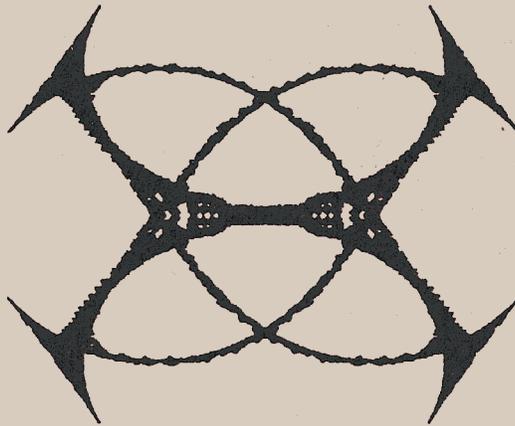


CSDL-R-2261

**A SEARCH-BASED APPROACH TO
STEERING SINGLE GIMBALLED CMGS**

by
Joseph A. Paradiso

August 1991



**Approved for public release;
distribution unlimited.**

NASA-JSC 3/6/91



The Charles Stark Draper Laboratory, Inc.
555 Technology Square, Cambridge, Massachusetts 02139-3563

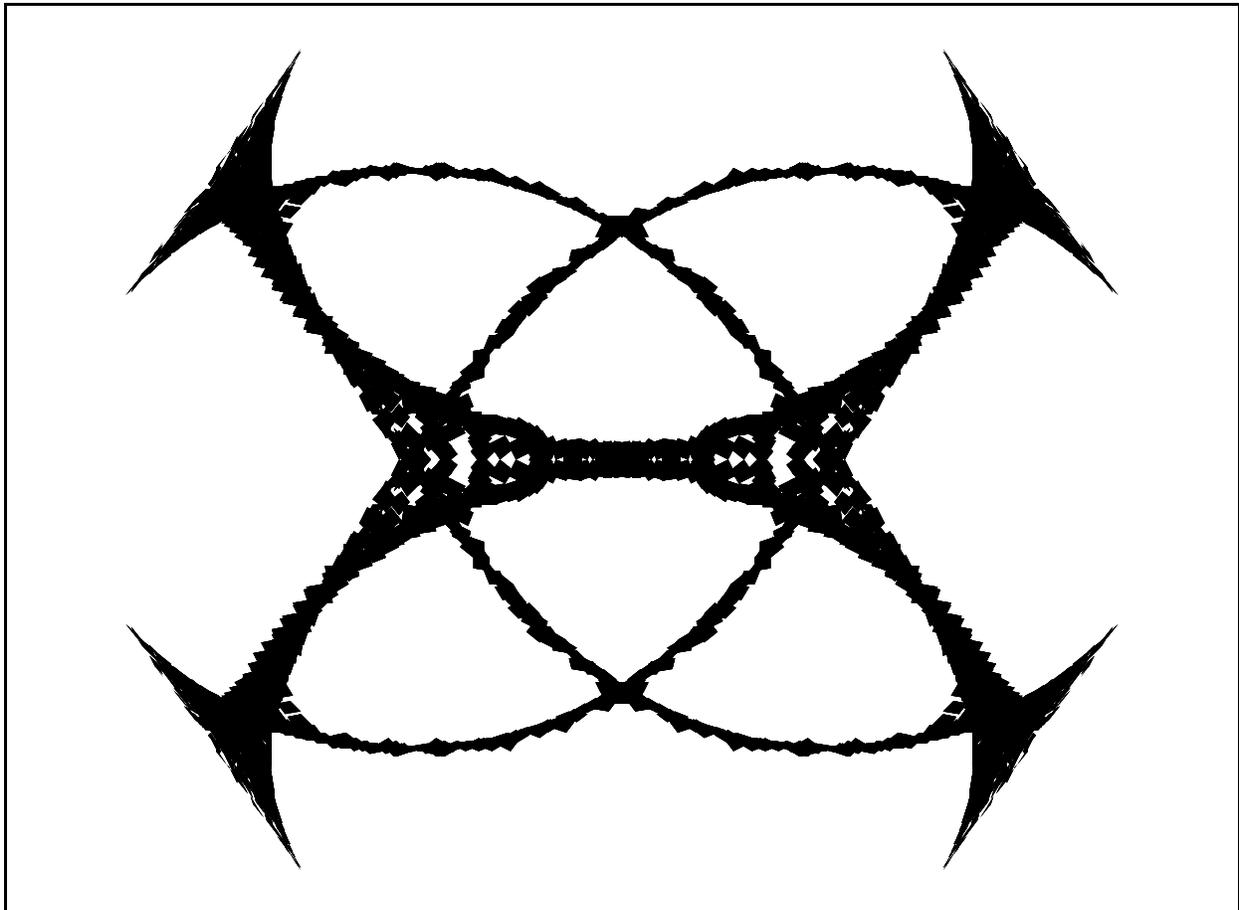
A Search-Based Approach to Steering Single Gimbaled CMGs

by

Joseph A. Paradiso



C.S. Draper Laboratory, Inc.



CSDL-R-2261

CSDL-R-2261

A Search-Based Approach to Steering Single Gimbaled CMGs

By

Joseph A. Paradiso

August, 1991

ABSTRACT

Single gimbaled Control Moment Gyroscopes (SGCMGs) are proposed as torque actuators that can effectively answer the attitude control requirements of future spacecraft. The kinematic properties of SGCMGs are reviewed, existing steering laws are summarized, and the structure of singular surfaces are examined for a 4-CMG pyramid-mounted array. A guided depth-first search that manages null motion about torque-producing trajectories calculated with a singularity-robust (SR) inverse is proposed as a practical feedforward steering law that can globally avoid (or minimize the impact of) singular states in minimally-redundant SGCMG systems. Cost and heuristic functions are defined to guide the search procedure in improving CMG trajectories. On-orbit implementation of the steering law is proposed as an extension to momentum management algorithms. A series of simulation examples are presented, illustrating the search performance for a 4-CMG pyramid-mounted array. Sensitivities of feedforward gimbal trajectories are examined in the presence of unmodelled disturbances, and techniques are proposed for avoiding excessive divergence. A "users manual" appendix describes the interactive Macintosh software package written to define, search over, and investigate SGCMG trajectories.

*The Charles Stark Draper Laboratory, Inc.
Cambridge, MA. 02139*

Table of Contents

- 1) Introduction 1
- 2) Kinematics & Control of Single Gimballed CMG Systems 5
 - 2.1) Background 5
 - 2.2) Formalism 8
 - 2.3) Envelopes, Singular Surfaces, and Closures..... 11
 - 2.4) SGCMG Steering Laws 28
- 3) Search-Based CMG Steering 35
 - 3.1) Analogy to Waypoint Planning 35
 - 3.2) Search Techniques 39
 - 3.3) Coordinates, Costs, and Heuristics 46
 - 3.4) Flight Implementation 55
- 4) Simulation Examples 60
 - 4.1) Simulation Tools 60
 - 4.2) CMG Search Examples..... 68
 - 4.3) Comparison with Local Steering Law 120
 - 4.4) Sensitivity to Unmodelled Disturbances 125
- 5) Conclusions 140

- Appendix: Software Implementation 142
 - A.1) Overview 142
 - A.2) Define H..... 144
 - A.3) CMG Search 149
 - A.4) Disturbance 165

- Acknowledgements*..... 175
- List of References* 177

1) Introduction

The next generation of manned and unmanned spacecraft[1] will require enhanced control algorithms in order to efficiently achieve their proposed mission objectives. Regularly coping with uncertainties in the orbital environment, dynamically changing spacecraft configurations (via docking and buildup), nonlinear actuator properties, and the need to tolerate potential hardware failures will mandate development of control strategies considerably beyond the available state-of-the-art. Because of the priority placed on minimizing cost, weight, and consumable requirements, future spacecraft will not always be able to rely on highly complex, multiply-redundant actuator systems (as is often now the case), but must employ more flexible and intelligent schemes that efficiently exploit all available onboard control capability. In other words, costly actuator "muscle" may be effectively replaced by simpler effectors with more control "intelligence".

Recent years have seen considerable advances in the application of computational solutions to nonlinear control problems. Many of these innovations are just beginning to impact certain engineering disciplines (i.e. robotics[2]), but their application to spacecraft remains largely unexplored. Rapid advances being made in computers and microelectronics indicate that the capability of spacecraft computation resources will continue to increase significantly, enabling onboard implementation of such adaptive and intelligent control algorithms.

The heuristic search[3] is one such technique that has recently matured considerably, spurred by a myriad of applications realized through advances in computation hardware. Search-based approaches have been proposed for real-time application in solving various path-planning problems. One such example is illustrated in Ref. [4], where an A* (A-star) search was applied to optimize an inter-goal "waypoint" trajectory for mission planning onboard a helicopter. Search techniques are currently applied in (or considered for) several spacecraft applications. The proposed Mars Rover[5] may use

search algorithms to navigate semi-autonomously across Martian terrain. Ground-based search procedures have already been used to realize fuel-optimal navigation for the Voyager and Galileo spacecraft[6]. An A* search has been applied to determine obstacle-avoiding trajectories for orbital spacecraft[7] and to dynamically optimize control and actuator usage for proposed hypersonic vehicles operating in an uncertain nonlinear environment[8].

Many promising applications of search-based path-planning algorithms may be found in addressing nonlinear spacecraft attitude control problems. Calculating optimal inverse kinematics of redundant actuators over global attitude trajectories is one such area where a search approach becomes tractable. Attitude commands resulting from a vehicle control scheme must be realized by a collection of onboard effectors. Future spacecraft will often employ a diverse array of actuators, most of which may be subject to various operational constraints and complicated optimization criteria. The demanding mission scenarios proposed for these vehicles suggest that new algorithms that address actuator nonlinearities and constraints (such as an efficient global search) will be needed to optimally specify actuator activity achieving the commanded vehicle response.

This study has examined methods of applying heuristic search techniques to perform adaptable, intelligent inverse kinematics and actuator management for spacecraft attitude control. In particular, Single Gimbal Control Moment Gyros (SGCMGs) were chosen as torque actuators. While many existing algorithms may be adequate for steering Double Gimbal Control Moment Gyro (DGCMG) arrays, such as envisioned for the NASA Space Station, no powerful techniques have been developed to manage minimally redundant arrays of SGCMGs. These devices, however, are ideally suited as momentum-exchange effectors for a wide class of future spacecraft because of their large torque output & momentum capacity. They offer significant cost, power, weight, and reliability advantages over DGCMGs, and could provide attitude control for a variety of future orbital spacecraft (torque requirements in many proposed moderate-sized spacecraft will surpass the capability of available reaction wheels). Due to the complicated nonlinear mapping between the input (gimbal) space and output (momentum) space, however, effective steering laws that reliably avoid problematic singular states have not been developed for minimally-redundant SGCMG systems, discouraging their application in many situations[9]. In order to fully exploit the capability of SGCMGs, intelligent steering laws must be developed that address the system nonlinearities and avoid singular states over global trajectories.

This effort has produced an effective steering algorithm for SGCMGs using search techniques commonly applied in solving trajectory optimization problems. Employing a momentum profile assumed to be forecast by a momentum management procedure (or

maneuver scheduler), the search-based steering law generates a set of feedforward gimbal trajectories that avoid singular configurations (or minimize the effect of any singularities that are unable to be avoided), while maintaining hard constraints on gimbal rates. While the CMGs follow the prescribed trajectories, a "watchdog" regulator task nulls local disturbances and monitors the divergence of gimbal trajectories from their feedforward values. If significant divergence is detected, the CMG planning algorithm is re-executed, starting from the current gimbal/momentum state. Assuming limited disturbance, the CMG search may only need to be performed up to a few times per orbit. For large, unscheduled vehicle maneuvers, however, the CMG search may be required to run immediately before the maneuver is executed. In general, the search algorithm has been seen to run fast enough for eventual on-line implementation; satisfactory gimbal trajectories are often obtained after only a few search trials.

Although feedforward techniques have been used to improve the performance of jet selections[10], they have seldom (if ever) been applied to steer CMGs. By using a global planning algorithm of this sort as a feedforward CMG steering law, one is effectively uniting many aspects of the CMG steering process with a momentum management algorithm. In the current implementation, however, this union isn't complete. Both tasks are still executed separately; i.e. the steering law uses the results of the independently calculated momentum management forecast. Several advantages could be reaped, however, by fully combining the actuator trajectory planner with the momentum management (i.e. solving both tasks together as one large planning problem). The minimally redundant SGCMG array is conjectured[11,12] to exhibit unavoidable singularities; i.e. gimbal paths starting from a given set of initial gimbal angles are unable to attain particular angular momentum states without encountering a singular configuration. The CMG search algorithm presented here will act in these cases to minimize the impact of the singular state; i.e. pass promptly through it while trying to keep the momentum errors to a minimum. If the search process were also able to account for vehicle angular momentum (i.e. as in momentum management), the vehicle attitude/momentum state prediction could be adjusted and scheduled together with the feedforward CMG gimbal state such that singularities could, in general, be avoided.

In addition, the momentum envelope of a symmetric SGCMG array can be skewed and distorted after a device failure, leading to a limited actuation capability along particular directions. Under such conditions, a standard momentum management scheme may be subject to severe control restrictions due to an assumption of spherical momentum capacity. A search-based approach, however, could directly account for the irregular shape of the momentum envelope, leading to superior reconfiguration performance. The search process

will certainly be complicated with the addition of the extra momentum and attitude variables, but with the introduction of clever heuristic functions and constraints, such a problem may be adequately satisfied well within a small fraction of an orbital period.

In order to determine the feasibility of such an approach, the effort discussed in this report examines only search-based CMG steering. The extension of these techniques to momentum management, however, is a promising topic for future research.

The inverse-kinematic "steering" problems posed by CMGs and robotic manipulators are essentially identical, as illustrated in Ref. [13]. Because of this correspondence, many of the abundant algorithms and paradigms developed for robotics can be applied to CMG steering and analysis. While search algorithms have also been applied to solve for nonlinear manipulator trajectories (i.e. Ref. [14]), many other powerful techniques have been directed toward solving robotics problems. Two methods that bear considerable promise for CMG steering are the application of genetic algorithms[15] for determining global CMG gimbal trajectories, and the use of neural networks[16] to locally perform singularity-avoiding inverse kinematics. These techniques are currently being researched, and CMG results should be eventually forthcoming.

The feedforward gimbal trajectories produced by the search process will be realized in the presence of unmodeled disturbances. The susceptibility of gimbal trajectories to unknown secular torques is examined in this report. In certain cases, particularly those with CMGs approaching the vicinity of a singular state, significant divergence from the feedforward path can occur, rapidly increasing after the singular encounter. Although several suggestions are proposed for adequately dealing with this situation, much fruitful analysis remains to be performed on the stability of projected CMG gimbal trajectories.

2) Kinematics & Control of Single Gimbaled CMG Systems

2.1) Background

Control Moment gyros are momentum exchange actuators that consist of a constant-rate flywheel mounted on a gimbal (or set of gimbals). By torquing the gimbal(s), the flywheel orientation is changed, thereby re-directing the rotor's angular momentum. The net momentum stored in a CMG array onboard a spacecraft is equal to the vector sum of all CMG rotor momenta. As individual gimbal angles are varied, the net CMG momentum may be continuously and deliberately adjusted. In accordance with conservation of angular momentum, any change in stored CMG momentum must be transferred to the host spacecraft. The instantaneous torque applied to the spacecraft by the CMG array is likewise a vector sum, with a term for each CMG, weighted by its gimbal rate. CMG kinematics are briefly introduced in this chapter; for a more involved discussion, see Ref. [17]. Gimbal acceleration torques, stiction, and higher order CMG effects are ignored in this analysis; for a detailed dynamic treatment, see Ref. [18].

The two standard CMG varieties are depicted in Figs. 1 and 2. Fig. 1 shows the single gimbaled device. From the hardware viewpoint, this is the simplest type of CMG. The rotor is constrained to rotate on a circle in a plane normal to the gimbal axis $\hat{\sigma}$, hence the orientation of its angular momentum vector is specified through the gimbal angle θ . Single gimbaled CMGs produce "torque amplification", where a relatively small torque needed along the gimbal axis to overcome the gimbal and flywheel inertias produces a much larger torque across the gimbal mount (hence directly into the spacecraft) as the rotor turns. Modern units are equipped with slip rings, enabling the gimbals to turn endlessly without encountering stops that limit angular excursion. SGCMGs generally are able to operate with large gimbal rates (units exist[19] with peak rates in excess of 1 rad/sec) in

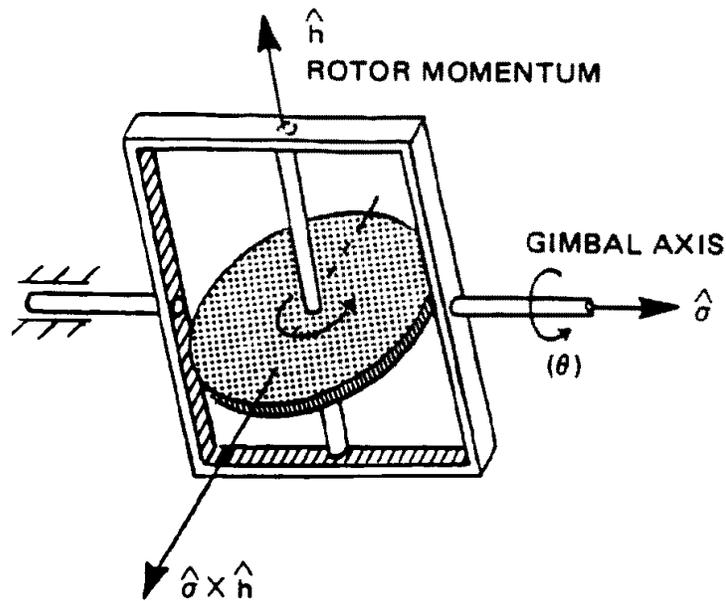


Figure 1: Single Gimballed CMG

order to exploit the amplification principle and couple large torques into a spacecraft. Because of the restricted freedom per device (i.e. torque & momentum constrained to a fixed plane), optimal inverse kinematics (termed "CMG steering") can be extremely complicated. A major application for SGCMGs is in situations where high torque output is needed about specific axes, thus a "scissored pair" configuration[20] can be adopted; SGCMGs aren't generally used to maintain simultaneous 3-axis control without excessive restriction or actuator oversizing.

Double gimballed CMGs (DGCMGs; Fig. 2) are considerably easier to use. The rotor is suspended inside two gimbals (Fig. 2 shows a standard "Euler mount"), hence the rotor momentum can be oriented on a sphere, along any direction (assuming no restrictive gimbal stops); they are not subject to the planar restriction of SGCMGs. The torque amplification advantage is effectively lost, however, because there is no "hard mount" to the spacecraft. Torque produced by driving one gimbal must generally be transferred to the spacecraft by backdriving the other. Double gimballed CMGs are generally significantly heavier, consume more power, are considerably more complicated (thus potentially less reliable), and appreciably more expensive than single gimballed devices. The steering problem is much simpler, however; because of the extra degree of freedom per device, DGCMGs can be easily managed by simple steering laws exploiting specific mounting protocols[21], pseudoinverse approaches[22], or linear programming[23]. Their main

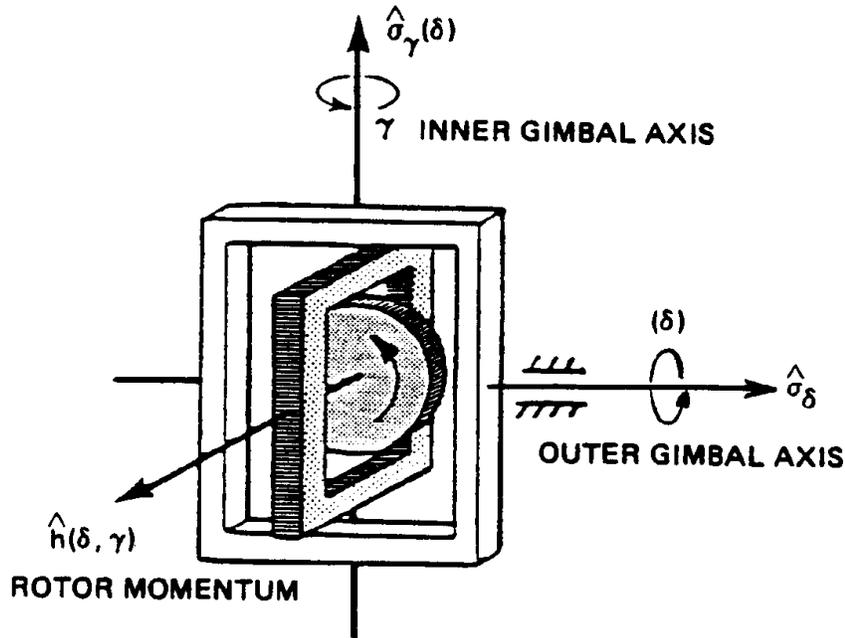


Figure 2: Double Gimbaled CMG

application is in the momentum management of large spacecraft[24,25,26], where a set of flywheels carrying copious momentum are lazily gimballed about, nulling small environmental torques over each orbit and cyclically exchanging significant amounts of angular momentum. DGCMGs were used to control Skylab attitude[27] and are proposed for use onboard the planned NASA space station, where devices have been baselined[28] at roughly 4000 ft-lb-sec rotor momentum and 5 °/sec peak gimbal rate.

Because of their myriad hardware advantages, consideration of SGCMGs has begun to erode traditional DGCMG applications; i.e. attitude control of large spacecraft. Examples are the Soviet MIR space station, stabilized by a 6 SGCMG array[29] onboard the KVANT astrophysics module. SGCMGs were also seriously considered for the NASA space station[30], but eventually lost to DGCMGs[9]. Because current steering laws experience difficulty avoiding singularities, these SGCMG systems must be grossly oversized to place all unescapable singular states outside of the required momentum reservoir, leading to considerably increased expense.

Reaction wheels are conventionally used to control the attitude of 3-axis stabilized satellites and smaller spacecraft. These devices vary the spin of a fixed-axis flywheel to transfer momentum and directly couple torque to a host vehicle. Although they are the simplest, lightest, least expensive, etc. of all momentum-exchange effectors, they achieve

lower bandwidth and much smaller torque capability than CMGs[31] (no amplification principle is at work; torques are coupled directly into the spacecraft). As larger spacecraft and satellites are constructed, SGCMGs may prove to be a necessary and cost-effective upgrade from reaction wheel systems. Before SGCMG arrays can compete efficiently in any of these arenas, however, a general steering algorithm must be developed that can successfully manage a minimally-redundant SGCMG array.

2.2) Formalism

Fig. 3 shows a drawing and schematic of a basic SGCMG. Define the total CMG momentum state to be the sum of all rotor momenta in the CMG array:

$$1) \quad \underline{\mathbf{h}}^T = \sum_{i=1}^N \underline{\mathbf{h}}_i \quad \begin{array}{l} N = \# \text{ of CMGs in array} \\ \underline{\mathbf{h}}_i = \text{Angular momentum of rotor } \#i \end{array}$$

The rotor momenta may be parameterized by a set of rotations:

$$2) \quad \underline{\mathbf{h}}_i = \underline{\mathbf{h}}_i^0 \cos(\theta_i) + (\hat{\sigma}_i \times \underline{\mathbf{h}}_i^0) \sin(\theta_i)$$

Where:

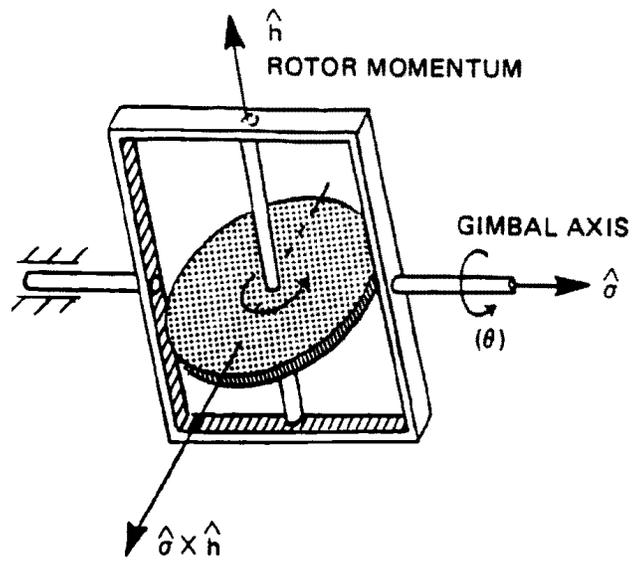
$\underline{\mathbf{h}}_i^0$ = Reference rotor position (at zero angle) for CMG #i

$\hat{\sigma}_i$ = Unit vector along gimbal axis for CMG #i

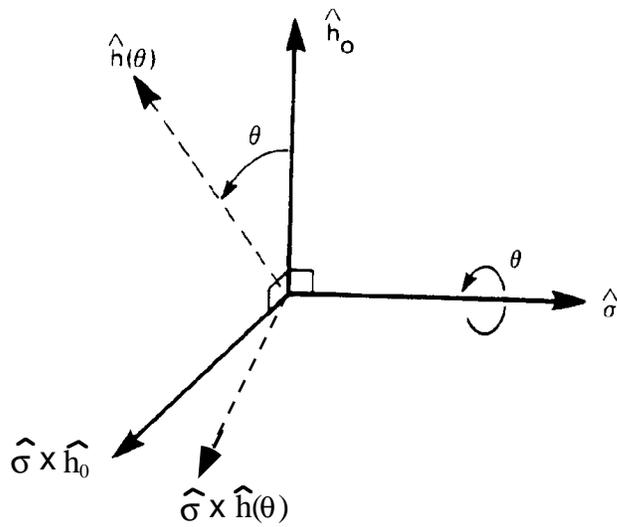
θ_i = Gimbal angle of CMG #i

When mounted on a spacecraft, the total system momentum remains constant in the absence of disturbance torques:

$$3) \quad \begin{aligned} \underline{\mathbf{H}}_s + \underline{\mathbf{h}}^T &= \text{constant} \\ \frac{d(\underline{\mathbf{H}}_s + \underline{\mathbf{h}}^T)}{dt} &= \underline{\mathbf{0}} \\ \left[\frac{d\underline{\mathbf{H}}_s}{dt} + \underline{\omega}_s \times \underline{\mathbf{H}}_s \right] + \left[\underline{\omega}_s \times \underline{\mathbf{h}}^T + \sum_{i=1}^N (\hat{\sigma}_i \times \underline{\mathbf{h}}_i) \dot{\theta}_i \right] &= \underline{\mathbf{0}} \end{aligned}$$



Drawing



Schematic

Figure 3: SGCMG Drawing & Schematic

$$4) \quad [\mathbf{I}] \dot{\underline{\omega}}_s = -\underline{\omega}_s \times \underline{\mathbf{H}}_s - \left[\underline{\omega}_s \times \underline{\mathbf{h}}^T + \sum_{i=1}^N (\hat{\sigma}_i \times \underline{\mathbf{h}}_i) \dot{\theta}_i \right]$$

Where:

$\underline{\mathbf{H}}_s$ = Spacecraft angular momentum

$\underline{\omega}_s$ = Spacecraft angular rate

$[\mathbf{I}]$ = Spacecraft inertia matrix

$\dot{\theta}_i$ = Gimbal rate for CMG #i

Eq. 3 states the system momentum conservation and Eq. 4 results from the consequent torque balance. The bracketed term at right in Eq. 4 describes the CMG reaction on the spacecraft. The first term in the bracket is caused by Euler coupling of the CMG rotors into the spacecraft rate; it arises from the spacecraft-fixed coordinate frame in which Eq. 4 is stated. This term is not commonly used in control laws (although may be exploited in nonlinear momentum management schemes[24]), and is subtracted from the CMG torque commands such that all gimbals are driven inertially. Torque and momentum profiles used in this report assume that commanded torques have been compensated via the equation below, effectively removing the spacecraft rate and zeroing the CMG coupling term.

$$5) \quad \underline{\tau}_{\text{cmd}} = \underline{\tau}_{\text{cmd}}^0 + \underline{\omega}_s \times \underline{\mathbf{h}}^T$$

The CMG output "control" torque is given by the second term in the bracket of Eq. 4. The torque amplification effect can be seen in the product of rotor momentum ($\underline{\mathbf{h}}_i$) and gimbal rate ($\dot{\theta}_i$); i.e. a small gimbal rate produces a sizable torque through the "leverage" of a large rotor momentum. The gimbal equations[32] don't involve the angular momentum $|\underline{\mathbf{h}}_i|$; gimbal torquers need only overcome the gimbal and flywheel inertias to produce a gimbal rate, hence output torque.

CMG control variables are the gimbal rates $\dot{\theta}_i$. One may state the CMG control constraint in terms of a commanded torque:

$$6) \quad \underline{\tau}_{\text{cmd}} = [\mathbf{J}(\underline{\theta})] \dot{\underline{\theta}}$$

The matrix $\mathbf{J}(\boldsymbol{\theta})$ is the Jacobian of the total CMG momentum \mathbf{h}^T with respect to gimbal angles. Its columns are the individual CMG output torque authorities, $\boldsymbol{\tau}_i = \hat{\boldsymbol{\sigma}}_i \times \mathbf{h}_i$, as in the summation of Eq. 4, where \mathbf{h}_i is a function of $\boldsymbol{\theta}_i$ given in Eq. 2. The CMG steering problem is effectively an inversion of Eq. 6; i.e. specify a set of gimbal rates that deliver $\boldsymbol{\tau}_{\text{cmd}}$ while meeting constraints (i.e. hardware limits on maximum gimbal rates) and managing CMG system redundancy in an "optimal" fashion, such that singular gimbal states and undesirable rotor & gimbal orientations are avoided.

2.3) Envelopes, Singular Surfaces, and Closures

For most relevant applications, the CMG array will be required to maintain full 3-axis attitude control. Situations exist, however, where the CMG rotors are configured such that the Jacobian in Eq. 6 loses rank, and control authority can not be projected along a particular axis. Such gimbal configurations, termed singular states, are clearly undesirable.

The simplest singular state is momentum saturation, defined as the "momentum envelope". This is a 3-dimensional surface that represents the maximum angular momentum attainable by the CMG array along any given direction. It is analogous to the "workspace" boundary of the robotic manipulator (i.e. the surface of maximum reach).

The mechanical analogy between manipulators and CMGs was illustrated in Ref. [13]. The output of both systems is a sum of rotated vectors *à la* Eq. 1. By respectively replacing torque, momentum, and gimbal angles by end effector velocity, position, and joint angles, one can mathematically interchange CMGs with robotic manipulators. Figure 4 shows a 3-link (thus minimally redundant) planar manipulator system. One can see that the vector sum of links (each rotated by joint angles θ_i) determines the end effector position.

An SGCMG system, however, is a collection of planar actuators used to control a 3-axis space. The exact mechanical analog to an SGCMG array is given in Fig. 5. Although each manipulator joint is only able to move in a plane, the combined action of all joints enables the end effector to span a 3-dimensional workspace. Since it's more difficult to interpret these 3-dimensional drawings, the simple planar manipulator will be used to illustrate the concepts and examples introduced in this section.

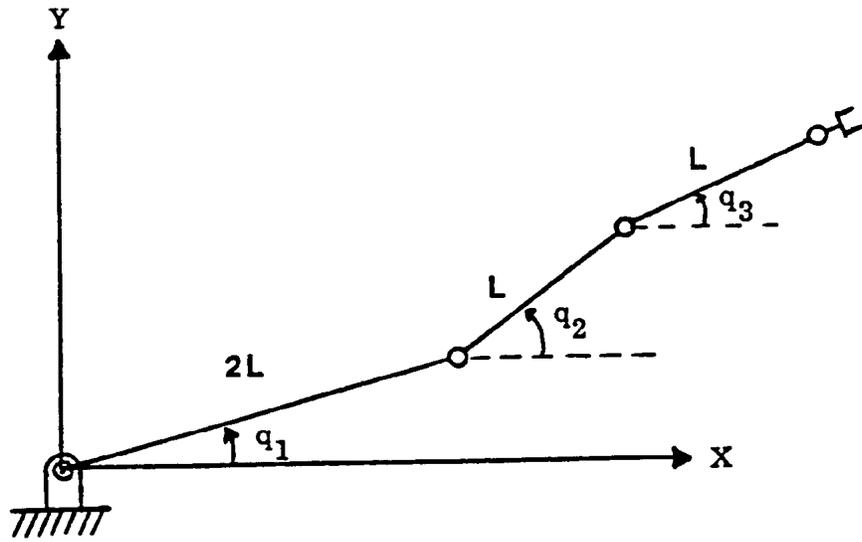


Figure 4: A 3-Link Planar Manipulator

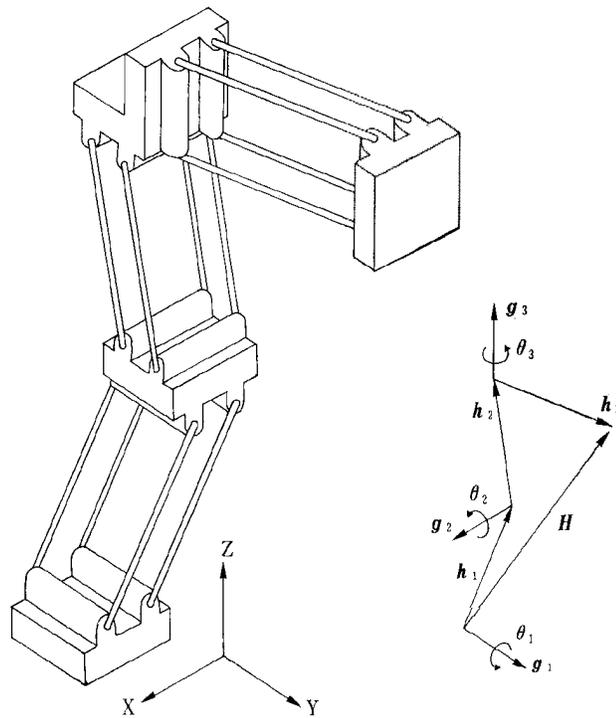


Figure 5: True 3-Axis Mechanical Analog to a SGCMG System

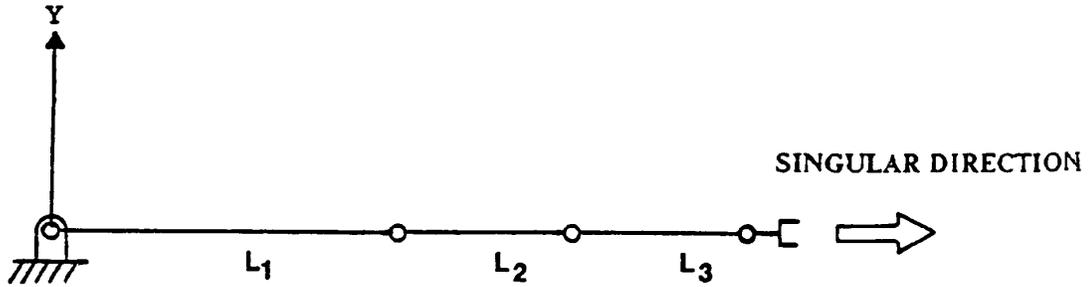


Figure 6: Planar 3-Link Manipulator in Saturation

Figure 6 shows the 3-link planar manipulator in saturation along the horizontal axis. The manipulator is seen to be "maximally stretched"; i.e. each link projects maximally onto the singular direction. The system's Jacobian loses rank in such a situation, and its span is orthogonal to the singular axis.

