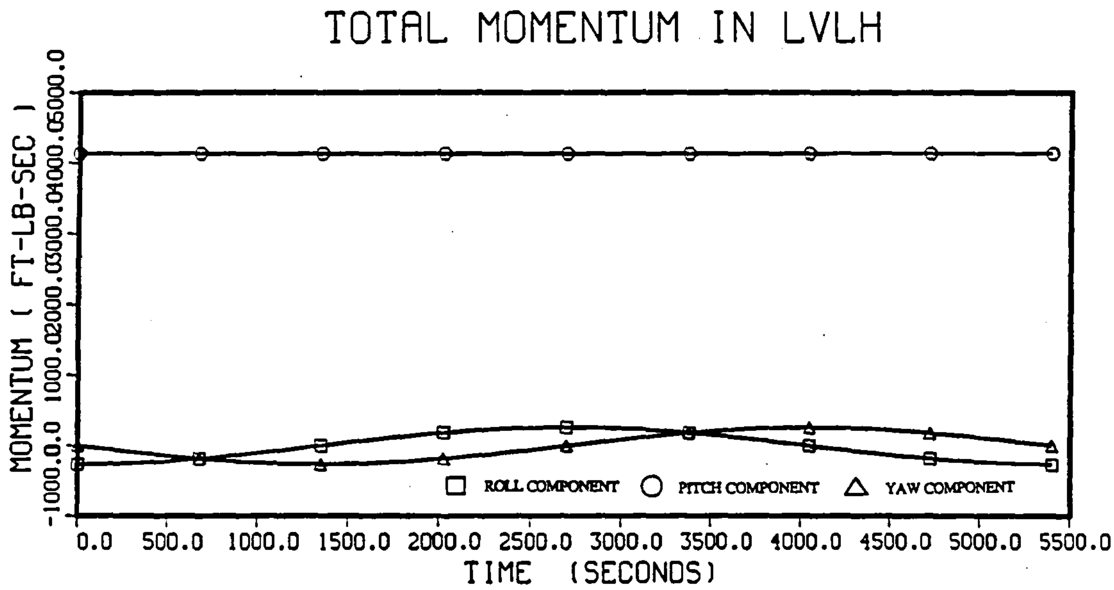


Figure 4.2: LVLH Momentum: Gravity Gradient Torque Only (513).

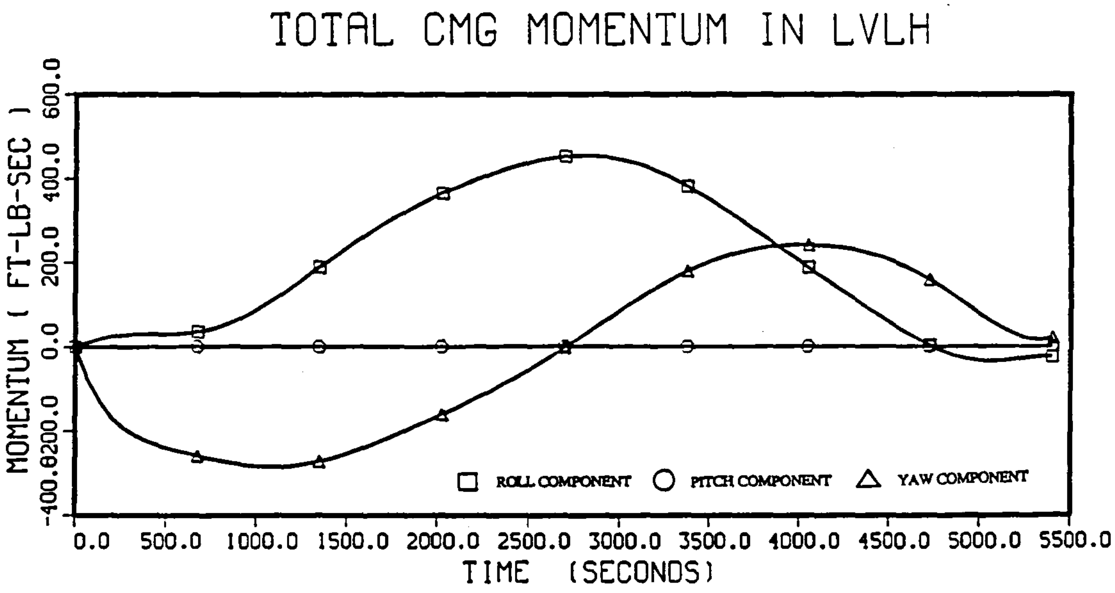


Figure 4.3: CMG Momentum in LVLH Frame: Gravity Gradient Torque Only (513).

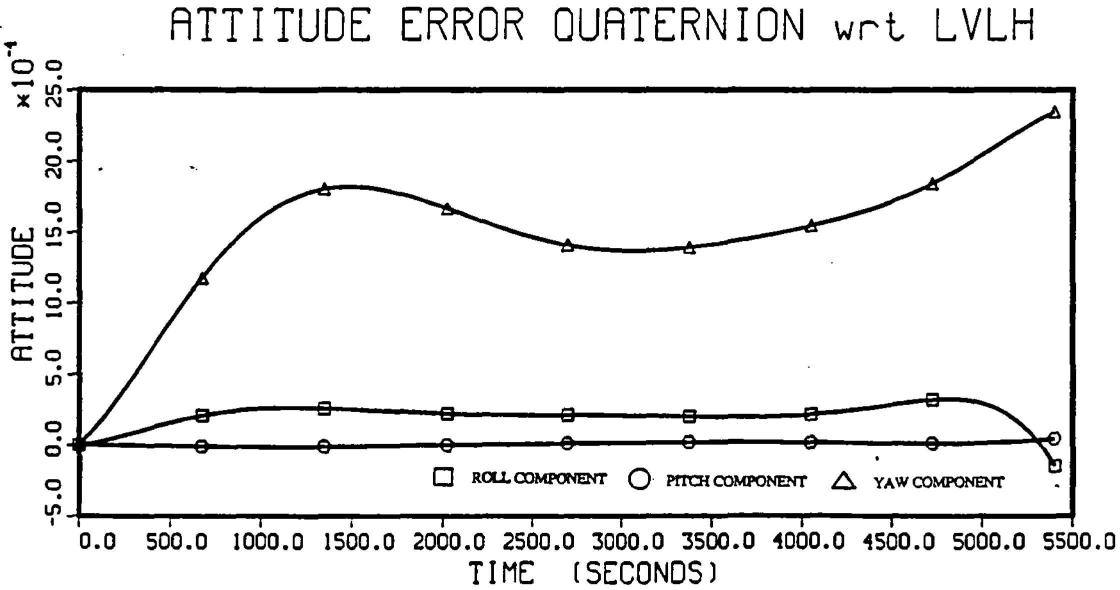


Figure 4.4: Attitude Errors: Gravity Gradient Torque Only (513).

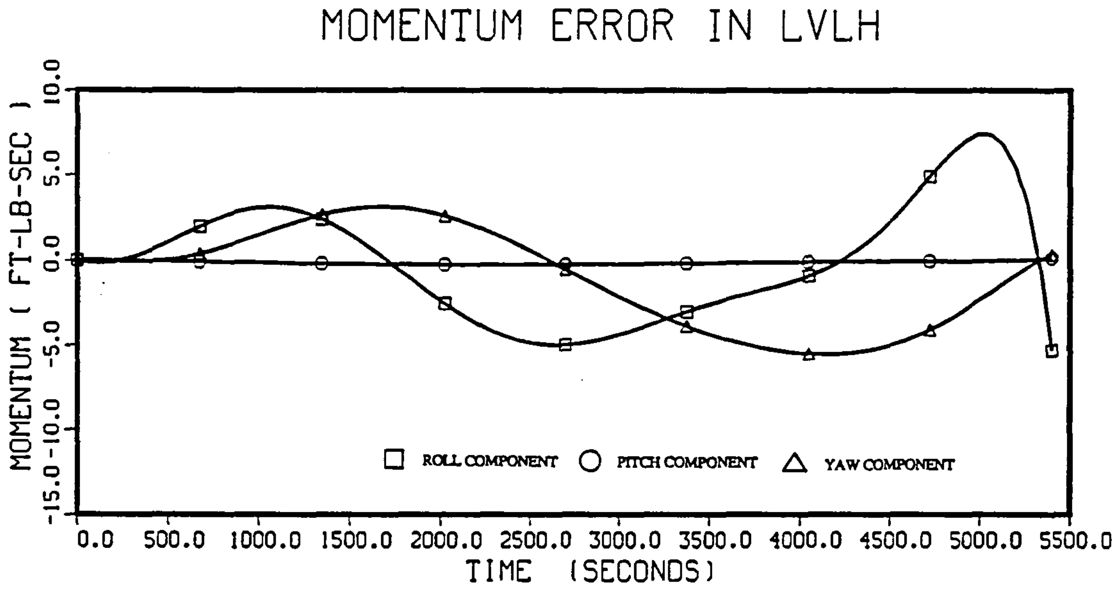


Figure 4.5: LVLH Momentum Error: Gravity Gradient Torque Only (513).

4.4.3 Momentum Management in the Presence of a Yaw Bias Torque

In this example, a constant torque of *0.1 ft-lb* is applied along the yaw direction in the LVLH reference frame. Torques are applied in LVLH coordinates rather that body coordinates because the momentum is expressed in this reference frame, and the results are more easily interpretted when it is not necessary to consider the influence of a nonzero attitude. The initial data and error penalties are the same as in the previous example, and the desired data are computed as described in the introduction to the examples.

Figure 4.6 shows that a momentum error of about *100 ft-lb-sec* quickly accumulates in the positive roll direction, along with a smaller error in the negative yaw direction. The reasons for this momentum transfer can be understood by examining the rate of change of momentum in inertial coordinates. If the effect of the external torque were not compensated for, the torque would cause the net momentum of the vehicle system to change at a rate of *0.1 ft-lb* in the positive yaw direction of the LVLH frame. The momentum transfer in the roll direction couples with the orbital rate to produce a momentum change in the negative yaw direction that approximately cancels the momentum transfer due to the external torque ($\omega_o \times h^L \approx -0.001 \cdot 100 \approx -0.1$). The necessary momentum transfer is produced by a positive deviation from the torque equilibrium attitude that can be seen in Figure 4.7.

Figure 4.7 also shows an attitude error accumulateing in the yaw direction, but is nulled before the final time. Since the error eventually stops increasing, the CMGs must be applying torques to the spacecraft to limit the attitude excursion. The influence of the CMGs can be understood from Figure 4.8 which shows the momentum transfer in body coordinates. Initially, The CMGs compensate for the external torque by transferring momentum into the negative roll and yaw directions. The momentum transferred in the negative roll direction produces the positive attitude error in yaw. Because the vehicle is unstable at the desired attitude, this error results in a positive torque in the LVLH frame that produces the momentum error seen in Figure 4.6. The momentum transfer in yaw produces a small positive displacement in yaw.

The attitude rate changes that occur at the beginning of the solution result in changes in the vehicle momentum approximately equal to the momentum transfer from the CMGs. Note that even a change of *500 ft-lb-sec* is small compared to the total momentum of the orbiting vehicle (about *4000 ft-lb-sec*), the net change in the pitch-yaw plane is about twice the nominal value. It is the momentum that lies in the plane perpendicular to the pitch axis that determines the amount of Euler coupling. A momentum management scheme based on a tangent linearization of the dynamical model would underestimate this torque by a factor of two.

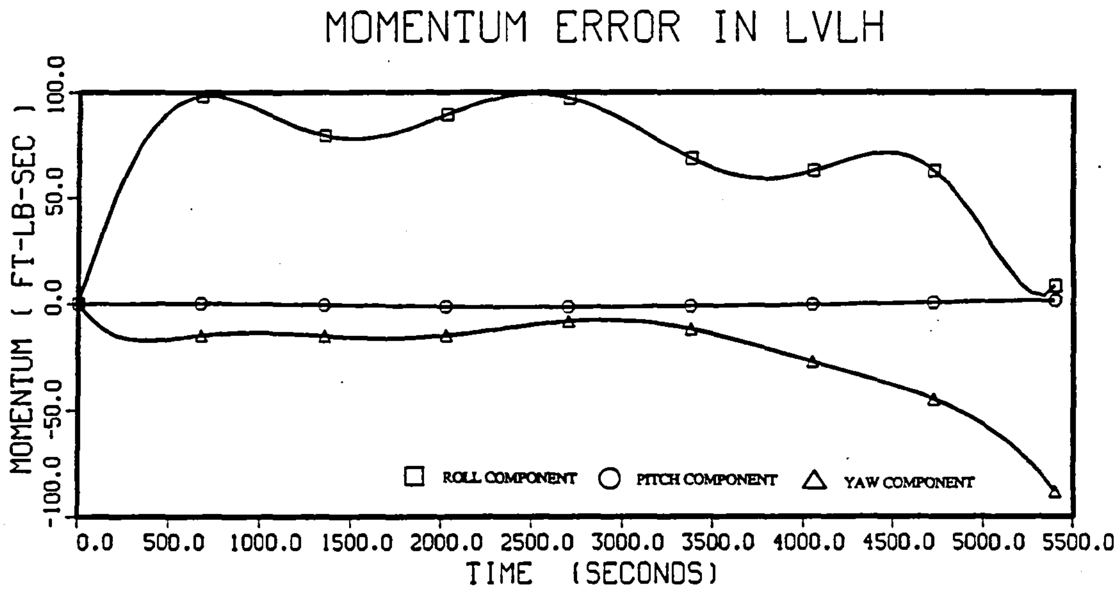


Figure 4.6: LVLH Momentum Error: *0.1 ft-lb* Torque about Yaw (566).

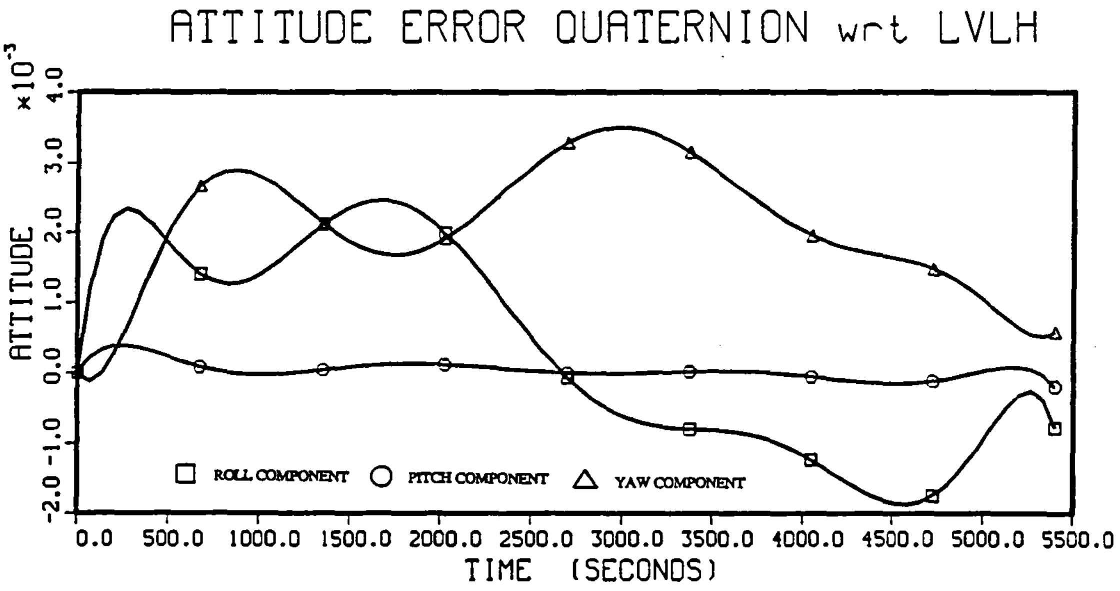


Figure 4.7: Attitude Errors: *0.1 ft-lb* Torque about Yaw (566).