The saturation state is impossible to avoid through managing CMG (or manipulator) redundancies. It must be directly addressed by the momentum management scheme (or task scheduler in the manipulator case).

Other singular configurations exist, however, with momenta (or end effector position) inside the envelope (or workspace). These are characterized by a combination of maximal and minimal link projections onto a given singular direction. A manipulator example is given in Fig. 7, which shows both singular (dotted) and nonsingular (solid) configurations for an end effector positioned within the envelope. Singular states can be coded with a binary character string representing the projection of each link onto the singular direction. A point on the envelope (i.e. Fig. 6) is labeled [+++], while the internal singularity of Fig. 7 is notated [++-]. Pursuing this convention, there are 2^N possible singular states for an N-link system (manipulator or CMG) along any given singular direction[33]. A CMG steering law (or inverse kinematic procedure) manages system redundancy to avoid internal singularities while answering torque commands.

By taking the determinant of the square matrix formed when multiplying the Jacobian by its transpose, one obtains a quantity (referred to as the singularity index[34], CMG gain[35], and miscellaneous other monikers) that reflects the Jacobian's rank. The square root is conventional in most formulations.

7)
$$m = \sqrt{|\mathbf{J}\mathbf{J}^T|}$$

NONSINGULAR CONFIGURATION

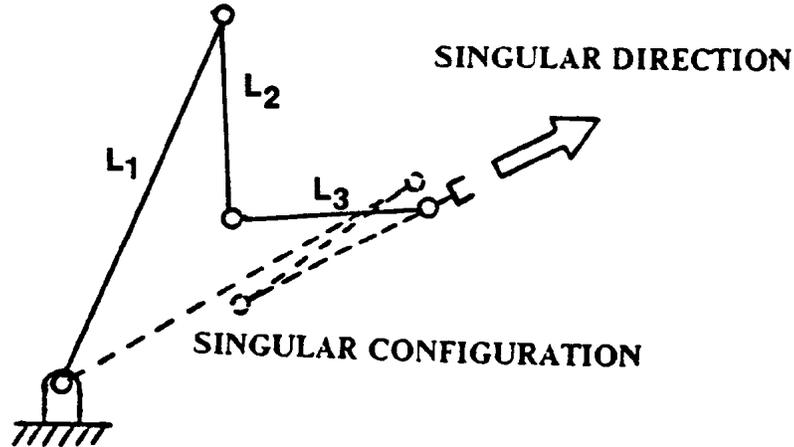


Figure 7: Singular & Non-Singular Manipulator Configurations

In the CMG case, $[\mathbf{J}\mathbf{J}^T]$ is a 3×3 matrix. As m approaches zero, the system is nearly singular, and an axis of control is effectively lost. This "lost axis" is along the eigenvector associated with the minimum eigenvalue of $[\mathbf{J}\mathbf{J}^T]$. The CMG gain " m " may also be expressed as the product of the singular values of \mathbf{J} (Ref. [13]). Blocks termed "minors" can be formed for redundant manipulators or CMG systems by grouping all distinct combinations of Jacobian column vectors into square matrices. These are 3×3 matrices for the CMG case (2×2 for planar manipulators). For an N -gimbal CMG system, there are $\binom{N}{3}$ possible minors [$\binom{N}{3} \Rightarrow$ binomial coefficient]. The signs and magnitudes of the minor determinants reflect the state of the relative "kinks" between the component links. This is illustrated in Fig. 8, which shows the same end effector position of a planar manipulator as realized by 3 different joint solutions. Each of these is characterized by a different sign pattern of minor determinants. Link motion that maintains such a given sign pattern is deemed a closure[13] or solution family[11]. Transitions between closures are accomplished through a singular minor (with zero determinant), as illustrated in Figure 8, where the manipulator is shifted from state A to state B (without moving the end effector; this is termed "null motion") through a singular ($L_2 L_3$) minor in state C. This has direct consequence in CMG systems, where shifts between different closures must sometimes occur through a singular state (i.e. *all* minors in the CMG system become singular; the CMG gain m may be expressed as the quadrature sum of the minor determinants). A detailed discussion on the role of minors in CMG systems is presented in Ref. [13].

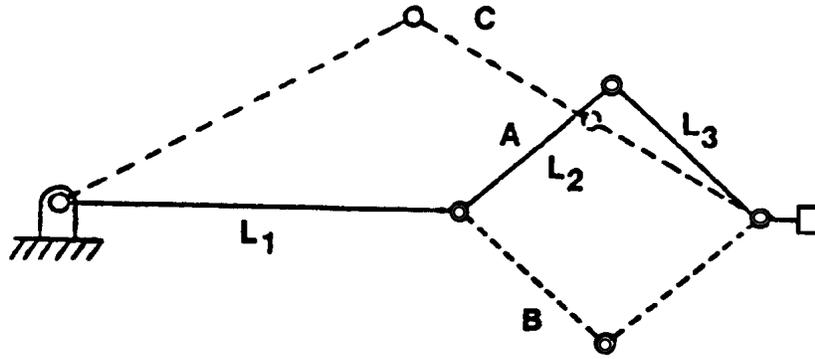


Figure 8: Transition Between 2 Closures Via Singular Minor

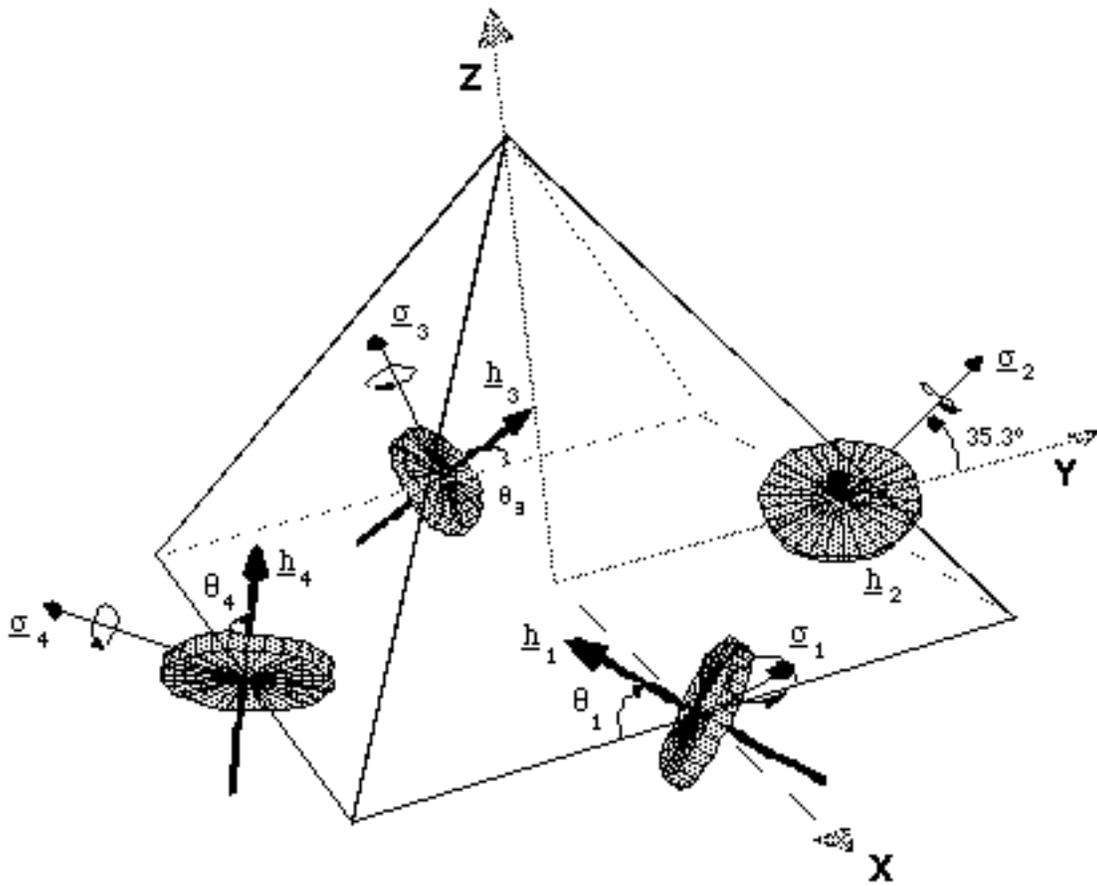


Figure 9: 4-CMG Pyramid Mounting Scheme

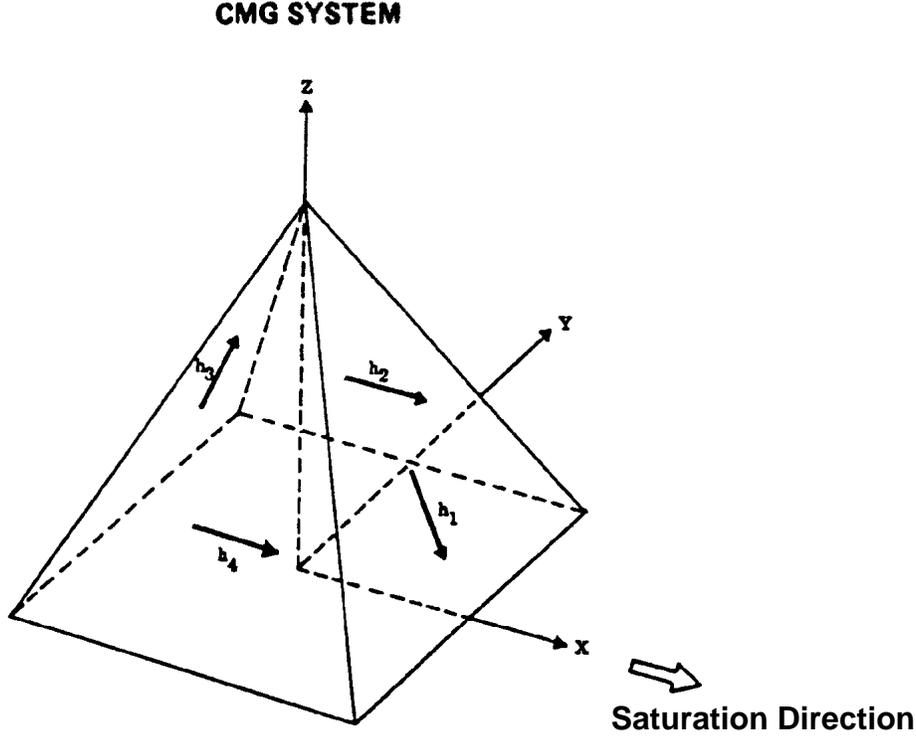


Figure 10: 4-CMG Pyramid in Momentum Saturation

The CMG mounting scheme used in this study is depicted in Fig. 9. It is a conventional "pyramid mount", where the CMGs are constrained to gimbal on the faces of a regular pyramid (the gimbal axes $\hat{\sigma}$ are orthogonal to the pyramid faces). Each face is inclined at 54.7° to the horizontal (thus gimbal axes are at the complement angle; i.e. 35.3°), yielding a momentum envelope that is roughly spherical for a 4-CMG array; i.e. the authority along the vertical (z) axis is similar to the authority that can be projected along x and y. Other mounting arrangements have been touted in the literature; i.e. rooftop mounts[36,37] or symmetric polyhedral mounts[29,36]. The benefits of these mountings (removal of all "inescapable" singular states inside the momentum volume) tend to be relevant for CMG arrays having more than minimal redundancy (more than 4 CMGs). Since the purpose of this study is to investigate a new steering law in the presence of all types of singularities, the standard 4-CMG pyramid is retained.

An example of a CMG arrangement that yields a point on the momentum envelope is given in Fig. 10 for the 4-CMG pyramid mounted system. One can see that the system is "maximally stretched"; i.e. all rotors project maximally onto the singular direction. A

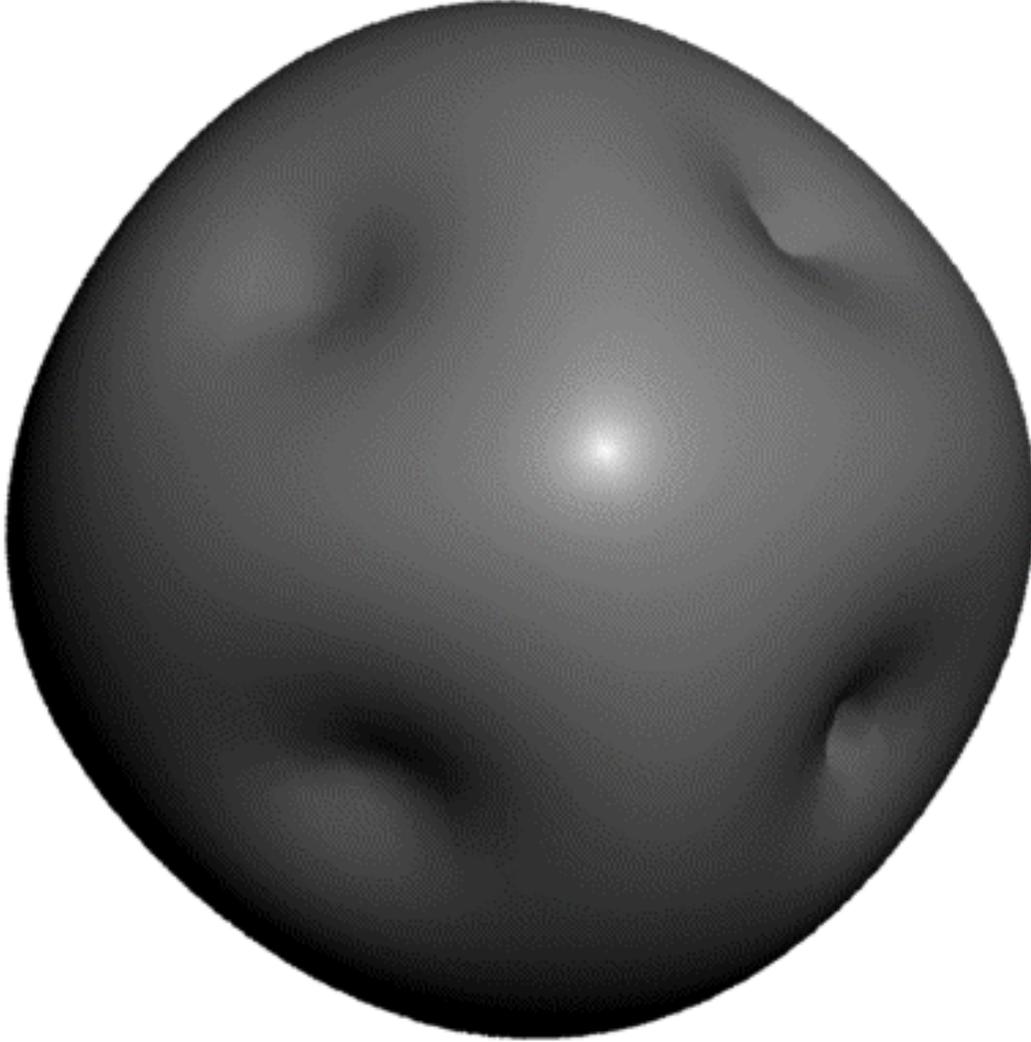


Figure 11: Momentum Envelope for 4-CMG Pyramid System

"cutting plane" technique[33,38] can be applied to calculate a set of points scattered about the momentum envelope or internal singular surfaces. With this method, the CMG orientation that is singular along the direction specified by a given vector \hat{v} is obtained by aligning the CMGs such that their output torques lie in the plane normal to \hat{v} (since the CMG output torques are always normal to their rotor vector \underline{h} and gimbal axis $\hat{\sigma}$, this is a straightforward projection). For a point on the envelope, CMG orientations are selected with rotors all projecting positively onto \hat{v} . The vector \hat{v} specifies the singular direction, and the sum of rotor vectors (\underline{h}^T , Eq. 1) specifies the momentum value.

Figure 11 displays the momentum envelope created by the 4-CMG pyramid of Fig. 9. Points were calculated using the above technique on a 4π spherical polar net (a subdivided polygonal net circa Ref. [39] would be a superior choice for even coverage of

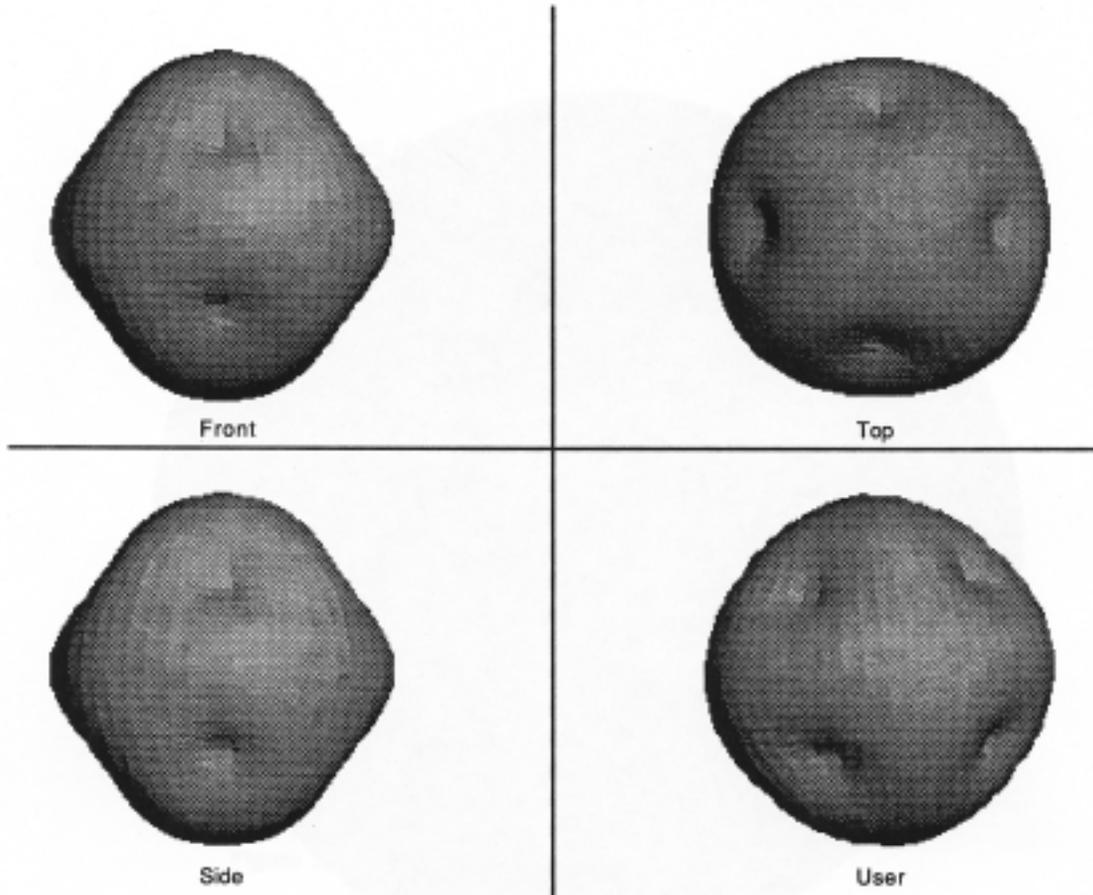


Figure 12: 4-View of Momentum Envelope for 4-CMG Pyramid

solid angle, but the author came across this technique after the graphics were already generated). Triangular polygons were formed by connecting the momenta calculated at adjacent points, and displayed using the Super 3D graphics program[40] and RenderMan[41], running on a Macintosh II computer. In comparison to the schematic of Fig. 9, the z-axis of the envelope emerges from the polar singularity visible at the "top" of the solid (since the object isn't too far removed from a sphere, this is readily discernable). The x and y axes are in the "equatorial" plane. The centers of the "dimples" puncturing the faces are aligned with positive and negative gimbal axes ($\pm \hat{\sigma}_i$) labeled in Fig. 9. For aid in visualization, a 4-view of this object is given in Fig. 12. Here, the "Top" view is looking down from the z-axis, while "Front" and "Side" views are looking along x and y. "User" is from the viewpoint of Fig. 11.

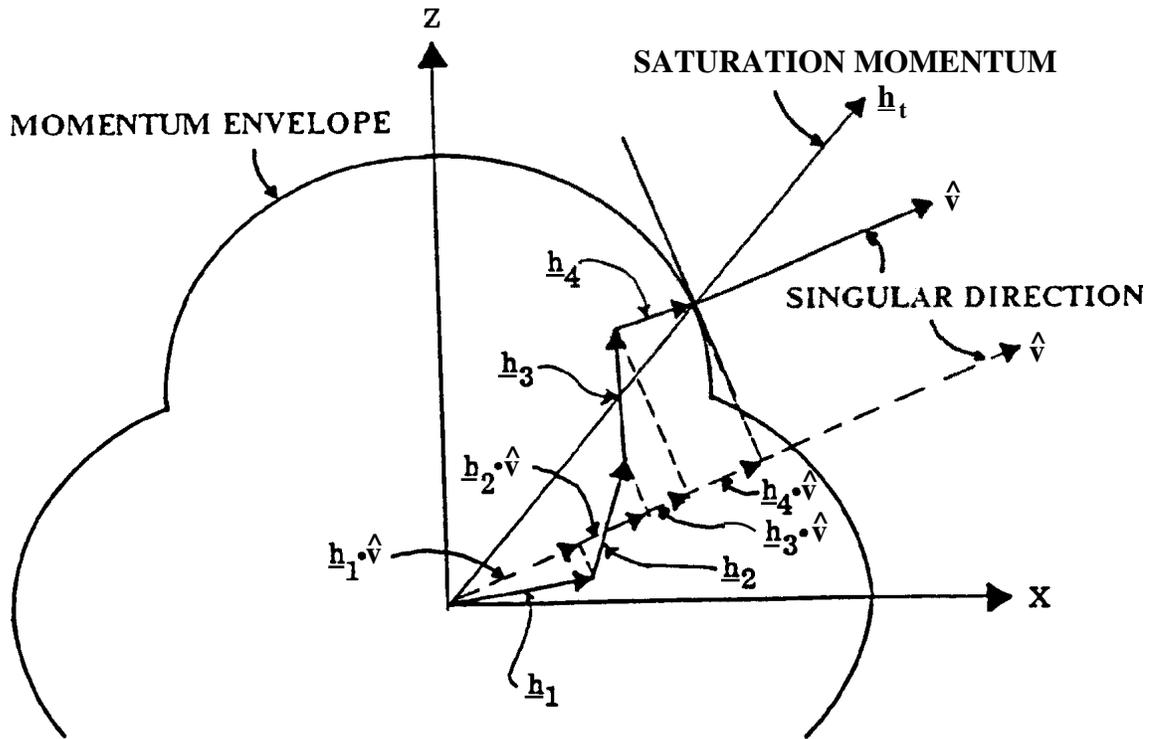


Figure 13: Singular Direction vs. Momentum Projection on XZ Plane

When calculating a momentum envelope or singular surface via the cutting-plane technique, the singular direction (which is normal to the surface) can be misaligned with the net momentum vector (which is the sum of all CMG rotor momenta). This is illustrated for the momentum envelope in the projection of Fig. 13, and is a property of SGCMG systems. The effect here is especially pronounced near the "dimples", where the momentum envelope departs considerably from a sphere.

These dimples seen on the solid depicted in Figs. 11 & 12 (and sketched in the projection of Fig. 13) are not part of the envelope as calculated via the cutting plane technique. If one pursues this method without modification, the dimples don't appear, leaving circular holes in their place. This is because they actually belong to the singular state having one rotor flipped with respect to the singular direction \hat{v} (termed a 2H state by the nomenclature of Refs. [34,38]; 4H states are on the envelope proper, with all rotors aligned). When \hat{v} is selected to point toward a "dimple", the rotor corresponding to the gimbal axis piercing the dimple can no longer contribute to the momentum along \hat{v} (since \hat{v} is now closely aligned with the dimple's $\hat{\sigma}$, the corresponding rotor is nearly orthogonal to \hat{v} , and can't project). In order to keep the net momentum along \hat{v} , the component of this rotor momentum normal to \hat{v} must be nulled by anti-aligning the rotor

from the opposite pyramid face. This creates a 2H singularity and reduces the momentum capacity of the CMG array near the gimbal axes, producing the dimples. In order to generate the solid plotted in Figs. 11 & 12 (dimples and all), initial 4H gimbal solutions obtained from the cutting plane technique (which aligned the singular direction with \hat{v}) were iterated along the envelope and dimple surfaces with an SR inverse (Eq. 13), until the total CMG momentum \underline{h}^T was made to align with the commanded vector (the original \hat{v}) from the polar net.

The momentum envelope for an array of double gimballed CMGs is perfectly spherical (in the absence of constraining gimbal stops). This remains true regardless of mounting arrangement and device failure status (provided at least two units are operational to span the 3-axis control requirement). The singular surfaces of DGMCG arrays are also spherical in momentum space. They can easily be avoided in most cases by steering away from rotor alignment expressed via a simple dot product objective[23], or preferring CMGs to be distributed at equal angles about the total stored momentum[21,42].

The single gimballed CMG situation is considerably different. Figs. 11 & 12 illustrated that the momentum envelope is somewhat non-spherical, as noted in the roughly octagonal shape and concave dimples intersecting the gimbal axes. This surface appears more complicated than the spheres characterizing DGCMGs and the polyhedra created in momentum space by reaction wheel systems[43]. Any mission requiring system operation near the envelope must account for its nonspherical shape in the resident momentum management scheme, otherwise peak momentum must be limited to the minimum radius. As CMGs are added, the envelope can be made to better approximate a sphere. In these cases, however, a device failure will break the mounting symmetry, causing the envelope to skew over and project unequally along control axes (see Ref. [12]). This is illustrated in Fig. 14, which shows a side view of a momentum envelope created by an irregular pyramid-mounted SGCMG system (the arrangement used is the same as shown in Fig. 9, except CMGs #3 & #4 are rotated about the z axis by an additional 45°). An envelope of this type might be expected after device failures on a multiply-redundant CMG system. The envelope now resembles a skewed ellipsoid; a minimum-radius spherical requirement would severely limit the stored momentum capacity. In order to fully exploit all control potential within an envelope as depicted in Fig. 14, the momentum management procedure must be able to employ non-uniform constraints on peak momentum loading along different axes. A search-based momentum manager, which might be obtained by extending techniques demonstrated in this report, or an approach able to infinity-norm bound the momentum on each axis (i.e. an ℓ^1 optimization, as in Ref. [44]), could provide the potential for realizing such a scheme.

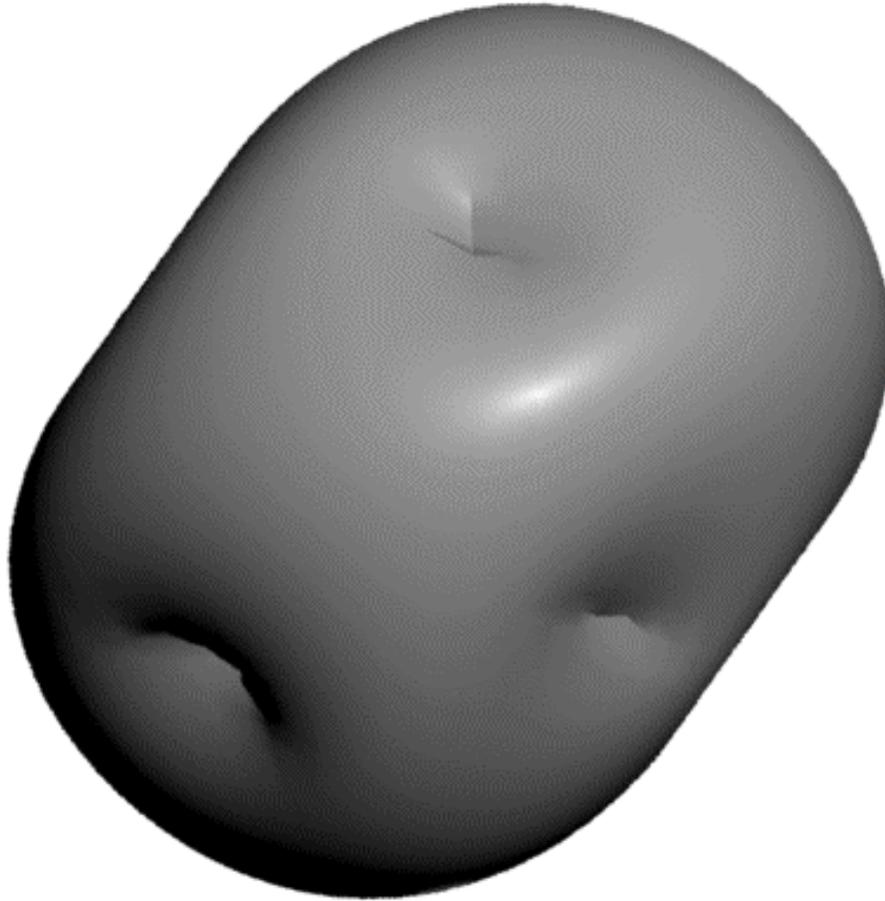


Figure 14: Momentum Envelope for Irregular 4-CMG Pyramid

Singular states are considerably more complicated for single gimballed CMGs. Figure 15 shows the 2H singular surfaces obtained in momentum space by anti-aligning one rotor with the singular direction (\hat{v}) for the 4-CMG pyramid of Fig. 9. As one can readily see, these surfaces are hardly spherical. Fig. 15 is actually composed of a union of four surfaces, each obtained by flipping one rotor to oppose \hat{v} , yielding the projection patterns [-+++], [+--+], [++-+], [+++ -]. Each such surface is aligned with its "flipped" gimbal axis, and connects the petals projecting on either side. These "petals" are actually the "dimples" seen on the envelope plots of Figs. 11 & 12 (Fig. 11 is rendered from the same viewpoint as Fig. 15, thus features may be compared directly). The intersection with a plane across each petal (normal to the petal's gimbal axis) becomes triangular as the base of the petal is approached. This phenomenon was noted in Ref. [33] and can be seen in Fig. 15. The 2H singular states patch smoothly to the 4H envelope at the perimeter of the petals, then "twist" at the petal base into 3 "ribbons" (formed from the sides of the aforementioned

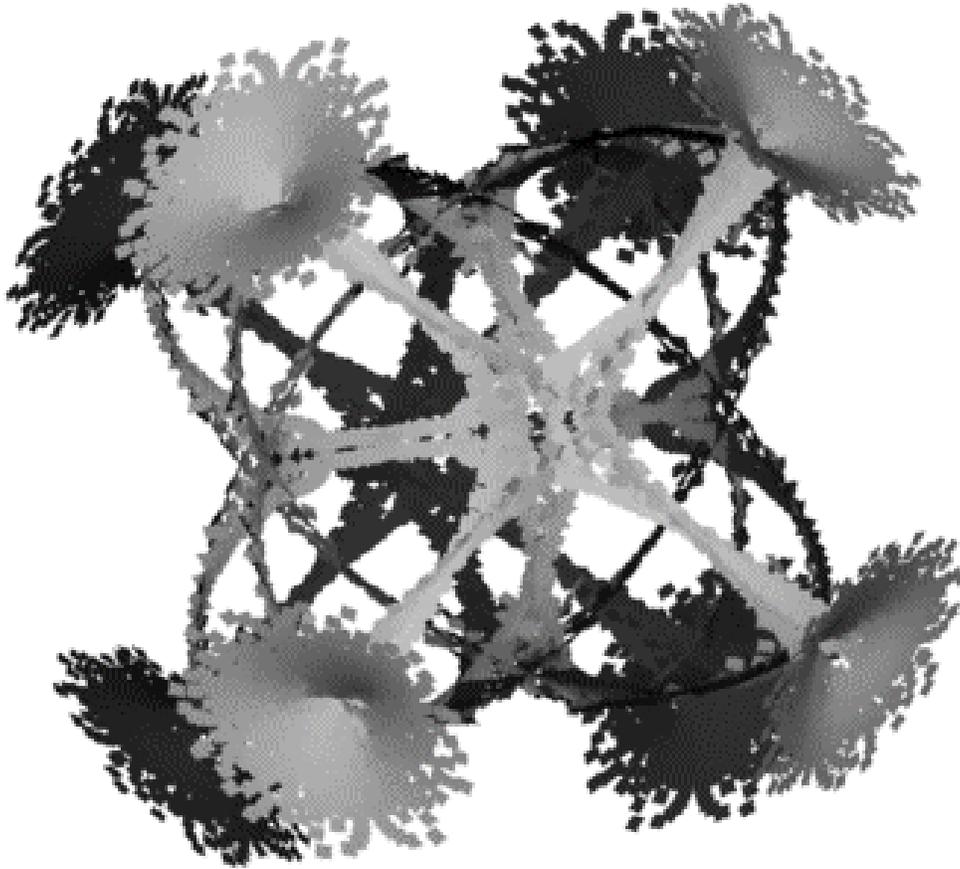


Figure 15: 2H Singular Surfaces (with one rotor flipped)

triangle) that arc around within the momentum envelope (becoming an internal singular state) to join the opposite petal. The four 2H singular surfaces are all interconnected near the pyramid base.

Because the mapping from the continuous polar net (defining \hat{v} in the domain space) to the momentum space of the singular surfaces is highly nonlinear, adjacent θ and ϕ points could not be connected together to form a "sheet" of polygons, as was possible for the envelope (adjacent points in θ, ϕ would often map into highly separated momenta). In order to generate these images of singular states, a small square is drawn at the midpoint of each momentum value calculated. The squares are oriented normal to the singular direction \hat{v} , and are not connected together. If the density of squares is sufficient, they begin to overlap and tile the singular momentum surface. The mapping between \hat{v} and the singular momenta is so nonlinear, however, that a uniform polar net in \hat{v} yields rather "spotty" and "granular" coverage in momentum space. This creates definite artifacts in these images. For example, the rough boundary of the petals in Fig. 15 is due to a lack of covering polygons; the \hat{v} -to- \underline{h} mapping is highly curved in these regions, thus a densely packed

bundle of input vectors maps to a dispersed collection of momenta \underline{h} . The petals would have a smooth boundary in an ideally rendered solid. Despite these drawbacks, if one views these figures at "arms length", a very good feeling can be gleaned as to the structure of these surfaces.

The "OH" singular state is formed by simultaneously flipping two rotors to oppose the singular direction (\hat{v}). Its momentum surface is thus smaller, and it fits within the 2H state pictured in Fig. 15. No parts touch the momentum envelope; this singular condition is entirely internal. There are two types of OH singularities for the 4-pyramid of Fig. 9; i.e. one created by flipping adjacent pairs of rotors (this includes two distinct surfaces: [+++] and [+--+]), and another made from flipping opposing pairs of rotors (only one surface, defined by [++-]). The OH "adjacent pairs" surfaces are shown in Fig. 16. Although the small squares produced as artifacts from the tiling algorithm discussed above are clearly visible, the actual momentum surfaces are smoothly connected. One must mentally "interpolate" and imagine a continuous tiling.