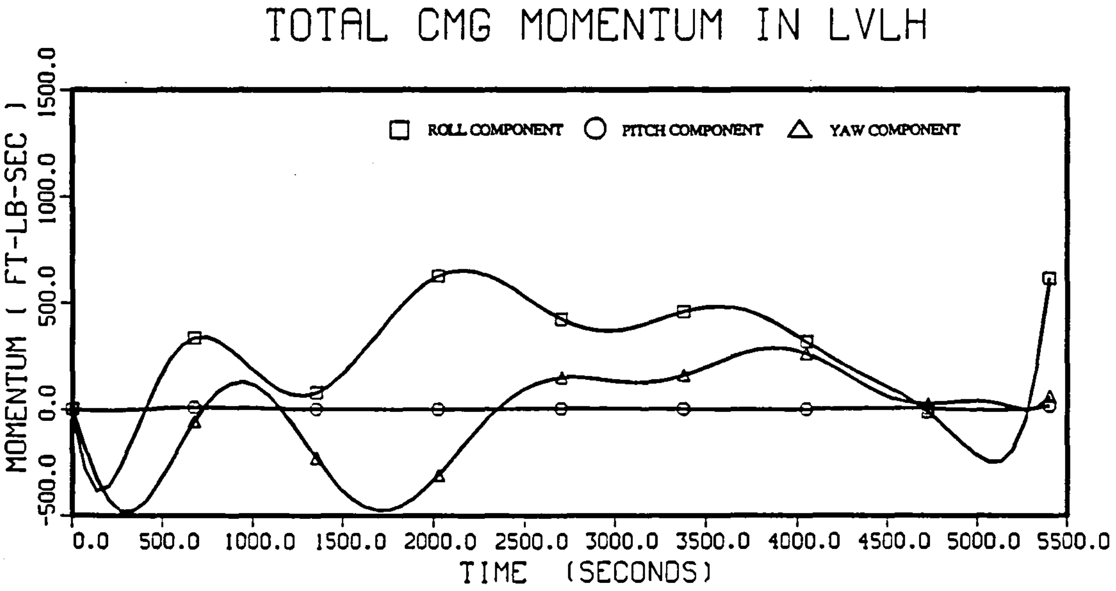


Figure 4.8: CMG Momentum in Body Frame: *0.1 ft-lb* Torque about Yaw (566).

4.4.4 Momentum Management in the Presence of an Aerodynamic Torque

In this section, the torque applied to the vehicle is chosen to be representative of aerodynamic torques that the vehicle is likely to encounter on-orbit. The applied torque is shown in Figure 4.9 and was chosen to be similar to a plot in [15] which was generated by a computational model of the fully erected space station. The sinusoidal variation in the torque is due to the variation in atmospheric density as the spacecraft orbits on the night and then the dayside of the Earth. More complicated behavior can be expected if effects such as solar panel rotation are taken into account.

The attitude error is shown in Figure 4.10. As the external torque about the pitch axis becomes less negative, the pitch component of the equilibrium attitude becomes more negative. The negative attitude error about this axis produces a negative torque on the vehicle that aids in tracking the equilibrium attitude and reduces momentum requirements on the CMGs. The attitude error about roll exploits gravity torques to cause the transfer of momentum in the negative roll direction that can be seen in Figure 4.11. In the previous example it was explained how momentum transfer in the pitch direction aids in compensating for torque disturbances in the yaw direction. The CMG momentum transfer in LVLH is shown in Figure 4.12. The momentum transfer at the starting time in the pitch and roll directions produces negative torques on the vehicle and yields the resulting attitude changes. Note once again that the momentum excursions are large enough that a linearized model will miscalculate the Euler coupling by a factor of between four and eight.

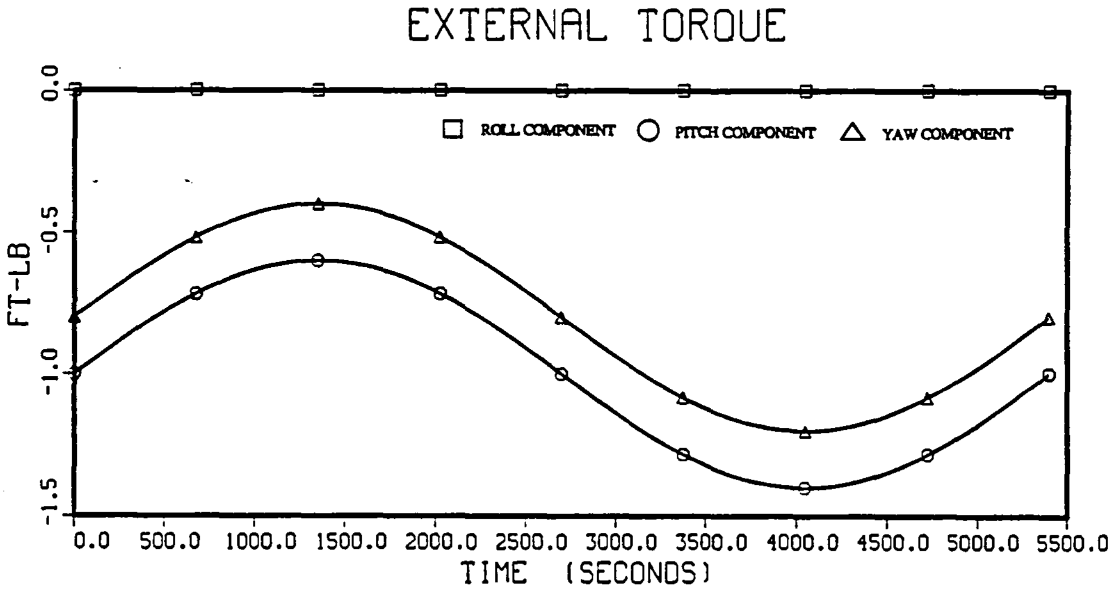


Figure 4.9: Representative Aerodynamic Torque in LVLH Coordinates (606).

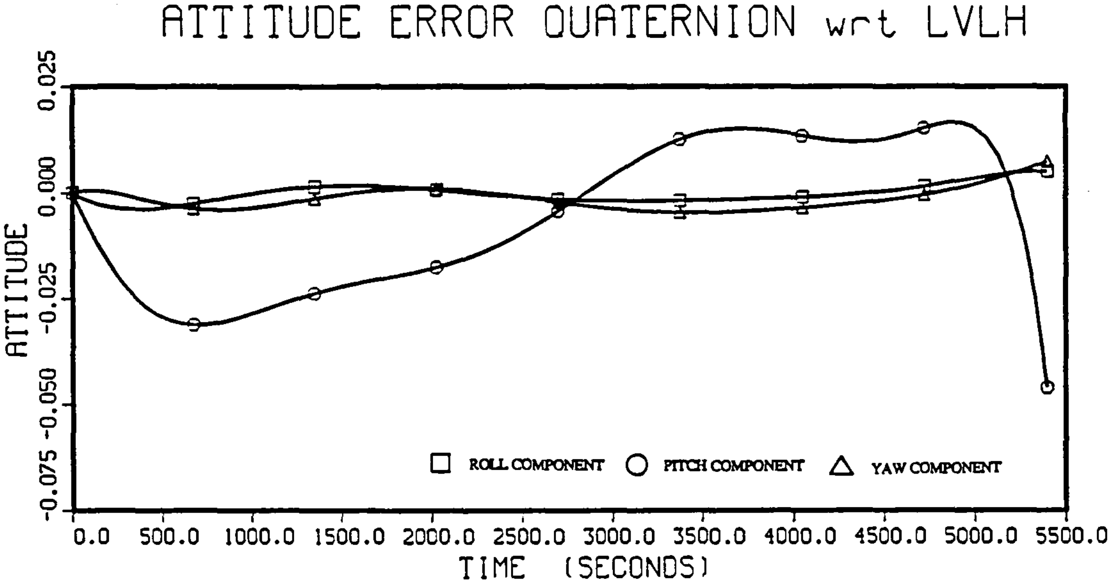


Figure 4.10: Attitude Errors: Aerodynamic Torque (606).

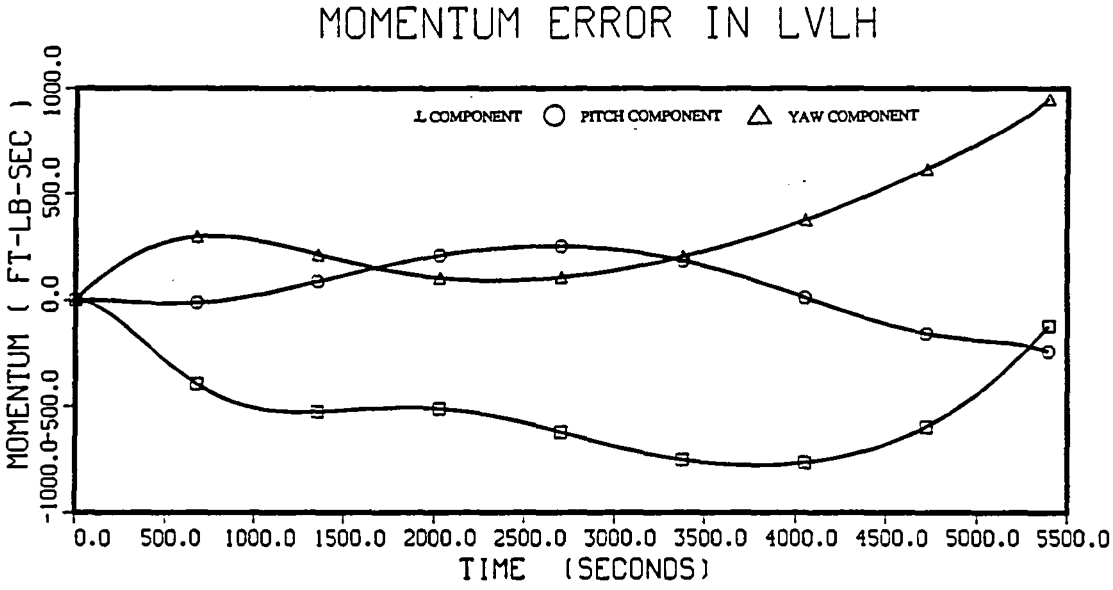


Figure 4.11: Momentum Errors: Aerodynamic Torque (606).

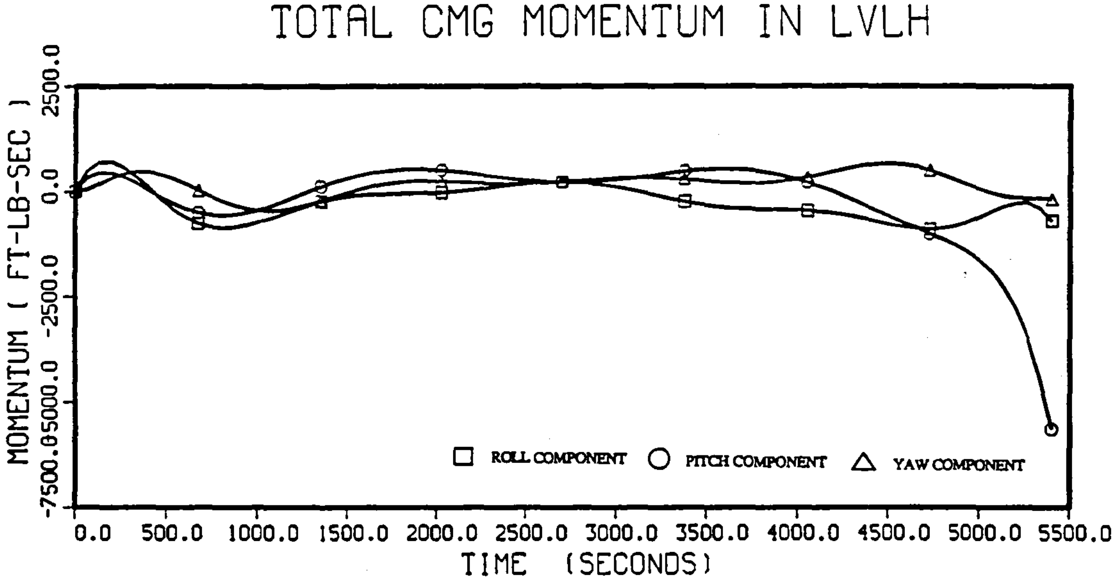


Figure 4.12: CMG Momentum Transfer: Aerodynamic Torque (606).

4.4.5 Momentum Management with Initial Momentum Error in Pitch

In this example, the CMGs are initialized with a net momentum storage along the pitch axis (i.e. parallel to the outer gimbal axes). Except for gravity, there are no other external torques acting on the vehicle. The initial gimbal angles are all *-0.1 radians*, and all other data are the same as in the previous examples.

Figure 4.13 shows that the initial momentum error is *2000 ft-lb* in the pitch direction, and that the momentum is smoothly removed by the end of the orbit. The momentum error reduction is achieved by manipulating the gravity gradient torque. The large pitch attitude error apparent in Figure 4.14 produces a negative gravity torque on the vehicle. Figure 4.15 shows that the CMGs provide the torques required to produce this attitude error, and then to prevent the attitude from diverging due to the unstable nature of the equilibrium.

Inspection of Figure 4.15 indicates that the CMGs exchange about *5000 ft-lb-sec* of momentum with the spacecraft along the pitch axis. As a consequence, the momentum stored in the rotating spacecraft, as well as the Euler torque on the spacecraft, more than double. Once again, a linearized model could not account for these effects without imposing restrictions on the allowable vehicle rates.

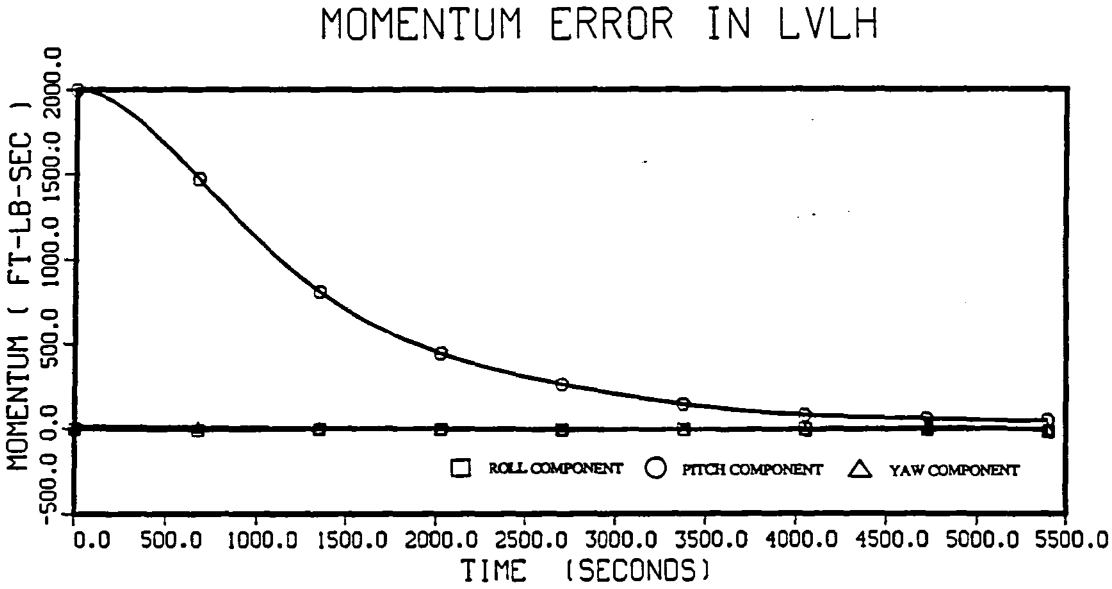


Figure 4.13: Momentum Errors: Initial Momentum Error in Pitch (569).

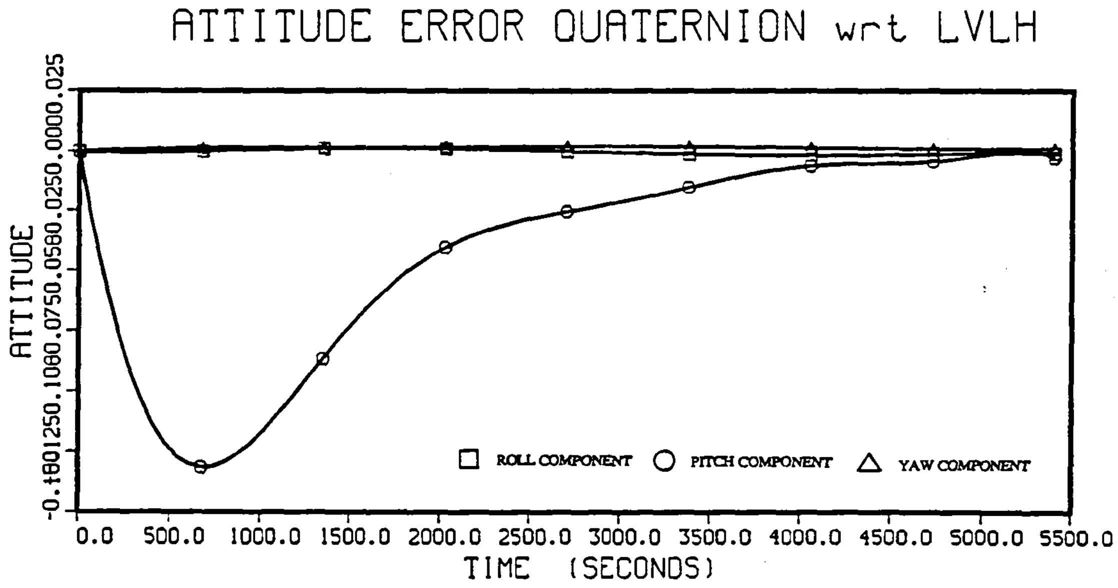


Figure 4.14: Attitude Errors: Initial Momentum Error in Pitch (569).