The nature of this object can be initially difficult to discern, due to its complex structure. After some examination, it can be seen to form a multi-ridged solid, with adjacent ridges intersecting at six apex points spaced regularly about its periphery. Making an organic analogy, its appearance is not unlike a ridged Brazil nut or star fruit. In order to improve the rendering, the orientation of this figure is rotated somewhat from that of Fig. 15. In this view, one is looking "down" at the xy plane from a vantage roughly 30° from the z-axis. The z-axis protrudes from the apex point that projects out of the page near the top of the object. The x and y axes are in the plane containing the other four apex points. All of the OH apex points patch onto the 2H state at the areas where the different 2H surfaces intersect; by mentally aligning the apex points in Fig. 16 with the intersections between the "ribbons" of Fig. 15 (4 at the vertices of a square in the xy plane, 2 along the $\pm z$ axis), one can see how the 2H and OH singularities interrelate.

The OH "opposite pair" surface is shown in Fig. 17. The curving lines visible at the fringes of the structure are tiling artifacts due to the nonlinear mapping discussed above; ideally, these planes should be solid and continuous. This figure is somewhat difficult to discern, again because of its multiplicity of protruding ridges and facets. It seems to consist of a polyhedron with ridges extending along its perimeters and intersecting at apexes, as was seen in Fig. 16. This surface is "flatter", however, and stays closer to the xy plane than the adjacent rotor states shown in Fig. 16. Figs. 16 & 17 were created from roughly similar viewing angles (Fig. 17 is seen from a vantage closer to the xy plane); by comparing the two, one can see that the apex points are identical, and the ridges seem to overlap.

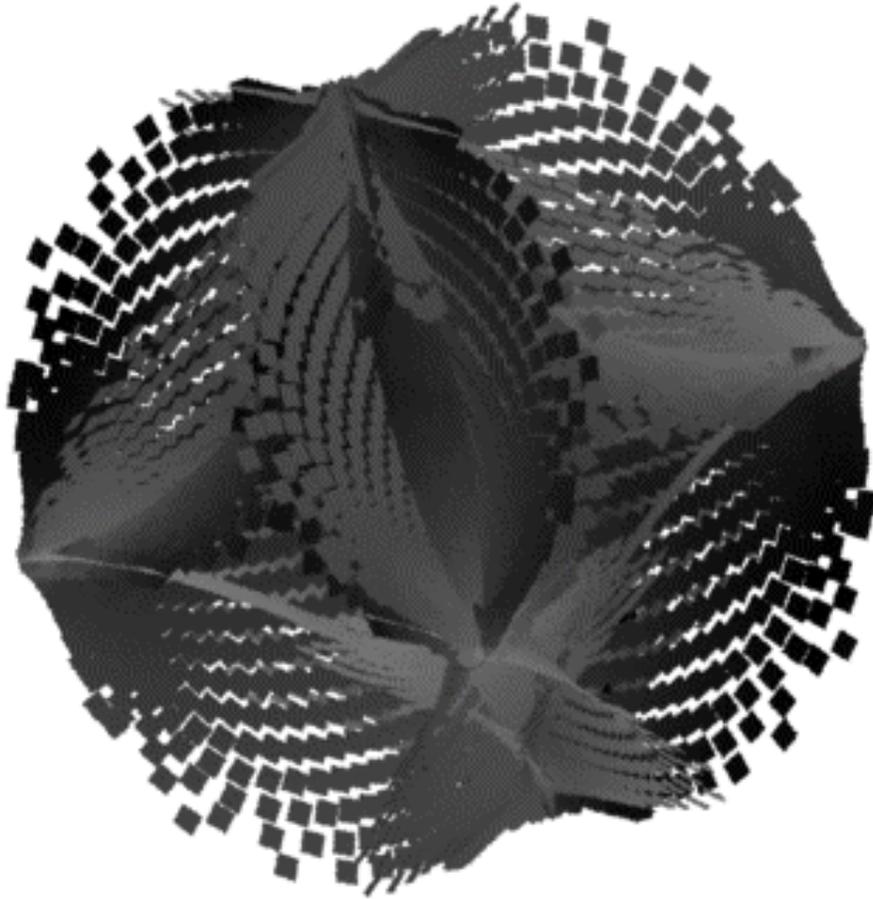


Figure 16: 0H Singular Surfaces with Adjacent Rotor Pairs Flipped

In order to summarize the interrelation of all singular states, Fig. 18 gives a cutaway view of the momentum envelope, with all of the singular momentum surfaces visible inside. The envelope is bisected by the xz plane (all points with $x > 0$ are plotted), and the viewpoint is adjusted such that one is looking at a small angle to the x -axis along the xy plane. The 2H singular surface can be seen to patch to the dimples on the envelope, then arc inside. The 0H surfaces can be seen to fill the void inside the 2H surfaces, and the ridges protruding from the 0H edges are seen to fill in the space between the 2H "ribbons" as they arc toward the dimples on the envelope. Immediately beneath these dimples, the gap between the 2H "ribbons" seems to be filled in by the 2H state itself; further in toward the center, the 0H state covers this area, as suggested above. This hints that the 2H and 0H states may patch together along these "ridges" (or "webs") between the 2H arcs; a gap can

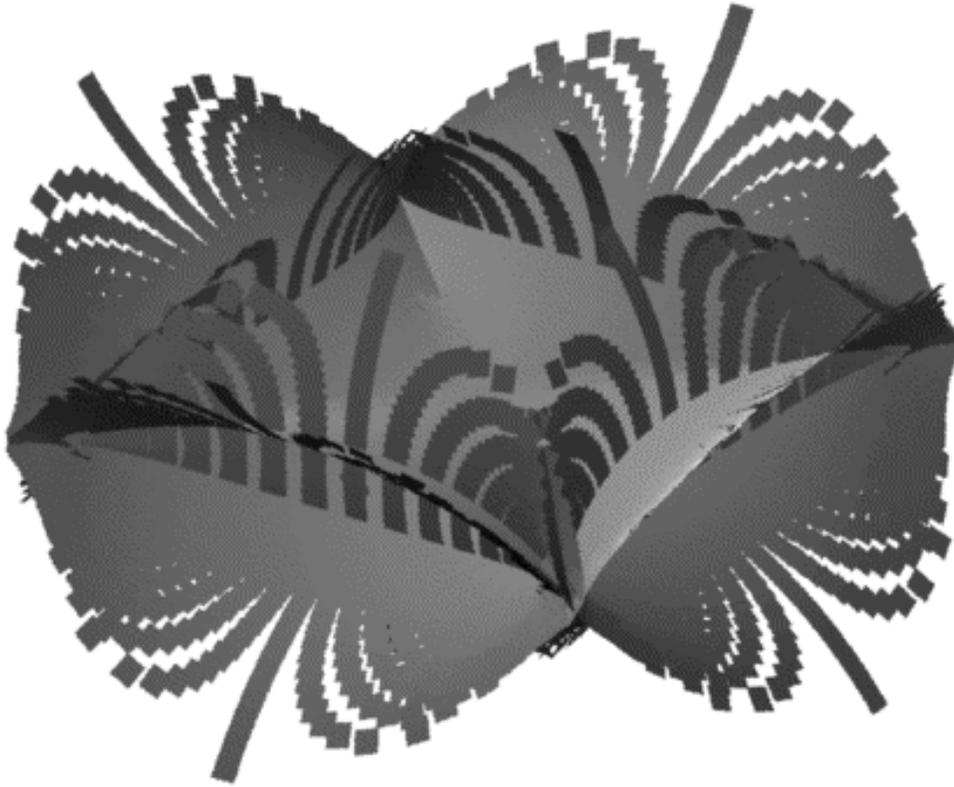


Figure 17: 0H Singular Surfaces with Opposite Rotor Pairs Flipped

be seen there in Fig. 18, but this lack of points may be due to large curvature on the nonlinear map between \hat{v} and \underline{h} , as discussed earlier.

Figure 19 shows another cutaway; this time the hemisphere of Fig. 18 is sliced into a quarter-sphere by the xy plane (now all plotted points have $x,y > 0$). The viewpoint is rotated by roughly 45° , such that the envelope segment is seen edge-on. The 2H state can be seen to patch onto the envelope dimples at the top, left, and right of the figure (the "eye-like" features left and right of center are cut 2H surfaces coming out of the page). The 0H surfaces are seen to fill in the region near the center.

A variety of planar singular surface projections for SGCMG pyramid-mounted arrays can be found in Ref. [45]. A detailed analysis of the properties of singular surfaces is presented in Ref. [46]. Singular surfaces become considerably more complicated[47] if the CMGs in an array are allowed to have different rotor momenta (not standard operation, but potentially a contingency in the case of degraded hardware or when using the CMG rotors for energy storage).

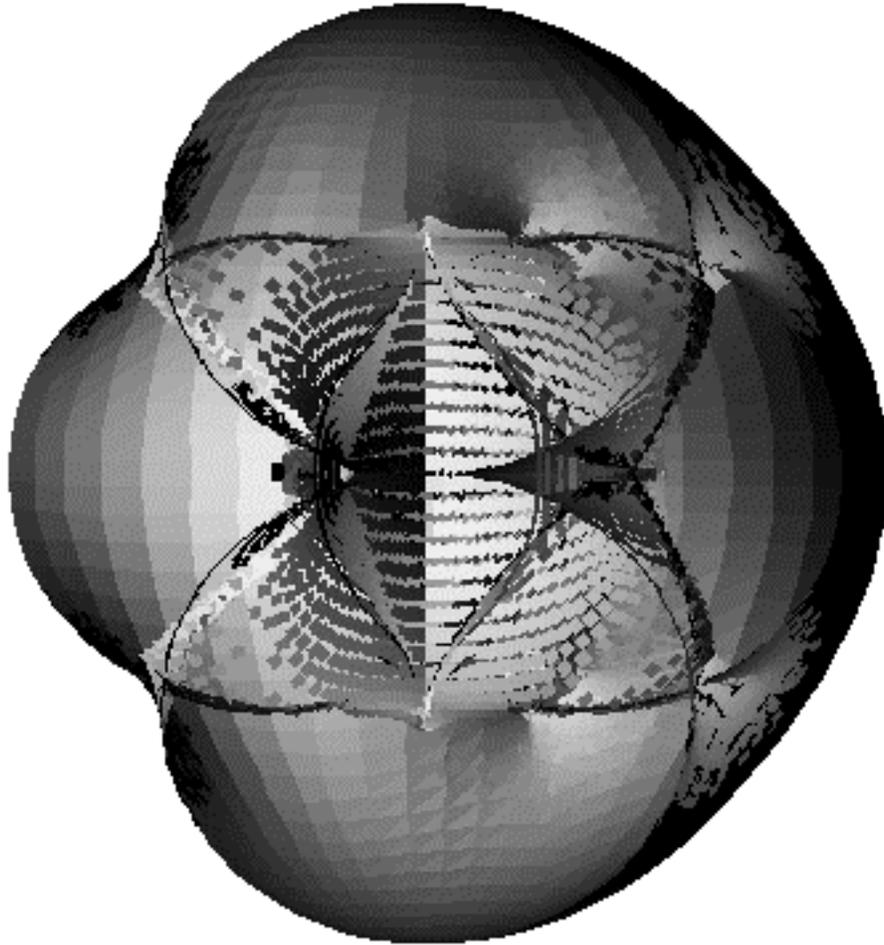


Figure 18: Envelope & Singular Surfaces; Cutaway View

Singular states fall into two categories; those that can be escaped through null motion (i.e. redistributing the CMG rotors to relieve the singularity without changing the momentum state of the spacecraft), and those that are inescapable through null motion. The latter "inescapable" states pose a major difficulty with SGCMG systems. They arise from the limited control capability per actuator; i.e. using a collection of single DOF devices to control a 3-axis space, as was mechanically illustrated in Fig. 5. With devices such as DGCMGs (2 DOFs per actuator) or planar manipulators (1 DOF actuators to control 2-axis space), inescapable singular states are either nonexistent or trivial (i.e. the envelope of any system is technically an inescapable singularity).

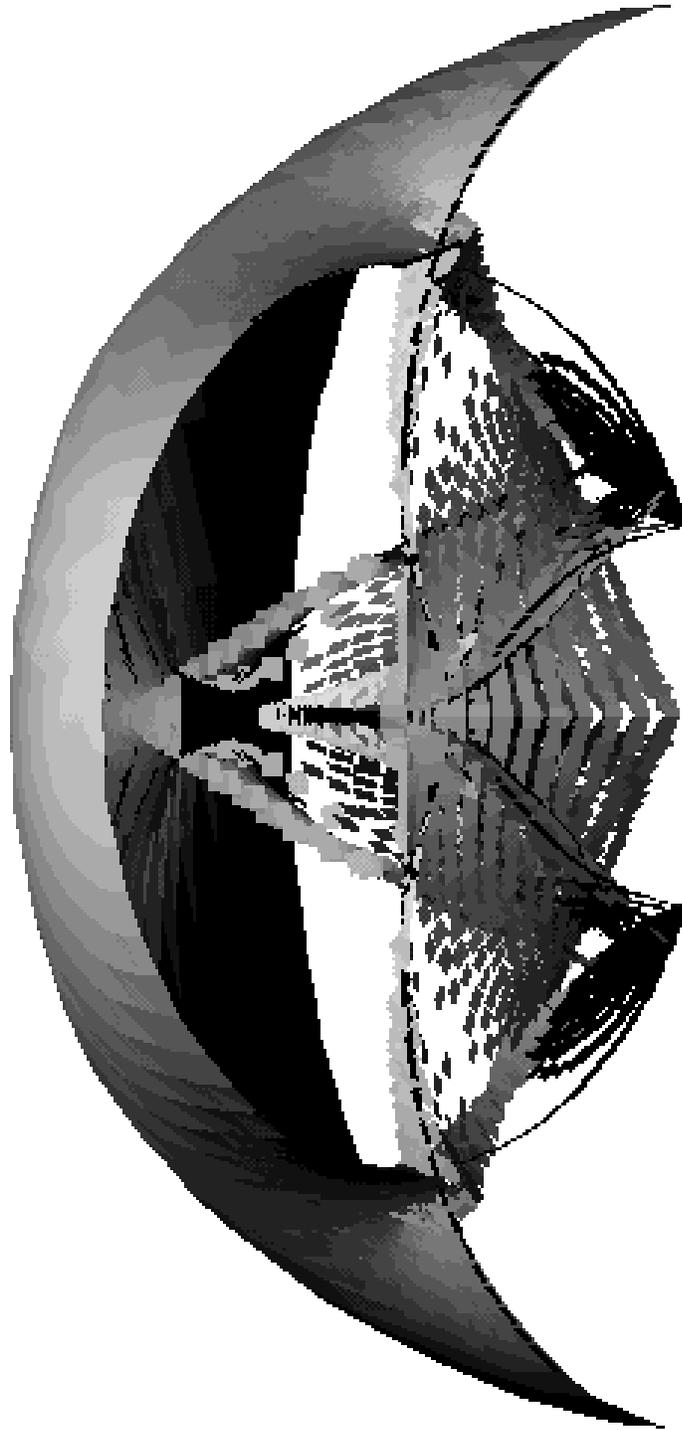


Figure 19: Momentum Envelope & Singular Surfaces; Quarter-Slice

Refs. [12,33,48,49,50] detail a quantitative method for testing singular surfaces for the possibility of null motion escape. These techniques involve expanding the total CMG momentum about a singular point to second order in null gimbal displacement, and examining the resultant quadratic form (see Sec. 4.1). If it is positive definite, the range of

null motion is bounded, and the singularity is inescapable (hence termed "elliptic"). Otherwise, the singularity may be altered by null motion and is potentially escapable (these are termed "hyperbolic"). Degenerate motion of the system, however, can alter the singular state (hence the singularity is classed hyperbolic), but not affect the CMG gain (m ; Eq. 7). Refs. [49,50] further test admissible null solutions by requiring null displacement to alter the m values, thus ensuring escape from the singular state.

This technique has been applied[33,49] to classify the singular states of the pyramid-mounted 4-SGCMG system. The 2H surfaces (Fig. 15) were generally found to be elliptic (inescapable), while the 0H surfaces (Figs. 16,17) were generally seen to be hyperbolic (escapable). Since they can not always be relieved without torquing the spacecraft, the 2H singularities can be especially problematic for SGCMG systems and steering laws. Proposed SGCMG deployments generally size the CMG array to keep the required momentum capacity within the boundary of these 2H singular surfaces. This considerably reduces the available momentum capacity (the effective reduction can be seen in Fig. 18). In order to operate in the regions of inescapable singularities, an intelligent steering law must be developed that avoids elliptical states where possible, or rapidly transits elliptical configurations, while minimizing their effects on the spacecraft momentum state and attitude control.

2.4) SGCMG Steering Laws

The purpose of a "conventional" CMG steering law is to determine CMG gimbal rates that answer instantaneous torque commands, while managing the system redundancy to maintain desirable (i.e. nonsingular) gimbal states. Most CMG steering laws calculate torque-producing gimbal rates with some variant of pseudoinverse.

$$8) \quad \dot{\underline{\theta}}_{\tau} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1} \underline{\tau}_{\text{cmd}}$$

The pseudoinverse produces the minimum 2-norm vector of gimbal rates $\dot{\underline{\theta}}_{\tau}$ that realize the commanded torque $\underline{\tau}_{\text{cmd}}$. If these gimbal commands are applied to a CMG system without modification, singularities are often encountered. Because of its minimum Euclidean-norm property, the pseudoinverse solution tends to mainly move CMG rotors with output torques projecting significantly onto the input command. Rotors with small

torque projection (i.e. aligned or anti-aligned with the command direction) are essentially kept in position, encouraging the formation of a singular state.

The pseudoinverse can be considered as a "particular" solution to the torque equation, Eq. 6. The corresponding homogeneous solution is produced through null motion, which describes gimbal motion that does not change the CMG momentum state and torque the spacecraft. Null motion may be calculated in a variety of manners; i.e. through a singular value decomposition[51], or a projection operator[34] as outlined below:

$$9) \quad \dot{\underline{\theta}}_N = [\mathbf{I}_N - \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1} \mathbf{J}] \underline{d}$$

Where:

$N = \#$ of CMGs

$\underline{d} =$ Arbitrary N -vector to be projected into nullspace of \mathbf{J}

$\mathbf{I}_N = N \times N$ Identity matrix

Since the null vectors $\dot{\underline{\theta}}_N$ are orthogonal to the row vectors of \mathbf{J} , null motion may also be effectively calculated as a cross product[33]. An independent null vector exists for each excess degree of freedom in the CMG system. The rank of the nullspace is thus:

$$\text{Rank}(\mathcal{N}) = N - \text{Rank}(\mathbf{J})$$

In the nonsingular case, \mathbf{J} always has rank 3. For a nominal 4-SGCMG system, there is thus only one null vector, and it can thus be summarized:

$$10) \quad \dot{\underline{\theta}}_N = (|c_2 \ c_3 \ c_4|, -|c_1 \ c_3 \ c_4|, |c_1 \ c_2 \ c_4|, -|c_1 \ c_2 \ c_3|)^T$$

The c_n vectors above represent the output torques of CMG #n; i.e. the n'th column of \mathbf{J} . The components of $\dot{\underline{\theta}}_N$ are determinants of 3x3 matrices or scalar triple products:

$$11) \quad |c_i \ c_j \ c_k| = c_i \times c_j \cdot c_k = \det[c_i \ c_j \ c_k]$$

This technique can be generalized to systems of more than 4 CMGs. For instance, for a 5-CMG system, one must calculate two null vectors. The first null vector can be obtained by zeroing the fifth gimbal rate and applying Eq. 10 on the other four gimbals.

The second null vector is then found by taking the cross product of the rows of a 4x5 matrix formed by including the first null vector as an additional row of \mathbf{J} (since the two null vectors should also be orthogonal). This again leads to a null vector with components calculated as determinants. As opposed to Eq. 10, the second null vector in the 5-CMG case will require the determinants of 4x4 matrices. This null vector will have five components, each of which will be calculated by a determinant including all columns of the augmented Jacobian, except for the column numbered the same as the null component being calculated (as in Eq. 10). For odd numbers of CMGs (N), the signs modifying all determinants are positive; for even N , the signs alternate, as in Eq. 10. More details on this technique are given in Ref. [33].

The particular and homogeneous solutions are summed to form a general relation that spans all possible CMG motions that attain the commanded torque:

$$12) \quad \dot{\underline{\theta}}_{\text{cmd}} = \dot{\underline{\theta}}_{\tau} + \sum_{i=1}^N \bar{k}_i \dot{\underline{\theta}}_{N_i}$$

The CMG steering law assigns weights \bar{k}_i to the system's null vectors $\dot{\underline{\theta}}_{N_i}$ to determine gimbal motion that avoids singular and problematic states. For the 4-CMG system, this amounts to picking a signed scalar variable, since there is only one null vector. Most (if not all) steering laws are local in nature, hence solve Eq. 12 for gimbal rates that are instantaneously "optimal" at the current gimbal configuration. The \bar{k}_i factors are thus picked at each timestep. An objective function can be defined to reflect desirable gimbal orientations, and the \bar{k}_i factors can be calculated by differentiating it with respect to null vector displacement.

Although such "gradient" techniques can work well with double gimballed CMGs[22], they generally have considerable difficulty in managing SGCMG systems of limited redundancy[13]. Steering laws that locally track the maximum of an objective function tend to often be drawn into singular states, especially if the objective is determined through a function of the CMG gain (Eq. 7). Although the current gimbal position seems locally "optimal", in the sense that any null displacement will lower the CMG gain (hence decrease the objective value), the torque command may be pulling the gimbal state into smaller gain and eventually a singularity. Using a familiar analogy, one is "marching" along a ridge that regionally seems to be the highest point (i.e. locally optimal). Eventually, the height of the ridge declines, until it ends up at a pond (i.e. singular state). In the SGCMG case, it seems that many of the locally optimal gimbal trajectories (i.e. "ridges") end up singular (i.e. "drenched"). In addition, singularities may "attract" nearby

gimbal trajectories. The singular state is, in a sense, "maximally stretched", in that all rotors project maximally or minimally onto the singular direction. In this regard, it can extract behavior similar to that encountered past momentum saturation. Since the authority about the singular axis is limited near a singular state, the gimbals can be quickly driven singular in order to produce any commanded torque components in the singular direction. If a feedback controller is wrapped around the steering law, this effect can be highly problematic. As the CMG gimbals drift out of the singular orientation, they are driven back into it to try and null errors accumulated about the singular axis, thus one tends to "hang up" or "lock" the gimbals in the singularity.

In order to avoid the above pitfalls, a variety of heuristic and analytic approaches have been developed to determine a singularity-avoiding null motion policy (i.e. a strategy for assigning the constants \bar{k}_i in Eq. 12). Two promising techniques were proposed in Ref. [34]. One of these is a simple strategy to steer individual CMG rotors to point toward the saturation singularity (i.e. direction of commanded or stored momentum). As discussed in the previous section (and illustrated in Fig. 7), internal singular states can be characterized by having one or more rotors (links) "flipped" to project against the singular direction. By adjusting the added null motion such that the rotors are "unkinked" where possible, and all eventually point into the saturation hemisphere, many internal singular states can ideally be avoided. Unfortunately, this technique can't be relied upon to skirt all singular states[13]. It often tends, however, to obtain gimbal trajectories that can be rapidly improved by nonlinear optimization, thus is used as a starting point for the search procedure discussed in the next chapters.

The other method proposed in Ref. [34] is a much more complicated procedure, where all internal singular states along a predicted torque history are characterized and ranked. Null motion is then added via the projection of Eq. 9 to avoid the singularity that is expected to be encountered first. Although this hints of being a global solution, in that one considers a predicted torque history and its consequences, it is not a nonlinear optimization, and picks the null factors \bar{k} using a linear gradient technique, which is known to fail.

A related approach has been formulated in Ref. [35] for a 4-CMG pyramid used to stabilize a balloon-borne telescope package. Here, a parameter table (potentially generated off-line) stores the signs of null displacements \bar{k} that will achieve a "globally maximal" CMG gain at various gimbal orientations. This table is indexed with the current gimbal angles, hence a "globally best" state must be determined for a wide range of input gimbal positions. Nonetheless, this method is still an instantaneous "tangent" approach, in that the predicted torque history is not taken into account. Gimbals are steered into a "best" orientation for the current momentum state, although attaining that "best" orientation at one

point in time may create a problem later and prevent the CMG system from making a nonsingular transition to a future commanded momentum. CMG gimbal trajectories may be considered as a path through various gimbal closures (introduced in the previous section). Although a certain gimbal closure may be "globally" optimal for a particular momentum value, the CMG system may have to transit through a singular state to attain a closure that yields a future commanded momentum. In general, the global momentum trajectory should be considered for best performance.

A variety of tangent-based techniques were applied to steer SGCMGs in Refs. [13,52]. Weighted pseudoinverses and several gradient-directed null motion algorithms were attempted. As mentioned earlier, strategies using local gradients often lead to singular states; this was indeed noticed. In fact, the technique that worked best is adding "undirected" null motion; i.e. assigning \bar{k} to be always positive (thus not forcing the system along a local gradient). The factor \bar{k} was adjusted in an unusual way; if the CMG gain m was above unity (normalizing the CMG rotor momenta $|\underline{h}_i|$ to 1), \bar{k} was made proportional to m^6 , otherwise \bar{k} was scaled as $1/m^6$. The above conditional adds null motion at both high m states (to keep the system moving, and hopefully not settling into configurations & trajectories becoming singular) and low m states (to avoid the singularities themselves). This technique, termed the "second inverse gain" method, has an obvious drawback in that null motion should as easily enter a singularity as leave one. Adding gimbal motion without following a local gradient seems, however, to yield better performance.

One of the main contributions from the effort of Refs. [13,52] was the application of the SR (Singularity Robust) inverse to CMG steering. The SR inverse is a modification to the pseudoinverse that has been proposed for robotic manipulators[53]. When the CMG gain begins to decrease, and the system goes singular, the SR technique adds a perturbation to the 3x3 $\mathbf{J}\mathbf{J}^T$ matrix in order to maintain rank and retain invertibility.

$$13) \quad \dot{\underline{\theta}}_{\tau} = \mathbf{J}^T (\mathbf{J}\mathbf{J}^T + \rho \mathbf{I}_3)^{-1} \underline{\tau}_{\text{cmd}}$$

Eq. 13 is identical to the pseudoinverse (Eq. 8), except for the term proportional to the 3x3 identity matrix. The constant of proportion, ρ , is generally made to be negligible when the CMG gain is high, and increase when the gain drops in the vicinity of a singular state. This term, when significant, will introduce torque errors. Gimbal rates, however, can always be calculated; Eq. 13 does not become singular with a drop in the Jacobian rank, as does Eq. 8. Singularities are thus avoided or "transited" in momentum space; torque errors are created near a singular state and hopefully compensated afterward when the CMG system re-achieves full control.

The effect of the SR inverse near a singular state is to drop the rank of the torque constraint (Eq. 6); gimbal rates are not generated to answer command components about the singular axis. This can create a problem when the torque command is parallel to the singular direction; all gimbal rates will be zero, and the system will essentially stop in the midst of the singularity[13]. One must then augment the SR inverse with an additional procedure that adds torque disturbance (or modifies the input command) to relieve this situation.

The search procedure described in the next chapter uses the SR inverse to calculate torque-producing gimbal rates. The feedforward gimbal trajectories are adjusted to minimize the torque error produced by the SR inverse, as will be discussed.

In order to assure operation without encountering inescapable singular states (i.e. Fig. 15), recent deployed and proposed SGCMG systems for large spacecraft rely on CMG arrays with a device count well above minimum redundancy (i.e. 5 or 6 wheels) and/or a momentum envelope sized appreciably beyond the mission requirement. SGCMGs are currently running on the MIR[29] and have been proposed[30] for the NASA space stations. Since singularity avoidance becomes much simpler after adding more CMGs[48] or restricting their range of operation, straightforward gradient-based algorithms and configuration-specific approaches begin to function adequately, and have both been simulated and/or applied with these vehicles.

Other steering concepts have been proposed and attempted. The system of Ref. [54] attempts to circumvent the singularity issue by driving subsets of SGCMGs in an array as virtual "scissored pairs", potentially resulting in excessive restriction on the CMG system and operation. The method of Ref. [55] involves calculating optimal initial gimbal angles for different torque requests issued to a CMG system. Despite this approach, the central problem of SGCMG steering remains; i.e. one must somehow transit to this "optimal" set of initial gimbal angles (representing a particular closure) from an arbitrary CMG orientation. In addition, the meaning of "optimal initial angles" becomes diffuse as the CMG array is loaded with secular momentum and torque commands vary in direction and magnitude.

In Ref. [17], the pseudoinverse was dispensed with entirely, and a linear program was used to steer DGCMGs, steer SGCMGs, fire reaction control jets, or manage any mixture thereof. The linear program has the advantage of being able to hard-bound gimbal rates, and finds a gimbal rate solution to the torque command that minimizes the 1-norm of an objective function, which may be formulated to reflect the optimality of a CMG configuration. It is still a linear gradient approach, however, and thus experienced considerable difficulty in steering minimally redundant arrays of SGCMGs.

In summary, most linear steering laws that locally answer instantaneous torque commands will have difficulty avoiding singular states in minimally redundant SGCMG systems. In order to improve performance, the ideal steering algorithm must be "global" in nature, and optimize the CMG gimbal motion over a predicted momentum trajectory. In Ref. [56], a variational method was proposed to accomplish this, but the implementation is quite complex, especially for onboard spacecraft operation. "Global" CMG steering over predicted momentum trajectories may be efficiently attained, however, by a search operation, as is detailed and demonstrated in the following chapters.

3) Search-Based CMG Steering

3.1) Analogy to Waypoint Planning

The use of search algorithms in this study of CMG steering was inspired by an analogy with waypoint determination for mission planning applications. When planning a mission that requires, for instance, an optimal aircraft trajectory[4,57], the problem is divided into three sections; goal planning, waypoint planning, and timeline management. Goal planning determines the selection of mission objectives (i.e. "targets") and the order in which they are to be visited. The waypoint planner finds an optimal path between goal nodes (i.e. minimizing needed fuel, avoiding threats, etc.). The timeline manager is a "watchdog" regulator task that monitors the aircraft state as the optimal trajectory is followed, rejecting disturbances and re-queuing the higher-level planners when encountering significant divergence from the predicted state (i.e. due to unmodeled winds leading to higher fuel usage, changes in the nature of anticipated threats, vehicle damage, etc.).

The adaptation of this strategy to CMG steering is diagrammed in Fig. 20. The goal planning function is accomplished by the momentum management algorithm, which schedules a momentum profile to be attained by the CMG system. The waypoint planner is realized by a search process that performs the singularity-robust inverse kinematics to determine CMG gimbal trajectories that follow the momentum commands. A regulator task monitors the divergence of the spacecraft dynamics and CMG gimbal motion from predicted values. For small disagreements, corrections can be applied directly to the CMG system. If larger divergence is evident, the CMG search must be re-executed to calculate a new gimbal path. If significant divergence is detected from the predicted environmental torque history, the momentum manager must be re-executed, and the waypoint search run with the updated momentum profile. Additional detail on spacecraft implementation is

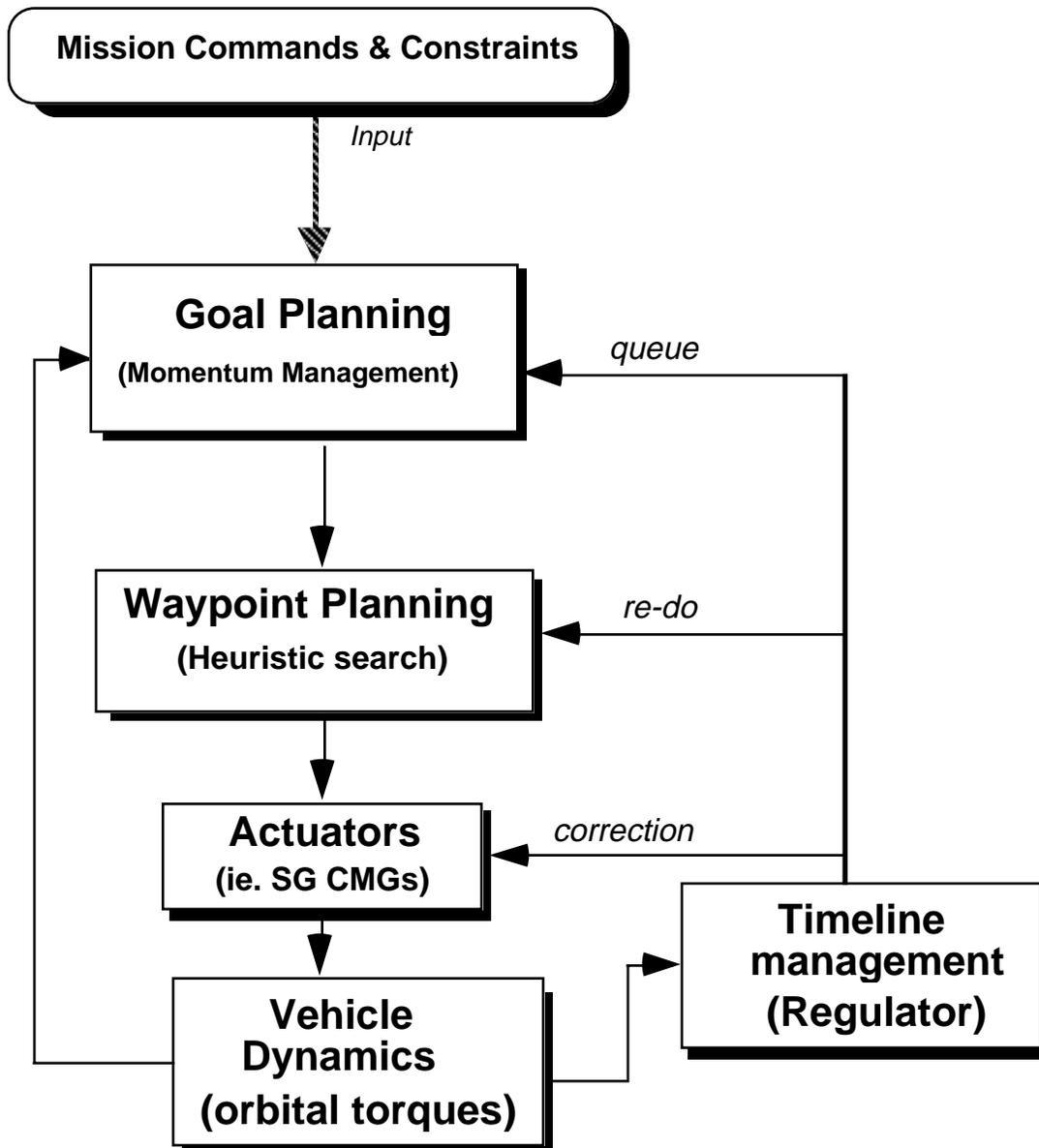
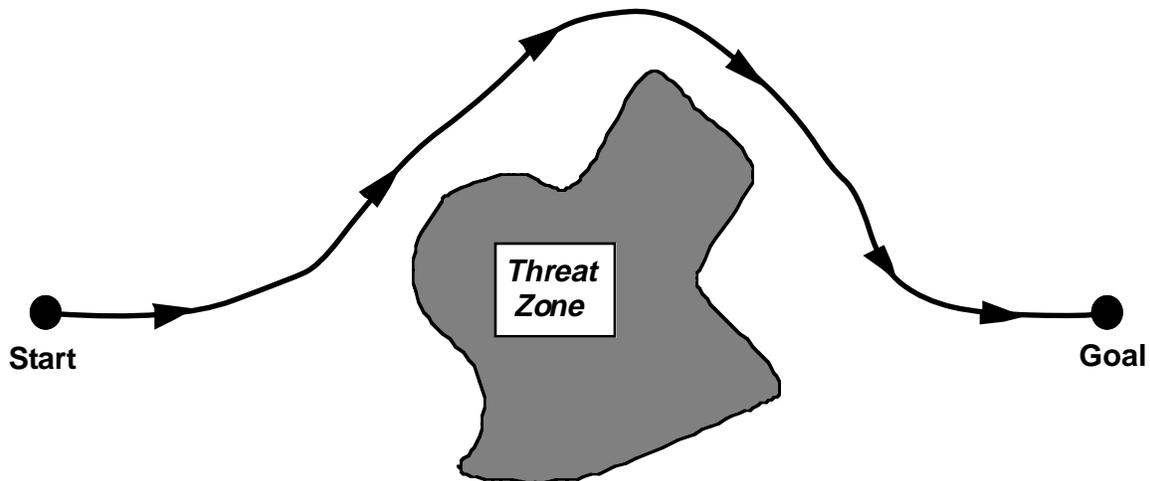


Figure 20: Mission Planning Implementation of Search-Based CMG Steering

given later in this chapter. Since this study has focussed on the waypoint planner, most of this chapter will be dedicated to describing its philosophy, structure, and function.