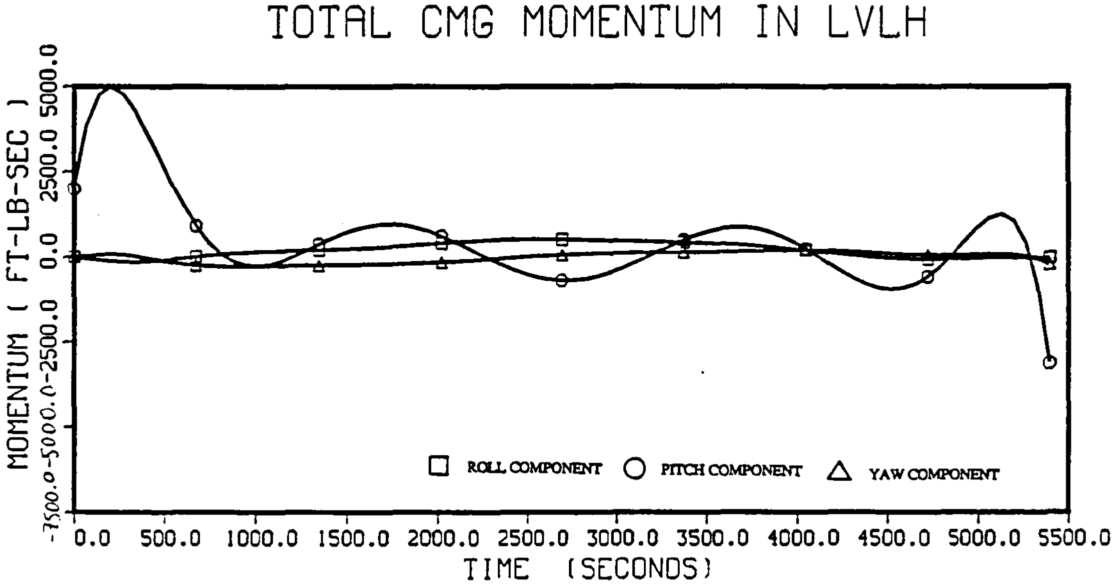


Figure 4.15: CMG Momentum Transfer: Initial Momentum Error in Pitch (569).

4.4.6 Momentum Management with Initial Momentum Error in Yaw

The problem solved in this example is the same as the previous example, except that the CMGs are initialized with net momentum storage in the yaw direction of the LVLH reference frame. The momentum error in yaw is obtained by initializing the outer gimbal angles of CMG#2 and CMG#3 at $\frac{\pi}{2}-0.1$ and $\frac{3\pi}{2}+0.1$, respectively. All other data are initialized as in the first example where there were neither external torques nor initial momentum errors.

Figure 4.16 shows that the optimal solution adequately removes the momentum error that was present at the initial time. Note that there is initially an increase in the momentum error in the pitch direction. The reason for this can be understood from equation (2.56), which shows that the rate of change of momentum in the yaw direction is equal to the product of the orbital rate with the momentum in the pitch direction. The attitude errors and the CMG momentum transfer shown in Figures 4.17 and 4.18, respectively, show that there is a significant amount of momentum exchange occurring, even though the rate-of-change of net momentum is quite smooth. Note also the correlation between the attitude error about roll, and the rate of net momentum change, which is due to the gravity torque.

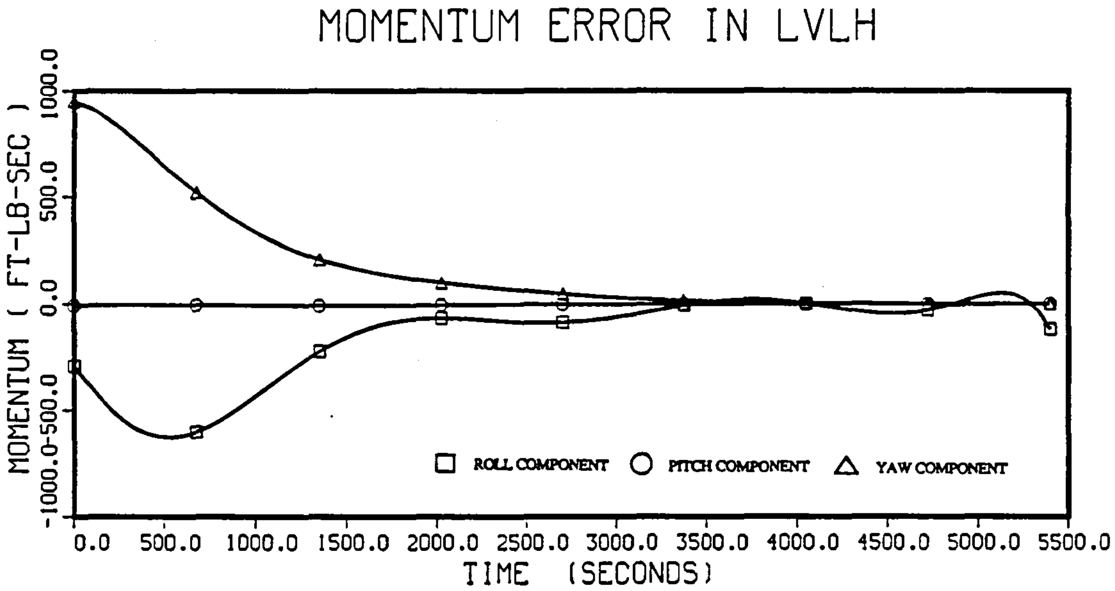


Figure 4.16: Momentum Errors: Initial Momentum Error in Yaw (570).

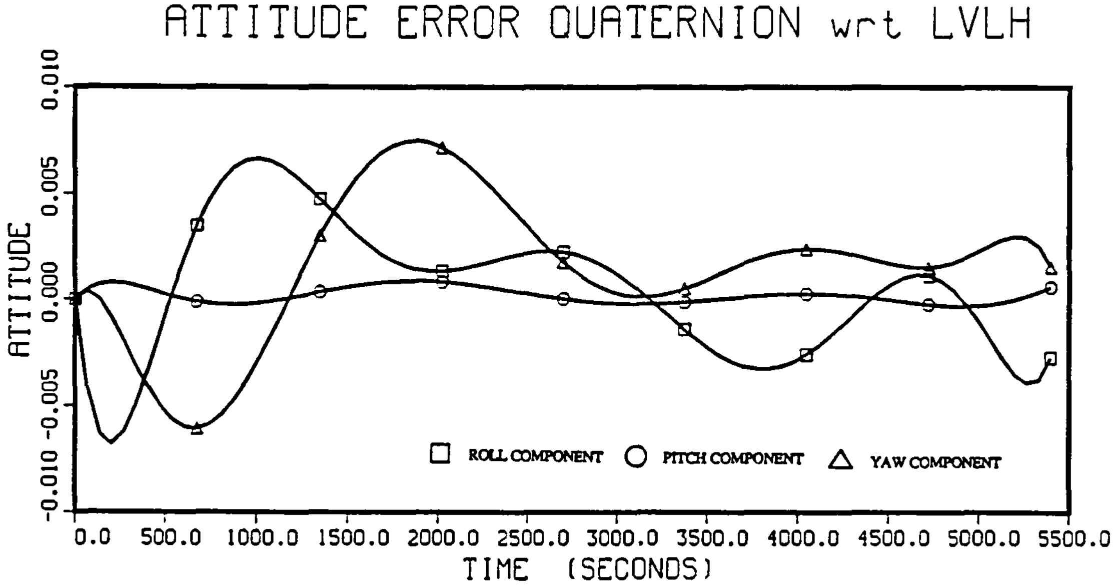


Figure 4.17: Attitude Errors: Initial Momentum Error in Yaw (570).

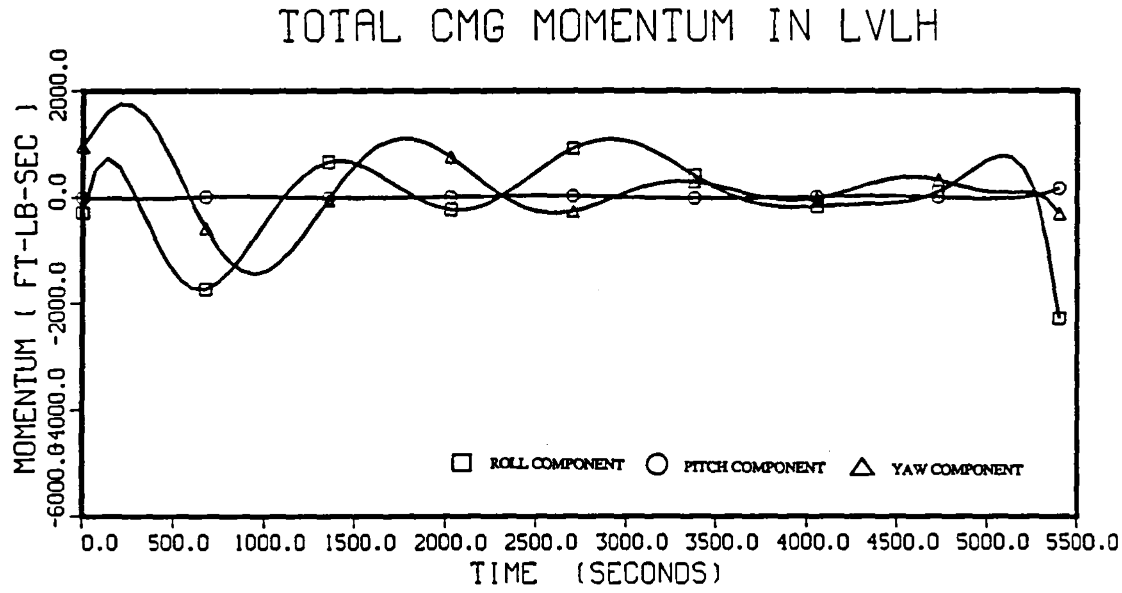


Figure 4.18: CMG Momentum Transfer: Initial Momentum Error in Yaw (570).

4.5 A Conceptual Spacecraft Control System Based on Feedback Linearization

In this section, an attitude control system is described that incorporates the optimal momentum management scheme and the CMG steering algorithm described in Chapter 4 into a control system that regulates not only the vehicle attitude, but also the stored momentum and the CMG configuration. The schematic structure of the system is shown in Figure 4.19. Functionally, the system can be seperated into a command generator, a feedback control system, and a plant whose dynamics are to be controlled.

In the case of interest here, the plant is a spacecraft equipped with CMGs. The command generator consists of the optimal momentum management scheme described in the previous chapter. The optimal momentum management problem is solved to obtain a state history for the vehicle to track over an orbit. Closed-loop control is required to regulate the actual spacecraft state to remain near the commanded state in the presence of uncertain environmental torques. The design of these feedback control laws has not been considered in this thesis. If errors in the environmental disturbances become so large that the control system has difficulty tracking the desired solution, the momentum manager can be called to generate a new solution that also desaturates any excess momentum stored in the system.

The functions of attitude control and momentum control have been shown separately in Figure 4.19. One reason for this is that it seems reasonable to consider momentum control as being subordinate to attitude control in typical on-orbit applications. Also, the control authority over attitude dynamics provided by the CMGs is likely to be greater than the authority gravity gradient torques have over momentum, implying that the two control systems may have widely differing bandwidths. Furthermore, there are a number of other actuation devices that may be used to control momentum that have not been considered here. An evaluation of the precise function of the feedback control systems along with procedures for obtaining useful designs has been left for future investigation.

Note that the approximation based steering algorithm has also been included in the control system shown in Figure 4.19. In this case, the approximation procedure outlined in Chapter 3 should be formulated using the affine control parameterization for the system model that includes the external torque, and the desired rates are computed by the optimal momentum manager. The approximation algorithm is still needed because there will always be distrubances that are not accounted for in the optimal solution. Even though the approximation algorithm should continue to monitor the CMGs for undesirable configurations, its primary function here is to distribute disturbances among the available CMGs so that the desired trajectory can be tracked.

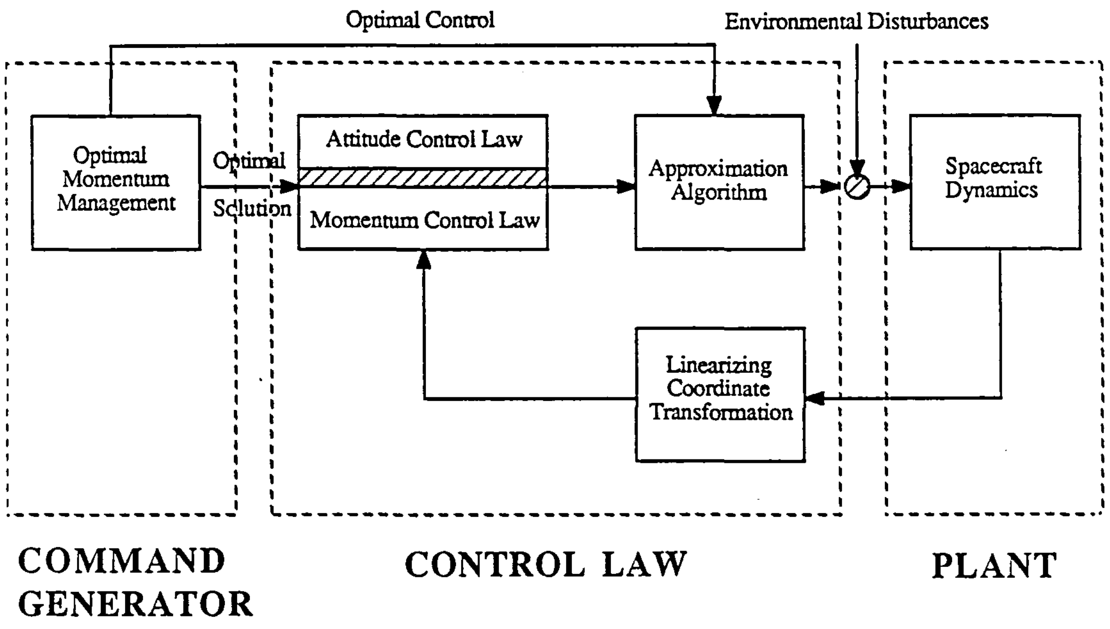


Figure 4.19: Schematic of an Attitude/Momentum Control System for a Spacecraft

Chapter 5

Summary and Recommendations for Future Research

The goal of this thesis was to develop an appoach to the control of spacecraft with CMGs in which the vehicle and the CMGs are treated as a combined system. In general, the equations for such a system are nonlinear and highly coupled. Feedback linearization techniques were used to transform appropriate nonlinear spacecraft models into equivalent linear systems. These linearized models were then used to develop practical schemes for solving the CMG steering problem and the optimal momentum management problem.

In Chapter 2, two spacecraft models were derived along with the state and input transformations that linearize them. The difference between these two models was that one assumed the spacecraft was an isolated system with inertially constant net momentum. This model included state equations for the vehicle attitude and the CMG gimbal angles. The second model incorporated the effect of the attitude dependent gravity gradient torque, and provided for the accomodation of arbitrary external torques. In addition to equations for the vehicle attitude and CMG state, this model included the net momentum as a state variable. The chapter concluded with the development of transformations that linearized both models.

In Chapter 3, an approach to CMG steering was developed utilizing the transformation that linearized the dynamics of the first spacecraft model. Although the system to be controlled had been made linear, the transformations themselves did not address certain critical issues relating to the steering problem. For systems in which there are redundant actuators, it was pointed out that the linearizing transformations effectively eliminate the redundancy. This is particularly undesireable when redundancy has been included to increase the flexibility of the system to be controlled. To remedy the situation, a method was developed to distribute control effort among redundant actuators in a feedback linearized system, and then applied to the spacecraft with CMGs.

The *normed approximation* algorithm developed in Chapter 3 for CMG steering was shown to have a number of advantages over existing algorithms. With this method, the problem of CMG steering law design involves defining how the CMGs should behave in a given situation, and then chosing a set of gimbal rates that produces this behavior. Conceptually, this problem is simpler than the conventional method of determining a function whose gradient defines the desired behavior. Using this approach, it is possible to develop a steering law that can recognize problem situations, and take specific action to remedy the difficuly. Another important feature of the approach is that it incorporates explicit bounds on the rates that can be produced by the gimbal drives (reflecting actual hardware limitations). With one exception, existing approaches scale the commanded gimbal rates to obtain a set of rates that do not violate physical constraints, thus failing to produce the required torque on the vehicle. A simple modification to the proposed algorithm was described that guaranteed the steering law would not fail when forced to operate near a singularity.

Examples were presented to demonstrate the behavior of the proposed steering algorithm performing a variety of attitude control functions. The first three examples illustrated the characteristics of the gimbal angle trajectories generated by the three normed approximation algorithms. The 1-norm and ∞-norm algorithms were found to exhibit undesirable oscillations in the present application. Because the 2-norm algorithm generally yielded smooth, well behaved solutions, it was used in the remaining examples. The fourth example demonstrated a situation in which the steering algorithm was able to deal explicitly with a problem situation that conventional gradient based methods would fail to recognize. The next two examples showed the algorithm redistributing the CMG rotors when they are initialized in poor configurations. In one of these examples, the algorithm was initialized in a singular configuration. The final example of this chapter showed the response of the vehicle/CMG system when a large onboard disturbance occurs. The disturbance was large enough to saturate several of the CMGs in rate, and the example demonstrates the advantage of enforcing rate limits on the individual gimbals.

In Chapter 4, an approach was developed to compute suboptimal trajectories for an optimal momentum management problem. The method used feedback linearizing transformations to restate an optimal control problem with a nonlinear dynamical constraint as an equivalent problem in which the system model was linear. The solution to the linear system was approximated using collocation techniques, and the optimal control problem approximated as an unconstrained nonlinear program. In particular, the optimal momentum management problem was reduced to an unconstrained, nonlinear, quadratic programming problem, and a computational algorithm for obtaining solutions was described.

A solution to the optimal momentum management problem yields a state and control history that maintains both the spacecraft attitude and momentum near desired states in the presence of external disturbances. The primary advantage of the proposed approach is that issues relating to attitude control and momentum management are treated in a combined framework that incorporates a nonlinear vehicle model. The importance of treating the nonlinear vehicle dynamics was highlighted in a number of examples.

The first two examples using the optimal momentum manager showed how the solutions manipulated the total momentum of the vehicle/CMG system to reduce the influence of external torques and Euler coupling on the vehicle. A third example included an external torque roughly representative of those the proposed NASA space station may encounter on-orbit. The last two examples illustrated how optimal solutions could exploit the gravity gradient torque to eliminate net momentum stored in the CMG system. In all but the first example, the variation of vehicle momentum was of the same order-of-magnitude as the nominal values. These variations are large enough to invalidate the small state excursion assumption made in the developement of existing schemes for momentum management.

This thesis has investigated the application of feedback linearization theory to two problems arising in the control of spacecraft with CMGs. While the approach to CMG steering developed in Chapter 3 is no more computationally intensive than existing algorithms, the computational overhead associated with the optimal momentum management scheme presented in Chapter 4 can be significant. This burden is offset by the fact that the computational algorithm developed here is very robust, thus fairly reliable estimates on convergence times can be made. The reliable convergence features of this approach to momentum managment make the technique suitable for autonomous implementation when turn-around time is not a critical issue. In the space station examples studied here, the momentum management algorithm must be scheduled for execution about once an orbit (i.e. *90 min*)

Chapter 4 concluded with a description of a proposed spacecraft attitude control system based on the feedback linearized dynamics of the spacecraft. Although the detailed design of the closed-loop control system has been left for future research, the roles of the optimal momentum manager and the approximation algorithm developed in Chapter 3 were emphasized. The optimal momentum manager played the role of a command generator, and would typically be called once per orbit, or when needed. The approximation algorithm was used to distribute environmental disturbances among the available CMGs.

One limitation inherent in the proposed approach to momentum management is that it does not allow for discontinuous disturbances or control action. Step or impulse disturbances represent an important class of torques associated with docking or configuration changes and are important to consider. Although high-order approximations can be used to deal with such disturbances, this is an inefficient approach. One possible strategy might be to use polynomial approximations on intervals that have their endpoints at the times when the discontinuities occur. Note that this technique could also be used to include actuators like jets in the momentum management problem, since their net effect on momentum appears linearly in both the nonlinear and the linearized models.

Future investigations might also consider methods of establishing an optimal trajectory that would be valid on succesive orbits. One approach would be to apply the present algorithm to a time interval containing several orbits, and choose a suitable subinterval of this solution. Another possibility would be to include the initial and final states as decision variables in the optimization, and to require that certain compatability conditions between the initial and final states be met. Such a technique should also eliminate the errors resulting from the polynomial approximation that were seen to occur near the final time in the examples that have been presented.

In the initial stages of this research, it was found that a criterion that measured the optimality of a CMG configuration could be chosen as a state variable in the linearized model. A number of simulations were done in which the gain of the CMG configuration was a state variable of the linearized system. One difficulty with this approach was that the transformation would fail to exist at points where this function attained a local optimum. In addition, when the CMG system has a large number of degrees-of-freedom, it is necessary to create a number of criteria that must be controlled independently. Although there may be limitations, further investigation of this idea may lead to useful steering algorithms for CMG configurations with few redundant degrees-of-freedom.

One important question left for future investigation involves how uncertainty in the vehicle model effects the stability of closed-loop control systems designed using feedback linearizing transformations. Results are available which guarantee that sufficiently small errors will not destabilize such control systems [56]. For practical purposes, it would be useful to have quantitative bounds on the uncertainty that these systems can accomodate. Another related issue is that the linearizing transformations assume perfect state information. In practice, all state variables may not be available, and this may pose an additional limitation in certain applications.

The manner in which the linearizing transformations assimilate perfect model information might also be exploited in an adaptive control scheme. In applications such as the proposed NASA space station, where the spacecraft will be erected over a long period of time, the performance robustness of any fixed control system is a delicate question. If accurate model information can be provided to the algorithm constructing the linearizing transformation, the transformations will adapt perfectly to the changing system.

In conclusion, this thesis has used feedback linearization techniques to develop approaches to the resolution of redundancy in actuator systems, and the solution of optimal control problems. Both of the problems studied in this thesis arise in various forms in other areas of control and systems theory. In particular, the similarity of spacecraft control with CMGs to the control of robotic manipulators has been discussed in some detail. The ideas developed in this thesis could similarly be applied to obtain practical solutions to problems in a number of related fields.

Appendix A

Notation

h	denotes a momentum vector
I	denotes an inertia tensor
I_n	denotes an identity matrix of dimension $n \times n$
M, N	used to denote a manifold
$dim(\cdot)$	dimension of a linear vector space
q	a quaternion
q_A^B	quaternion of reference frame B wrt. A
q_I	quaternion of a body fixed frame wrt. an inertial frame
q_L	quaternion of a body fixed frame wrt. an LVLH frame
LVLH	Local-Vertical/Local Horizontal
$\mathbf{R}^n$	the real n-tuple space
$x, f, \ldots$	lowercase italics are the workhorses meaning is defined by context, e.g. n-tuple, function, etc.
$\dot{x}$	time derivative of x

$A \cdot b$	dot used to emphasise an arithmetic product or matrix multiplication
$\widehat{\ }$	unit vector, e.g. $\widehat{h}$
$*$	denotes the dual of a vector space
$\mathbf{L}_f$	denotes the Lie derivative with respect to the vector field f
$\perp$	denotes the annihilator of a distribution
(U, ϕ)	a coordinate system with coordinate chart ϕ defined on U
$\triangleq$	defined equal to or defined by
δ_{ij}	Kronecker delta, $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$
$\mathcal{F}, \mathcal{G}, \ldots$	calligraphic letters denote distributions
θ	used to denote a set of CMG gimbal angles
ω	denotes and angular rate
$\circ$	composition, $f \circ g \triangleq f(g)$
^{T}	denotes transpose
$\min\limits_{a} \ J(a)$	minimize $J(a)$ with respect to a
$\max\limits_{a} \ J(a)$	maximize $J(a)$ with respect to a
$[f, g]$	Lie bracket of the vector fields f and g
$ad_f^n g$	n-fold Lie bracket $ad_f^0 g \triangleq g$, $ad_f^n g \triangleq [f, ad_f^{n-1} g]$

Appendix B

Solution of the Feedback Linearization Problem

In this appendix the external linearization problem is precisely defined and solved. Before discussing the problem, requisite background concepts and definitions are presented. The problem solved here is the same as that solved in [19]. The derivation presented here, however, follows the geometric approach of [57]. Conditions are stated which guarantee the existence of coordinate changes in the state and input spaces which together with nonlinear feedback linearize the state equations of a locally controllable nonlinear system.

B.1 Background from Differential Geometry

In this section, the notation and concepts used to derive the results of subsequent sections are presented. In the next subsection, the notion of a differentiable manifold is introduced, along with a number of structures defined on manifolds. A completely integrable manifold is defined and the Frobenius Theorem is stated and interpreted. Most of the definitions and results can be found in greater detail in texts such as [58], [59], and [60], or in the appendix of [57].

B.1.1 Manifolds, Vector Fields, and Covector Fields

The notation $f : A \longrightarrow \mathbf{R}^n$ denotes a vector valued function defined on an open set $A \subset \mathbf{R}^m$ taking values in the real n-tuple space. Such a function is called smooth, differentiable, or of class C^∞ if all partial derivatives of all orders of the component functions $f_1, \ldots, f_n$ exist and are continuous. Nearly all of the results which follow are true assuming that all n^{th}-order partial derivatives of the component functions are continuous for n sufficiently large. The *derivative* (or Jacobian) of f is denoted $\mathbf{D}f(x)$ and is given by the n by m matrix of partial derivatives

$$[\mathbf{D}f(x)]_{i,j} \;=\; \left[\frac{\partial f_i}{\partial x_j}\right]_{(x)} \tag{B.1}$$

In the case where $m = n$, and the matrix $\mathbf{D}f(x)$ is nonsingular at x, f is called a diffeomorphism. By the Inverse Function Theorem, differentiability of f and nonsingularity of (B.1) at x guarantee that the function f has a smooth inverse on an open neighborhood of $f(x)$.

A smooth *k-manifold* M in $\mathbf{R}^n$ is a set of points in $\mathbf{R}^n$ having the property: Given $p \in M$, there are open sets A, B in $\mathbf{R}^n$ containing p, and a diffeomorphism $g : A \longrightarrow B$ such that

$$g\left(A \cap (\mathbf{R}^k \times 0^{n-k})\right) \;=\; B \cap M \tag{B.2}$$

The notation $X \times Y$ denotes the *cross product* of the sets X and Y; this is the set of ordered pairs of the form $[x, y]$ where $x \in X$ and $y \in Y$. In particular, the set $\mathbf{R}^k \times 0^{n-k}$ denotes the set of n-tuples for which the last $n-k$ elements are zero. This definition of a manifold is more restrictive than usually employed in abstract discussions; the structure just defined is often called a *regular submanifold* of $\mathbf{R}^n$.

From a practical point of view, most dynamical systems can be thought of as evolving in an n-dimensional *state space*, and since the linearization theory to

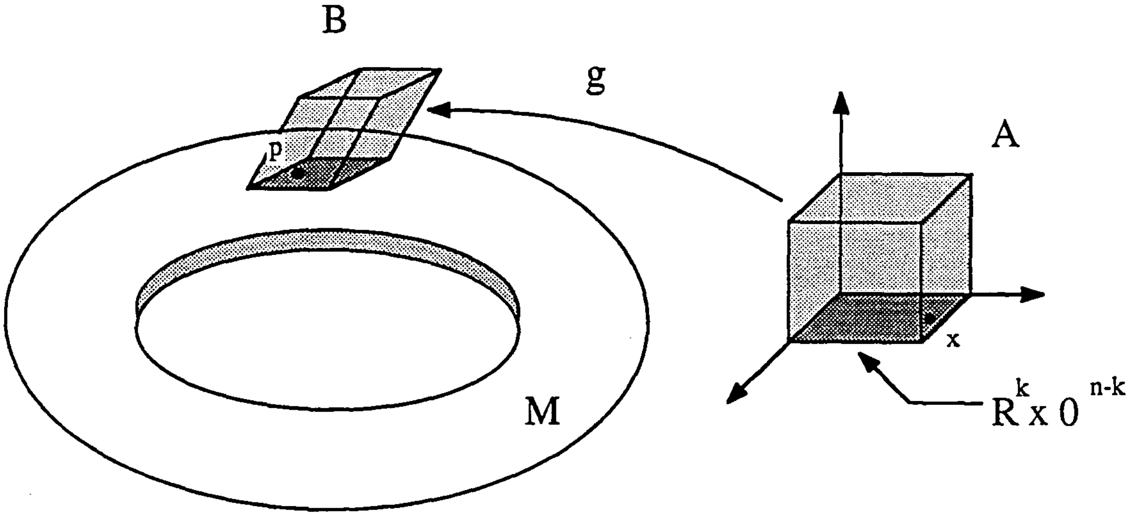


Figure B.1: A Two-dimensional Manifold in $\mathbf{R}^3$

be developed is local, it is not necessary to have a definition which characterizes manifolds more generally. The definition of a manifold used in this discussion is illustrated in Figure B.1 for a manifold consisting of a torus in three-dimensional space. The function defined by the rule

$$\alpha(x) \triangleq g(x, 0, \ldots, 0), \; x \in \mathbf{R}^k \tag{B.3}$$

is called a *coordinate patch* on M, and its inverse $\phi(p) = \alpha^{-1}(p)$ is called a *coordinate chart* on M. Note that because the manifold is a subset of $\mathbf{R}^n$, the range of the coordinate patch α is an n-tuple with $n \geq k$. Since the derivative $\mathbf{D}\alpha$ is not a square matrix, the mapping α will not generally have an inverse on $\mathbf{R}^n$. The coordinate patch does have an inverse on an open set of points *in the manifold.* Similar remarks apply to the coordinate chart which is a map $\phi : \mathbf{R}^n \longrightarrow \mathbf{R}^k$.

Based on these definitions, a manifold can be envisioned as a smooth surface in $\mathbf{R}^n$, and a coordinate patch is an invertable function that maps points in a k-dimensional Euclidean space onto the manifold, M. The coordinate patch and coordinate chart can be thought of as defining the relationship between points on a flat map covered with a rectangular grid and a spherical globe. The definition says that for each point on the manifold, p, there is an open set $U \subset M$ containing p, and a coordinate chart ϕ defined on U; the pair (U, ϕ) is called a coordinate system at p. The principal advantage of defining a manifold in a Euclidean space is that functions defined on the manifold can be extended to functions defined on the surrounding space in a natural way. This will simplify notation when it is necessary to manipulate functions defined on the manifold; definitions can be expressed in the *local coordinates* of the Euclidean space surrounding the manifold rather than in the coordinates of the range space of the coordinate chart.

A tangent vector v, with components $\{v_1, \ldots, v_n\}$ at a point x in $\mathbf{R}^n$, can be thought of as an operator on differentiable functions. If $\lambda : \mathbf{R}^n \longrightarrow \mathbf{R}$ is a smooth function on a neighborhood of x, then $v(\lambda)$ is the real number which is the directional derivative of λ in the direction of the vector $\{v_1, \ldots, v_n\}$. In the standard coordinate system for $\mathbf{R}^n$ - which is itself a manifold - $v(\lambda)$ is given by

$$v(\lambda) = \sum_{i=1}^{n} v_i \left(\frac{\partial \lambda}{\partial x_i}\right)_x \tag{B.4}$$

Denoted $T_x\mathbf{R}^n$, the tangent space to the manifold $\mathbf{R}^n$ at x is the set of all tangent vectors at x. The tangent space is a real, linear vector space with scalar multiplication and addition defined by

$$(c_1 v_1 + c_2 v_2)(\lambda) \triangleq c_1 v_1(\lambda) + c_2 v_2(\lambda) \quad \text{for } c_1,\ c_2 \in \mathbf{R} \tag{B.5}$$

The set of tangent vectors which are directional derivatives along each of the n coordinate axes in $\mathbf{R}^n$ are denoted by

$$\left(\frac{\partial}{\partial x_1}\right)_x, \ldots, \left(\frac{\partial}{\partial x_n}\right)_x \tag{B.6}$$

and form a basis for the space $T_x\mathbf{R}^n$ in much the same way that the vectors $e_i \triangleq [0_1, \ldots, 1_i, \ldots, 0_n]^T$ form a basis for $\mathbf{R}^n$. In fact, the e_i are the components of the basis vectors of the tangent space; consequently, the vector space stucture of $T_x\mathbf{R}^n$ can be envisioned with the same geometric structure as the vector space $\mathbf{R}^n$. It is possible to define the tangent space at a point on a k-manifold by introducing a coordinate system, (U, ϕ), at a point $p \in M$. Denoted T_pM, this is also a real vector space. Associated with the coordinate chart, there are k tangent vectors at p defined by

$$\begin{aligned} \left(\frac{\partial}{\partial \phi_i}\right)_p (\lambda) &\triangleq \left(\frac{\partial \lambda \circ \phi^{-1}}{\partial x_i}\right)_{x=\phi(p)} \qquad \text{for } 1 \leq i \leq k \\ &= \sum_{j=1}^{n} \left(\frac{\partial \lambda}{\partial y_j}\right)_p \cdot \left(\frac{\partial \phi_j^{-1}}{\partial x_i}\right)_{x=\phi(p)} \\ &\triangleq [\mathbf{D}\lambda][\mathbf{D}\phi_i^{-1}]_{x=\phi(p)} \end{aligned} \tag{B.7}$$

where $\mathbf{D}\phi_i^{-1}$ denotes the i^{th} column of the derivative of ϕ^{-1} and y denotes a point in the space containing the manifold. If ϕ^{-1} is imagined as mapping the i^{th} coordinate axis onto a parameterized curve on the manifold, the i^{th} basis vector takes the derivative of λ in the direction of the tangent to the curve at p. In other words, if ϕ^{-1} is viewed as a function of a single variable, say $t = x_i$, then the tangent to the curve can be interpretted as a velocity vector. In local coordinates, this velocity vector is given by $\mathbf{D}\phi_i^{-1}$. Since ϕ is a diffeomorphism of open sets, the set of directions associated with the k tangent vectors (B.7) are independent and lie in a k-dimensional planar surface tangent to M at p. These k tangent vectors form a basis for the tangent space of M at p called the *natural basis* induced by the coordinate system. Every element of T_pM can be expressed in the form

$$v = \sum_{i=1}^{k} v_i \left(\frac{\partial}{\partial\phi_i}\right)_p \tag{B.8}$$

If λ is a smooth function defined on an open neighborhood of $p \in M$, and $v \in T_pM$, then $v(\lambda)$ is the derivative of λ in a direction which is tangent to the surface of M at p. For manifolds in $\mathbf{R}^n$, it is useful to associate a tangent vector with a direction in $\mathbf{R}^n$ and a point on the manifold.