The operation of the waypoint planner is illustrated in the planar example of Fig. 21, where an optimized "threat-free" path is determined between start and goal nodes. In the aircraft case, threats have the standard connotation; i.e. regions to be avoided where adversaries have concentrated sufficient firepower to create appreciable probability of scoring a hit and jeopardizing the mission. For CMGs, the notion of a "threat" may be



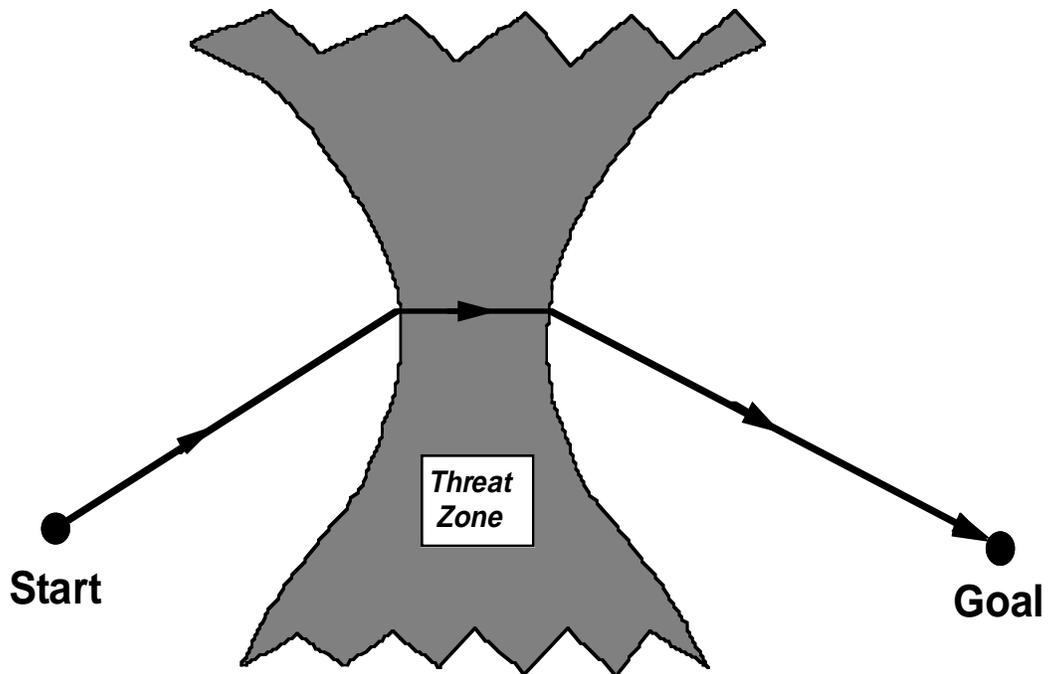
- Find "Optimal" path between sequential pair of goals
 - Avoid threat zones, minimize fuel
 - Use heuristically-guided A* Search algorithm

Figure 21: Waypoint Planner for Optimal Determination of Aircraft Trajectory

replaced with a singular gimbal state, creating a waypoint planner that will try to calculate singularity-avoiding gimbal paths between sets of start and goal nodes.

In the aircraft planning problem, a situation can be envisioned where threats can not be entirely avoided because of constraints imposed by aircraft maneuverability and the allowed span of trajectories. A situation of this sort is portrayed in Fig. 22. The dark zone in the middle of the paper represents an extended threat zone between start and goal nodes. Since the aircraft can not fly around the threat here, the optimal planner will pick a path that minimizes time spent in the threat zone, thereby choosing the least risk to aircraft and mission.

A direct analogy again exists in the CMG world. As discussed in Sec. 2.3, inescapable "elliptic" singular states exist in single gimballed CMG systems. If one is sufficiently close to such a singularity, or in a particular gimbal closure, there may be no path between start and goal nodes that is able to avoid the singular state. The situation is illustrated in Fig. 23. If the null degrees of freedom (vertical axis) are used for singularity avoidance, and the torque command (horizontal axis) extends through an elliptic singular state, there may be no null path around the singularity from particular initial gimbal configurations. Since the CMG waypoint search uses the SR inverse, most such singularities may be transited, at the expense, however, of introducing torque and momentum errors. In these cases, as with the aircraft planner, the CMG waypoint search will try to select null motion such that the effects of the singularity are minimized; i.e. the



- No path past threat
- ☞ Minimize risk & time spent in threat

Figure 22: Aircraft Waypoint Planner Response to Unavoidable Threat

gimbal orientation at the singular state is adjusted such that momentum perturbations are kept small and the singular condition is transited as rapidly as possible.

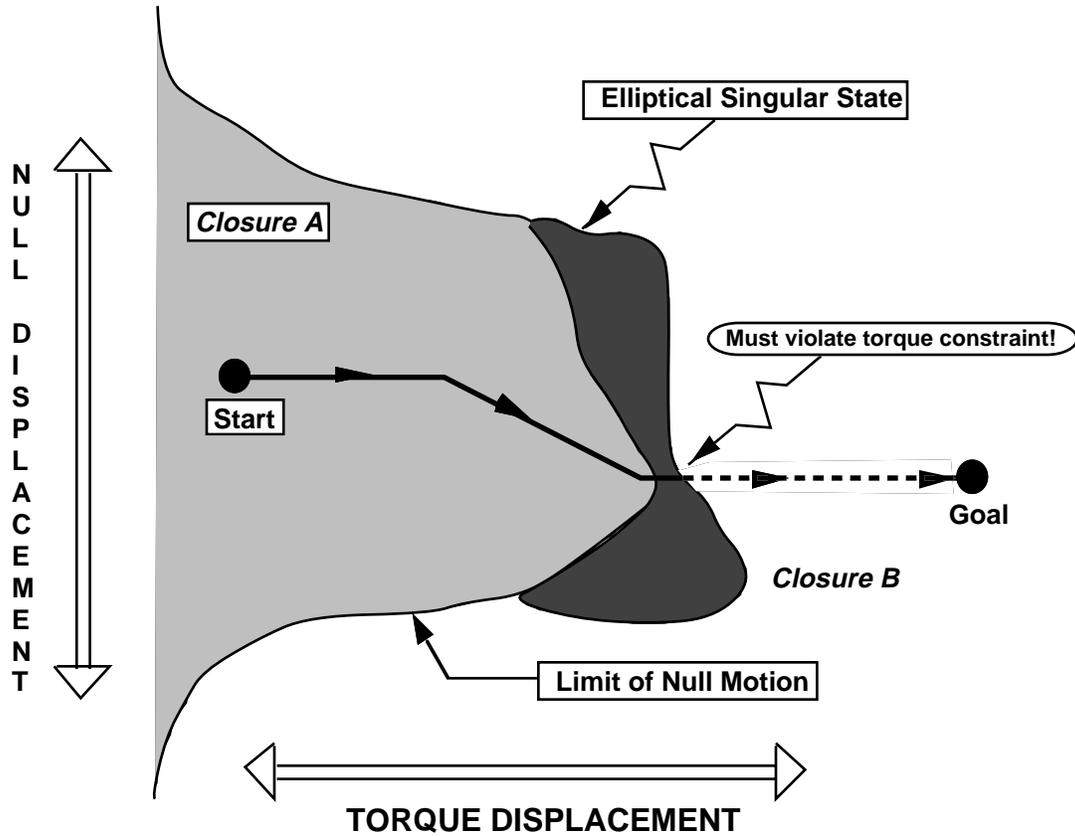
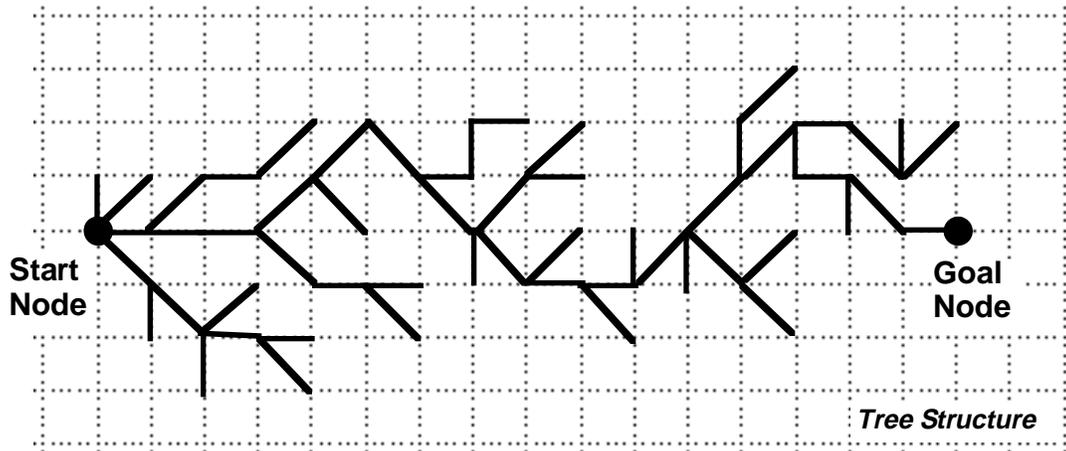


Figure 23: CMG Waypoint Planner Response to Unavoidable Singularity

3.2) Search Techniques

The aircraft mission planner of Refs. [4,57] used an A^* search to determine the optimal vehicle trajectory between start and goal nodes. Discretized vehicle positions (i.e. "nodes") are defined on a planar grid containing the start and goal locations (a 3-dimensional grid is used if altitude is also taken into account). Beginning at the start, the vehicle position is propagated through adjacent nodes until the goal is reached. As different nodes are expanded, a linked list is created; each node has a pointer to its predecessor "parent" node. When the search terminates at the goal, the optimal path is reconstructed by propagating backwards through this list from goal to start. A coarse illustration of this concept is given in Fig. 24, which shows a typical "tree" structure produced by a search as it solves for a trajectory.



- **Expands path through grid from start node to goal node**

Figure 24: Tree Structure for an A* Search Over a Planar Grid

The A* search consults an objective function to assign a "cost" to each node and solve for an optimal trajectory. The objective for a node j consists of two components; a trajectory cost g_j that specifies the penalization accumulated from the start to node j (representing the optimality of the trajectory at node j 's position), and a "heuristic" estimate of the cost to completion from node j to the goal. This may be summarized:

14)
$$c_j = g_j + h_j$$

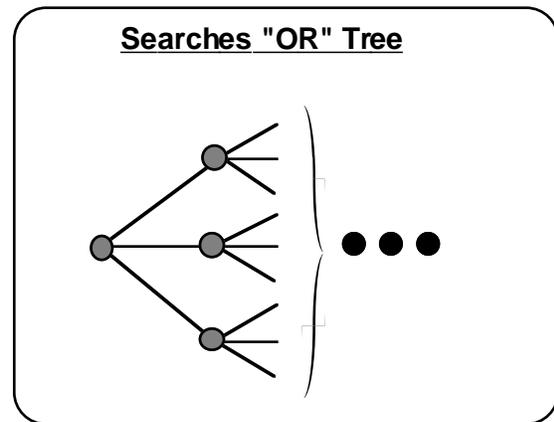
g_j = "Cost" of node # j from start

h_j = Estimated cost to complete path from node # j to goal.

The logic flow for a simplified A* tree search is given in Fig. 25. Initially, all nodes are expanded that originate from the starting node (i.e. for the mission planner, nodes are explored that represent the discretized vehicle positions nearest to the starting point). If the goal is hit, the search exits with the optimal path, propagating from goal-to-start, node-by-node, as mentioned previously. Otherwise, all "open" nodes (i.e. unexpanded nodes at the edge of the tree) are scanned for the smallest evaluation of c_j (assuming an objective minimization). This node is expanded, checked for "goal" status, and the process continues as discussed above.

A* Search Process

- Search process is a "diffusion" across the tree
- Optimal solution guaranteed if heuristic estimate *admissible*



Simplified Implementation

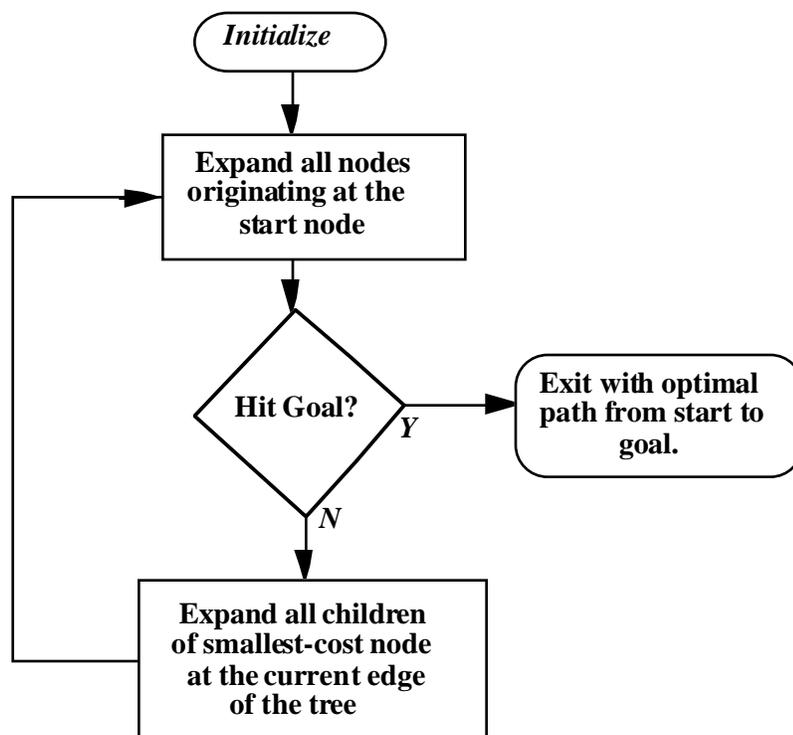


Figure 25: Simplified Logic for an A* Search

The addition of a predictive heuristic function to the objective calculation gives the A* search its special character. An appropriately defined heuristic will effectively prune

away paths leading down "blind alleys", enabling computational resources to be concentrated on the most promising alternatives. The optimal tree-search problem is known to be "NP-hard"; i.e. the computation requirement needed in applying a blind search grows exponentially with the size of the problem. The introduction of appropriate heuristics reduces the difficulty in arriving at a solution. In many problems, the search complexity can be reduced sufficiently to approach linear dependence on the problem dimensions. A tradeoff exists between the "aggressiveness" of a heuristic definition and the optimality of the resulting solution. If a heuristic is defined to readily sidestep search branches and paths, a solution will be obtained quickly (and the search will require less storage space for expanded nodes), although this solution may be significantly suboptimal. If, on the other hand, the heuristic is defined to be more admissible, thus less apt to discourage search branching, the solution will be closer to optimal, although it may require considerably more computation time and node storage. This type of optimality vs. computation tradeoff appears in several nonlinear optimization strategies, and may be effectively managed in an A* formalism by adjusting the heuristic contribution. By setting all h_j to zero, one has a uniform-cost search; if, in addition, all c_j are set to equal their level in the search tree (i.e. c_j is set to 1 plus the level of the parent node), a breadth-first search will result. If the c_j are set to the negative of their level in the search tree, a depth-first search is produced. An informed depth-first search may also be approached by defining an overly-aggressive heuristic that prefers a particular search path. More detail on A* and other search variations may be found in Ref. [3].

As mentioned previously, the basic aircraft mission planner works to minimize fuel usage and avoid threat zones. These quantities may be readily accounted for in the objective. The g_i factor may be defined to represent the amount of fuel and integrated threat status encountered "so far" in traversing the tree (i.e. moving the vehicle) from the start to node j . The heuristic function h_j is defined to represent the expected fuel needed to complete the voyage from the location of node j to the goal (i.e. if straight-line travel is assumed, an admissible heuristic is produced), with perhaps some amplitude added to reflect the anticipated threat status that could be encountered enroute. Since the mission planner's search will take place within a bounded 2 or 3 dimensional region, the total amount of discretized positions (or nodes) will remain quite finite. In this case, however, a node may have different trajectories passing through it (i.e. several parents). This situation may be dynamically accommodated while solving the search, as discussed in Ref. [3]. Because the number of nodes in this problem is significantly limited, cost values for all nodes may be pre-calculated through Eq. 14 and stored in a table that is consulted during

the real-time search execution. This minimizes the computation load, and the mission planning search is seen to finish quite rapidly, as demonstrated in Refs. [4,57].

In contrast, however, the CMG search is somewhat more difficult. As will be defined in the next section, the coordinates are not specified in a simple Euclidean space, thus the search is across a tree, with "child" nodes connected only with their immediate parents. The density of nodes thus increases exponentially with the depth of the tree (the problem is classically NP-hard). This discourages the possibility of "pre-calculating" node costs and parameters for all nodes; costs must be calculated dynamically, as the nodes are expanded. In addition, a true A^* search, as described above, can experience difficulty with such a fine tree structure. If a singular state (i.e. "threat zone") is encountered in the midst of the search tree, where the node density is high, the search can become "bogged down" expanding the multitude of nonsingular nodes surrounding the border of the singularity. When operating in a region where the node density is high, the A^* search can repeatedly expand nearby good gimbal states into the high cost region (i.e. singular state or objective "hill"), and spend much effort exploring the myriad of fruitless possibilities available in trying to avoid the singularity. Since the A^* search examines all "open" nodes before any new node is expanded, the needed computation time can appreciably increase as more open nodes are added. Consequently, in cases where a difficult singular condition is positioned within a region of the tree with high nodal density, the search can become burdened with expanding dead-end paths and not produce a solution trajectory until inordinate amounts of computer time and storage are expended.

If one lets the A^* search solve for a CMG trajectory, the resulting solution will be optimal, provided that any employed heuristic is sufficiently admissible. The downside to this feature is that it may take practically forever to arrive at this solution, especially considering situations such as portrayed above. The CMG trajectory, however, need not be optimal; a suboptimal gimbal trajectory will suffice, so long as it avoids (or reduces to tolerable levels) the impact of a singular gimbal configuration. This argument has led to the CMG adaptation of a guided depth-first search.

The guided depth-first search somewhat resembles a gradient-based null motion CMG steering law with corrective look-back; i.e. local gradient solutions are pursued until they get into trouble, at which point the trajectory is examined, perturbed at a promising point, and again gradient-propagated. A summary of the depth-first search strategy is shown in the diagram of Fig. 26. Before the search is actually executed, a series of trial trajectories are calculated using various gradient techniques. These trajectories are propagated locally node-by-node; i.e. at each step, the child node that best exhibits the assigned criteria is chosen for expansion. The trials thus produce an initial population of

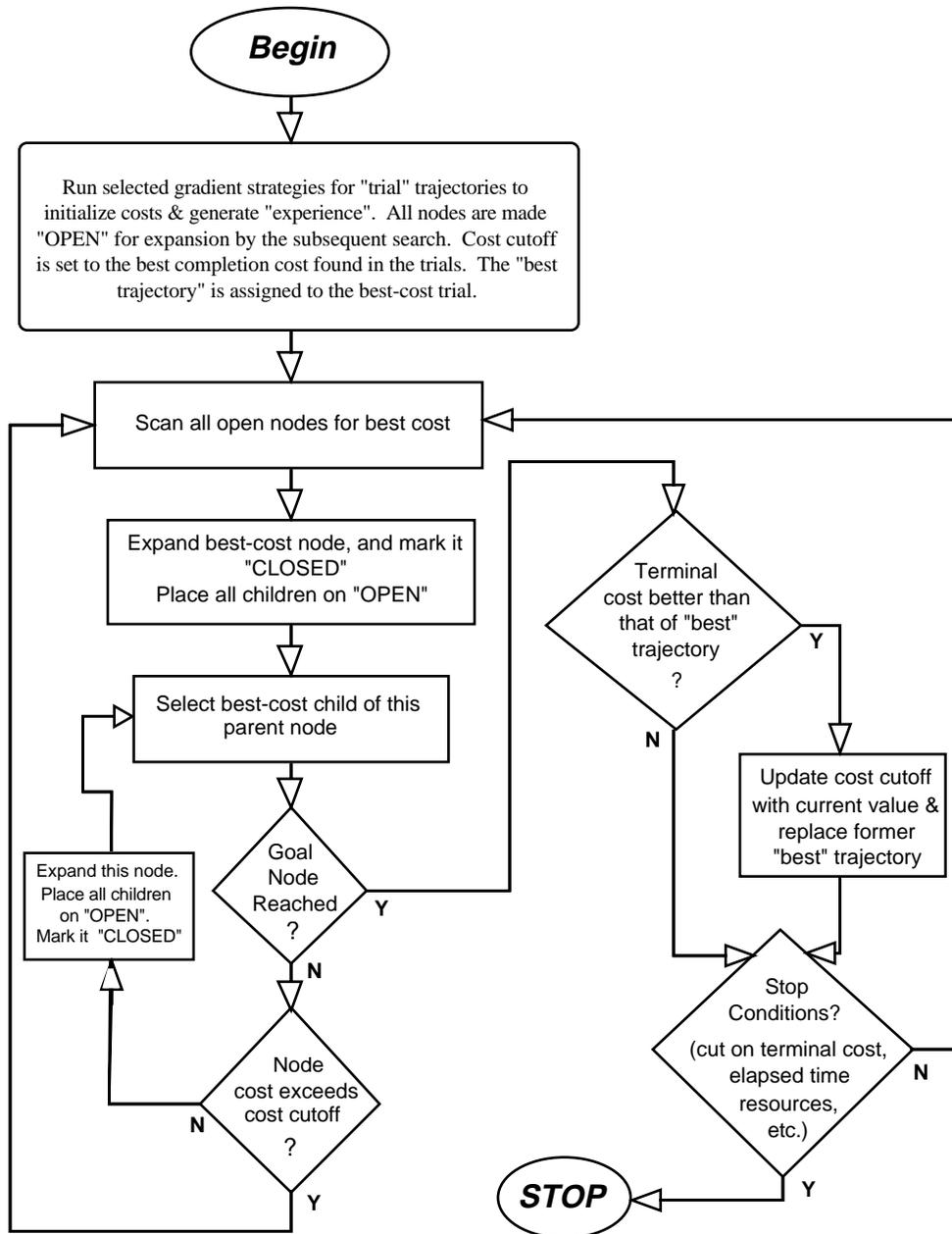


Figure 26: Flow Diagram for Guided Depth-First CMG Search

open nodes representing gradient solutions; these may be "grafted upon" (i.e. they may be chosen for expansion) and improved by the search process. In addition, the winner of the trials produces an initial "best trajectory", providing a cost cutoff with which to judge and prune subsequent search attempts.

The search process itself is then begun; the list of all open (i.e. unexpanded) nodes is scanned (initially derived from the trial trajectories), and the best-cost candidate is expanded and closed. The search then proceeds in a gradient-guided depth-first fashion, picking the best immediate child for subsequent expansion at each level. This policy continues until either the goal is reached (i.e. the end-state momentum is achieved) or the net trajectory cost exceeds the current cost cutoff. In the former case, the "best trajectory" definition and cost cutoff are updated with the newer values, provided that the new terminal cost surpasses the previous best result. The list of all open nodes is again scanned for the best cost value, and the guided depth-first search continues anew from that point.

Candidate CMG gimbal trajectories are continually produced with this search technique. The initial trajectories can be of fairly poor quality, especially in cases with difficult singular configurations dominating the current gimbal region, but as the search progresses, the quality of the accepted trajectories gradually improves; the cost cutoff, updated after each trajectory is completed, grows more stringent, and the search wastes less time exploring poor-cost paths. The search process can be stopped when the allotted time and/or computation resources are exhausted, or when the terminal trajectory cost improves past a preset threshold.

The above discussion illuminates a major advantage of the guided depth-first search over A^* for real-time implementation in this application. As discussed earlier, the A^* search can take a very long time to arrive at a solution, due to the high node density implicit in the CMG problem. When extended into a region of increasing cost, the A^* formulation tends toward a guided breadth-first search, which can be extremely time consuming. An efficient, predictive, heuristic function may speed the A^* operation somewhat, but can be difficult to formulate for CMG systems (apart from the cost-cutoff pruning discussed above). Once the A^* search produces a solution, it will be near-optimal, but the lengthy wait may be totally impractical. The depth-first solution, on the other hand, provides no claim of optimality, but quickly produces solutions that progressively improve. The search process can be halted at any time, and the "currently best" trajectory implemented with some anticipation of its performance.

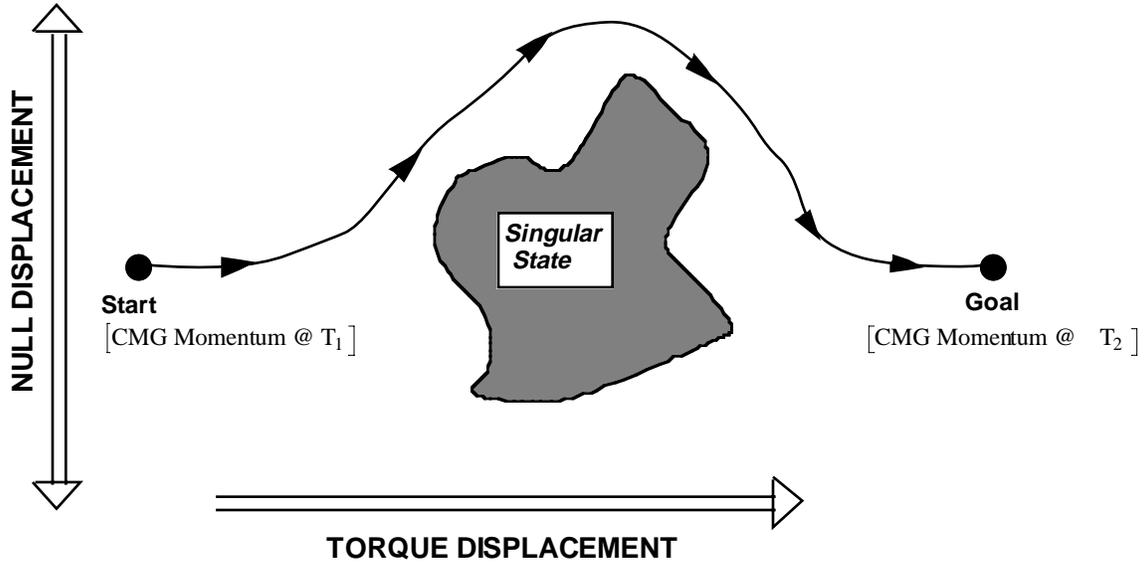


Figure 27: Coordinate Definition for CMG Waypoint Search

3.3) Coordinates, Costs, and Heuristics

The nodes on the tree searched by the processes sketched in the previous section represent CMG gimbal orientations. If the gimbal angles themselves (θ_i of Eq. 2) are used for variation in the search strategy, the combinatoric possibilities (i.e. one search variable for each actuator) would preclude rapid search execution. Fortunately, the kinematics of the problem simplify this issue. The 3-axis torque/momentum constraint (Eq. 6) intrinsically removes three degrees of freedom from the search, leaving the system redundancy, expressed as null motion, available for variation.

This is illustrated in Fig. 27 as a CMG implementation of the waypoint task that was introduced in Fig. 21. The horizontal axis represents increasing time. At each point on this axis, the momentum manager has assigned a stored momentum to the CMG system, hence the difference between these momentum commands at adjacent points may be scaled by their time difference to produce a commanded torque. CMG gimbal rates can be generated to answer this torque via the SR-inverse (Eq. 13). The CMG search then specifies the direction and magnitude of the null motion (\vec{k}_i , Eq. 12) added at each discretized timestep in order to avoid singularities and troublesome gimbal conditions. The

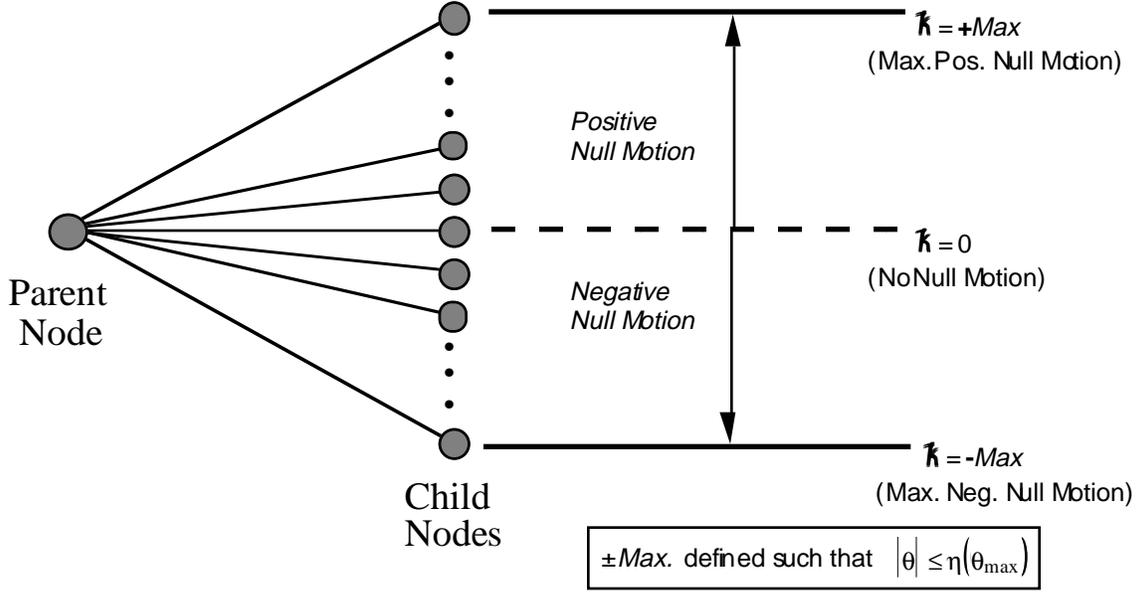


Figure 28: Node Structure for CMG Search

dimension of the nullspace is the number of CMGs minus the degree of the torque/momentum constraint (generally equal to 3). For a 4-CMG system, the search is consequently run over a scalar variable, thus it is simple to implement, and can execute quickly. The 4-CMG situation is depicted in Fig. 27. The vertical axis represents the signed null component (\hat{k}). The search assigns a value to \hat{k} at each discretized point to avoid (or minimize the impact of) singular states, thus the vertical axis represents the free coordinate used to divert away from problematic configurations in gimbal space. Consequently, the feedforward output of the CMG search is the discretized function of applied null motion over time; i.e. $\hat{k}(t_i)$.

The set of child nodes expanded from a parent thus span a range of possible null motion, as depicted in Fig. 28. The center node represents gimbal motion performed between the two timesteps without adding null motion; i.e. $\hat{k} = 0$ in Eq. 12, and the SR-inverse solution is applied without modification. The nodes extending above (and below) the center denote transitions between parent and child momentum states that introduce positive (and negative) null motion via a signed value of \hat{k} . The nodes are generally equally spaced in \hat{k} , and the range in \hat{k} spanned between upper and lower nodes is normalized such that the node corresponding to maximum (or minimum) null motion runs the gimbals to within a preset fraction (η) of their peak hardware rate limits. The dynamic range in gimbal rate accessible to the search is thus set by the number of child nodes per parent.

If the SR-inverse solution exhibits at least one CMG gimballed above its peak rate, and this situation can be alleviated by adding null motion, the $\bar{k} = 0$ node is suppressed (i.e. not made available to the search selection), and only null motion nodes with gimbal rates under peak are allowed. Wherever possible, this policy restricts the search to avoid selecting nodes that violate hardware rate constraints. If the addition of null motion is unable to reduce the worst-case gimbal rate to below peak, we essentially relent, and only the $\bar{k} = 0$ node is made available to the search.

Under these coordinates, the "terminal node" is actually any node at the last timestep (i.e. generally where the CMGs achieve their final commanded momentum). All nodes at the rightmost edge of the tree (as per our convention in the figures) are at the last timestep, hence can be considered "terminal". They all achieve the final momentum command, unless their associated trajectory had become mired in a singular state.

As more CMGs are added, the dimension of the nullspace is likewise incremented, and the problem can become much more complicated; i.e. the amount of nodes (hence storage) needed can rise with the power of the search dimension, which is equal to the number of null degrees of freedom. Fortunately, the SGCMG kinematics again allow a loophole out of this situation. As discussed in Chapter 2, singularities become much easier to avoid as more CMGs are added. Even simple gradient steering techniques begin to function adequately with 5 or more SGCMGs in a system. The search process may not need to work very hard to improve a gradient solution in a larger CMG system, thus the search can be constrained without sacrificing performance; i.e. the null vectors can be searched over independently, resulting in a linear (as opposed to exponential) increase in the number of nodes vs. search dimension. Other methods may also be brought to bear to add further constraints and gain more simplification; i.e. a single-variable search may be performed over only one null vector, unless a seemingly insurmountable singularity is encountered, at which point the search switches to varying another null vector. Additional approaches may readily be defined to exploit specific CMG characteristics and reduce the order of a multi-degree-of-null-freedom CMG search. In general, however, the major challenge in steering SGCMGs resides with the minimally-redundant 4-CMG system, hence this setup is used in all of the examples discussed in the following chapter.

As mentioned above, each node expanded by the search represents a specific CMG gimbal orientation. A numerical "cost" can be associated with each node to express the optimality of the corresponding gimbal position, as quantified below:

$$15) \quad c_n = W_1 (m_{min})_n - W_2 \sum_{j=1}^n \left(\frac{1}{m_j} \right) - W_3 \sum_{j=1}^n |\underline{h}_{res(j)}|^2 - W_4 \sum_{j=1}^n (\Delta \dot{\theta}_{over})_j - W_5 \sum_{j=1}^n J_{null}(j)$$

Where:

m_j = CMG gain (re. Eq. 7) at node j.

m_{min} = Minimum value of CMG gain over path through node j.

$\underline{h}_{res(j)}$ = Momentum residual (commanded vs. delivered) at node j.

$(\Delta \dot{\theta}_{over})_j$ = Net gimbal rate above hardware limit at node j.

$J_{null}(j) = |\zeta^{(j)}| + |\zeta^{(j)} - \zeta^{(j-1)}|$

$\zeta^{(j)}$ = Null motion added at node j.

The objective function given in Eq. 15 is defined to be *maximized* in an optimization; i.e. the search works to maximize the c_n of the terminal node. The summations in Eq. 15 add their arguments back along the tree (tracing parents through the linked list) from node n to the start node. They represent sums over the nodes along the unique path through the tree that terminates at node n, hence are not sums over all expanded nodes. These sums need not be calculated completely anew at every encountered node, but can be recursively updated; i.e. when a new node is expanded, the current value of the summation argument is added to the previous sum stored at its parent node. The various terms in Eq. 15 act to produce specific CMG steering response, and will be discussed individually below.

The first term is proportional to the minimum value of CMG gain over the trajectory through node n (it's essentially the infimum of m). It is generally weighted somewhat heavily, thus the search works extensively to maximize the worst-case gain value, thereby avoiding singular states.

The second term approximates the integral of the inverse gain. Since it's added negatively, the search works to minimize this quantity. It represents the amount of time spent in the vicinity of a singular state; the longer the trajectory lingers at low gain, the higher this penalty becomes. The $(1/m_j)$ that is summed here is clamped by an upper limit of 10 (for $m_j < 0.1$, the singularity is sufficiently severe that differences no longer matter). In addition, if the $(1/m_j)$ falls below 2, it is set to zero; this term is defined to penalize time spent in low-gain states, not reward high-gain trajectories (this is accomplished by the terminal cost relation, Eq. 16).

The third term is the amount of momentum residual accumulated over the trajectory through node n. As mentioned in Chapter 2, the SR-inverse (which calculates the

torque-producing rates) can hang-up in a singular state, causing considerable divergence between commanded and delivered CMG momenta. This term penalizes nodes attached to trajectories that have been significantly displaced from their commanded momentum state. The CMG momentum at node n is calculated from the node's gimbal configuration and subtracted from the commanded value to form the residual, which is recursively accumulated over the trajectory.

The fourth term acts to prevent nodes from being selected that exhibit CMG gimbals running over their maximum allowed rates. Since its weight is set very high, this term is assigned the utmost importance in the optimality criterion. If all CMG gimbals are running below peak hardware limits, $\Delta \dot{\theta}_{over}$ is zero, and no amplitude is contributed to the objective. As the gimbals surpass their peak limits, $\Delta \dot{\theta}_{over}$ is set to the net excess rate, hence the penalty increases the more the rate constraint is violated. The over-rate penalty is also summed recursively across the trajectory, hence represents all excess rates encountered through node n . In general, the use of the SR-inverse and the node expansion policy applied when adding null motion (see discussion of Fig. 28) prevent peak rates from being exceeded. Situations can arise, however, where a high-torque command isn't realized sufficiently by a particular gimbal configuration, causing an over-rate response. This objective contribution acts to forbid such states where possible; otherwise, if a solution with legitimate rates is not found, the search works to minimize the net 1-norm of accumulated excess rate. Of course, when implementing over-rate solutions, the rates are clipped to their hardware maxima, producing momentum errors in subsequent nodes that are likewise penalized by the $|\underline{h}_{res}|$ term.

The final term in Eq. 15 reduces undesirable side-effects of null motion. By itself, the pseudoinverse solution usually generates a very smooth and clean gimbal rate profile, due to its intrinsic 2-norm minimization. Gimbal rates can increase when adding null motion; with a small torque command, the added null motion can often dominate the solution, especially when the null freedom is discretized to a small number of allowed levels, as it is under the node structure imposed by our search protocol (i.e. Fig. 28). If it's not needed to skirt a singularity, one would thus prefer adding little or no null motion. This is encouraged by the first term in the expression for J_{null} (i.e. the sum of null amplitudes across the trajectory). Even potentially worse than large unneeded null motion is a null policy that often switches the null vector rapidly on & off, or continually ramps it between positive and negative extremes. These solutions will produce appreciable chatter in the system that can vibrate the spacecraft, consume power, and perhaps eventually degrade the CMG hardware. These considerations are addressed by the second term in the expression for J_{null} . This term represents the magnitude of the difference between null

motion added at successive nodes of the trajectory (i.e. a discrete time derivative of the null motion factor $\dot{k}(t)$). This term acts to penalize trajectories exhibiting high-frequency gimbal chatter and prefers solutions with more constant, steady gimbal motion.

The search software is capable of adding another term into Eq. 15 that is proportional to the value of m calculated at node n . This is essentially an "instantaneous" cost, since it has no "memory" of its parent trajectory, as the other terms do. This contribution is only added when the search is propagating a local depth-first trajectory, and is omitted when calculating the terminal cost (as discussed below), or scanning for open nodes upon which to graft a new trajectory. This term does indeed cause the depth-first trajectories to follow the local gradient, but was seen to generally impede convergence to a superior global solution in the test examples, thus its weight is set at zero and it is not included in the objective calculation (hence it's not listed in Eq. 15).

When the search propagates a trajectory through to the terminal momentum state, a special "terminal cost" is calculated:

$$16) \quad c^t = c_n + W_6 \left(\frac{1}{N} \right) \sum_{j=1}^n (m_j)$$

Here, c_n is the conventional cost of node n (which is terminal), as calculated from Eq. 15. The summation is defined recursively over the trajectory (as with Eq. 15), and its argument m_j is the CMG gain. After dividing by the number of trajectory steps (N), a mean CMG gain is calculated. Adding this term into the cost definition, trajectories with higher average gain tend to be favored in an objective maximization. The terms in the node costs (Eq. 15) worked mainly to penalize low-gain states, and not explicitly reward trajectories with high average gain. The mean gain is accounted for here; after a trajectory has been propagated to the terminal state, the terminal cost is calculated via Eq. 16, and compared to the terminal cost of the previous "best" trajectory. If it is found to be better (i.e. larger), this new trajectory inherits the "best" status. This process was illustrated in the logic of Fig. 26.

The conventional cost (Eq. 15) of the terminal node on the "best trajectory" is also preserved as the "cost cutoff", which is used to suppress exploration of inferior paths, as discussed in Sec. 3.2. As the search progresses, better paths are discovered, hence this cutoff becomes more stringent, and the search grows more efficient.

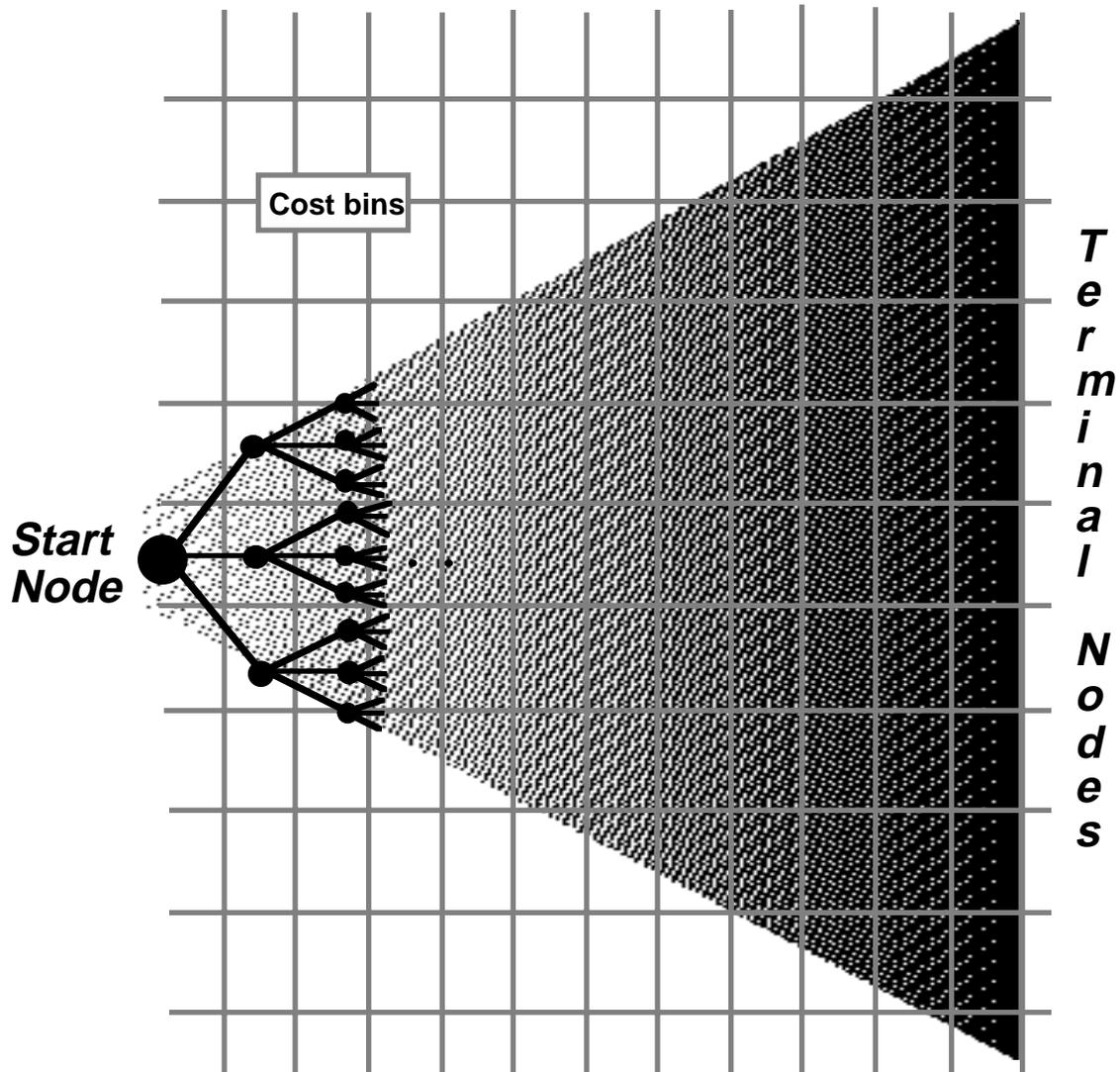


Figure 29: Example of Tree Segmentation for the Grid Cost Calculation

Optimizations of this sort can fall prey to local minima; i.e. in some cases, the search can rapidly converge to the neighborhood of a quasi-optimal trajectory and devote all of its effort into expanding nodes in the same region of the tree, potentially missing better solutions that are far removed. In order to circumvent this myopia, another cost amplitude has been developed to augment the objective. A 2-dimensional histogram is maintained that counts the number of times the search scans all open nodes and returns to start a path from discretized zones in the search tree. This array is addressed via two indices. One of these coordinates is simply the timestep (i.e. tree level or horizontal [x] axis in the figures). The other coordinate is the integer value of the integrated null motion coordinate, with the

displacement added at each timestep first scaled by the number of children per parent node (vertical axis [y] in the figures). Each time the search starts a path (i.e. grafts onto an open node) from a point in the tree, the histogram bin addressed by its x and y coordinates is incremented. A qualitative depiction of histogram quantization of x and y tree positions is given in Fig. 29 for a single degree-of-freedom search with 3 children per node.

When searching all open nodes to select the node on which to graft a new search path, a weighted value of the histogram contents (addressed at the node in question) is subtracted from the objective. If the search has frequently revisited a localized region of the tree, its addressed histogram bin will contain a large value, hence nodes in this neighborhood will have inferior cost, and the search will be encouraged to graft a new path elsewhere. Ideally, a cost of this sort will generate a search strategy that spreads its trials across the tree. The logarithmic dependence of the y-coordinate scale on the negative x-coordinate, however, will compress a large number of nodes into single bins at the rightmost regions of the tree, where the node density is very high (the net number of nodes at any timestep is the number of child nodes per parent raised to the power of the tree level [or x-coordinate]). As a result, this cost contribution has its greatest discrimination at the lower levels of the tree, before the branching density grows too large.

As the search progresses, it should be allowed to narrow its scrutiny down to a more local region, thus this cost amplitude is exponentially reduced each time the list of open nodes is examined. In summary:

$$17) \quad c_n^0 = c_n - \alpha W_G M(IX, IY)$$

Where:

c_n^0 = Node cost used when testing all open nodes for new graft point.

c_n = Node cost per Eq. 15.

α = Exponentially decreasing weight. Initially $\alpha = 1$, and after each node scan,
 $\alpha \leftarrow \gamma \alpha$, where $\gamma < 1$ is the attenuation factor.

W_G = Static cost amplitude.

M = Tree visitation "grid" histogram.

IX = Tree level (timestep or horizontal coordinate in Figs.).

IY = Logarithmically scaled integral of node index in the null motion (vertical coordinate).

The definition of a predictive heuristic function for a CMG system is somewhat difficult. Known characteristics of particular SGCMG arrays could be loaded into a large

lookup table, and consulted online to construct values for h_j . This has not been attempted in the current study; the actual h_j in Eq. 14 is essentially zero, and the node cost is defined through Eqs. 15-17. An effective h_j function, however, is dynamically "learned" via the adjustment of the cost cutoff, which accomplishes the duty of a heuristic in pruning fruitless search paths; i.e. the search doesn't waste resources exploring a path if its cost drops below that of a trajectory that has previously made it to the terminal state. The search thus effectively learns its "cutoff" heuristic; initially it is set very low, but as trajectories are generated, it quickly rises and becomes more discriminating. In order to set this cutoff appropriately and gain a base of open nodes that may be grafted upon and expanded, several guided depth-first trial trajectories are run to the terminal state before the search itself is begun. The first three are quite straightforward; one with no null motion (only applying the SR-inverse), one with null motion set to its positive limit, and another with null motion set to its negative limit. If more than three children are allowed per node, other trial trajectories are performed, with null motion set to the same intermediate level at each step (the total number of such trials performed is equal to the number of children per node). The next trial is a straightforward gradient approach; i.e. the best-cost child node is always selected for expansion. The final trial is a discrete version of the "indirect avoidance" method of Ref. [34]. If all child nodes project all their CMG rotors into the hemisphere of the torque command, the best-cost child node is selected for expansion, as in the previous gradient scheme. If, however, child nodes exist with at least one CMG rotor projecting negatively onto the torque request, the node exhibiting gimbal positions that yield the smallest magnitude of net negative rotor projection is selected for expansion. In this fashion, the rotors are "unkinked" and move into the hemisphere of the torque command, thereby disfavoring standard singular CMG states with antiparallel rotor alignments (as discussed in Chapter 2).

After these trial trajectories are completed, the search begins in earnest to modify and improve on them. Additional measures can cut the complexity of the search process to increase execution speed and reduce needed node storage. One such technique is to reduce the depth of the tree, thereby exponentially reducing the nodal density at the terminal state. The CMG search doesn't necessarily require a high bandwidth on null motion changes; the CMG system can be forward integrated at a significantly finer level than the tree's horizontal node spacing, thus search decisions are made at intervals that are several integration steps long. By matching the horizontal node spacing to the needed speed of null motion response, the intermediate gimbal states, thus the density of the search tree, can be reduced appreciably, and near real-time applications become more feasible.

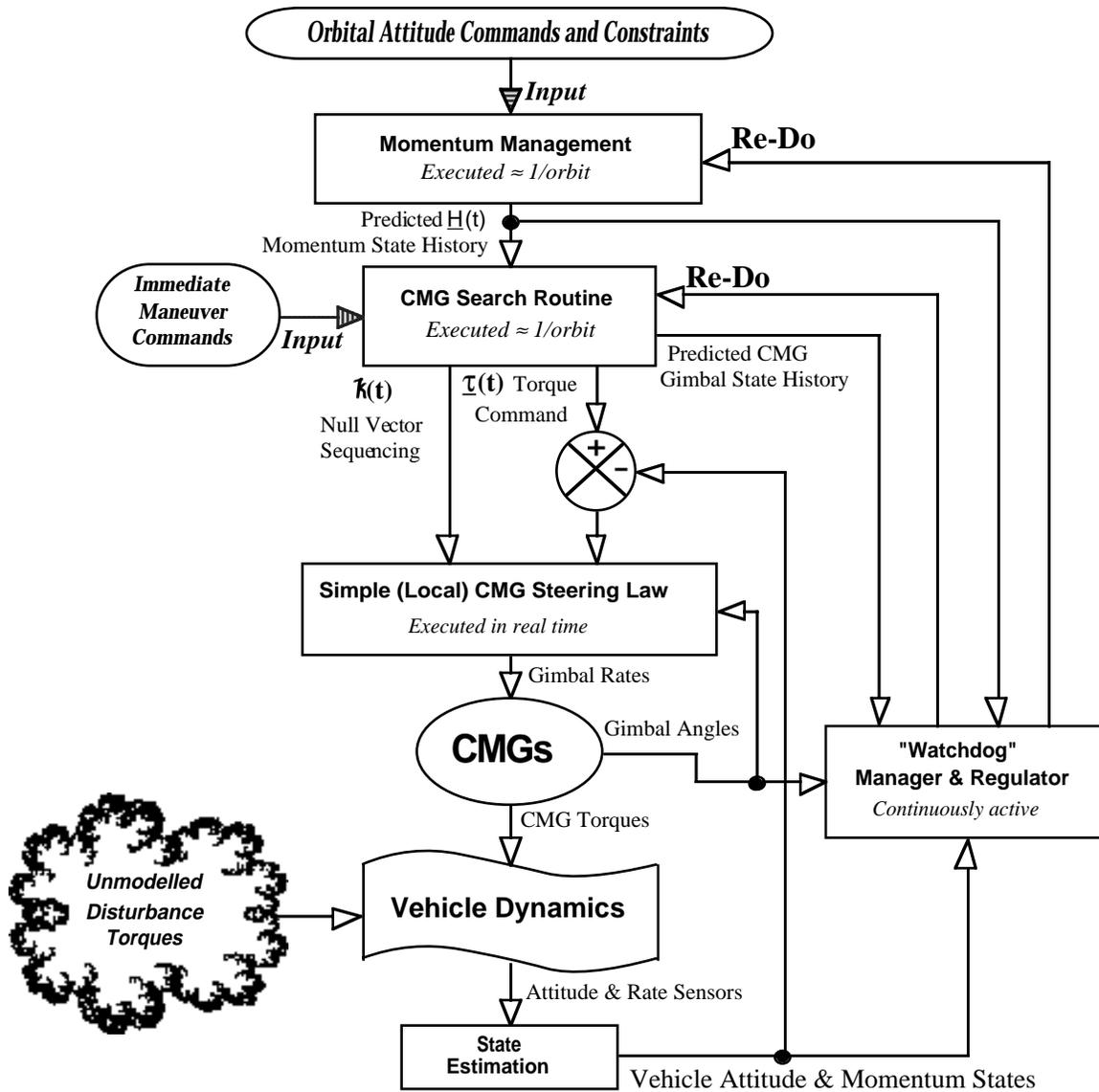


Figure 30: Logic for Orbital Implementation of Search-Based CMG Steering

3.4) Flight Implementation

A diagram is given in Fig. 30 to illustrate the on-orbit application of the search-based CMG steering approach. In most respects, Fig. 30 is a more detailed CMG-specific adaptation of the design that was introduced in Fig. 20. A forecast of the anticipated CMG-stored momentum (i.e. a 90-minute orbit-to-orbit prediction) is assumed to be produced by an onboard momentum manager. The CMG search routine is executed

to find a feedforward gimbal trajectory that answers this momentum profile. Ideally, the search is only needed to run once per orbit, and more often (as required) if significant unmodeled disturbance torques are present. The search uses its allotted time to globally perfect gimbal trajectories; since the depth-first approach is used, the search may be halted at any time, and the best calculated solution put into service.

The outputs of the search are torque command $\underline{\tau}(t)$ and null vector scaling $\bar{k}(t)$ histories that are implemented by a simple CMG steering law, running in a standard high-rate real-time loop. This is a tangent steering law (circa Eq. 12), that calculates both the SR-inverse (for $\underline{\theta}_t$ in response to the input torque command), and null vector(s) from the currently measured CMG gimbal state. Its input torque command is formed by compensating the current value of scheduled torque output from the CMG search and momentum management routines with an estimate of any disturbance via a loop closed around the vehicle dynamics (needed to force the spacecraft to track the commanded state). In the ideal case (without any disturbance), the vehicle accurately follows the prescribed trajectory, and the simple steering law drives the CMGs exactly as anticipated in the CMG search.

The singularity-avoiding output of the CMG search is a discrete time history of the null vector scaling $\bar{k}(t)$. As the low-level steering law answers the commanded torque via the SR-inverse, it also calculates a null motion vector $\dot{\underline{\theta}}_N$ from the estimated gimbal state at each iteration. Instead of weighting the null motion locally, as performed for singularity avoidance in local steering laws, the value of $\bar{k}(t)$ is taken from the search output (at the current time "t", and interpolated from the coarser timesteps used in the search). In this fashion, the steering law uses the global search information to manage null motion and evade singular conditions.

The local nature of the steering law (i.e. calculation of torque and null solutions relative to the instantaneous gimbal state, together with feedback around the vehicle dynamics) always drives the system to track the input torque command. The fed-forward null amplitude $\bar{k}(t)$, however, is calculated by the search, relative to a predicted gimbal angle trajectory. If the actual gimbal trajectory, as managed by the local steering law, diverges appreciably from the predicted path used in the search (i.e. due to unmodeled disturbances), the fed-forward null amplitude no longer has any relevance, and the CMG system will not necessarily produce the anticipated singularity characteristics. Situations of this sort will be detected by the watchdog regulator task, which compares the actual gimbal angles with their predictions. If significant divergence is detected, several provisions may be taken.

The simplest remedy is merely to re-start the search from the current gimbal position, preferably with an updated disturbance model. If one is running at typical momentum management rates (i.e. 90 minute period), such a technique may well be feasible. Otherwise, if a more rapid response is called for (as in the midst of a tight maneuver), other methods could be pursued. The easiest would be to revert to a tangent technique for manipulating the null motion; since the feedforward $\bar{k}(t)$ is no longer valid for the current gimbal configuration, a tangent null technique, such as those described earlier, may deliver more predictable, albeit limited, performance. As discussed in the previous chapter, steering laws can be attracted toward, and lock into, nearby singular states, particularly when driven around a feedback regulator that tries to null disturbance and off-axis torques. In cases where a singularity is closely skirted (i.e. the gain drops low, circa below 0.8), the system may be prevented from being driven completely singular and locking up by disabling the feedback and driving the CMGs open-loop while in the singular region. In this case, the singular state can be quickly transited, and feedback control re-established (hence nulling any accumulated errors) when the gain again rises on the other side of the singularity. Simulations have shown this technique to often avoid gimbal lock, although at the potential expense of introducing some momentum error.

Several observations on effects of modeling error are drawn from the simulation examples presented in the next chapter. In general, it has been noted that the potential for gimbal divergence is most acute if the system is forced to closely skirt or transit a singular state (i.e. the search was unable to find a null path around it, so settled for the best path through). The gimbal trajectory predictions become much less reliable after the trajectory goes singular or nears a singularity. Because of the nature of such nonlinear dynamics, there is no real means to predictively "see" through a singular state, where several gimbal closures (thus possible trajectories) are joined.

This actually aids our cause somewhat. The locations of probable rapid divergence are known ahead of time to be at the gimbal states where the search had indicated proximity to a singularity. This allows some contingencies to be taken; i.e. the gimbals can be constrained to follow a particular path when traversing the singular state such that the gimbal state is pre-known both before and after the singularity is crossed. The results of two feedforward searches can then be applied; one leading up to the singular state, and another from the opposite side of the singularity onward. This discussion will be elaborated in the next chapter.

If one removes the CMG search block from the implementation diagram of Fig. 30, the logic is essentially identical to that of a standard orbital autopilot. The watchdog regulator is still needed, but it now monitors the momentum state of the vehicle and

re-queues the momentum manager upon discovering significant divergence from prediction. The logic of Fig. 30 pulls aspects of the CMG steering (i.e. the nonlinear optimization) up to the level of momentum management, where we now calculate projected gimbal trajectories, as well as momentum storage.

Details on the current software implementation are given in the next chapter on simulation results. Execution requirements seem encouraging for application of such schemes on future spacecraft. The simulations showed promising results when limiting the

Variable Name	Definition	Min. Storage (Bytes)
Gain Value	Current value of CMG Gain	1
Min. Gain	Minimum of CMG gain across Trajectory	1
Parent Pointer	Pointer to parent node	2
Gimbal Angles	Current Gimbal Angles (4 total)	4 x 1 ¹ / ₂ = 6
Open Flag	Flag to denote open status of node	1/2
Step #	Search step at this node	1
Null Value	Null motion (\vec{k}) added at this node	1 ¹ / ₂
# Children	Number of Children attached to this node	1/2
Saturation Value	Fraction of total CMG momentum used	1
Integral Cost	integrated inverse gain at this node	1 ¹ / ₂
X Position	X-position in search tree (for plots)	2
Residual	Integrated Momentum residual at this node	1
Inull	Integer null motion index (\pm # children)	1/2
IntDsp	Integrated null motion (signed)	1
OvrRte	Net Gimbal rates over limit	1
Null Sum	Integrated absolute null motion	1
lpat	Sign pattern of gimbal angle motion	1/2
GnInt	Average CMG gain across trajectory	1

Table 1: Node Variables and Minimum Storage Requirements

number of expanded nodes to under 6000 (2000 search steps, each expanding 3 child nodes per parent). The nodal data structure used in the test software is listed in Table 1. A set of values for these parameters is defined at each node that has been opened. The current test software reserves a full 32-bit machine word for each of these values, which is somewhat wasteful. An estimate of the minimum bit field (rounded to 4-bit increments)

needed to manage a 6000 node search and calculate the objective per Eqs. 15-17 with sufficient accuracy has been listed in the right column. These projections indicate that the total bytes/node needed sums to 24, although at least 3 of these quantities (4 bytes) aren't really required to be stored (the Saturation Value, X Position, and Intdsp), as they are mainly used in making plots (one can become even more aggressive and eliminate other variables, although at some cost in calculation). Assuming, then, that a data structure of roughly 20 bytes per node is adequate for managing the search, a 6000-node search operation can be performed in roughly 120 Kbytes (which can be released to other applications after the search concludes).

On a Macintosh II computer (68020/68881 processor) running extremely inefficient, diagnostically-oriented, user-friendly software, the search was seen to usually approach convergence to its best solution within 1-2 minutes. A large gain in speed can easily be realized through tighter coding of the search algorithm. The search can be additionally hastened through application of parallel, dedicated processors, such as are now being proposed for eventual use in space-based robotics[58], resulting in execution times of seconds or less, and enabling near real-time operation.

4) Simulation Examples

4.1) Simulation Tools

In order to gain experience with the application of a directed search to SGCMG steering, an interactive simulation package has been assembled for the Macintosh II computer. Three major programs have been written. A trajectory specification routine allows the user to interactively define commanded momentum histories (i.e. as would arise from a momentum manager or maneuver scheduler). The user can draw momentum trajectories interactively on the screen with the mouse, or manually type in 3-momentum values at specified key points (the program interpolates linearly between them). Another program performs the CMG search about these momentum commands to locate satisfactory gimbal trajectories, which are analyzed in the final program for sensitivity to unmodeled bias torques. Details on software implementation are given in the appendix.

The CMG mounting scheme used in this study was depicted in Fig. 9. It is a conventional "pyramid mount", where four CMGs are constrained to gimbal on the faces of a regular pyramid (the gimbal axes $\hat{\sigma}$ are orthogonal to the pyramid faces). Each face is inclined at 54.7° to the horizontal (thus gimbal axes are at the complement angle; i.e. 35.3°), yielding a momentum envelope (Fig. 11) that is roughly spherical for a 4-CMG array; i.e. the authority along the vertical (\hat{z}) axis is similar to the authority that can be projected along \hat{x} and \hat{y} . The minimally-redundant 4-SGCMG configuration encounters the most difficulty with singular states, thus presents a significant challenge for CMG steering laws, and is used exclusively in these tests. Most examples assume that the CMGs start with initial gimbal angles of zero (as seen in Fig. 9), except where explicitly noted. The CMGs are all defined to have unit momentum, and a peak gimbal rate of 1 radian/sec. No gimbal stops are imposed (although they can be directly accommodated under the search framework).

The momentum command profile is specified at 30 discrete time-steps in these examples (although other timestep spacings can be defined in the trajectory generation program and automatically accommodated in the search and disturbance routines). The search integrates its gimbal trajectory (i.e. calculates the SR inverse and null motion) twice per step (a linear interpolation is performed to effectively double the number of commanded points). At each point, the momentum residual from the previous integration is added into the new command, forcing the CMGs to track the input momentum profile (in an actual spacecraft implementation, this could be replaced by something like a PID controller to maintain commanded attitude rather than commanded momentum). Search decisions are made after every four integration steps (thus any trajectory is 15 search nodes deep), although nodes are produced (that can be later "grafted upon" and expanded) every two integration steps. All of the above search parameters and CMG definitions can be readily varied interactively, as detailed in the Appendix.

The examples shown here allow only 3 children per parent node (i.e. negative null motion, no null motion, or positive null motion). Although the test software is able to employ an arbitrary branching density, this "ternary divert" strategy was seen to perform adequately, and minimizes the search complexity and execution requirement. The null motion scaling (η in Fig. 28) was chosen such that at least one CMG was driven at 70% of its peak gimbal rate in nodes with null motion added (in nodes without null motion, the rates are produced by the SR-inverse alone). Referring to Eq. 15, the objective weights used in this study were $W_1 = 20$, $W_2 = 3$, $W_3 = 2$, $W_4 = 100$ (sec/rad), and $W_5 = 0.05$. The weight on the average gain cost (Eq. 16) was $W_6 = 1.8$. This mixture of weights was empirically arrived at; after many simulations, it was seen to generally produce good trajectories fairly quickly. The highest weight is on the gimbal overrate contribution, thus trajectories exceeding gimbal rate limits are emphatically avoided. The next most severe cost is on the trajectory's minimum gain value; the search initially works quite hard to bring this value up to a reasonable level, away from the clutch of a singular state. Solutions are then selected for higher average gain, low residuals, lower null motion addition/changes, etc., as discussed in Sec. 3.3. The grid cost coefficients (Eq. 17) were set at $W_G = 5.$, $\alpha = 0.95$ (causing a fairly fast decay of the grid cost contribution).

If the CMG gain m exceeds 1, the weight ρ on the identity term of the SR-inverse (Eq. 13) was set to zero, producing a pseudoinverse calculation. As a singularity is approached, however, and m drops below unity, ρ is set to $0.1/m$, and is not allowed to grow beyond a ceiling of 0.2 (these settings are similar to those used in the work of Refs. [13,52]). These weights, thresholds, and ceilings can readily be changed during execution

of the CMG search routine, but the values quoted above are used in all of the tests shown here.

The program was limited to 2000 search steps (i.e. parent-to-child expansions) or to producing 10,000 nodes, whichever came first (at 3 children per parent, the 2000-step limit created only 6000 nodes, thus these examples never encountered the 10,000-node limit). This produces fairly modest search requirements; with data packing it should be possible to run within roughly 120 Kbytes of node storage (as mentioned in Sec. 3.4). The search is seen to execute quite quickly (coded in Absoft FORTRAN 2.4 [Ref. 59] on the Macintosh II). Considerable trajectory improvement is usually seen within a score or two of search operations (i.e. under a minute of execution with the inefficient test code, which also generates graphics and writes data files while conducting the search). A full 2000-step search (which is generally not needed to produce workable trajectories) will complete within 5-10 minutes. Tightly coded software can easily speed the execution considerably, possibly enabling real-time operation.

In analyzing the performance of CMG trajectories, it is important to know whether the commanded trajectory approaches momentum saturation. In general, a CMG saturation index may be defined as the magnitude ratio of the current stored momentum \underline{h}_s to the maximum momentum possible to project along the direction of \underline{h}_s . This may be described as $s = |\underline{h}_s| / |\underline{h}_m|$, where \underline{h}_s is the sum of current CMG momenta and \underline{h}_m is maximum momentum that the CMG array can project along the \underline{h}_s direction. For double gimballed CMGs without restrictive gimbal stops, s is very easy to calculate, since \underline{h}_s is a simple vector sum and $|\underline{h}_m|$ is just the scalar sum of rotormomenta. For single gimballed CMGs, however, the situation is somewhat more complicated, since the rotors can not generally be precisely aligned with any orientation of \underline{h}_s . This gives rise to the dimples on the momentum envelope surface, illustrated in Figs. 11 & 12, and discussed in Sec. 2.3.

A method for approximating s has been derived in Ref. [17] for single gimballed CMGs. Since the SGCMGs can not always align their rotors precisely along the desired direction, the closest possible alignment is used in calculating their contribution to the momentum envelope. When oriented in this fashion, SGCMGs can produce a finite momentum component perpendicular to the desired final state. This must also be absorbed by the CMG system, and is accounted for by being subtracted in quadrature from the projected momentum envelope. The calculation of the effective $|\underline{h}_m|$ may thus be detailed:

$$18) \quad |\underline{h}_m| = \sqrt{(\underline{h}_p \cdot \hat{h}_s)^2 - |\underline{h}_p \times \hat{h}_s|^2}$$

Where:
$$\underline{h}_p = \sum_{j=1}^N \underline{h}_{p(j)} \quad (\text{sum of projected momenta})$$

$$\underline{h}_{p(j)} = |\underline{h}_j| \mathit{unit} \left[\underline{h}_s - \left(\hat{\sigma}_j \cdot \underline{h}_s \right) \hat{\sigma}_j \right]$$

The vector $\underline{h}_{p(j)}$ represents the orientation of CMG rotor #j that has the maximum projection onto the desired final state \underline{h}_s . The value of $|\underline{h}_m|$ calculated in Eq. 18 is used to form the ratio with $|\underline{h}_s|$ and thus calculate the saturation index s . This quantity is used by the search to detect proximity to the momentum envelope; if s rises beyond 0.95 when a node is expanded, the momentum envelope is assumed to be nearby, and the CMG gain value m of the parent node is adopted so the search will not confuse the (unavoidable) momentum saturation with an (avoidable) internal singular state, and thus include it in the optimization.