Let (U, ϕ) and (V, ψ) be coordinate systems about a point p on an m-manifold M. The two coordinate charts induce different bases for the same tangent space T_pM. The relationship between the two bases can be found by expressing each basis vector $(\frac{\partial}{\partial\psi_i})_p(\lambda)$ in terms of the basis induced by (U, ϕ).

$$\begin{aligned}\left(\frac{\partial}{\partial\psi_i}\right)_p(\lambda) &= \left(\frac{\partial(\lambda\circ\phi^{-1}\circ\phi\circ\psi^{-1})}{\partial y_i}\right)_{y=\psi(p)} \\ &= \sum_{j=1}^{m}\left(\frac{\partial(\lambda\circ\phi^{-1})}{\partial x_j}\right)_{x=\phi(p)}\cdot\left(\frac{\partial(\phi_j\circ\psi^{-1})}{\partial y_i}\right)_{y=\psi(p)}\end{aligned}$$

$$= \sum_{j=1}^{m} \left[\left(\frac{\partial(\phi_j \circ \psi^{-1})}{\partial y_i} \right)_{y=\psi(p)} \right] \left(\frac{\partial}{\partial \phi_j} \right)_p (\lambda) \tag{B.9}$$

The function defined by $f = \phi \circ \psi^{-1}$ is a diffeomorphism and defines a change of coordinates about $p \in M$. If $\{v_1, \ldots, v_m\}$ and $\{w_1, \ldots, w_m\}$ denote the components of the same tangent vector in the bases induced by (U, ϕ) and (V, ψ), respectively, then (B.9) and the requirement that

$$v = \sum_{i=1}^{m} v_i \left(\frac{\partial}{\partial \phi_i} \right)_p (\lambda) = \sum_{j=1}^{m} w_j \left(\frac{\partial}{\partial \psi_j} \right)_p (\lambda) \tag{B.10}$$

imply that

$$\left\{ \begin{matrix} w_1 \\ \vdots \\ w_m \end{matrix} \right\} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_m} \end{bmatrix} \left\{ \begin{matrix} v_1 \\ \vdots \\ v_m \end{matrix} \right\} = \mathbf{D} f(x)\, v \tag{B.11}$$

The notion of a change of basis can be generalized to an arbitrary mapping between two manifolds. Let N and M be smooth manifolds of dimension n and m, and let $f : N \longrightarrow M$ be a smooth mapping. The *differential* of f at $p \in N$ is a linear map $f_* : T_p N \longrightarrow T_{f(p)} M$ defined by

$$(f_*(v))(\lambda) \triangleq v(\lambda \circ f) \tag{B.12}$$

for λ is a smooth real-valued function. An analogous computation to (B.9) can be applied to $v \in T_p N$ and $w \in T_{f(p)} M$ to show that the component vectors are related by $w = [\mathbf{D} f(x)]\, v$.

A *vector field* on a k-manifold M is a map assigning a tangent vector to each point on M. A smooth vector field $v(q)$ is such that for each point p, there is a coordinate chart (U,ϕ) about p, and k real-valued smooth functions $v_1(q),\ldots,v_k(q)$ defined on U such that

$$v(q) \;=\; \sum_{i=1}^{k} v_i(q)\left(\frac{\partial}{\partial\phi_i}\right)_q \tag{B.13}$$

for all $q \in U$. It is often useful to think of the vector field as defined by the set of component functions $\{v_i(q)\}$ and to associate the vector field with the column vector of component functions taking the form

$$v(q) \;=\; \left\{\begin{array}{c} v_1(q) \\ \vdots \\ v_n(q) \end{array}\right\} \tag{B.14}$$

It is a customary abuse of terminology to refer to the vector of component functions as a vector field; it is more accurate to say that they *define* a vector field. If the manifold under consideration is $\mathbf{R}^n$, the vector of n component functions can be naturally associated with a direction in the space.

The *Lie derivative* of a scalar valued function λ with respect to a vector field is denoted $\mathbf{L}_v\lambda$ and defined to be

$$(\mathbf{L}_v\lambda)(p) \;=\; (v(p))(\lambda) \tag{B.15}$$

The Lie derivative can be expressed in local coordinates yielding

$$(\mathbf{L}_v\lambda)(p) = \begin{bmatrix} \frac{\partial\lambda}{\partial x_1} & \cdots & \frac{\partial\lambda}{\partial x_n} \end{bmatrix}_p \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}_p \tag{B.16}$$

If one of the component functions of the coordinate chart is used in place of λ, then (B.16) implies that $\mathbf{L}_v\phi_i$ is the projection of the components of v on the gradient of ϕ_i. This implies that (B.13) can be written

$$v(q) = \sum_{i=1}^{k} \mathbf{L}_v\phi_i(q)\left(\frac{\partial}{\partial\phi_i}\right)_q \tag{B.17}$$

The vector space of all smooth tangent vectors on a manifold can be given the structure of a *Lie algebra* by defining the *Lie bracket* of a pair of vector fields. The Lie bracket of two vector fields v, w is another vector field denoted $[v, w]$ and defined by

$$([v,w](p))(\lambda) \triangleq \mathbf{L}_v(\mathbf{L}_w(\lambda))(p) - \mathbf{L}_w(\mathbf{L}_v(\lambda))(p) \tag{B.18}$$

In local coordinates, direct computation shows that the components of the vector field $[v, w]$ are given by

$$[v,w](x) = [\mathbf{D}w(x)]\{v(x)\} - [\mathbf{D}v(x)]\{w(x)\} \tag{B.19}$$

It is instructive at this point to examine the connection between vector fields, as defined here, and differential equations. It is customary to consider a system of

n autonomous equations of the form

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1(t), \ldots, x_n(t)) \\ &\vdots \\ \frac{dx_n}{dt} &= f_n(x_1(t), \ldots, x_n(t)) \end{aligned} \tag{B.20}$$

A parameterized curve $x(t) \in \mathbf{R}^n$ is a called a solution if it *satisfies* (B.20). Viewed as directions in $\mathbf{R}^n$, both sides of (B.20) can be associated with vector fields in a natural way, namely,

$$f(x) \triangleq \sum_{i=1}^{n} f_i(x_1, \ldots, x_n) \left(\frac{\partial}{\partial x_i} \right) \tag{B.21}$$

and

$$X(t) = \sum_{i=1}^{n} \frac{dx_i(t)}{dt} \left(\frac{\partial}{\partial x_i} \right) \tag{B.22}$$

Note that $f(x)$ is being used to denote a tangent vector field in this context, the $f_i(x)$ are component functions. At any particular time, $X(t)$ is also a tangent vector at the point $x(t)$ to the 1-manifold which is the solution; the components $\frac{dx_i}{dt}$ are merely the expression of X in the local coordinates of the manifold. Recalling the definition of the differential, (B.12), and regarding the solution $x(t)$ as a mapping, $x(t) : \mathbf{R}^1 \longrightarrow \mathbf{R}^n$, it is possible to conclude that

$$X(t) = x_* \left(\frac{\partial}{\partial t} \right) \tag{B.23}$$

and since (B.21) and (B.23) must be equal along the solution, it is necessary that

$$x_*(t)\left(\frac{\partial}{\partial t}\right)_t = f(x(t)) \tag{B.24}$$

This leads to the general notion of an *integral curve* of a *differential equation on a manifold*, M. A smooth curve, $\sigma : (t_0, t_1) \longrightarrow M$ is an *integral curve* of the vector field f defined on the M if

$$\sigma_*\left(\frac{\partial}{\partial t}\right)_t = f(\sigma(t)) \tag{B.25}$$

for all t in the open interval (t_0, t_1). Clearly, an integral curve is simply a generalized notion of a solution to (B.20). The advantage of this description is that it is a coordinate-free notion of a differential equation (i.e. state variables have neither been defined nor chosen).

The dual space to T_pM is called the *cotangent space* and is denoted T_p^*M, the *tangent covectors* are linear operators on tangent vectors that take values in the real numbers. The value of $v^* \in T_p^*M$ at $v \in T_pM$ is denoted $\langle v^*, v\rangle$. The space T_p^*M has the structure of a real vector space with addition and scalar multiplication defined by

$$\langle c_1 v_1^* + c_1 v_2^*, v\rangle = c_1 \langle v_1^*, v\rangle + c_2 \langle v_2^*, v\rangle \ , \quad c_1, c_2 \in \mathbf{R} \tag{B.26}$$

For a smooth function $\lambda : M \longrightarrow \mathbf{R}$, there is a way of associating the differential

λ_* at p with a covector. Since $\lambda_*(v)$ can be written in the form

$$\lambda_*(v) = c(v)\frac{\partial}{\partial x} \tag{B.27}$$

and since λ_* is a linear operator, the scalar function $c(v)$ must be a linear function of v. Since $c(v)$ is a linear operator on tangent vectors, there must be a covector $(d\lambda)_p$ such that

$$\lambda_*(v) = \langle (d\lambda)_p, v\rangle \frac{\partial}{\partial x} \tag{B.28}$$

In local coordinates, the component in this expression is

$$\langle (d\lambda)_p, v\rangle = \begin{bmatrix} \frac{\partial\lambda}{\partial x_1} & \cdots & \frac{\partial\lambda}{\partial x_1} \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \tag{B.29}$$

The last equality implies that

$$\begin{aligned} \left\langle (d\phi)_p, \left(\frac{\partial}{\partial\phi_j}\right)\right\rangle &= \left(\frac{\partial}{\partial\phi_j}\right)_p (\phi_i) \\ &= \frac{\partial(\phi_i \circ \phi^{-1})}{\partial x_j} \\ &= \frac{\partial x_i}{\partial x_j} \\ &= \begin{cases} 0 & \text{if} \quad i \neq j \\ 1 & \text{if} \quad i = j \end{cases} \end{aligned} \tag{B.30}$$

and the covectors $\{d\phi_1, \ldots, d\phi_n\}$ form a basis for the cotangent space which is dual to the basis for the tangent space. Any cotangent vector can be represented in this basis by an expression of the form

$$v^* = \sum_{i=1}^{k} v_i^* \, (d\phi_i)_p \tag{B.31}$$

Note that there are natural bases for the tangent space and the cotangent space induced by a coordinate chart on the manifold. If v and v^* are expressed in the natural basis induced by the same coordinate chart at p, then

$$\langle v^*, v \rangle = \sum_{i=1}^{k} v_i^*(p) \cdot v_i(p) \tag{B.32}$$

and it is natural to associate the operation of a cotangent vector on a tangent vector with an inner product on $\mathbf{R}^k$.

As in the case of a vector field, is is possible to define a covector field on a k-manifold. A *covector field* on a manifold is a map assigning to each point of the manifold a cotangent vector. A smooth cotangent field $v^*(q)$ is such that for each point $p \in M$, there is a coordinate system (U, ϕ) about p and k real-valued, smooth functions $v_1^*(q), \ldots, v_k^*(q)$ defined on U such that

$$v^*(q) = \sum_{i=1}^{k} v_i^*(q) \, (d\phi_i)_p \tag{B.33}$$

for all $q \in U$. Smooth covector fields are also known as 1-forms. The notation d is a standard notation for the *differential* operator on k-forms. Since a real-valued function is often defined to be a 0-form, (B.29) indicates that it is natural to think of $(d\lambda)$ as a linear operator on tangent vectors; a 1-form. A smooth covector field is called an exact one-form if $v^*(p) = d\lambda$ for some smooth real valued λ. Later in this chapter, it will be shown that the linearization problem is solved by a certain set of exact 1-forms; the computational problem is to find the associated 0-forms. Allowing for a slight abuse of terminology, a row vector of n real functions will often be referred to as a covector field. If the manifold happens to be $\mathbf{R}^n$, the operation $\langle v^*, v\rangle$ can be thought of as an inner-product of vector valued functions, $v^*(x)\, v(x)$.

In proving the theorems in the next section, there will be a need for a Leibnitz-type formula relating Lie derivatives of functions, Lie brackets (which are Lie derivatives of vector fields), and Lie derivatives of 1-forms. Let $v^*(x)$ denote the (row) vector of component functions of a covector field v^*, and let $v(x)$ denote the (column) vector of component functions of a vector field v. The Lie derivative of v^* with respect to v is a covector field denoted $\mathbf{L}_v v^*$. In local coordinates, the components of $\mathbf{L}_v v^*$ are given by the row vector of scalar functions

$$\mathbf{L}_v v^*(x) \;=\; v(x)^T[\mathbf{D}v^*(x)] + v^*(x)[\mathbf{D}v(x)] \tag{B.34}$$

Let f and g denote smooth vector fields and let v^* denote a smooth covector field, the Leibnitz-type formula that will be needed is

$$\mathbf{L}_f \langle v^*, g\rangle \;=\; \langle \mathbf{L}_f v^*, g\rangle \;+\; \langle v^*, [f, g]\rangle \tag{B.35}$$

If v^* is an exact one-form, then $v^* = d\lambda$ for some real-valued function λ, and (B.35) can be rewritten in the two equivalent forms

$$\mathbf{L}_f \langle d\lambda, g\rangle = \langle \mathbf{L}_f(d\lambda), g\rangle + \langle d\lambda, [f,g]\rangle \tag{B.36}$$

$$\mathbf{L}_f\mathbf{L}_g\lambda = \langle d(\mathbf{L}_f\lambda), g\rangle + \langle d\lambda, [f,g]\rangle \tag{B.37}$$

The two equalities $\langle d\lambda, f\rangle = \mathbf{L}_f\lambda$ and $\mathbf{L}_f d\lambda = d(\mathbf{L}_f\lambda)$ are used to write (B.36) in the form (B.37).

B.1.2 Distributions and the Frobenius Theorem

For $d \leq m$, a smooth d-dimensional distribution $\mathcal{D}$ on an m-dimensional manifold M is a smooth mapping which assigns to each point $p \in M$ a subspace $\mathcal{D}(p)$ of T_pM. A more insightful definition comes by considering a set of smooth, independent vector fields $\{\tau_1, \ldots, \tau_d\}$ defined on M. These vector fields define a distribution on M according to the rule

$$\mathcal{D} \triangleq span\{\tau_i : i \in [1, \ldots, d]\} \tag{B.38}$$

The notation $span\{\ldots\}$ denotes the subspace of a vector space which is the set of all linear combinations of elements of the argument list. Just as a tangent vector can be associated with a vector in a Euclidean space, $\mathcal{D}(p)$ can be associated with a subspace of Euclidean space with origin fixed at p. The subspace spanned by the set of *directions* $\{\tau_i\}$ is the set of directional derivatives associated with elements in $\mathcal{D}(p)$. The dimension of a distribution $\mathcal{D}$ at p is the dimension of the subspace $\mathcal{D}(p)$ of T_pM. A point $p \in M$ is a regular point of a distribution if there is an open neighborhood $U \subset M$ containing p such that the dimension of the distribution is constant on U. A vector field is contained in a distribution on a manifold, written

$\tau \in \mathcal{D}$, if $\tau(p) \in \mathcal{D}(p)$ for all $p \in M$. If $\mathcal{D}$ is a smooth d-dimensional distribution in a neighborhood U of a regular point p, then there is a set of d vector fields $\{\tau_i : i \in [1, \ldots, d]\}$ and d smooth functions $\{c_i : i \in [1, \ldots, d]\}$ such that every $\tau \in \mathcal{D}$ can be represented in the form

$$\tau = \sum_{i=1}^{d} c_i \tau_i \tag{B.39}$$

on U. The set of d vector fields on U is called a *local generator* for the distribution.