A method has been often discussed in the literature (Refs. [12,13,33,48,49,50]) through which a singular state can be tested to determine whether it is escapable or inescapable via null motion. Omitting the derivation, which is given in all of the above references, the basic technique involves testing the quadratic form:

$$19) \quad [Q] = [N]^T [P] [N]$$

$$\text{Where: } [P] = \text{diag}(\underline{h}_i^s \cdot \underline{u}), \quad i = 1, 2, \dots, N$$

In the case of a singular 4-SGCMG system, $[Q]$ will be a 2x2 matrix, thus will produce two eigenvalues. If both are positive, $[Q]$ is positive definite, thus the singularity is classed "elliptic", and inescapable via null motion. Otherwise, null motion should theoretically be able to relieve the singular condition (other inescapable degenerate conditions may be possible if $[Q]$ is semidefinite, as discussed in [50], but these shouldn't occur in a 4-SGCMG system with equal rotor momenta). The matrix $[N]$ is the nullspace basis, and, in the vicinity of a singularity, it is composed of two 4-vectors. These vectors may be found through a singular value decomposition of the augmented CMG Jacobian (retaining the two right row vectors corresponding to the two smallest singular values as the null basis[51]). The vector \underline{u} denotes the singular direction, and it can be approximated by taking the mean orientation of cross products between the CMG Jacobian's columns. The vectors \underline{h}_i^s are merely the CMG momentum vectors \underline{h}_i calculated at the singular state.

The two eigenvalues of $[Q]$ can be plotted across candidate gimbal trajectories to yield additional information on singular states that are encountered. Since the ingredients $[\mathbf{N}]$ and \mathbf{u} are meaningless away from a singular condition, the eigenvalues are "windowed" on the plots by scaling by $(1 - m)^2$ [where if $m > 1 \Rightarrow m \equiv 1$]. In this fashion, the plotted eigenvalues are forced to zero away from a singularity (where they are ill-defined), and allowed to take finite values near a singular state, where their polarities and magnitudes attain meaning.

A singularity analysis is presented with examples that encounter singular states. This is a set of plots that include the eigenvalues mentioned above, a projection of the torque command onto the singular direction (to keep the plotted projection close to zero away from a singular region, where it has no definition, it is also scaled by the $(1 - m)^2$ factor used above), and a sum rotor orientations that determines how rotor vectors are mutually "kinked"; i.e. whether the system is in a 0H, 2H, or 4H state.

Topographic plots of CMG gain m taken around a specific gimbal trajectory are one of the major tools used in summarizing the operation and performance of the CMG search. These plots are frequently used here, and can be initially somewhat confusing, thus an explanation is provided below, before the start of the next section. Figure 31 shows such a topographic plot, with a legend that maps the shading level to a range in the CMG gain. This shading thus indicates the quality of the CMG state that is derived by varying the amount of null motion added to the SR inverse solution (i.e. sweeping the value of \bar{k} in Eq. 12). The darker the underlying shade, the better the corresponding CMG state (i.e. the darkness of the shading is discretely proportional to the CMG gain m). At each vertical step, these plots show the gain of CMG states that will be produced from the trajectory's previous CMG state (i.e. at the parent node below) by answering the current torque command with the SR-inverse, while sweeping null motion (\bar{k}) up to the maximum allowed amount in both positive (right) and negative (left) directions. The sampling on these plots is somewhat coarse (thus the jagged blocks), but still adequate to glean a good representation of the trajectory's performance. These plots are similar in nature to the topographic projections labeled $\mathcal{L}(\mathbf{H})$ that appeared in Ref. [35].

The overlaid curve follows the chosen node-to-node trajectory. When the trajectory is displaced at left or right of center, null motion is being added into the solution (center is no null motion, left is negative null, right is positive null). The example of Fig. 31 allowed up to 5 values of \bar{k} to be specified (i.e. there were 5 children per parent node), thus the overlaid trajectory is capable of attaining 5 positions along the horizontal axis (i.e. two at left, two at right, and center). Although the CMG search routine can be specified to allow an arbitrary number of children per parent node (and this is automatically reflected in plots

such as Fig. 31), the examples shown here used only three nodes per parent (as mentioned earlier), thus the topographic plots shown in this section will have overlaid curves that can attain only 3 horizontal values; i.e. left edge, center, and right edge. The trajectory starts at the initial CMG state at the bottom of the plot, and achieves the final commanded momentum state (provided that the system wasn't hung up enroute in a singularity) at the top edge. The character string printed at lower right summarizes the node-to-node progression of the overlaid trajectory. A "0" denotes a step taken without adding any null motion. A "+" or "-" denotes maximum null motion added in the corresponding direction, while ">" or "<" denote the addition of intermediate null motion values. The filename from which the momentum command was taken is printed at lower left.

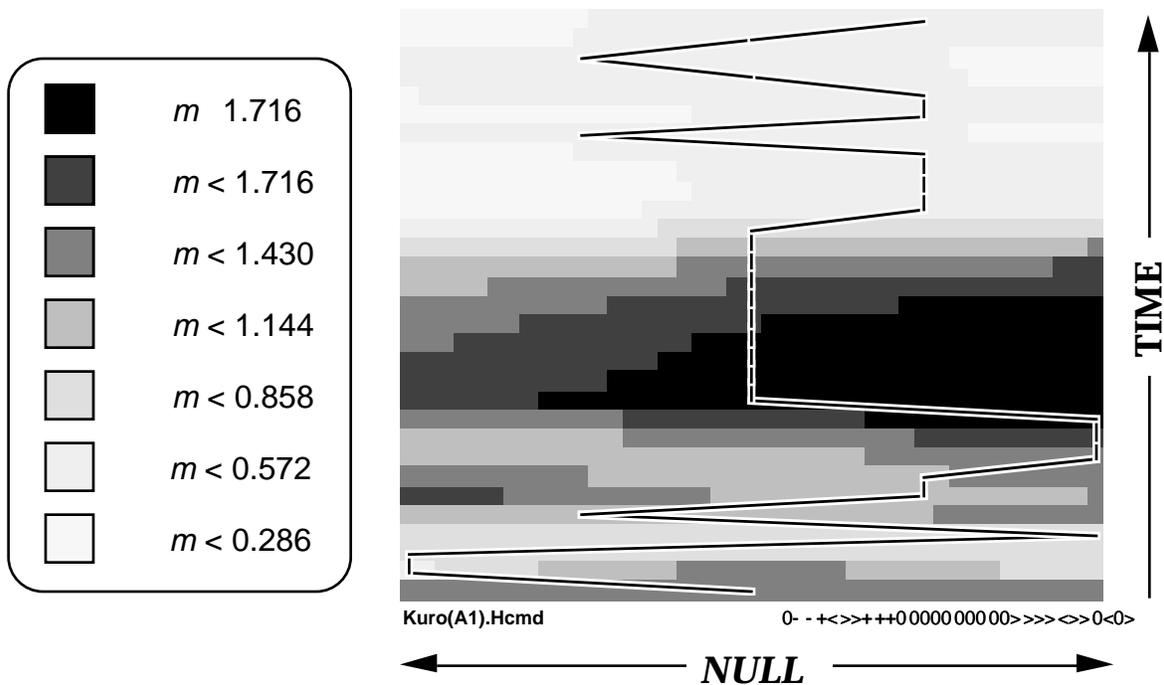


Figure 31: Key to Topographic CMG Gain Plots

One must bear in mind that these topographic maps are not taken over simple Euclidean planes (as is the case with analogous plots shown in trajectory optimization applications[4]), but are 2-dimensional slices projected along nonlinear paths (i.e. around the plotted gimbal trajectory) in a 4-dimensional gimbal space. As such, they can be counterintuitive. For example, in a Euclidean plane, a trajectory that starts out at the center, then moves left, then an equal amount right before returning to center again will wind up at exactly the same end state as a trajectory that stayed at center all the way through. This is not necessarily so here, as the gimbal configuration attained by pulsing null motion back

gradient on these plots, but generally had trouble locating superior solutions. As mentioned in Chapter 2, such gradient-based steering laws can have significant difficulty overcoming local minima. Using the objective of Eq. 15, solutions were seen to initially raise the minimum m encountered on the trajectory (thus avoid the singularity), then work on raising the value of m elsewhere (as encouraged by the integral inverse gain term in Eq. 15 and the terminal cost of Eq. 16), usually achieving much better results than with the objective driven by ∇m .

The examples shown here do not consider a spacecraft model and/or equations of motion (circa Eqs. 3,4). The feedforward angular momentum profile output from a momentum manager (here substituted by the trajectory definition routine) is assumed to already include orbital effects (such as Euler coupling), thus the search and disturbance routines produce CMG gimbal motion that follows these momentum trajectories directly, and do not need to consider spacecraft dynamics. The spacecraft reaction to the resulting CMG momentum history can be calculated by Eqs. 3 and 4.

The tests summarized in the following section show the CMG search performance for a variety of momentum command sequences which are known to encounter singular states. These gimbal trajectories are examined for their sensitivity to unmodeled secular torques in the final section of this chapter.

4.2) CMG Search Examples

In this section, the performance of the CMG search will be exhibited and analyzed through several simulation examples. The first example to be investigated employs a constant torque command about the \hat{x} axis. Such cases have been studied previously [11,13,49,52,55] for the 4-SGCMG pyramid, and several existing steering laws can encounter difficulty with the unescapable singular state that lurks at a net CMG momentum of $h_x = 1.15$.

The momentum command sequence is plotted in Fig. 33a. The solid ramp is the commanded linear increase in \hat{x} CMG momentum (realized by integrating the constant torque), while the dotted line along the horizontal axes indicates that the \hat{y} and \hat{z} momenta are to remain zero. The plot in Fig. 33b shows the saturation index (Eq. 18) for a CMG trajectory that attained the commanded final-state momentum. This command sequence uses slightly over half of the available CMG momentum, and finishes far from the 100% saturation limit (depicted by the dotted line).

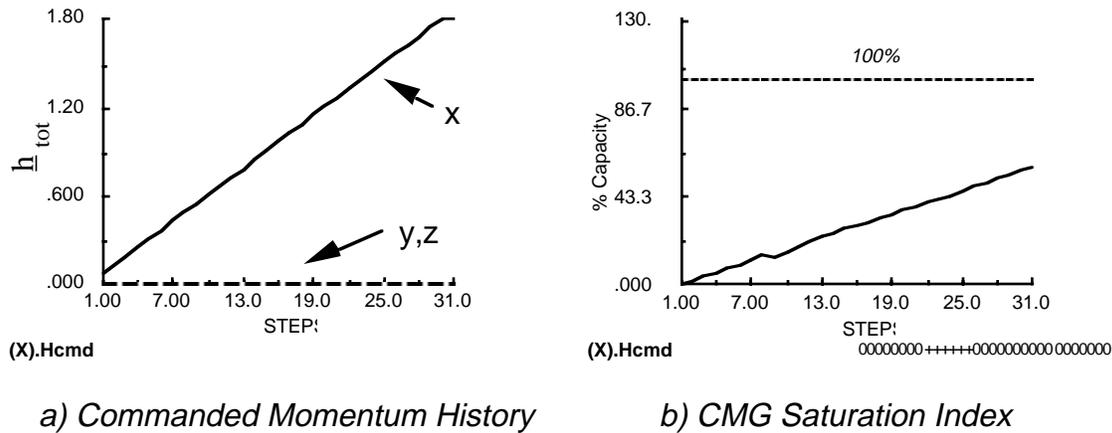


Figure 33: Momentum Command & Saturation Index for Constant \hat{x} Torque

The initial trial gimbal solution pursued by the search startup routine uses no null motion, and applies only the SR-inverse to calculate gimbal rates. The singularity mentioned above is readily encountered at $h_x = 1.15$, as expected. Fig. 34 presents an analysis of this trajectory and singular state, using some of the quantities that were introduced in the previous section. Fig. 34a plots both eigenvalues of the Q matrix defined in Eq. 19 (scaled by an inverse function of the CMG gain such that significant values are plotted only in singular regions). As the singular state develops (beyond step 13), both eigenvalues are seen to be positive, indicating a definite Q , hence an "elliptic" singular state

that is inescapable via null motion. Since the singular direction is totally aligned with the momentum (torque) command, as seen in the rise to unity of the projection plot in Fig. 34b, the SR-inverse is unable to produce gimbal rates of significant magnitude after the singularity is approached, hence the system remains "locked" in the singular configuration (some small rates are created, leading to the changes in the eigenvalues of Fig. 34a, but the system essentially stays singular).

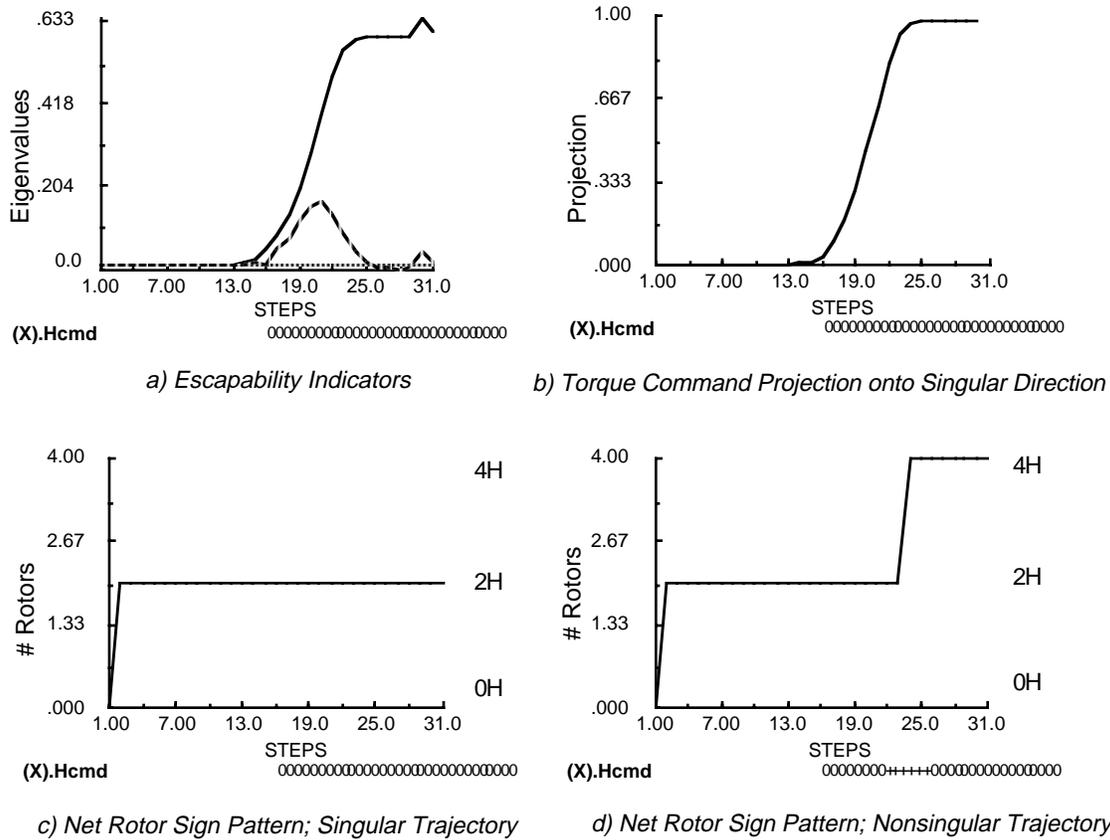


Figure 34: Singularity Analysis for Constant \hat{x} Torque

The bottom two plots of Fig. 34 show the sum of the signs of rotor projections onto the net CMG momentum, yielding an insight into how "kinked" the CMG linkage has become at each trajectory step. Using the terminology of Chapter 2, a sum of 0 yields a 2-against-2 "OH" state, a sum of 1 yields a 3-against-1 "2H" state, while a sum of 4 indicates all rotors pointing into the same hemisphere (a "4H" state). The plot at left (Fig. 34c) shows the sign sum for the SR-only trajectory; one can see that the singularity is a "2H" rotor configuration, as expected for most elliptic states. The plot at right (Fig. 34d) shows the sign sum for a trajectory that successfully achieves the terminal momentum state. One can see that the CMG system transits the "2H" configuration

(through a nonsingular state, as will be seen in the later plots), and finishes with a mutually directed "4H" rotor arrangement.

Figs. 35-38 depict how the "currently best" trajectory adopted by the search evolves as the search progresses. Each plot corresponds to the "best" trajectory found thus far; the results of the initial "trial" with no null motion added (a) are shown in the upper left corners, and the other solutions adopted as "best" are shown sequentially as they usurp their predecessor. The terminal cost of the corresponding trajectory (Eq. 16) and the number of nodes currently expanded are listed in the captions of each plot.

Fig. 35 shows the topographic gain plots, as were discussed in Sec. 4.1. Plot (a) is the initial trial consisting of the SR inverse only, with no added null motion. Thirty nodes were expanded at this trial, and an objective evaluation (Eq. 16) of -14.2 was achieved. The gimbal angle plot (Fig. 37a) indicates that the commanded \hat{x} torque was attained by "scissoring" two opposing gimbals, while leaving the other two nearly unaltered. The performance was quite poor, as seen from the plot of the CMG gain (Fig. 36a); a hard singular condition was held from timestep 22 onward (this is the unescapable singularity at $h_x = 1.15$). Since the singular direction is aligned with the commanded torque (as was shown in Fig. 34a), the SR inverse produces no gimbal rates after the singularity is achieved, and sizeable momentum errors are accumulated. This is seen in the plot of net CMG momentum (Fig. 38a), which shows the \hat{x} momentum plateauing before the terminal momentum state is reached (yet the CMG system is still well below momentum saturation).

The following accepted trajectory (b) is a trial result employing a constant negative null motion. The terminal momentum command is reached, and the cost is considerably better (-0.23), but the system still skirts singular states (although avoids being hung up), and exhibits a noisy momentum profile and large added null motion. These drive the cost negative, hence there is still ample room for improvement.

The next accepted trajectory (c) is an early result from the search. The performance is considerably better, excepting a quick brush with a singular state at step #22. The search thereupon rapidly improves the solution in steps (d-g). The dip in the CMG gain that indicated the approach of the singular state is quickly removed, and the average CMG gain is considerably increased, producing a highly controllable gimbal state. The last trajectories (e,f,g) demonstrate the operation of the null motion objective component. Since the CMG gain is high throughout all of these trajectories, the null objective contribution becomes significant. Frequent changes in null motion are disfavored. The final solution (g), after 456 node expansions, was able to avoid the singularity with a single null motion pulse, achieving an objective evaluation of +21.0.

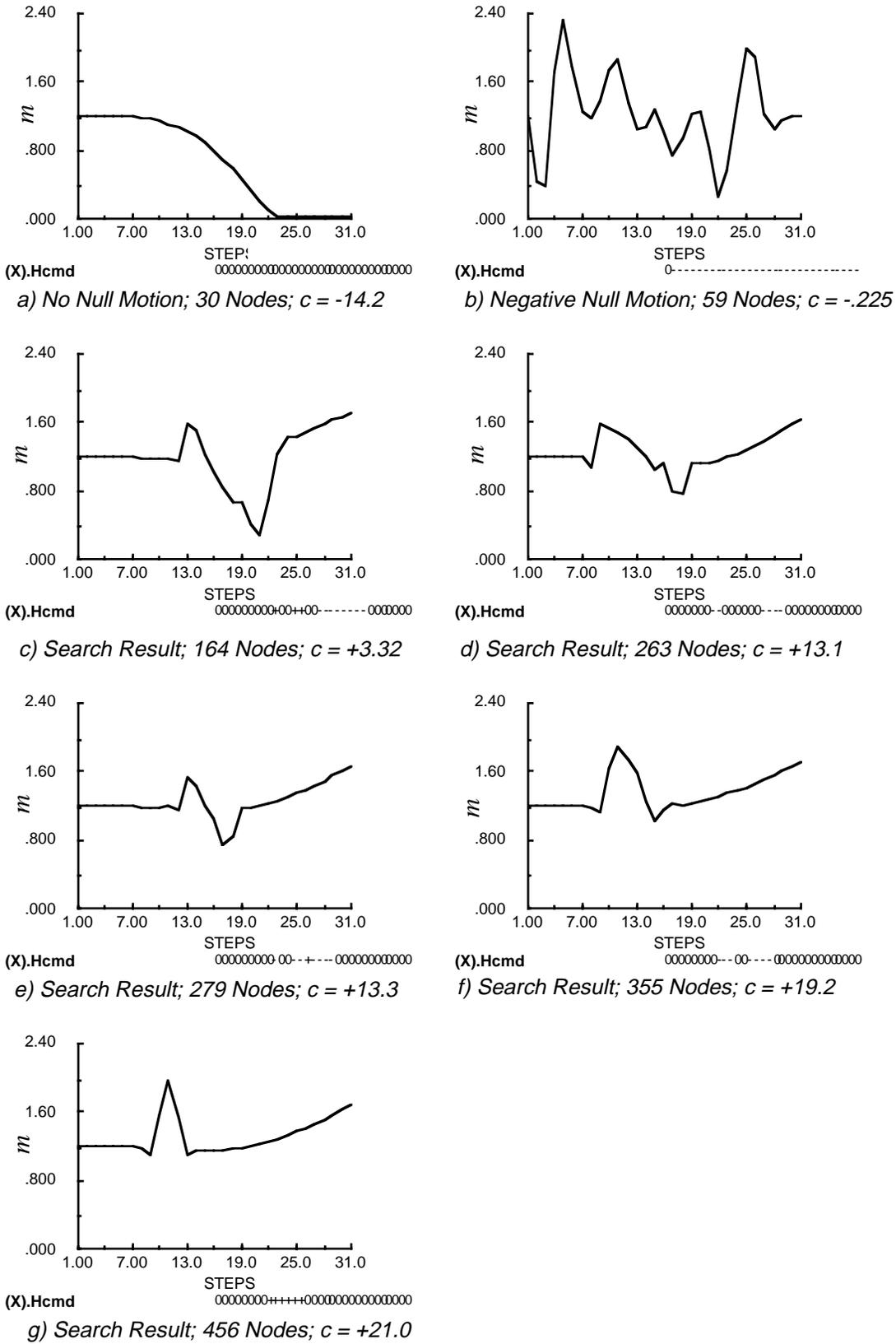
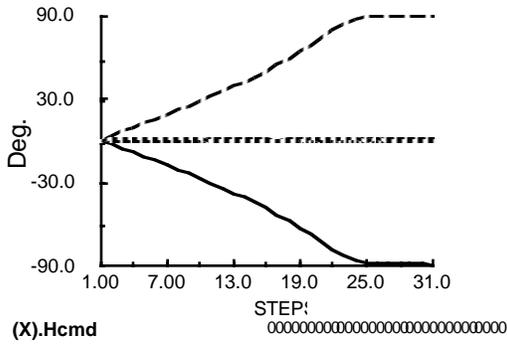
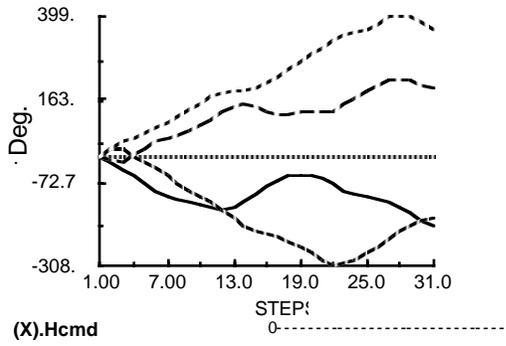


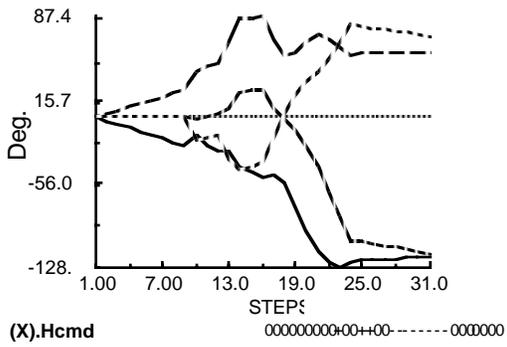
Figure 36: Evolution of CMG Gain for Constant Torque Along \hat{x} Axis



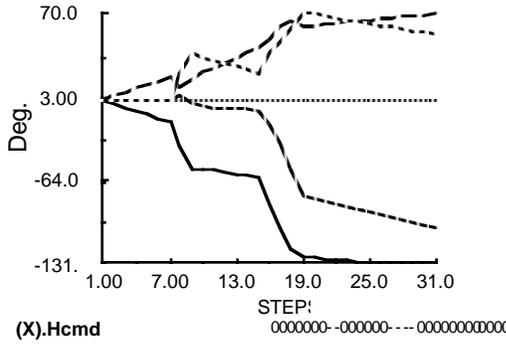
a) No Null Motion; 30 Nodes; $c = -14.2$



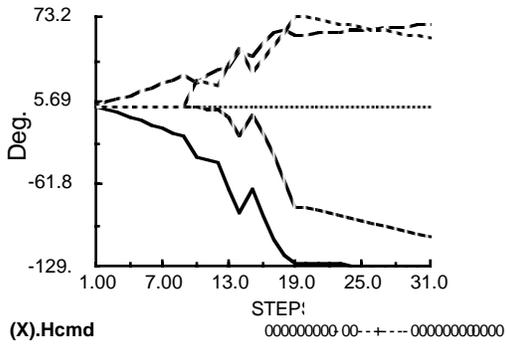
b) Negative Null Motion; 59 Nodes; $c = -.225$



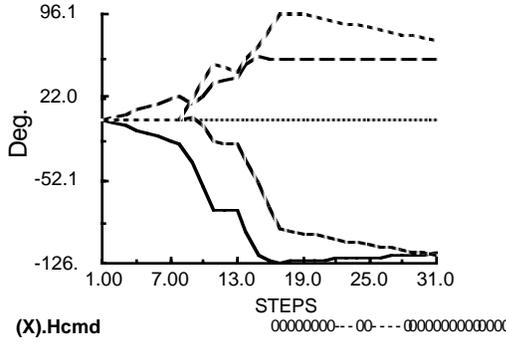
c) Search Result; 164 Nodes; $c = +3.32$



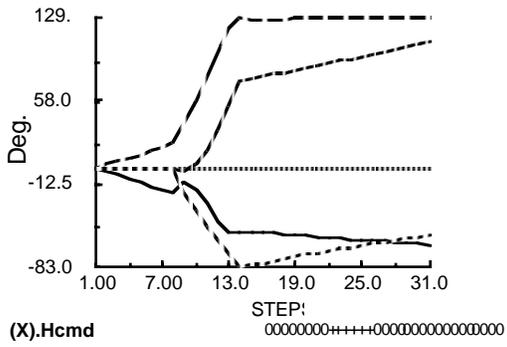
d) Search Result; 263 Nodes; $c = +13.1$



e) Search Result; 279 Nodes; $c = +13.3$

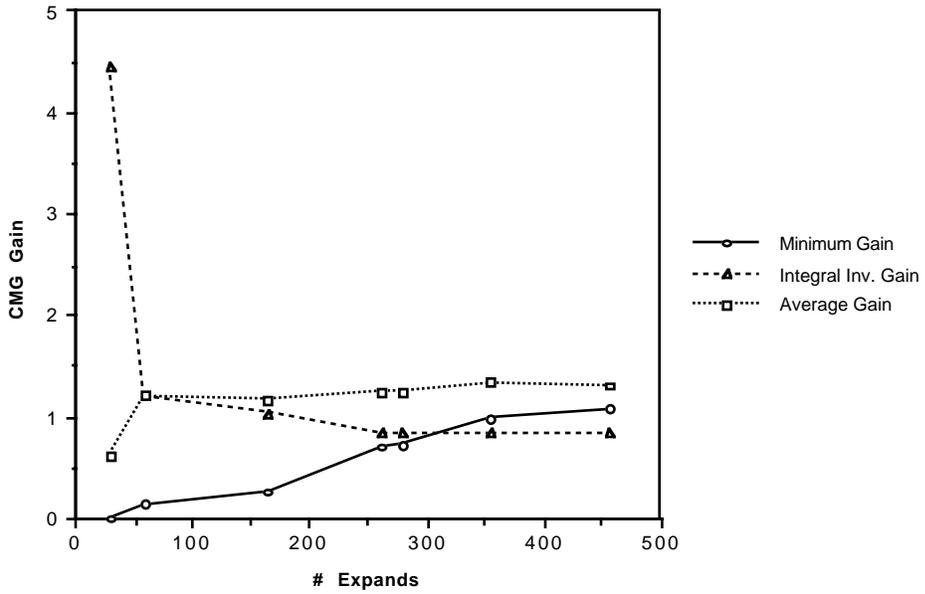


f) Search Result; 355 Nodes; $c = +19.2$

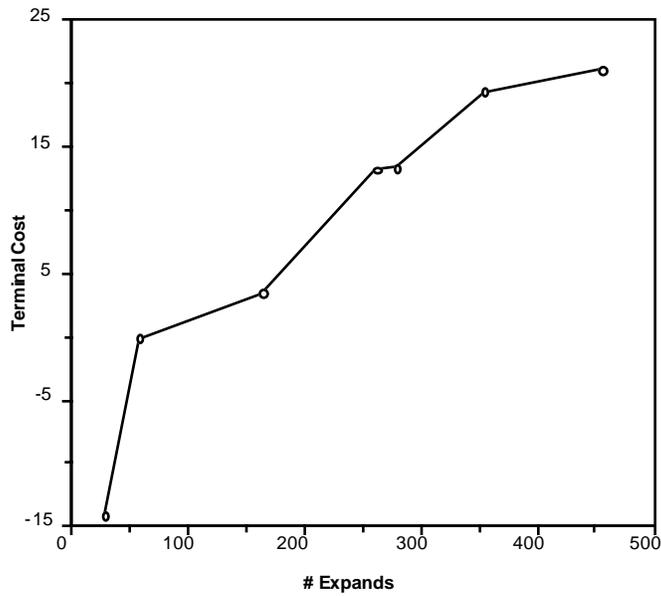


g) Search Result; 456 Nodes; $c = +21.0$

Figure 37: Evolution of CMG Gimbal Angles for Constant Torque Along \hat{x} Axis



a) CMG Gain Improvement



b) Trajectory Costs

Figure 39: Summary of Search Operation for Constant Torque About \hat{x} Axis

This example is the same as given in the short report, Ref. [60]. Due to space limitations, however, Ref. [60] omitted the plots for results b & d. As this document has

no such length constraint, all intermediate search trajectories are presented in the examples shown here.

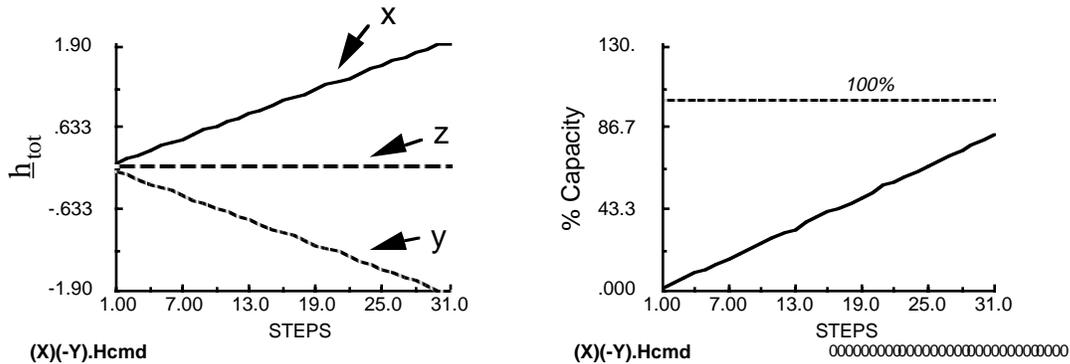
Small "bumps" are seen in the momentum profiles of Fig. 38 (apart from their somewhat wrinkly appearance, which is only an artifact of the Macintosh graphics); these can be most clearly discerned as slight kinks in the linear \hat{x} ramp as seen in the later trajectories (d-g). They are associated with the introduction of null motion, and are caused by errors in the calculated null displacements (these are "tangent" calculations, exactly valid only at one set of gimbal angles). They can be reduced by a finer integration of the CMG kinematics (which will cause the search to take proportionally longer), or by specifying a smaller limit on the null gimbal rate scaling (η in Fig. 28). Since momentum errors are feedback-compensated between successive search steps, steady-state momentum shifts are rejected (to control vehicle attitude, however, a second-order feedback loop will be necessary). The momentum residual contribution to the objective of Eq. 15 also works to minimize these errors. Such coarse integration steps (and resultant small momentum errors) appear to be adequate for the search; when the search results are realized in a higher-bandwidth controller (Fig. 30) running a faster integration on finer timesteps, the momentum errors are better compensated, and little divergence is generally seen from the gimbal profile forecast by the search (see Sec. 4.4).

Since these examples employ a 3-state null amplitude ($\bar{k} \in -,0,+$), the gimbal angles will be driven in an impulsive fashion. If it becomes problematic, this situation can be relieved in a variety of manners; i.e. allowing the search to pick intermediate values of \bar{k} and yielding wider dynamic range, or by searching over the *change* in \bar{k} rather than the absolute \bar{k} (and limiting this change to a modest value).

Figure 39 shows a summary of the search performance; parameters are plotted for the accepted "best trajectories" as a function of the number of nodes expanded before the trajectory was discovered. The terminal cost of each trajectory is plotted in (b). Here we see a steady improvement in cost, seeming to begin saturating at the final discovered trajectory (note that the last trajectory was found after 500 expands, but the search continues through 2000 expands; no better trajectories were found). The primary factors going into the terminal cost are plotted for each successive trajectory in (a). The "minimum gain" is the infimum of m over the trajectory (maximized by the search), the "integral inverse gain" is the accumulated $1/m$ that was described in the discussion of Eq. 15 (this quantity expresses the amount of time spent in a singular region, and is minimized by the search), and the "average gain" is the correction due to mean m that was added to the objective in Eq. 16 (maximized by the search). The first solution plotted at left (corresponding to the trial without null motion) is truly abysmal, with large integrated

inverse gain (indicating lots of time spent in a singular situation), zero minimum gain (indicating the singular presence), and low average gain. A large improvement in the integral inverse gain was attained after the next solution, since the trajectory was no longer hung-up in a singularity. These quantities are all gradually improved by the search until they start to show a plateau at the last solution.