A distribution is *involutive* if the Lie bracket of any pair of vector fields is contained in the distribution. The statement that a distribution is closed under the bracketting operation is equivalent to requiring that the Lie bracket of two vector fields contained in the distribution can be expressed locally in the form of (B.39). If f is a vector field and $\mathcal{D}$ a distribution on M, then $[f, \mathcal{D}]$ is the distribution defined by the rule

$$[f, \mathcal{D}] \triangleq \{[f, \tau] : \forall\, \tau \in \mathcal{D}\} \tag{B.40}$$

It is natural to also introduce the objects which are dual to distributions. A *codistribution* $\mathcal{D}^*$ on a manifold M is a mapping assigning to each point of M a subspace of $T_p^* M$. If $\mathcal{D}^*$ is a smooth codistribution of dimension d at p, then there is a set of d independent covector fields τ_i^* defined on a neighborhood U containing p such that

$$\mathcal{D}^* = span\{\tau_i^* : i \in [1, \ldots, d]\} \tag{B.41}$$

Associated with a distribution $\mathcal{D}$ is a codistribution called the *annihilator* of $\mathcal{D}$, denoted $\mathcal{D}^{\perp}$, and defined by the rule

$$\mathcal{D}^{\perp} \triangleq \{v^* \in T_p^*M \,:\, \langle v^*, v\rangle = 0 \text{ for all } v \in \mathcal{D}(p)\} \tag{B.42}$$

If $\mathcal{D}$ is a d-dimensional distribution on an n-dimensional manifold, then $dim(\mathcal{D}) + dim(\mathcal{D}^{\perp}) = n$. In the Euclidean domain, this statement says that the dimension of a subspace plus the dimension of the orthogonal subspace is equal to the dimension of the entire space.

A nonsingular d-dimensional distribution $\mathcal{D}$ on an n-manifold N is *completely integrable* if at each $p \in N$ there exists a coordinate chart (U, ϕ) with coordinate functions $\phi_1, \ldots, \phi_n$ such that

$$\mathcal{D}(q) \;=\; span\left\{\left(\frac{\partial}{\partial \phi_1}\right)_q, \ldots, \left(\frac{\partial}{\partial \phi_d}\right)_q\right\} \tag{B.43}$$

for all $q \in U$. If there exists a coordinate chart (U, ϕ) such that (B.43) is satisfied, then it is possible to partition U into *slices* passing through any $p \in U$ defined by

$$S_p \triangleq \{q \in U \,:\, \phi_i(q) = \phi_i(p) \text{ for } d+1 \leq i \leq n\} \tag{B.44}$$

The slices S_p are d-dimensional manifolds which have the property that their tangent spaces coincide with the subspaces $\mathcal{D}(p)$ at each point of U. Another important aspect of completely integrable distributions is that they are involutive. This can be seen by considering the vector of component functions of an arbitrary tangent vector $\tau(x) = \{\tau_1(x), \ldots, \tau_n(x)\}^T$ in a completely integrable distribution. Definition (B.43) implies that there is a change of coordinates under which the last $n-d$

component functions are identically zero; therefore, the last $n - d$ rows of $\mathbf{D}\tau(x)$ are identically zero. Applying (B.19) to two such vector fields implies that the last $n-d$ component functions of the bracket are also zero and the bracket is contained in the distribution.

Clearly involutivity is a necessary condition for a distribution to be completely integrable; it is in fact true that it is also a sufficient condition.

Frobenius Theorem: A nonsingular distribution is completely integrable if and only if it is involutive.

The proof of sufficiency is fairly involved and can be found in any of the references cited in the introduction to this section.

The Frobenius Theorem leads to an algorithm for the construction of a change of coordinates that satisfies (B.43). Let $\mathcal{D}$ be a nonsingular involutive d-dimensional distribution, (U, ϕ) a coordinate system about $p \in N$, and $\{\tau_i : i \in 1, \ldots, d\}$ a local generator for $\mathcal{D}$ on U. In the natural basis of the chart, the τ_i can be represented by

$$\tau_i = \sum_{j=1}^{n} (\mathbf{L}_{\tau_i}\phi_j) \left(\frac{\partial}{\partial \phi_j}\right) \tag{B.45}$$

The objective is to find a new coordinate system (V, ψ) about p such that $\mathcal{D} = span\{\frac{\partial}{\partial\psi_1}, \ldots, \frac{\partial}{\partial\psi_d}\}$. In this basis, the set of local generators is given by

$$\tau_i = \sum_{j=1}^{n} (\mathbf{L}_{\tau_i}\psi_j) \left(\frac{\partial}{\partial \psi_j}\right) \tag{B.46}$$

Since the coordinate transformation must satisfy (B.43), it is necessary that

$$\mathbf{L}_{\tau_i}\psi_j = 0 \quad \text{for } 1 \leq i \leq d \text{ and } d+1 \leq j \leq n \tag{B.47}$$

In order to insure that the manifold have the same rank in the new coordinates, the matrix defined by

$$K = \begin{bmatrix} \mathbf{L}_{\tau_1}\psi_1 & \dots & \mathbf{L}_{\tau_1}\psi_d \\ \vdots & \ddots & \vdots \\ \mathbf{L}_{\tau_d}\psi_1 & \dots & \mathbf{L}_{\tau_d}\psi_d \end{bmatrix} \tag{B.48}$$

must have rank d on V. The coefficients of the d generators on V are given by

$$\begin{aligned} \mathbf{L}_{\tau_i}\psi_j(x) &= \sum_{k=1}^{n} \left[(\mathbf{L}_{\tau_i}\phi_k) \circ \phi^{-1}(x)\right] \frac{\partial\psi_j \circ \phi^{-1}(x)}{\partial x_k} \\ &= \sum_{k=1}^{n} \tau_{i,k}(x)\frac{\partial \xi_j}{\partial x_k} \\ &\triangleq [\mathbf{D}\xi(x)]\,[T(x)] \end{aligned} \tag{B.49}$$

where ξ has been introduced to denote the change of coordinates on U. The last equation, together with the requirements (B.47) and (B.48), yield the system of linear partial differential equations

$$[\mathbf{D}\xi(x)]\,[T(x)] = \begin{bmatrix} K(x) \\ 0_{(n-d)\times d} \end{bmatrix} \tag{B.50}$$

whose solution yields the change of coordinates function.

A solution of (B.50) has been interpreted as defining a transfomation under which a distribution has certain properties; there is a dual interpretation of the equation which indicates that solutions also characterize a certain codistribution. Solving the system of partial differential equations (B.50) is equivalent to finding $n-d$ real-valued functions $\xi_{d+1}, \ldots, \xi_n$ on a neighborhood V of p with the property that the covectors $d\xi_{d+1}(p), \ldots, d\xi_n(p)$ are linearly independent and

$$\langle d\xi_i(p), \tau_j(p)\rangle \;=\; 0; \quad \text{for } d+1 \leq i \leq n \text{ and } 1 \leq j \leq d \tag{B.51}$$

for all $p \in V$. The conditions (B.51) imply that the set of $n-d$ covector fields $\{d\xi_i\}$ are linearly independent on V, and span the $(n-d)$-dimensional codistribution $\mathcal{D}^{\perp}$. This interpretation of the Frobenius theorem will be used extensively in the proof of the linearization theorem stated in the next section.

B.2 Solution of the Feedback Linearization Problem

In this section, the linearization problem is solved for the nonlinear system of equations

$$\dot{x} \;=\; f(x) \;+\; \sum_{i=1}^{m} g_i(x)u_i \;\triangleq\; f(x) \;+\; G(x)u \tag{B.52}$$

where $f, g_1, \ldots, g_m$ are smooth vector fields, $x \in \mathbf{R}^n$, and $u \in \mathbf{R}^m$. The goal is to construct a state transformation $z = F(x)$ and an input transformation $u = \alpha(x) + \beta(x)v$ such that the system associated with the pair (z, v) in Figure 2.4 is linear. Two systems related in this way will be called *feedback equivalent*. For the transformation to be invertible, it is also necessary that the matrices $\beta(x)$ and $\mathbf{D}F(x)$ be nonsingular. Upon substitution into (B.52), the transformations (F, α, β)

must satisfy

$$\begin{aligned}
\dot{z} &= \left[\mathbf{D}F \circ (F^{-1}(z))\right] \cdot \left[(f + \sum_{i=1}^{m} g_i\alpha_i) \; + \; \sum_{i=1}^{m}(G\beta)_i\, v_i\right] \circ (F^{-1}(z)) && \text{(B.53)}\\
&= Az + \sum_{i=1}^{m} b_i v_i && \text{(B.54)}\\
&\triangleq Az + B\,v && \text{(B.55)}
\end{aligned}$$

In this section, the problem is solved assuming that (A, B) is a controllable pair.

All controllable linear systems can be transformed using feedback and linear coordinate changes in the state and input spaces to a system in Brunovsky form [61]. This is a canonical form where the A matrix is block diagonal, and the i^{th} block is a $\kappa_i \times \kappa_i$ matrix of the form

$$A_i = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \tag{B.56}$$

The corresponding column of the B-matrix is

$$b_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{B.57}$$

where the nonzero entry corresponds the the last row of the submatrix A_i. Note that all linear systems having a particular Brunovsky form are related by a transformation of the type being allowed for nonlinear systems, namely, feedback and coordinate changes in the state and input variable. If it is desired to solve the linearization problem for a particular pair, (A_d, B_d), it is sufficient to solve the problem for the pair (A, B) in Brunovsky canonical form. Without loss of generality, it can therefore be assumed that the system matrices (A, B) of (B.55) are in Brunovsky form.

The Brunovsky canonical form is a multivariable generalization of the *phase variable* form for single-input linear systems, where feedback has been used to place all system eigenvalues at zero. Figure B.2 shows a realization of a multi-variable system in Brunovsky form, where the κ_i are assumed to be in decreasing order. The set of integers κ_i correspond to the Kronecker indices of a minimal realization [37]. Inspection of (B.53) and (B.54) indicates that the feedback linearization problem is

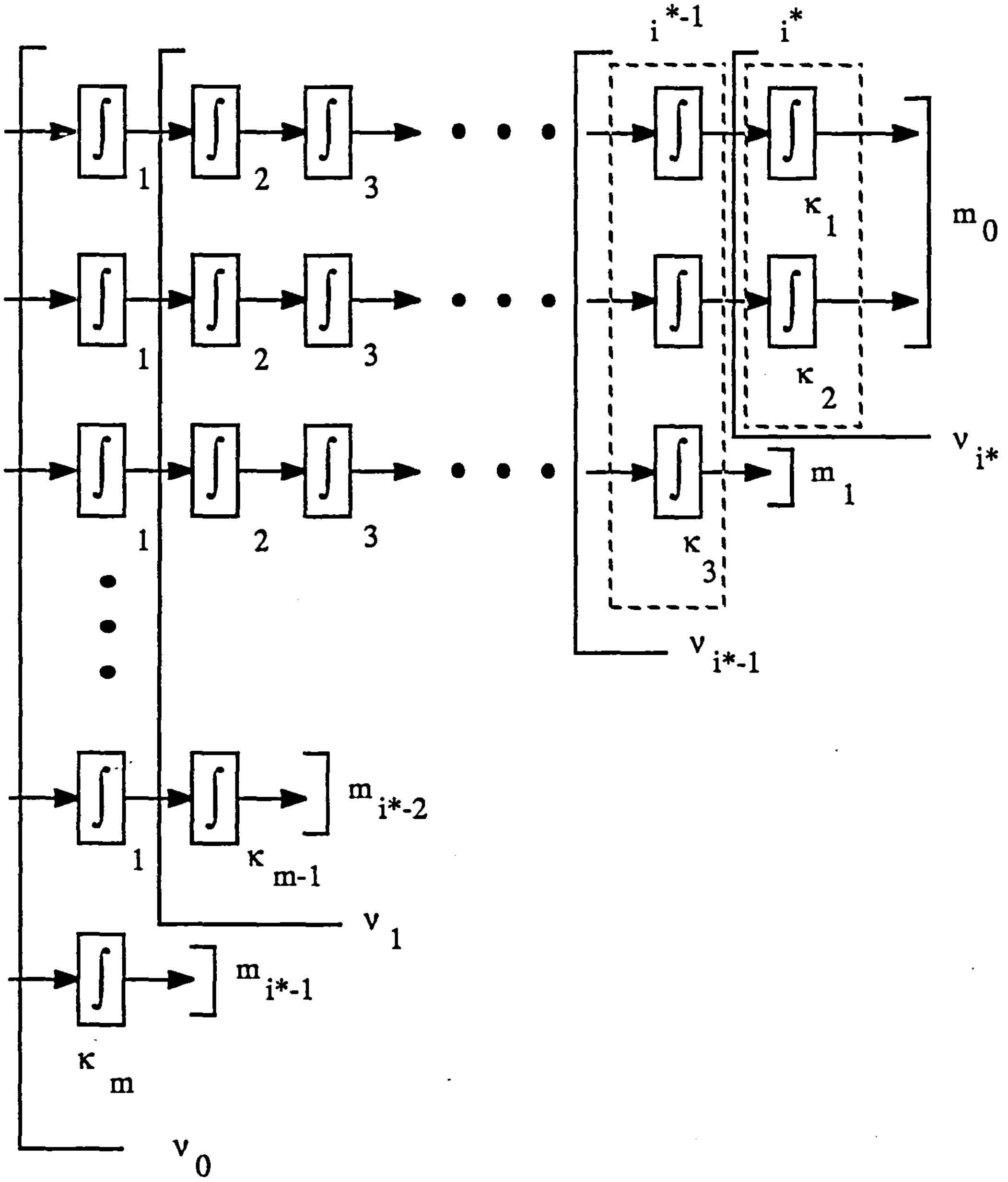


Figure B.2: Realization of a Brunovsky Canonical Form.

solvable if there exists a solution to the system of partial differential equations

$$\left[[\mathbf{D}F] \cdot [f + \sum_{i=1}^{m} g_i \alpha_i] \right] \circ (F^{-1}(z)) = A\, z \tag{B.58}$$

$$[\mathbf{D}F] \cdot [G\beta]_i \circ (F^{-1}(z)) = b_i \text{ for } 1 \leq i \leq m \tag{B.59}$$

on an open neighborhood about some point. In this section, necessary and sufficient conditions are derived which guarantee the existence of solutions to these equations.

Before proceding with the proof, it is necessary to state several definitions that will be used. Given the set of vector fields $f, g_1, \ldots, g_m$, the set of distributions $\mathcal{G}_0, \mathcal{G}_1, \ldots$ are defined recursively by

$$\mathcal{G}_0 \triangleq span\{g_1, \ldots, g_m\} \tag{B.60}$$

$$\mathcal{G}_i \triangleq \mathcal{G}_{i-1} + [f, \mathcal{G}_{i-1}] \tag{B.61}$$

If each of the distributions $\mathcal{G}_i$ is nonsingular, then

$$\mathcal{G}_i = span\{ad_f^k g_j : \ 0 \leq k \leq i, \ 1 \leq j \leq m\} \tag{B.62}$$

where the operator $ad_f^k g_j$ is the vector field constructed by the recursion $ad_f^0 g_j \triangleq g_j$ and $ad_f^k g_j \triangleq [f, ad_f^{k-1} g_j]$. The importance of the result (B.62) is that it yields an algorithm for constructing a basis or local generator for the distribution $\mathcal{G}_i$. It also yields an algorithm for constructing the dual basis for $\mathcal{G}_i^{\perp}$. In other words, if the vector fields $ad_f^k g_j$ span an m-dimensional subspace of the tangent space to an n-manifold, there will be $n - m$ exact covector fields $\{d\lambda_i\}$ - by the Frobenius Theorem - such that $\left\langle ad_f^k g_j, d\lambda_i \right\rangle = 0$ for each basis element of $\mathcal{G}_i$. Since the one-forms annihilate the basis elements of the distribution, every linear combination of the $\{d\lambda_i\}$ annihilates every vector in the distribution $\mathcal{G}_i$; therefore, the exact one-forms are a basis for $\mathcal{G}_i^{\perp}$. The previous argument is used extensively in the proof of the linearization theorem.

The sequence of integers $\nu_0, \nu_1, \ldots$ is defined by setting

$$v_0 \triangleq dim\ \mathcal{G}_0 \tag{B.63}$$

$$v_i \triangleq dim\ \mathcal{G}_i - dim\ \mathcal{G}_{i-1}, \quad i \geq 1 \tag{B.64}$$

$$\tag{B.65}$$

If ν_{i^*} denotes the last nonzero element in the sequence $\{\nu_i\}$, and if $dim\ \mathcal{G}_{i^*} = n$, then

$$\sum_{i=0}^{i^*} \nu_i = n \tag{B.66}$$

The set of indices $\{m_0, \ldots, m_{i^*}\}$ is defined according to

$$\begin{aligned} m_0 &= \nu_{i^*} \\ m_0 + m_1 &= \nu_{i^*-1} \\ m_0 + m_1 + m_2 &= \nu_{i^*-2} \\ &\vdots \\ m_0 + \cdots + m_{i^*} &= \nu_0 \end{aligned} \tag{B.67}$$

If $dim\ \mathcal{G}_{i^*} = n$, it follows that

$$dim\ \mathcal{G}_{i^*-i}^{\perp} = i \cdot m_0 + (i-1) \cdot m_1 + 2 \cdot m_{i-2} + m_{i-1}, \quad 1 \leq i \leq i^* \tag{B.68}$$

The last set of indices that is needed is given by

$$\begin{array}{lll} \kappa_i = i^* + 1 & \text{if} & 1 \leq i \leq \nu_{i^*} \\ \kappa_i = i^* & \text{if } m_1 > 0 \text{ and} & \nu_{i^*} + 1 \leq i \leq \nu_{i^*-1} \\ \kappa_i = i^* - 1 & \text{if } m_2 > 0 \text{ and} & \nu_{i^*-1} + 1 \leq i \leq \nu_{i^*-2} \\ & \vdots & \\ \kappa_i = 1 & \text{if } m_{i^*} > 0 \text{ and} & \nu_1 + 1 \leq i \leq \nu_{\nu_0} \end{array} \tag{B.69}$$

The interpretation of the three sets of indices is made more clear by referring to Figure B.2. The number i^* is the number of *banks* of integrators in the figure. The κ_i are the number of integrators in the i^{th} string of integrators, and hence, the Kronecker indices of the realization. The set of integers ν_i denote the number of state variables that are integrated in the $(i+1)^{\text{th}}$ bank of integrators - counting from the left. The m_i are the number of state variables coming out of the $(i+1)^{\text{th}}$ bank of integrators - counting from the right - whose integrals are not state variables.

The conditions under which the feedback linearization problem is solvable can now be stated [57].

Theorem: The feedback linearization problem is solvable on a neighborhood U of a point p if and only if p is a regular point of $\mathcal{G}_i$ for all $i \geq 0$, $dim(\mathcal{G}_{i^*}) = n$, and the distribution $\mathcal{G}_i$ is involutive for all i such that $m_{i^*-i-1} \neq 0$.

The requirement that p be a regular point of the distributions implies that the qualitative characteristics of the dynamical system are similar on a neighborhood of p. If the differential equation (B.52) describes a physical system, the points which are not regular points often indicate pathological conditions (e.g. a change in the number of independent state variables). The requirement that $dim(\mathcal{G}_{i^*} = n$ insures that the nonlinear system does have n independent state variables and is locally controllable. The involutiveness of certain $\mathcal{G}_i$ are the critical conditions which will be interpretted in the following proof.

Without loss of generality, it can be assumed that the set of vector fields $g_1, \ldots, g_m$ is independent, or equivalently, that $v_0 = m$. It will be seen from the construction which follows that the number of control variables necessary to solve the problem equals the dimension of the distribution $\mathcal{G}_0$. If one of the g_i is not independent, it can be dropped by assuming $u_i \equiv 0$ from (B.52) without affecting the construction of the linearizing transformation. The issue of redundant input variables will be discussed in more detail in Chapter 4. The sufficiency of the preceeding theorem is proven in detail. Since the proof is constructive, the argument can be reversed to show necessity of the stated conditions. The proof of necessity is omitted since it is available in the literature, and since the construction used in the sufficiency proof is used in the next section to linearize the two systems of equations derived in the previous chapter.

The most difficult part of the proof is the construction of a coordinate transformation about the point p. By assumption, the dimension of the distribution $\mathcal{G}_{i^*}$ is n, and the distribution $\mathcal{G}_{i^*-1}$ is involutive and has dimension $n - m_0$; these assumptions imply that $dim\ \mathcal{G}_{i^*-1}^{\perp} = m_0$. From the Frobenius Theorem, there is a neighborhood U_1 of p and m_0 exact one-forms on U_1 that span $\mathcal{G}_{i^*-1}^{\perp}$. Since the m_0 covector fields are exact, each can be associated with one of m_0 smooth functions

h_{0i} such that

$$\langle dh_{0i}, v\rangle = 0 \text{ for } 1 \le i \le m_0 \text{ and } \forall\, v \in \mathcal{G}_{i^*-1} \tag{B.70}$$

In particular, this is true for the elements $ad_f^k g_j$ for $1 \le j \le m$ and $0 \le k \le i^* - 1$; consequently, the $m_0 \times m$ matrix

$$M_0(x) \; = \; [m_{ij}^0(x)] \; \triangleq \; \left[\left\langle dh_{0i}, ad_f^{i^*} g_j\right\rangle(x)\right] \tag{B.71}$$

has rank m_0 on U_1. If this were false at some q, there would be a set of nonzero real numbers $a_1, \ldots, a_{m_0}$ for which

$$\sum_{i=1}^{m_0} a_i \left\langle dh_{0i}, ad_f^{i^*} g_j\right\rangle(q) \; = \; \left\langle \sum_{i=1}^{m_0} a_i\, dh_{0i}, ad_f^{i^*} g_j \right\rangle(q) \; = \; 0 \tag{B.72}$$

for all $j \in [1 \ldots m]$. Taken with (B.70), (B.72) implies that there is a covector field $d\lambda$ such that

$$\langle d\lambda, v\rangle \; \triangleq \; \left\langle \sum_{i=1}^{m_0} a_i dh_{0i}(q), v \right\rangle(q) \; = \; 0 \;\; \forall\, v \in \mathcal{G}_{i^*}. \tag{B.73}$$

Since $dim(\mathcal{G}_{i^*}) = n$ implies that $dim(\mathcal{G}_{i^*}^{\perp}) = 0$, the covector $d\lambda = 0$, and independence of the $\{dh_{0i}\}$ requires that $a_1 = \cdots = a_{m_0} = 0$ contradicting the assumption.

Now consider the distribution $\mathcal{G}_{i^*-2}$ of dimension $n - 2m_0 - m_1$ and the codistribution $\mathcal{G}_{i^*-2}^{\perp}$ of dimension $2m_0 + m_1$. The Leibnitz formula (B.36) can be applied to the covector fields dh_{0i} to show that

$$-\left\langle d\mathbf{L}_f h_{0i}, ad_f^k g_j\right\rangle \; = \; \left\langle dh_{0i}, ad_f^{k+1} g_j\right\rangle - \mathbf{L}_f \left\langle dh_{0i}, ad_f^k g_j\right\rangle \; = \; 0 \tag{B.74}$$

for $1 \le j \le m$, $1 \le i \le m_0$, and $0 \le k \le i^* - 2$ on the set U_1. This necessarily implies that

$$\langle d\mathbf{L}_f dh_{0i}, v\rangle = 0 \text{ for } 1 \leq i \leq m_0 \text{ and } \forall\, v \in \mathcal{G}_{i^*-2}^{\perp} \tag{B.75}$$

It is also true that the set of covector fields

$$dh_{01}, \ldots, dh_{0m_0}, d\mathbf{L}_f h_{01}, \ldots, d\mathbf{L}_f h_{0m_0} \tag{B.76}$$

are linearly independent for all $x \in U_1$. If the assertion were not true, there would be two sets of m_0 real numbers, $\{a_i\}$ and $\{b_i\}$, such that

$$\sum_{i=1}^{m_0} a_i\, dh_{0i}(q) \;+\; \sum_{i=1}^{m_0} b_i\, d\mathbf{L}_f h_{0i}(q) \;=\; 0 \tag{B.77}$$

at some point $q \in U_1$. This implies that

$$\begin{aligned} 0 \;&=\; \left\langle \textstyle\sum_{i=1}^{m_0} a_i\, dh_{0i}(q) \;+\; \sum_{i=1}^{m_0} b_i\, d\mathbf{L}_f h_{0i}(q), ad_f^{i^*-1} g_j \right\rangle \\ &=\; \left\langle \textstyle\sum_{i=1}^{m_0} b_i\, d\mathbf{L}_f h_{0i}(q), ad_f^{i^*-1} g_j \right\rangle \\ &=\; -\;\left\langle \textstyle\sum_{i=1}^{m_0} b_i\, dh_{0i}(q), ad_f^{i^*} g_j \right\rangle \end{aligned} \tag{B.78}$$

for all $1 \leq j \leq m$. The second equality follows from linearity of $\langle\cdot,\cdot\rangle$ and (B.70), and the third equality comes from the fact that $\mathbf{L}_f d\lambda = d(\mathbf{L}_f \lambda)$ and the Leibnitz formula. The last equality can only be satisfied if all the b_i are zero. Since $b_i = 0$ for all i and the dh_{0i} are independent, (B.77) can only be satisfied if $a_i = 0$ for each i.

If $m_1 = 0$, then we have constructed $2m_0$ vectors that span $\mathcal{G}_{i^*-2}^{\perp}$. If $m_1 > 0$, $\mathcal{G}_{i^*-2}$ is involutive and the Frobenius theorem guarantees that there are m_1 functions $h_{11}(x), \ldots, h_{1m_1}$ such that the covector fields $dh_{11}(x), \ldots, dh_{1m_1}$, together with (B.76) are all linearly independent and

$$\langle dh_{1i}, v\rangle = 0 \;\text{ for }\; 1 \le i \le m_1 \;\text{ and }\; \forall\, v \in \mathcal{G}_{i^*-2} \tag{B.79}$$

on some neighborhood $U_2 \subset U_1$ of p.

The $m_1 \times m$ matrix $M_1(x)$ is defined to be

$$M_1(x) \;=\; [m_{ij}(x)] \;=\; \left[\left\langle dh_{1i}, ad_f^{i^*-1} g_j\right\rangle (x)\right] \tag{B.80}$$

By construction, the $(m_0 + m_1) \times m$ matrix given by

$$\begin{bmatrix} M_0 \\ M_1 \end{bmatrix} \tag{B.81}$$

has rank $m_0 + m_1$ at each point in the set U_2. If the statement were not true, there would be a set of real numbers a_i, b_i such that at some point $q \in U_2$

$$\begin{aligned} 0 \;&=\; \left\langle \textstyle\sum_{i=1}^{m_0} a_i\, dh_{0i}, ad_f^{i^*} g_j\right\rangle (q) \;+\; \left\langle \textstyle\sum_{i=1}^{m_1} b_i\, dh_{1i}, ad_f^{i^*-1} g_j\right\rangle (q) \\ &=\; \left\langle - \textstyle\sum_{i=1}^{m_0} a_i\, d\mathbf{L}_f h_{0i} \;+\; \textstyle\sum_{i=1}^{m_1} b_i\, dh_{1i}, ad_f^{i^*-1} g_j\right\rangle (q) \\ &\triangleq\; \left\langle d\lambda, ad_f^{i^*-1} g_j\right\rangle (q) \end{aligned} \tag{B.82}$$

The second equality follows by the same trick used in (B.78). In combination with (B.82), the fact (B.79) implies that $d\lambda$ must be contained in the annihilator of $\mathcal{G}_{i^*-1}$. Since $\mathcal{G}_{i^*-1}$ is spanned by the covectors dh_{0i}, $d\lambda$ must be a linear combination of them. Because of the independence of the covectors (B.76) and (B.79), (B.82) can only be true if all a_i and b_i are zero, a contradiction.

Continuing this procedure leads to a set of covector fields

$$\begin{array}{l} dh_{01},\ldots,dh_{0m_0},d\mathbf{L}_f h_{01},\ldots,d\mathbf{L}_f h_{0m_0},\ldots,d\mathbf{L}_f^{i^*-1}h_{01},\ldots,d\mathbf{L}_f^{i^*-1}h_{0m_0} \\ dh_{11},\ldots,dh_{1m_1},\ldots,d\mathbf{L}_f^{i^*-2}h_{11},\ldots,d\mathbf{L}_f^{i^*-2}h_{1m_1} \\ \vdots \\ dh_{(i^*-1)1},\ldots,dh_{(i^*-1)m_{(i^*-1)}} \end{array} \tag{B.83}$$

There will be

$$i^*m_0+(i^*-1)m_1+\cdots+2m_{i^*-2}+m_{i^*-1} \;=\; n-m \tag{B.84}$$

of these covector fields, and by construction, the set is independent on some neighborhood U containing the point p. Associated with each nonzero m_i there is an $m_i \times m$ matrix

$$M_i \;=\; [m_{jk}]_i(x) \;\triangleq\; \left[\left\langle dh_{ik}, ad_f^{i^*-i}g_j\right\rangle\right](x) \tag{B.85}$$

such that the matrix defined by

$$\begin{bmatrix} M_0 \\ \vdots \\ M_{i^*-1} \end{bmatrix} \tag{B.86}$$

has rank ν_1 at all $x \in U$.

The covector fields given by

$$d\mathbf{L}_f^{i^*}h_{01},\ldots,d\mathbf{L}_f^{i^*}h_{0m_0},\ldots,d\mathbf{L}_f h_{(i^*-1)1},\ldots,d\mathbf{L}_f h_{(i^*-1)(m_{i^*-1})} \tag{B.87}$$

together with (B.83) yield a set of $(n-m)+m_{i^\bullet-1}$ linearly independent one-forms. If there are in fact n such one-forms, the construction is complete; otherwise, $m_{i^\bullet} > 0$ and there must exist a set of covector fields $h_{i^\bullet 1}, \ldots, h_{i^\bullet m_{i^\bullet}}$ which combined with (B.83) and (B.87) span the n-dimensional cotangent space at p.

At this point, the construction of a coordinate transformation is complete; the remaining step is merely to realize this fact. The n functions $\mathbf{L}_f^k h_{ij}(x)$ have been constructed such that in local coordinates, their derivatives are independent about p. If these n functions are regarded together as a mapping on an n-dimensional manifold, the inverse function theorem guarantees that the mapping has a smooth inverse about p. This mapping is therefore a coordinate transformation about p. Consider one of the functions $h_{ij}(x)$, what is the interpretation of the Lie derivative $\mathbf{L}_f h_{ij}(x)$ in the context of (B.52). Assuming that $u = 0$, $\mathbf{L}_f h_{ij}(x)$ can be expressed in local coordinates as

$$\begin{aligned} \mathbf{L}_f h_{ij}(x) &= \sum_{k=1}^{n} \frac{\partial h_{ij}}{\partial x_k} f_k \\ &= \sum_{k=1}^{n} \frac{\partial h_{ij}}{\partial x_k} \frac{dx_k}{dt} \end{aligned} \tag{B.88}$$

Clearly, the Lie derivative $\mathbf{L}_f h_{ij}(x)$ is the time derivative of $h_{ij}(x)$ as the nonlinear system evolves without the influence of controls. More formally, the coordinate transformation is constructed by letting

$$\begin{array}{lcllll} y_{i1}(x) & \triangleq & h_{0i}(x) & if & 1 \leq i \leq \nu_{i^\bullet} & \\ y_{i1}(x) & \triangleq & h_{1(i-\nu_{i^\bullet})}(x) & if & m_1 \geq 0 \text{ and } \nu_{i^\bullet}+1 \leq i \leq \nu_{i^\bullet-1} & \\ & \vdots & & & & \\ y_{i1}(x) & \triangleq & h_{i^\bullet(i-\nu_1)}(x) & if & m_{i^\bullet} \geq 0 \text{ and } \nu_1+1 \leq i \leq \nu_0 & \end{array} \tag{B.89}$$

Figure B.3 shows the relationship between the functions constructed in the proof of the linearization theorem and the state-variables of the Brunovsky canonical form. The function $y_{i1}(x)$ maps the state variable x to the output of the i^{th} string of integrators in the Brunovsky form shown in Figure B.3. The state variables associated with the i^{th} string of integrators of the system in Brunovsky form are given by

$$\begin{aligned} y_{i1}(x) &= h_{0i}(x) \\ y_{i2}(x) &\triangleq \dot{y}_{i1} = \mathbf{L}_f h_{0i}(x) \\ &\vdots \\ y_{i\kappa_i}(x) &\triangleq \dot{y}_{i(\kappa_i-1)} = \mathbf{L}_f^{\kappa_i-1} h_{0i}(x) \end{aligned} \tag{B.90}$$

for $1 \leq i \leq m$ and the coordinate change in the state-space is

$$z \triangleq [y_{11}(x), \ldots, y_{1(\kappa_1-1)}(x), \ldots, y_{m(\kappa_m-1)}(x)]^T \triangleq F(x) \tag{B.91}$$