In the next example, the CMGs respond to equal torques simultaneously commanded about the $+\hat{x}$ and $-\hat{y}$ axes (\hat{z} torque remains zero). A set of singular states that are extremely difficult to avoid are known to be encountered through this command[52]. The momentum command and saturation index are plotted for the initial trial trajectory (which successfully made it to the terminal momentum state) in Fig. 40. The saturation level (Fig. 40b) at the terminal state is seen to be significantly higher than in the previous example (Fig. 33b), but the CMG system still remains well within its momentum envelope.



a) Commanded Momentum History

b) CMG Saturation Index

Figure 40: Momentum Command & Saturation Index for Constant \hat{x} , $-\hat{y}$ Torque

A singularity analysis for this example is given in Fig. 41. The left column shows results for the initial trial trajectory (with no null motion), and the right column gives results for the final trajectory deemed "best" by the search before it quit. The eigenvalues plotted in the top row (a) are clearly of opposite sign during the singular encounters of both trajectories, indicating null-motion escapable "hyperbolic" singularities. There is no significant projection of the torque command onto the singular direction, however, as seen by the near-zero value of the plots shown in the middle row (b). This indicates that the singularity will not affect the CMG output, since no torque is needed along the singular axis, hence the system will "sail through" this singular configuration.

Any disturbance torques that might project onto the singular direction will alter this situation, however, and could produce a large deleterious effect during the singular

encounter. The rotor sign sums (c) show that both trajectories transitioned rapidly from 0H to 4H. This transition occurred through the singular state (an "unkinking") in both solutions. In the best solution found by the search, this closure transition occurred earlier.

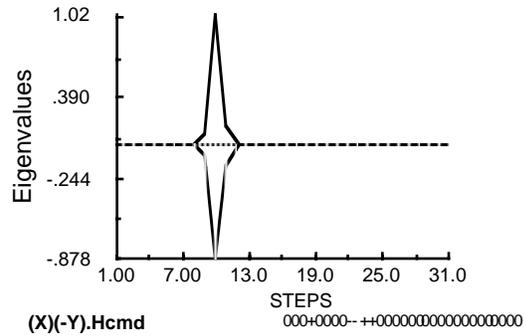
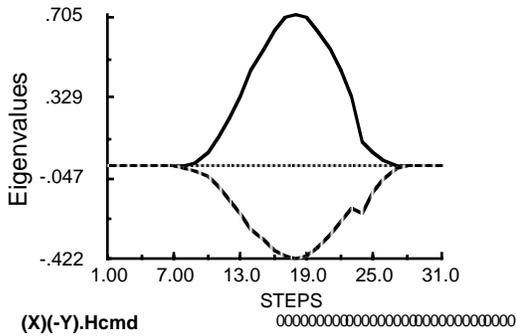
The evolution of the "best" search solution is shown in Figs. 42-45. The initial trial with no null motion (a) produces the commanded \hat{y} torque by scissoring all 4 gimbals, as seen in Fig. 44. The CMG gain plot (Fig. 43), shows that a singularity is achieved at timestep 18. Because the singular direction is not aligned with the torque command in this case, the system is able to transit through the singularity and complete the momentum command (Fig. 45). Nonetheless, the singular region is potentially unstable, and the objective encourages this encounter to be removed or minimized.

The next accepted attempt (b) is the trial with all negative null motion. Although a singular state is not so closely approached, near-singular encounters are frequent and significant momentum residual is accumulated. The following accepted trajectory (c) is the trial which tries to locally "unkink" the rotor projections through the method of Ref. [34]. Here, an initial shot of null motion pulls the system past a singularity at the start of the trajectory, after which a very high CMG gain is achieved (summarized nicely in the topographic plot of Fig. 42). The following solution (d) is located by the search, and uses excessively changing null motion to further reduce the time spent near the singular state and keep m large. This is additionally improved on; solutions (e-g) specify similar gimbal trajectories (a variant on the unkinking technique) that spend only a very brief period near a singular configuration, with minimal null pulsing added. Solutions e-g vary subtly; the search adjusted the time at which the null pulsing occurred by only a single step or two (yielding a small improvement in m , as seen in Fig. 43), but kept the same form for $\bar{k}(t)$.

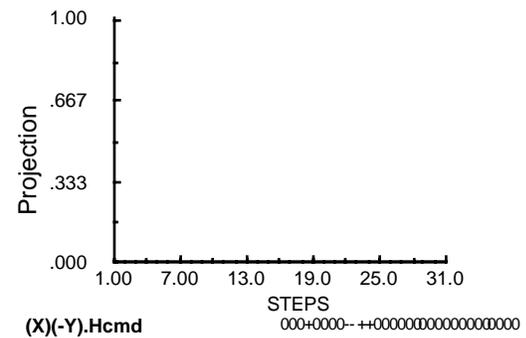
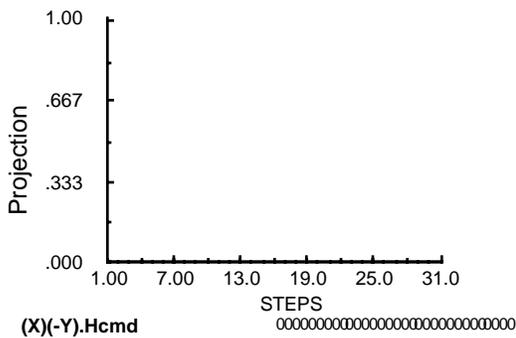
In this example, the singular state appears to be unavoidable with this command sequence and the constraints (i.e. peak gimbal rates) imposed on the CMG system (even though the singularity was hyperbolic in nature). The search did its best, however, to minimize both the approach to the singularity and the period of time spent at low CMG gain, as dictated by the objective of Eq. 15 and predicted by the discussion associated with Fig. 23. All solutions were able to attain their commanded terminal momentum state, as seen in Fig. 45.

Initial Trajectory

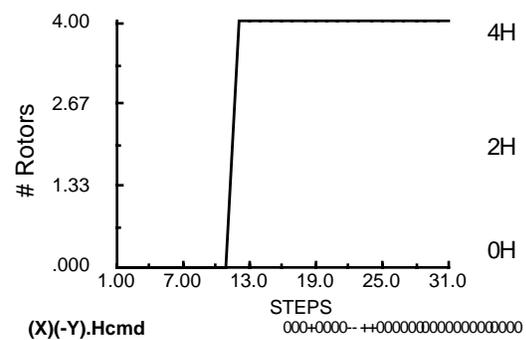
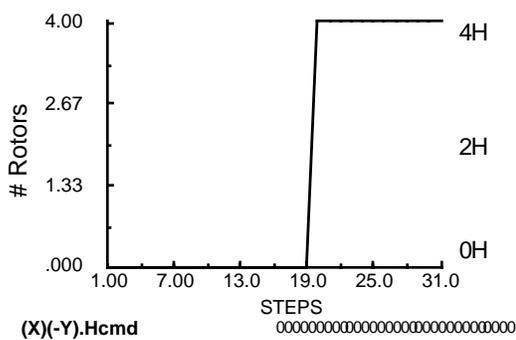
Final Trajectory



a) Escapability Indicators

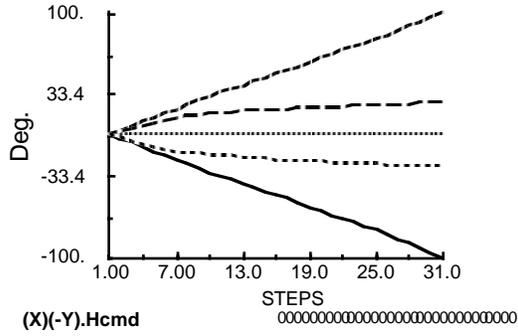


b) Torque Command Projection onto Singular Direction

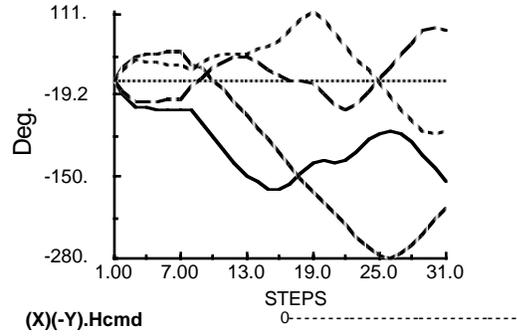


c) Net Rotor Sign Pattern; Singular Trajectory

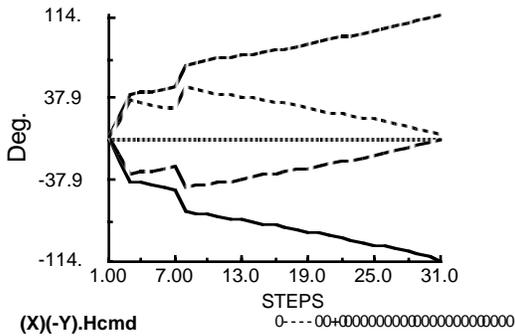
Figure 41: Singularity Analysis for Constant $\hat{x}, -\hat{y}$ Torque



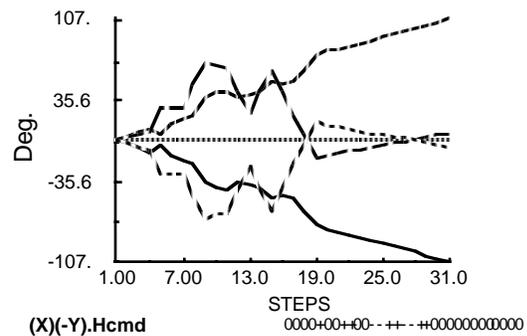
a) No Null Motion; 30 Nodes; $c = -8.57$



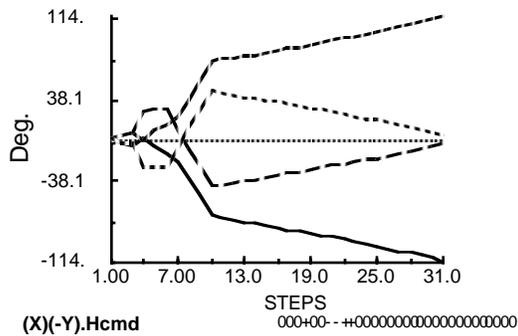
b) Negative Null Motion; 59 Nodes; $c = -8.17$



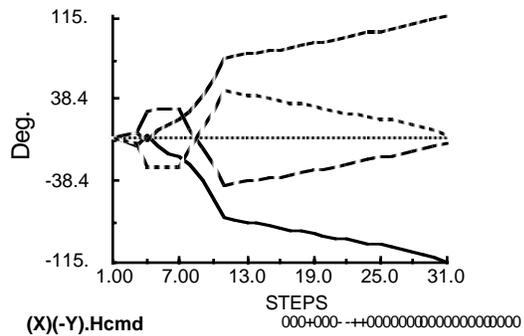
c) Cornick Unkinking; 144 Nodes; $c = -2.03$



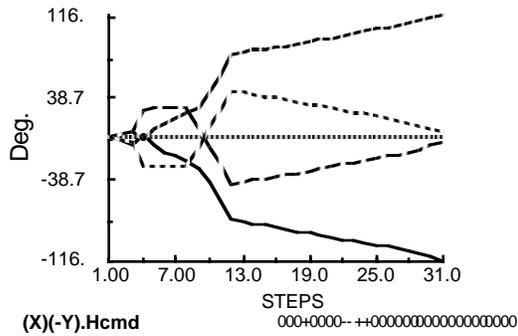
d) Search Result; 219 Nodes; $c = -1.69$



e) Search Result; 428 Nodes; $c = +0.19$



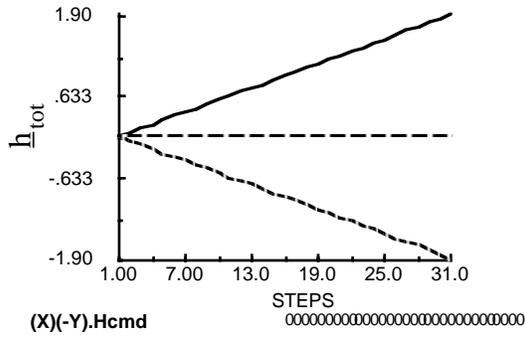
f) Search Result; 756 Nodes; $c = 0.19$



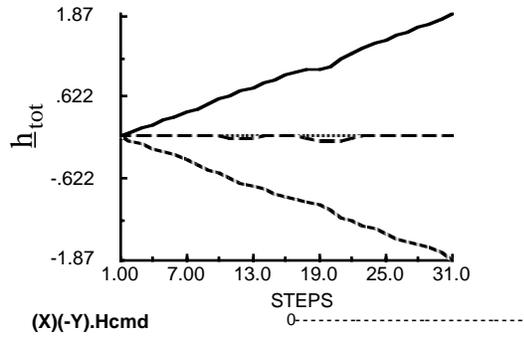
g) Search Result; 1075 Nodes; $c = +0.26$

Figure 44: Evolution of CMG Gimbal Angles for Const. Torque Along

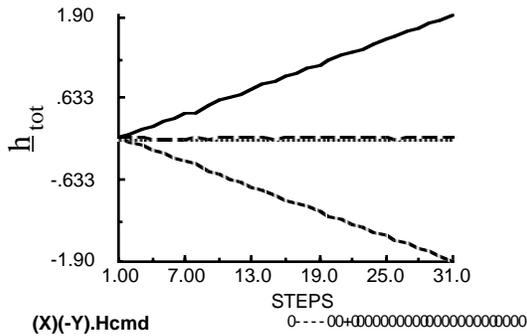
$\hat{x}, -\hat{y}$ Axes



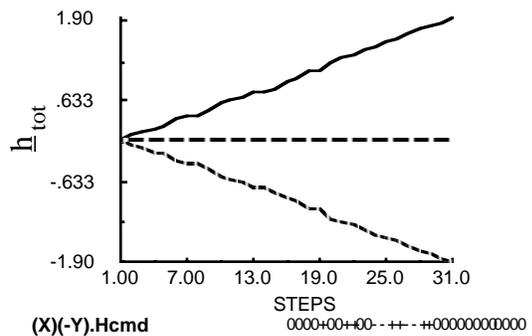
a) No Null Motion; 30 Nodes; $c = -8.57$



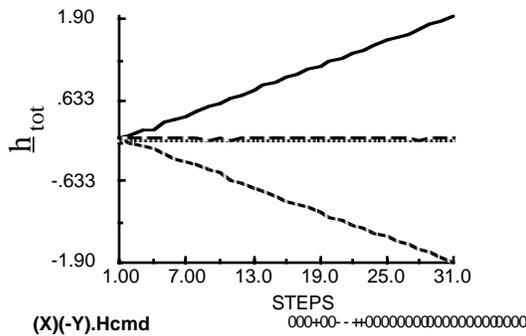
b) Negative Null Motion; 59 Nodes; $c = -8.17$



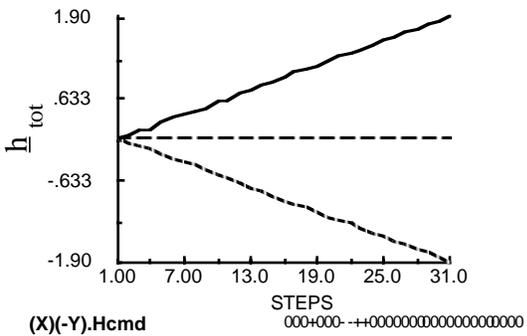
c) Cornick Unkinking; 144 Nodes; $c = -2.03$



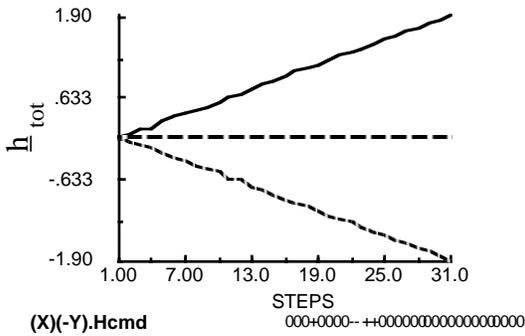
d) Search Result; 219 Nodes; $c = -1.69$



e) Search Result; 428 Nodes; $c = +0.19$



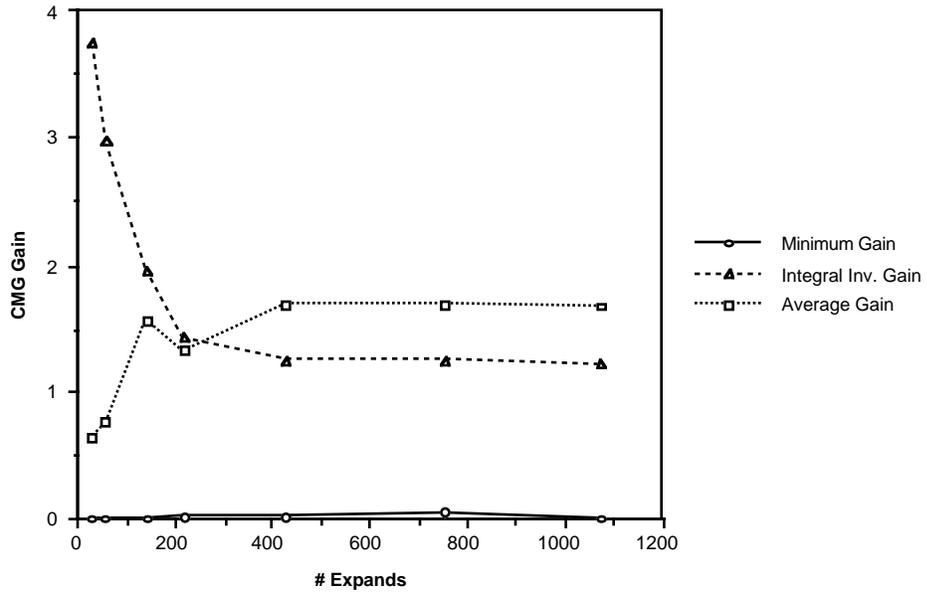
f) Search Result; 756 Nodes; $c = 0.19$



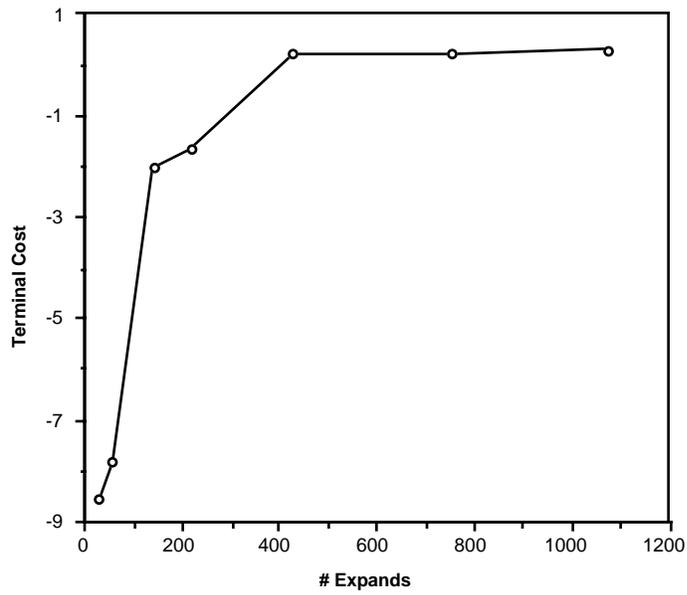
g) Search Result; 1075 Nodes; $c = +0.26$

Figure 45: Evolution of CMG Stored Momentum for Const. Torque Along

$\hat{x}, -\hat{y}$



a) CMG Gain Improvement



b) Trajectory Costs

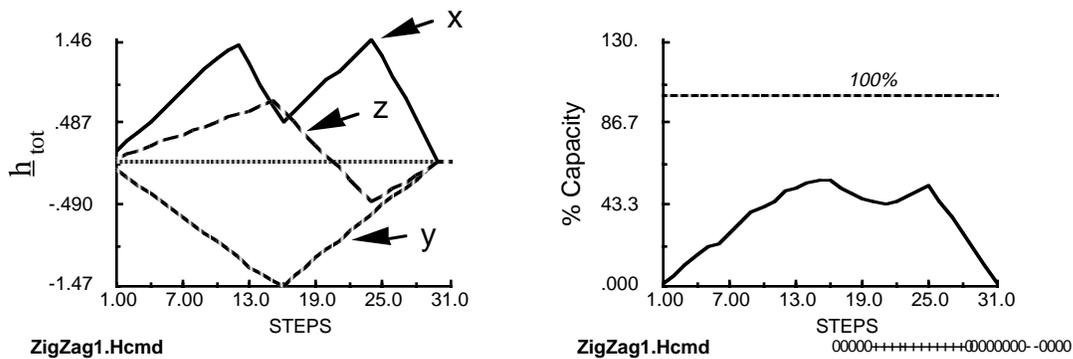
Figure 46: Summary of Search Operation for Const. Torque About

\hat{x} , $-\hat{y}$ Axes

Fig. 46 shows the parametric summary of search performance for this example. The cost function is seen to quickly improve after the initial trial solutions (first 3 points), then grow better as the search executes, and reaches a plateau with the last three trajectories, which have only subtle cost difference. In Fig. 46a, one can see that the cost improvement was due mainly to lower integral inverse gain (meaning less time spent near a singularity) rather than a larger minimum gain, which stayed near zero.

This example was also presented in Ref. [60], where (again due to space constraints) trajectories b, e, & g were omitted (b is a trial with little consequence, and solutions e,f,g are nearly identical).

The next example employs a cyclic command sequence about all 3 axes (thus it's dubbed the "ZigZag" maneuver). It was quickly drawn with the mouse as a rough approximation of a cyclic torque sequence that may be vaguely typical of the orbital environment (ie., gravity gradient and aerodynamic torques). This sequence is somewhat interesting, as it transits the vicinity of several singular states. The momentum command and saturation index (for the final solution) are shown above in Fig. 47. The CMG system is only driven to half of its momentum capacity, thus stays well below saturation.

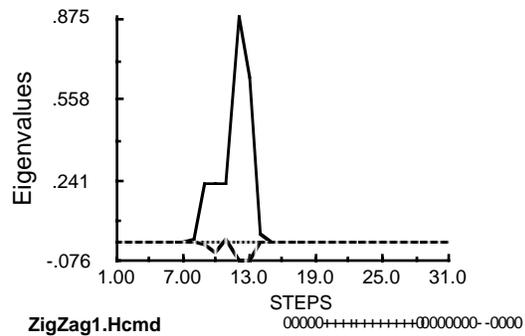
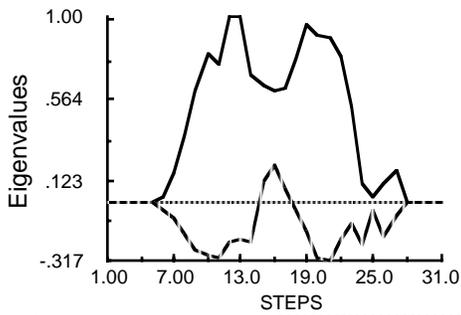


a) Commanded Momentum History b) CMG Saturation Index
Figure 47: Momentum Command & Saturation Index for Zig-Zag Sequence

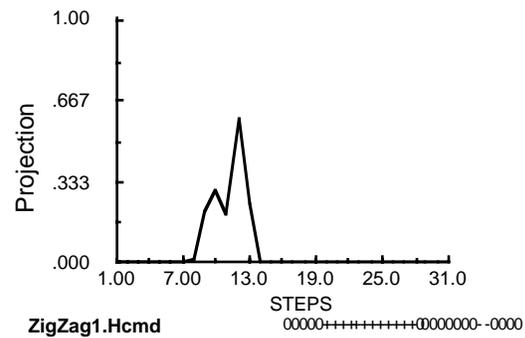
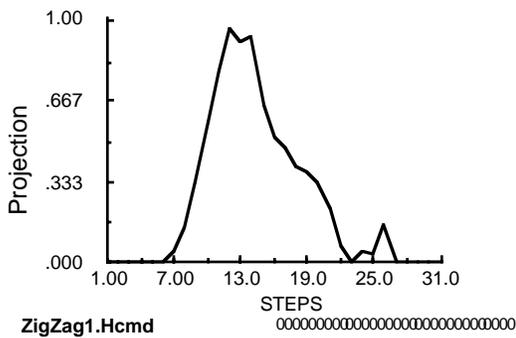
The singularity analysis for this system is shown in Fig. 48 (initial and final trajectory results are given). In the initial no-null motion case, the plot indicates a hard singularity (becoming elliptic, as both eigenvalues go positive in (a)), and a large torque projection onto a singular direction (b), certainly a recipe for trouble. The final trajectory displays a narrower singular encounter with less torque projection onto the singular direction. The rotor configurations (c) were seen to stay near the "kinked" 2H level in the initial trajectory (which can lead to difficult singularities), but were moved out to superior 4H orientations in the final solution.

Initial Trajectory

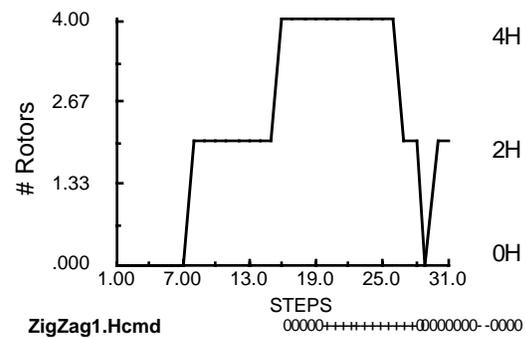
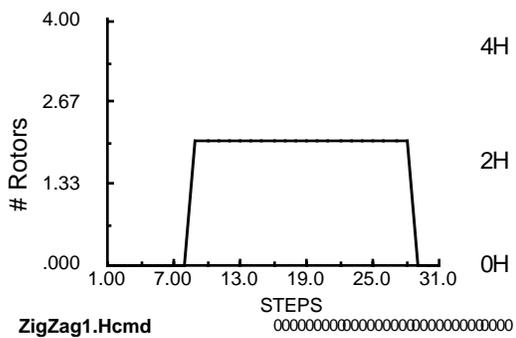
Final Trajectory



a) Escapability Indicators

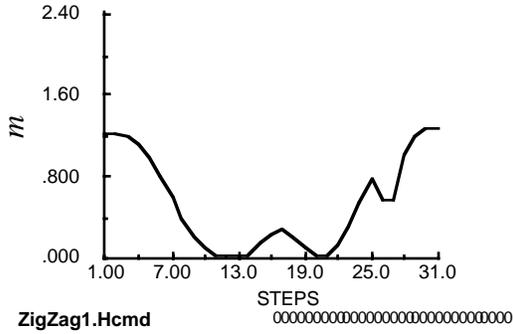


b) Torque Command Projection onto Singular Direction

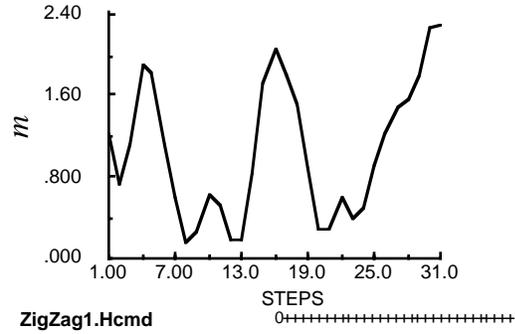


c) Net Rotor Sign Pattern; Singular Trajectory

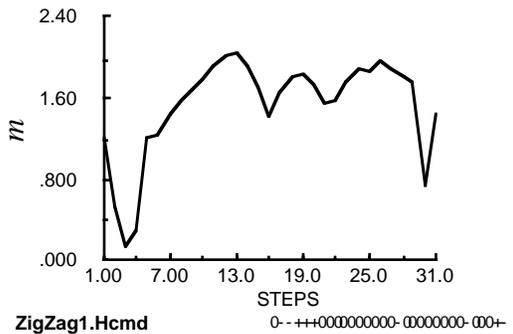
Figure 48: Singularity Analysis for Zig-Zag Maneuver



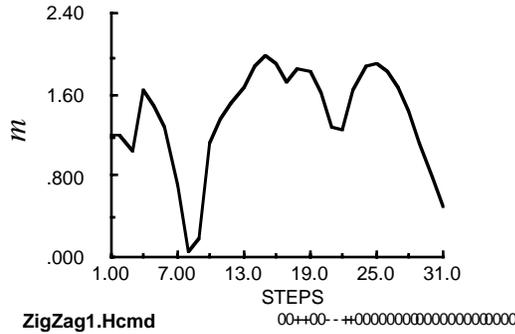
a) No Null Motion; 30 Nodes; $c = -20.0$



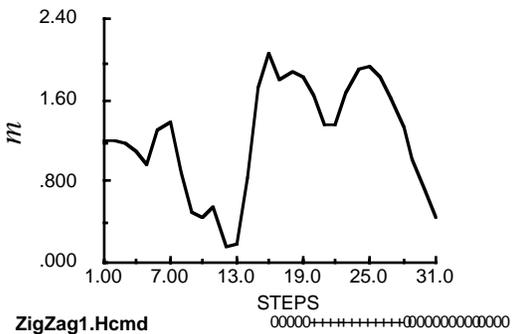
b) Positive Null Motion; 88 Nodes; $c = -5.13$



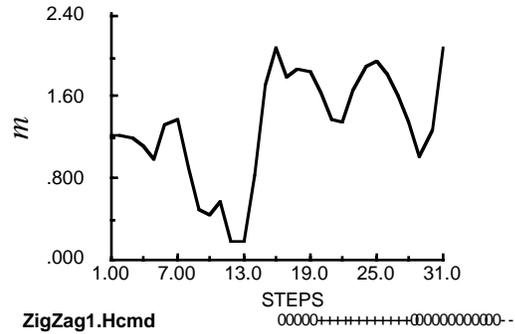
c) Cornick Unkinking; 144 Nodes; $c = -0.89$



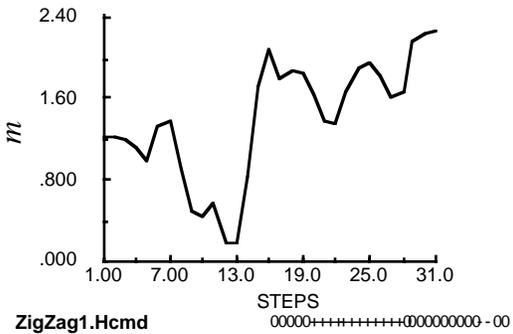
d) Search Result; 377 Nodes; $c = -0.68$



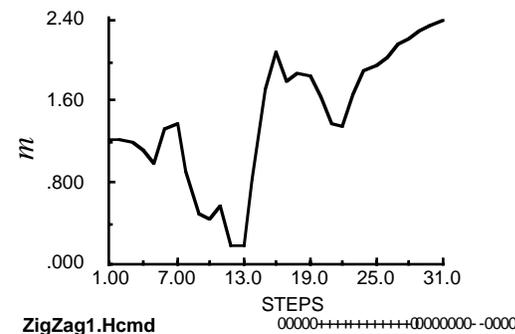
e) Search Result; 1153 Nodes; $c = +0.14$



f) Search Result; 1209 Nodes; $c = +0.30$

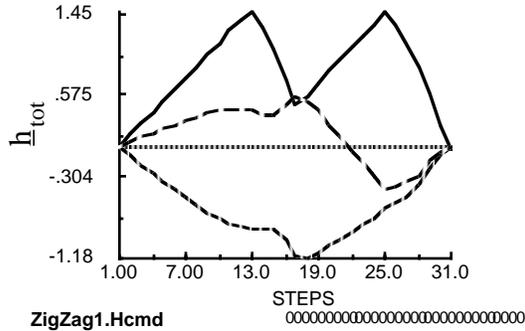


g) Search Result; 1275 Nodes; $c = 0.40$

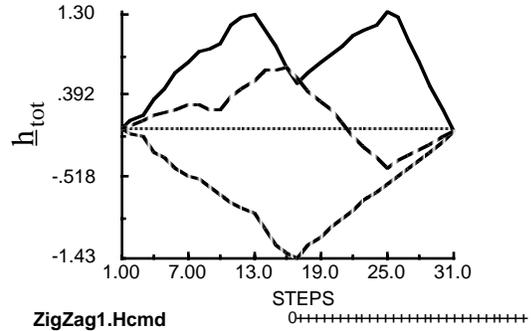


h) Search Result; 1281 Nodes; $c = +0.50$

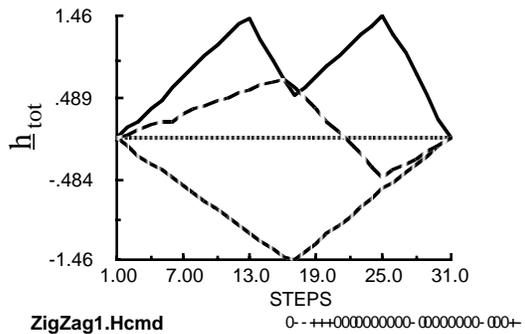
Figure 50: Evolution of CMG Gain for Zig-Zag Maneuver



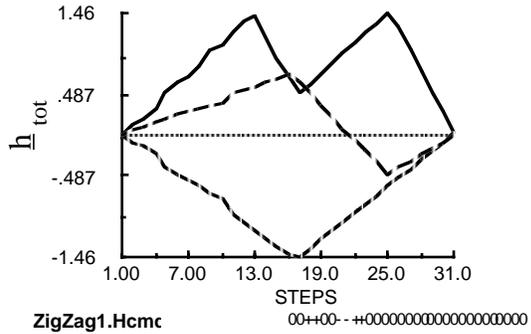
a) No Null Motion; 30 Nodes; $c = -20.0$



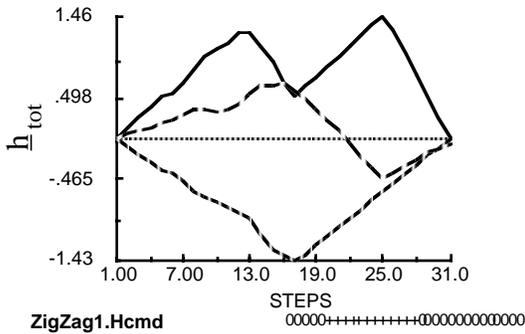
b) Positive Null Motion; 88 Nodes; $c = -5.13$



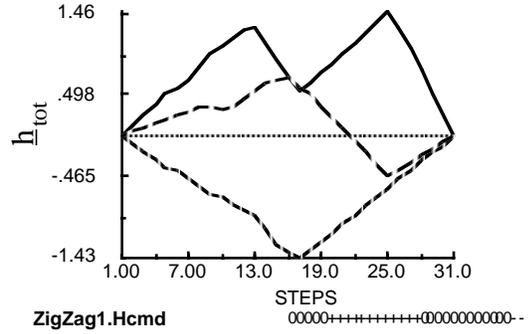
c) Cornick Unkinking; 144 Nodes; $c = -0.89$



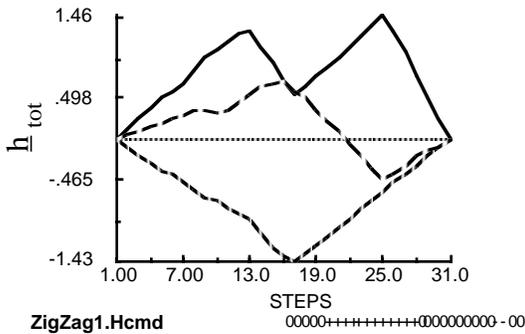
d) Search Result; 377 Nodes; $c = -0.68$



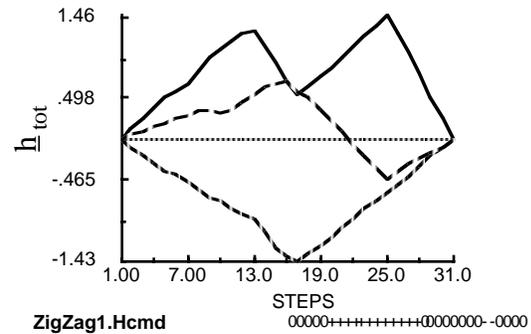
e) Search Result; 1153 Nodes; $c = +0.14$



f) Search Result; 1209 Nodes; $c = +0.30$

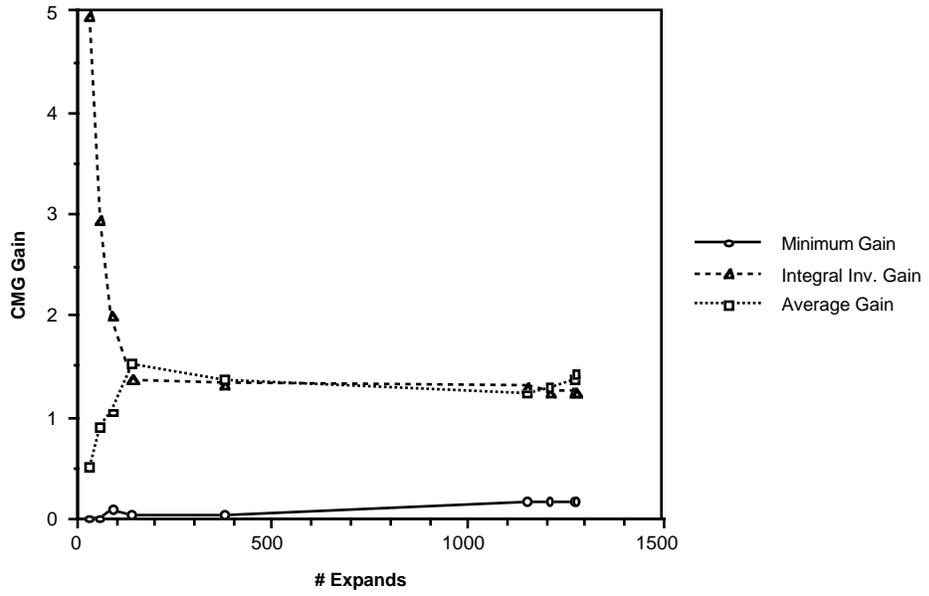


g) Search Result; 1275 Nodes; $c = 0.40$

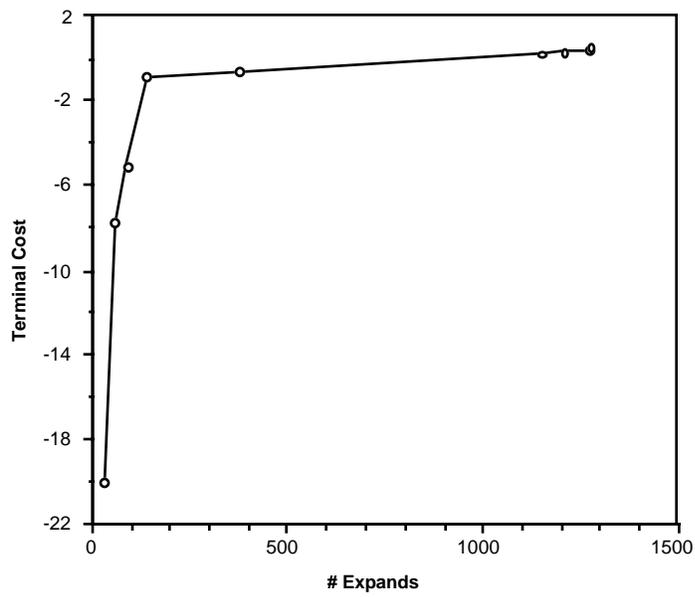


h) Search Result; 1281 Nodes; $c = +0.50$

Figure 52: Evolution of CMG Stored Momentum for Zig-Zag Maneuver



a) CMG Gain Improvement



b) Trajectory Costs

Figure 53: Summary of Search Operation for Zig-Zag Maneuver

The "best" zig-zag solutions found by the search are shown in Figs. 49-52. The initial trial is indeed seen to linger in a singular configuration, as seen in Fig. 50a, resulting in considerable distortion of the resultant momentum (Fig. 52a). No space remained on these pages to include the trial with negative null motion only, which was accepted as "best" immediately after the initial trial with no null motion. The all-negative trajectory was of mediocre quality, and was immediately supplanted by the trial with all positive null motion (shown as trajectory (b)). This trajectory is somewhat better, but still spends considerable time near singular states. The "unkinking" trial (c) shows definite improvement; the singular encounter is limited to a short pulse at the beginning of the trajectory, where the gimbals were "snapped" through a singular state and into a better orientation. The search results (d-h) worked to improve this solution. Because of its high weight in the objective, the search worked most vigorously to raise the minimum gain value, achieving a small degree of success by trajectory (h). The objective contribution that tries to increase the average trajectory gain participated in raising the m value of the terminal nodes in the last few trajectories accepted.