The final part of the proof is to construct a linearizing feedback. Consider one of the coordinate transformations $y_{i1}(x)$. From the construction of the coordinate transformation, the dual products

$$\left\langle d\mathbf{L}_f^k y_{i1}, g_j \right\rangle = 0 \tag{B.92}$$

for $1 \leq i, j \leq m$ and $0 \leq k \leq \kappa_i - 2$. Together with the fact that $\langle d\lambda, v \rangle = \mathbf{L}_v \lambda$, (B.92) implies that

$$\left\langle d\mathbf{L}_f^k y_{i1}, g_j \right\rangle = \mathbf{L}_{g_j} \mathbf{L}_f^k y_{i1} \tag{B.93}$$

Because $\mathbf{L}_g \mathbf{L}_f^k y_{i1}(x) = 0$ for $0 \leq k \leq \kappa_i - 2$, the Lie derivative $\mathbf{L}_f^k y_{i1}(x)$ depends only on $\mathbf{L}_f^{k+1} y_{i1}(x)$. This is equivalent to $\mathbf{L}_{\dot{x}} \mathbf{L}_f^k y_{i1}(x) = \mathbf{L}_f^{k+1} y_{i1}(x)$ for the specified

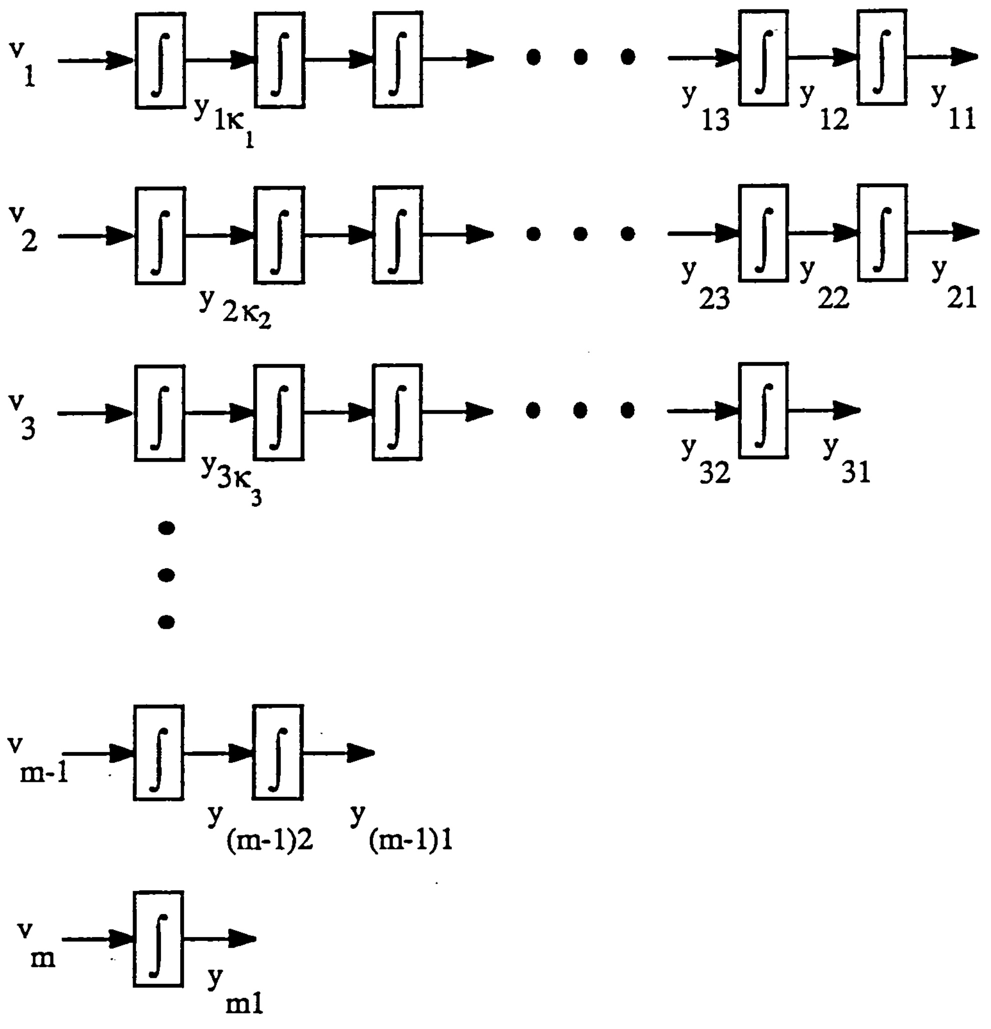


Figure B.3: Canonical Form of Linearized System with State-variables.

range of k. The remaining step is to compute the feedback and the input transformation such that the control variables $v_i \triangleq \mathbf{L}_{\dot{x}}\mathbf{L}_f^{\kappa_i - 1} y_{i1}(x)$ can be used to specify the derivatives of the variables $y_{i(\kappa_i-1)}$. This is accomplished by first computing the value of v_i associated with a specified state-control pair (x, u) yielding

$$\begin{aligned} \mathbf{L}_{\dot{x}}\mathbf{L}_f^{\kappa_i-1} y_{i1}(x) &= \mathbf{L}_{f+Gu}\mathbf{L}_f^{\kappa_i-1} y_{i1}(x) \\ &= \mathbf{L}_f\mathbf{L}_f^{\kappa_i-1} y_{i1}(x) + \mathbf{L}_{Gu}\mathbf{L}_f^{\kappa_i-1} y_{i1}(x) \\ &= \mathbf{L}_f^{\kappa_i} y_{i1}(x) + \textstyle\sum_{j=1}^{m} \mathbf{L}_{g_i u_i}\mathbf{L}_f^{\kappa_i-1} y_{i1}(x) \\ &= \mathbf{L}_f^{\kappa_i} y_{i1}(x) + \textstyle\sum_{j=1}^{m} \left\langle d\mathbf{L}_f^{\kappa_i-1} y_{i1}(x), g_i \right\rangle u_i \end{aligned} \qquad (B.94)$$

The m expressions for the individual components v_i can be combined to obtain a single vector-matrix equation

$$v = \begin{bmatrix} \mathbf{L}_f^{\kappa_1} y_{11}(x) \\ \vdots \\ \mathbf{L}_f^{\kappa_m} y_{m1}(x) \end{bmatrix} + \left[\left\langle d\mathbf{L}_f^{\kappa_i-1} y_{i1}(x), g_j \right\rangle\right] u \triangleq w(x) + A(x)\, u \qquad (B.95)$$

By construction, the matrix $A(x)$ is nonsingular on the neighborhood U containing p; therefore, (B.95) can be solved for the u which produces a desired v. Defining

$$u = \alpha(x) + \beta(x) v \quad where \quad \begin{aligned} \alpha(x) &= -A^{-1}(x) \begin{bmatrix} \mathbf{L}_f^{\kappa_1} y_{11}(x) \\ \vdots \\ \mathbf{L}_f^{\kappa_m} y_{m1}(x) \end{bmatrix} \\ \beta(x) &= A^{-1}(x) \end{aligned} \qquad (B.96)$$

the feedback linearization problem is solved by the state coordinate change $F(x)$, the state feedback $\alpha(x)$, and the linear change of coordinate in the input space $\beta(x)$.

Bibliography

[1] T. Vargas. *Attitude Control Augmentation of Spacecraft in low Earth Orbit Utilising Aerodynamic Forces.* Master's thesis, Massachusetts Institute of Technology, 1981. CSDL-T-760.

[2] N. Bedrossian. *Steering Law Design for Redundant Single Gimbal Control Moment Gyro Systems.* Master's thesis, Massachusetts Institute of Technology, 1987. CSDL-T-965.

[3] G. Margulies and J. Aubrun. Geometric theory of single-gimbal control moment gyro systems. *J. of Astronautical Sciences*, 26(2):159–191, April-June 1978.

[4] J. Paradiso. A highly adaptable steering/selection procedure for combined CMG/RCS spacecraft control. In *10'th Annual AAS Guidance and Control Conf.*, January-February 1987. AAS 87-002.

[5] H. Kennel. *Steering law for parallel mounted double-gimballed control moment gyros - revision A.* Technical Report NASA TM - 82390, NASA, January 1981.

[6] T. Yoshikawa. *A steering law for three double-gimbal control moment gyro systems.* Technical Report NASA TM - 82390, NASA, January 1981.

[7] K. Kurokawa, N. Yajima, and S. Usui. A new steering law of a single-gimbal cmg system of pyramid configuration. In *X'th IFAC Symposium on Automatic Control in Space*, 1981.

[8] D. Cornick. Singularity avoidance control laws for single gimbal control moment gyros. In *1979 AIAA Guidance and Control Conf.*, Aug. 1979.

[9] Y. Nakamura and H. Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *ASME J. Dyn. Sys., Meas., and Cont.*, 108:163–171, 1986.

[10] B. Colburn, L. White, and J. Boland III. Some simple cmg steering laws for spacecraft attitude control systems. In *9'th Annual Asilomar Conf. in Circuits, Systems, and Computers*, Nov. 1975.

[11] B. Colburn and L. White. Computational considerations for a spacecraft attitude control system employing control moment gyros. *AIAA J. of Spacecraft Dynamics and Control*, 14(1):45–53, 1977.

[12] H. Kennel. *Angular momentum desaturation for ATM/LM/CSM configuration using gravity gradient torques.* Technical Report NASA TM X - 53764, NASA, August 1968.

[13] D. Geller. Continuous momentum management for the space station power tower configuration. In *1985 AAS/AIAA Astrodynamics Specialist Conf.*, Aug. 1985.

[14] P. Hattis. Predictive momentum management for the space station. *J. Guidance*, 9(4):454, July-Aug. 1986.

[15] L. Bishop, R. Bishop, and K. Lindsay. Proposed CMG momentum management scheme for space station. In *AIAA Guidance, Navigation, and Control Conf.*, 1987.

[16] L. Lenorovitz. Soviets use new gyros to stabilize MIR Station. *Aviation week and space technology*, November 1987.

[17] H. Kennel. *A control law for double-gimballed control moment gyros used for space vehicle attitude control.* Technical Report NASA TM X - 64536, NASA, August 1970.

[18] Architectural control document: guidance, navigation, and control system. Space Station Program Office, January 1987. ACD no. 3/JSC 30259.

[19] L. Hunt and R. Su. Local transformations for multi-input nonlinear systems. In *Joint Automatic Control Conf.*, 1981.

[20] T. Dwyer. Exact nonlinear control of large angle rotational maneuvers. *IEEE Trans. on Auto. Control*, 29(9):769–774, September 1984.

[21] T. Dwyer. Exact nonlinear control of spacecraft slewing maneuvers with internal momentum transfer. *AIAA J. Guidance*, 9(2):240–247, March 1986.

[22] G. Meyer, R. Su, and L. Hunt. Application of nonlinear transformations to automatic flight control. *Automatica*, 20(1):103–107, 1984.

[23] G. Meyer. The design of exact nonlinear model followers. In *1981 JAC Conference*, 1981.

[24] T. Tarn and et. al. Nonlinear feedback in robot arm control. In *23rd Conference on Decision and Control*, Dec. 1984.

[25] B. Persson. Estimating aerodynamic drag and center of pressure. 1987. C. S. Draper Laboratory internal memo: Space Station 87-10.

[26] D. Whitney. The mathematics of coordinated control of prostheses and manipulators. *ASME J. of Dyn. Systems, Meas. and Control*, 94:303–309, 1972.

[27] M. Nagurka and V. Yen. Fourier-based optimal control of nonlinear dynamical systems. In *ASME Winter Annual Meeting*, December 1987.

[28] L. Brand. *Vector and Tensor Analysis.* John Wiley and Sons, 1947.

[29] J. Paradiso. An efficient OFS-compatible kinematic model for generalized CMG's. March 1985. C. S. Draper Laboratory internal memo: Space Station 85-9.

[30] P. Hughes. *Spacecraft Attitude Dynamics.* John Wiley and Sons, 1986.

[31] A. Krener. On the equivalence of control systems and the linearization of nonlinear systems. *SIAM J. Control*, 11(4):670–676, November 1973.

[32] R. Brockett. Feedback invariants for nonlinear systems. In *Proc. VII IFAC Congress*, pages 1115–1120, Helsinki, 1978.

[33] A. Isidori and A. Ruberti. On the synthesis of linear input-output responses for nonlinear systems. *Systems and Control Letters*, 17–22, January 1984.

[34] A. Krener, A. Isidori, and W. Respondek. Partial and robust linearization by feedback. In *Proc. 22nd IEEE CDC Conference*, pages 126–130, 1983.

[35] L. Hunt, R. Su, and G. Meyer. Global transformations of nonlinear systems. *IEEE Trans. on Auto. Control*, 28(1):24–31, January 1983.

[36] A. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems and Control Letters*, 47–52, June 1983.

[37] T. Kailath. *Linear Systems. Prentice-Hall Information and System Sciences Series*, Prentice-Hall, 1980.

[38] E. Bergmann, S. Croopnick, J. Turkovich, and C.Work. An advanced autopilot concept. *J. of Guidance and Control*, 2(3):161–168, May-June 1979.

[39] T. Dwyer, G. Lee, and Ning Chen. Exact nonlinear model followers for the control of industrial robots. In *Robotic Intelligence and Productivity Conf.*, Nov. 1983.

[40] T. Macloed, K. Rigsby, and M. Friedman. Rephased dual keel mass properties and TEA analysis. LEMSCO, August 1986.

[41] M. Hopkins and E. Hahn. Autonomous momentum management for the cdg planar space station. In *AIAA* 23rd *Aerospace Science Meeting*, January 1985. Reno, Nevada.

[42] M. Athans and P. Falb. *Optimal Control: An Introduction to the Theory and Its Applications.* McGraw-Hill, Inc., 1966.

[43] A. Bryson and Y. Ho. *Applied Optimal Control.* Hemisphere Publishing Corp., 1975.

[44] R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[45] R. Su. On the linear equivalents of nonlinear systems. *Systems and Control Letters*, 2(1):48–52, July 1982.

[46] G. Dahlquist and A. Bjoerck. *Numerical Methods. Prentice-Hall Series in Automatic Computation*, Prentice-Hall, 1974.

[47] B. Finlayson. *The Method of Weighted Residuals and Variational Principles.* Volume 87 of *Mathematics in Science and Engineering*, Academic Press, 1972.

[48] J. Vlassenbroeck and R. Van Dooren. A chebyshev technique for solving nonlinear optimal control problems. *IEEE Trans. on Auto. Control*, 33(4):333–340, April 1988.

[49] C. Lanczos. *Applied Analysis.* Prentice Hall, Inc., 1956.

[50] K. Wright. Chebyshev collocation methods for ordinary differential equations. *Chem. Eng. Sci.*, 22:1483–1501, 1967.

[51] C. Hargraves and S. Paris. Direct trajectory optimization using nonlinear programming and collocation. *AIAA J. Guidance*, 10(4):338–342, 1987.

[52] M.J.D. Powell. *A fast algorithm for nonlinearly constrained optimization calculations.* Volume 630 of *Lecture Notes in Mathematics*, Springer-Verlag, 1977.

[53] M. Urabe. Numerical solution of multi-point boundary value problems in chebyshev series: theory of the method. *Numer. Math.*, 9:341–366, 1967.

[54] J. Villadsen and W. Stewart. Solution of boundary value problems by orthogonal collocation. *Chem. Eng. Sci.*, 22:1483–1501, 1967.

[55] H. Woo, H. Morgan, and E. Falangas. Momentum management and attitude control design for a space station. *J. of Guidance and Control*, 11(1):19–25, Jan.-Feb. 1988.

[56] R. Su, G. Meyer, and L. Hunt. Robustness in nonlinear control. In *Differential Geometric Control Theory*, pages 268–297, Birkhauser, 1982.

[57] A. Isidori. *Nonlinear Control Systems: An Introduction.* Volume 72 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, 1985.

[58] F. Warner. *Foundations of Differentiable Manifolds and Lie Groups.* Volume 94 of *Graduate Tests in Mathematics*, Springer-Verlag, 1983.

[59] W. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, 1975.

[60] R. Abraham, J. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications. Global Analysis: Pure and Applied*, Addison-Wesley Publishing Co., 1983.

[61] P. Brunovsky. A classification of linear controllable systems. *Kybernetika*, 6:173–188, 1970.

End of Document