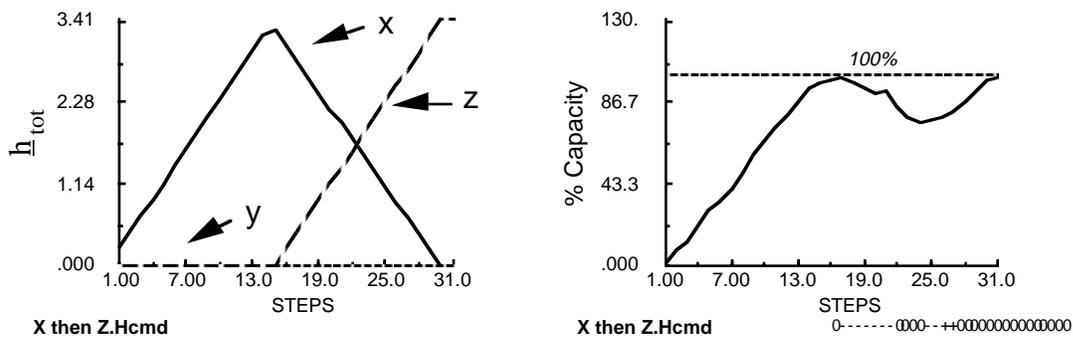
The final solution located by the search consisted of a broad positive null pulse followed by a briefer negative transient (Fig. 49). The long null pulse, coupled with the close skirting of the singularity, did produce some small distortion of the CMG momentum history (Fig. 53h). On search runs that weighted the momentum residual more heavily in the objective, the final solution had less momentum error; the last solution accepted tended to be a minor variant of the "unkinking" trial of trajectory (c), which is quite faithful to the momentum command sequence (Fig. 53c).

Figure 46 depicts a summary of how the "best" search solution evolved. Looking at the trajectory costs in plot (b), one sees that the largest benefit occurred in the 4 trial solutions (0 null, - null, + null, unkinking). Plot (a) indicates that that these benefits came mainly from a decrease in the integral inverse gain, insinuating that the later trajectories spent less time near singularities. The search then worked to increase the minimum gain value, and had only marginal success, as can be seen in the lower curve in plot (a) and the slowly increasing objective function in plot (b). The increase in average m due to the higher gain of the terminal states in the last accepted trajectories can be seen as a small upturn in the closely-dotted curve plotted in (a).

The next example takes the CMG system up to its envelope, and passes the 2H elliptical singular states seen as the "ribbons" leading to the "petals" or "dimples" that join the envelope in Fig. 15. These states can create difficulty in attaining net CMG momenta near the envelope, as related by Ref. [61].

The momentum command history is summarized in Fig. 54a; it is a triangle along the \hat{x} axis, superimposed atop a delayed ramp about the \hat{z} axis (\hat{y} momentum is commanded to remain zero). Referring to the pyramid mount (Fig. 9), this command sequence causes the CMG momentum state to move out to the envelope in the xy plane along \hat{x} , then move up toward the tip of the pyramid along \hat{z} (thus it's titled " \hat{x} then \hat{z} "). This sequence causes an approach to the "dimple" on the envelope that is oriented along the gimbal axis facing toward \hat{x} (see Fig. 15), and the associated 2H internal singular states.

The two approaches to momentum saturation are seen in Fig. 54b. The first saturation is along \hat{x} , and the second is along \hat{z} . The small kink seen between the two saturation encounters in Fig. 54b is due to the proximity of the CMG momentum state to a "dimple" in the envelope centered around the gimbal axes.



a) Commanded Momentum History

b) CMG Saturation Index

Figure 54: Momentum Command & Saturation Index for " \hat{x} then \hat{z} "

The singularity analysis for this example is shown in Fig. 55. The initial trajectory (with no null motion) is plotted at left. Two singular approaches are clearly seen to dominate the solutions. The singular states are seen to be mainly elliptic (owing to both eigenvalues staying generally positive), and project strongly onto the torque command (b), which will yield large momentum errors. The CMG system spends most of its time in a 2H state, which is ripe for elliptic singularities.

The situation seems much better with the final trajectory, plotted at right. Most singularities seem to be nearly avoided, excepting the envelope itself, which is encountered near step #16 and step #30 (since this state is elliptical by definition, note that both eigenvalues are strongly positive in its vicinity). The rotors stay mainly in an "unkinked" 4H state, which yields superior singularity characteristics. The short dip into a 2H state near step #21 is associated with the approach to an envelope "dimple".

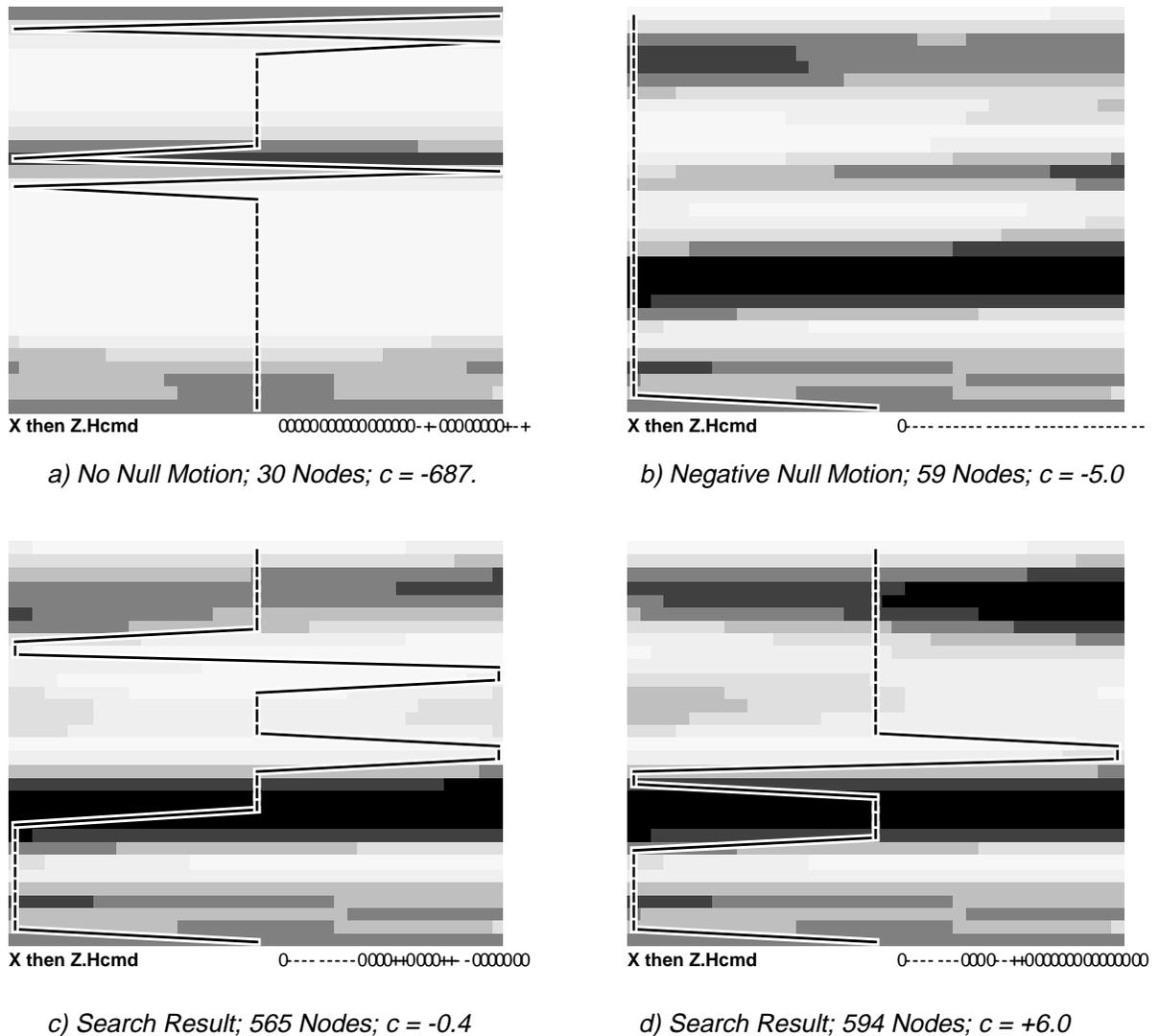
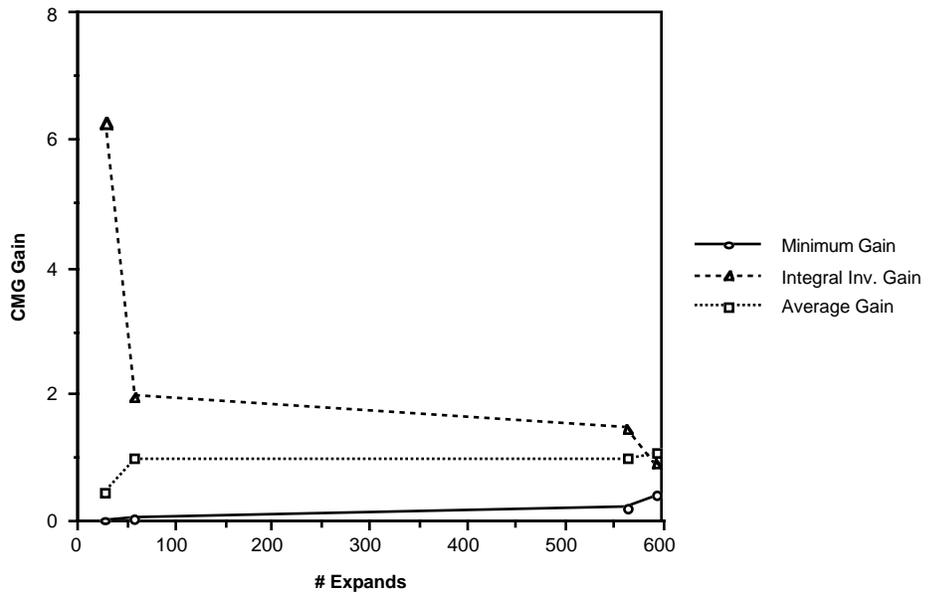
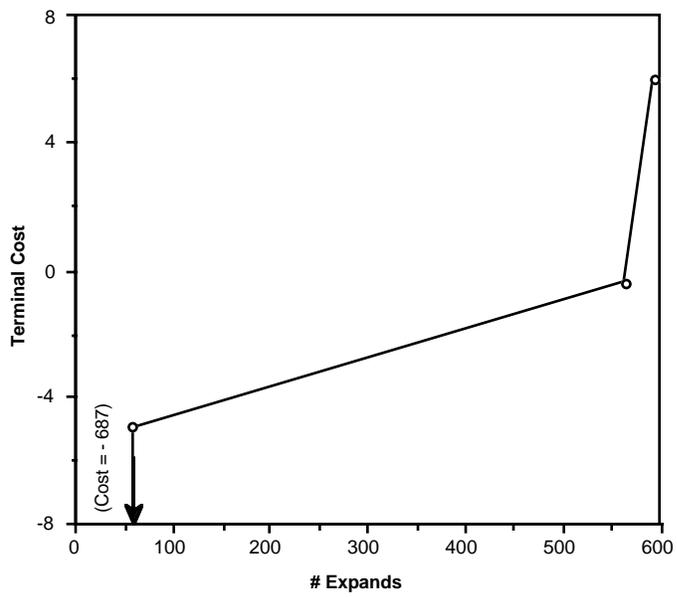


Figure 56: Evolution of Search Solutions for " \hat{x} then \hat{z} "

Figs. 56-59 show how the "best" solution evolves as located by the search process. Although a full 2000 nodes were expanded, the final solution was located after 594 nodes were opened and only four trajectories were accepted. The initial trial with no null motion (a) spent much of its time being singular (the pulses of null motion added in the midst of the trajectory were to reduce the excessive gimbal rates that were calculated there by the SR-inverse). The next accepted solution (b) is the trial with all negative null motion. Although it wings past the initial singular state (step #6) and avoids being trapped, it still approaches singularities, in particular the elliptic singularity near the "dimple" at step #21.



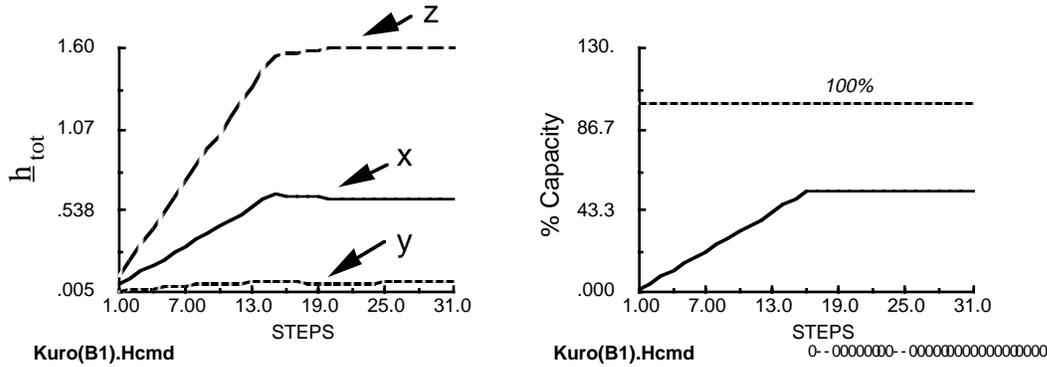
a) CMG Gain Improvement



b) Trajectory Costs

Figure 60: Summary of Search Operation for " \hat{x} then \hat{z} "

The next series of examples are maneuver sequences suggested by Ref. [61] that are designed to take the CMG system past various internal elliptic singular states. The first in this series is summarized in Fig. 61. This momentum sequence stays well inside the envelope, and only approaches roughly 50% saturation, as seen in Fig. 61b.



a) Commanded Momentum History b) CMG Saturation Index

Figure 61: Momentum Command & Saturation Index for "Kuro B"

The singularity analysis is given in Fig. 62 for the initial trial trajectory (with no null motion added). Since the eigenvalues both become positive in (a), an elliptic singular state has indeed been attained, as also supported by the rotor projection plot in (c), where the system stays in a 2H configuration. The large values seen in Fig. 62b indicate that the torque command projects significantly onto the singular direction. The rotor projection plot is given in Fig. 62d for the final trajectory found by the search. It is nonsingular, and can be seen to quickly transition to an "unkinked" 4H rotor orientation.

The evolution of the "best" search trajectories are detailed in Figs. 63 through 66. The elliptical state is seen clearly in plot (a), which shows the system slowly transitioning through the singularity (as the torque command was not precisely aligned with the singular direction, the SR inverse was able to generate gimbal rates, although produced large torque errors, as seen in Fig. 66a). The next trial (negative null motion) is somewhat better, but still exhibits a voyage through a singularity and significant torque error. The gradient trial is the next accepted, and produces a definite improvement [one must bear in mind that this trajectory follows the gradient of the objective (Eq. 15; we see the minimum gain contribution having the most impact here), and not the local ∇m , as is plotted under the trajectories in the topographic maps (Fig. 63)]. The search quickly improves the solution, finishing with a trajectory that raises the m value immediately at the trajectory start, and

keeps a high m (near its maximum of 2.4 for this CMG configuration) throughout the remainder of the the command sequence.

Looking at the plots in Fig. 65, it is obvious that this example employed a different set of initial gimbal angles. In order to encounter the elliptic singular state with this torque command, the CMGs were started in a zero-momentum state with angles of $[-60^\circ, 60^\circ, 120^\circ, -120^\circ]$.

A summary of the search performance for this example is given in Fig. 67. The three trials were seen to produce the most cost improvement (as they avoided the singular state), and the search operation progressively increased the average trajectory gain. This command sequence turned out to be a particularly easy challenge for CMG steering laws. The gradient trial was already seen to produce a workable solution. If the objective is modified to include the local ∇m , the gradient trial becomes almost perfect in this case, producing a solution of similar cost to the final search result shown here (g). This is a significant exception; in the other examples of this section, searches that emphasized the local ∇m in their objective were seen to encounter difficulty or delay in locating solutions of the quality attained through the objective of Eq. 15.

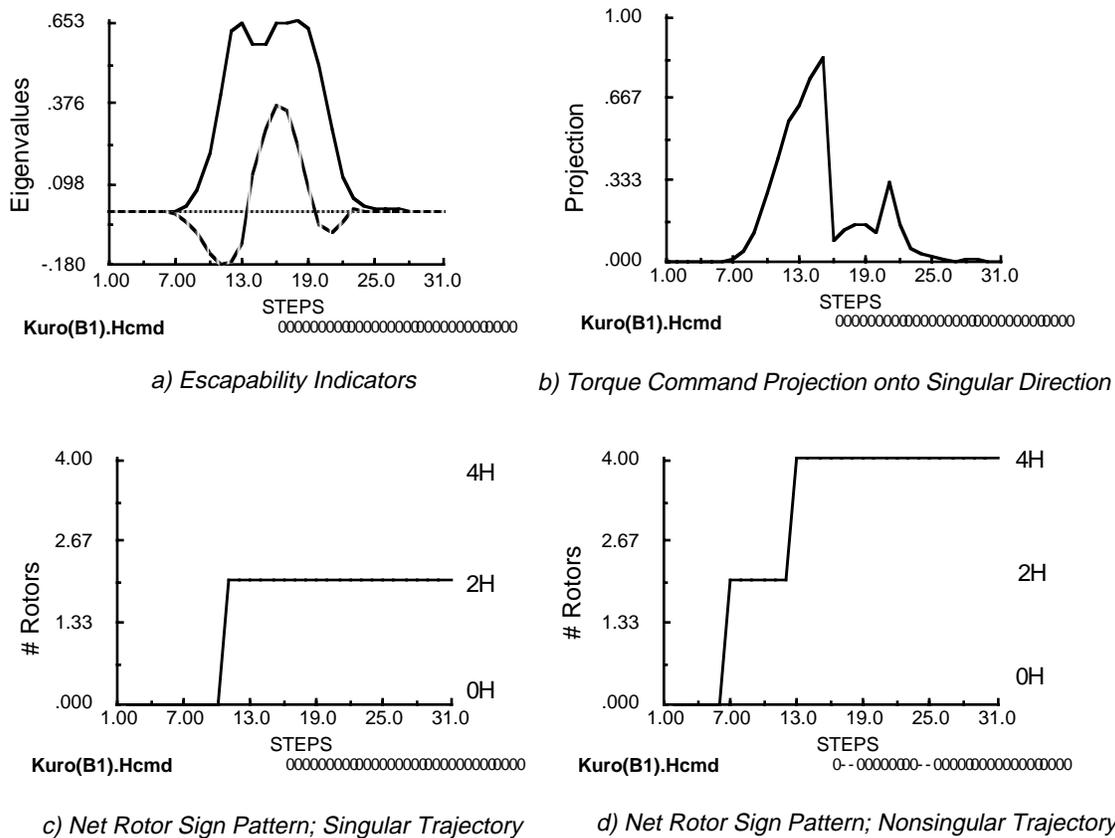
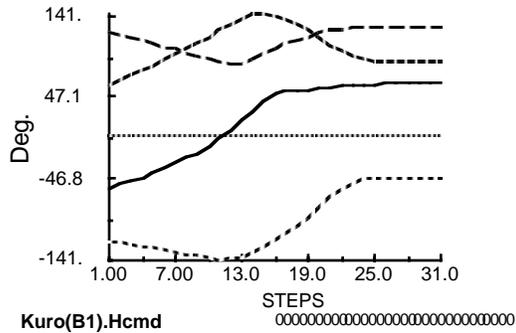
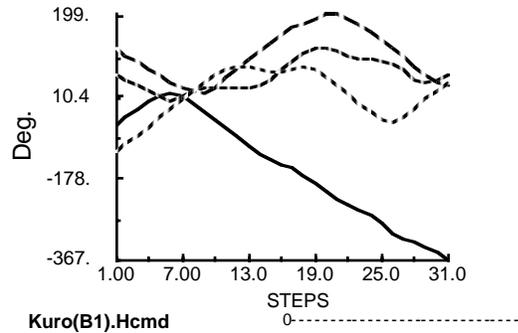


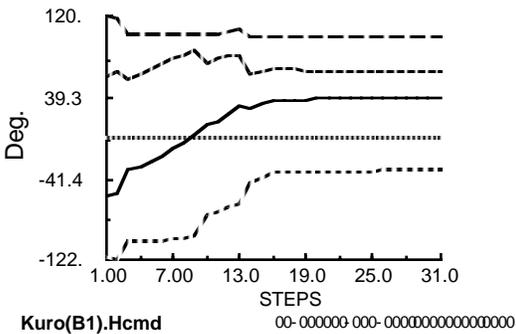
Figure 62: Singularity Analysis for "Kuro B"



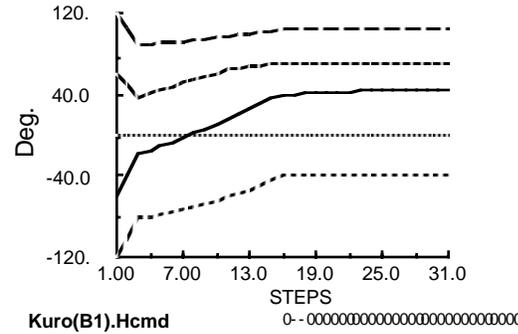
a) No Null Motion; 30 Nodes; $c = -9.89$



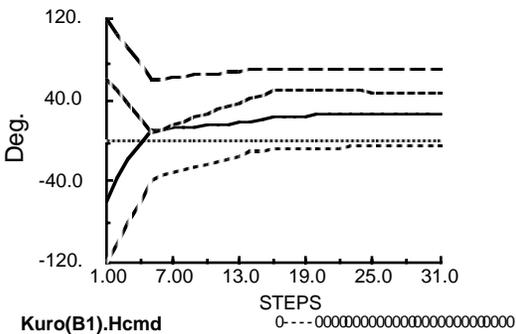
b) Negative Null Motion; 59 Nodes; $c = -1.47$



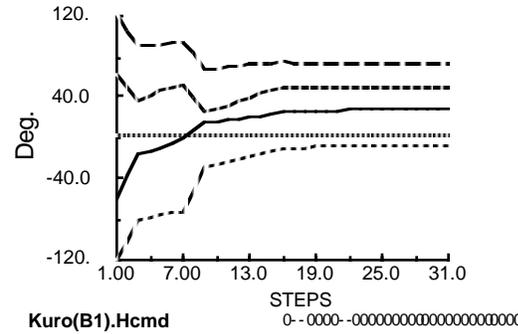
c) Gradient; 116 Nodes; $c = +23.5$



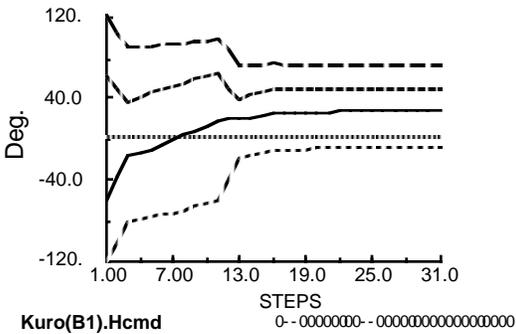
d) Search Result; 183 Nodes; $c = +24.0$



e) Search Result; 345 Nodes; $c = +24.2$

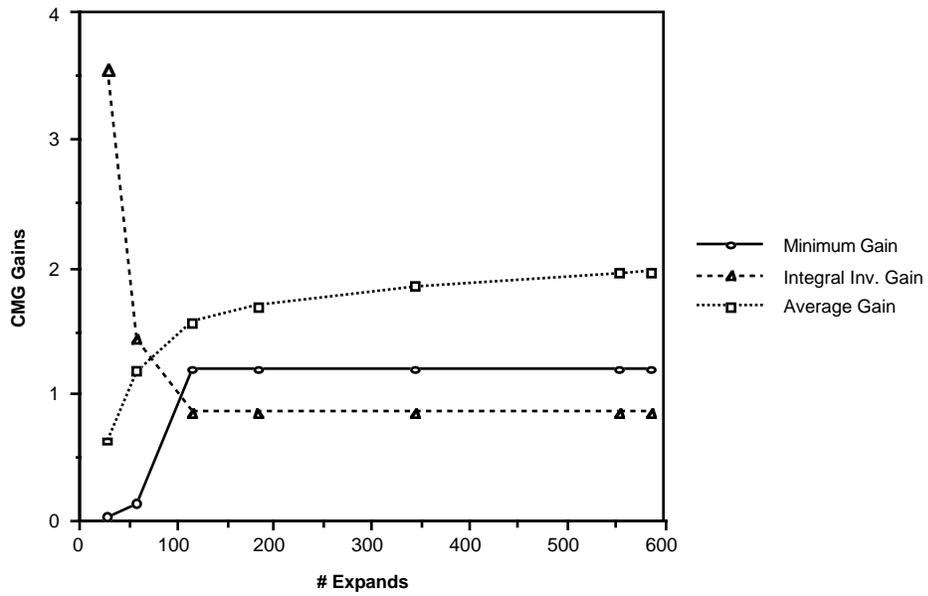


f) Search Result; 553 Nodes; $c = +24.2$

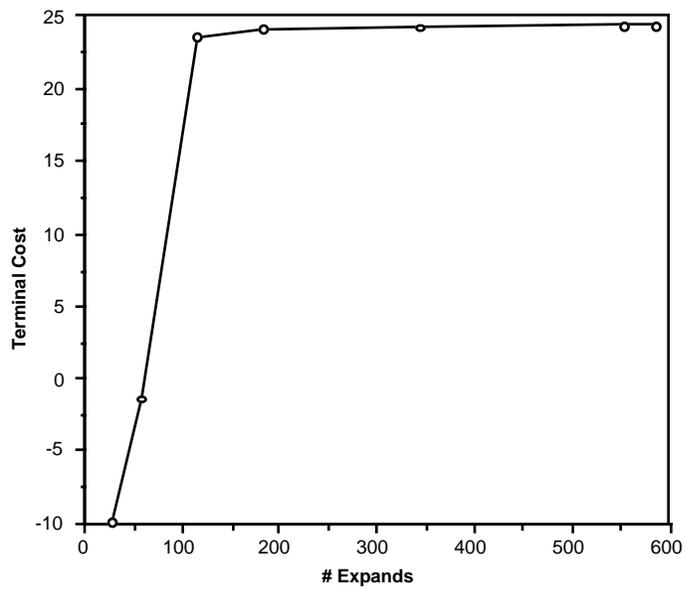


g) Search Result; 588 Nodes; $c = +24.3$

Figure 65: Evolution of CMG Gimbal Angles for "Kuro B"



a) CMG Gain Improvement



b) Trajectory Costs

Figure 67: Summary of Search Operation for "Kuro B"

The next example also comes from Ref. [61], and is designed to explore an elliptical singularity with a smaller momentum, located closer to the \hat{z} axis. The momentum command sequence is summarized in Fig. 68, along with the saturation index for the final trajectory found by the search (which achieved the terminal momentum command without difficulty).

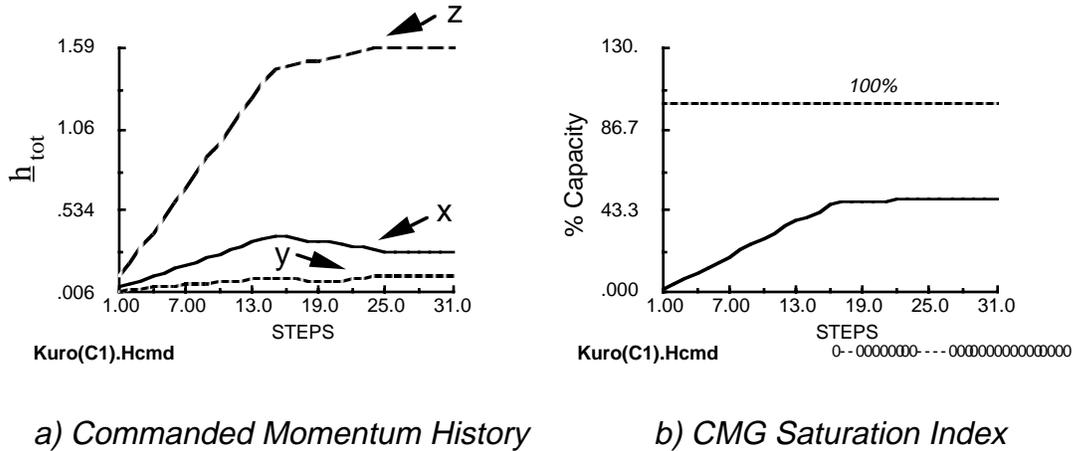


Figure 68: Momentum Command & Saturation Index for "Kuro C"

The singularity analysis for the initial trajectory is shown in Fig. 69. As both eigenvalues become positive (a), the singularity is indeed classed as elliptic. Since the torque command projects nearly totally onto the singular direction (b), the SR-inverse will have difficulty producing relevant gimbal rates. As expected, the rotors are seen to stay kinked in a "2H" state during the singular encounter (c), although rapidly transition to the mutually-pointing "4H" orientation in the final trajectory found by the search (d), which did not encounter a singular state.

The progression of search solutions is documented in Figs. 70-73. The topographic map and gain plot for the initial trial solution (Figs. 70a,71a) clearly show that the system remains in the singular condition for a large part of the command sequence. Although some semblance of the command momentum is achieved (Fig. 73a), the CMG system has difficulty in achieving prompt response in the singular region. The next three accepted solutions are the trials; negative null (b), positive null (c), and gradient (d) [remember that this gradient isn't from ∇m , but is calculated through Eq. 15]. The gradient trial is seen to avoid the singular state, and is improved on by the succeeding three accepted search trajectories.

The final solution (g) exhibits two negative null motion pulses; one at the start of the command sequence, to put the CMG system into a high gain state, and another midway through the sequence that switches the CMG closure in order to achieve a high gain state

throughout the remainder of the trajectory. Looking at the plot of the rotor configuration for this trajectory in Fig. 69d, one notices that the switch from a "2H" state to a "4H" state was caused by this later null pulse. The concurrent small dip in m that is seen in Fig. 71d (near step #15) indicates several CMG minors passed through zero during this closure change, as discussed in Sec. 2.3.

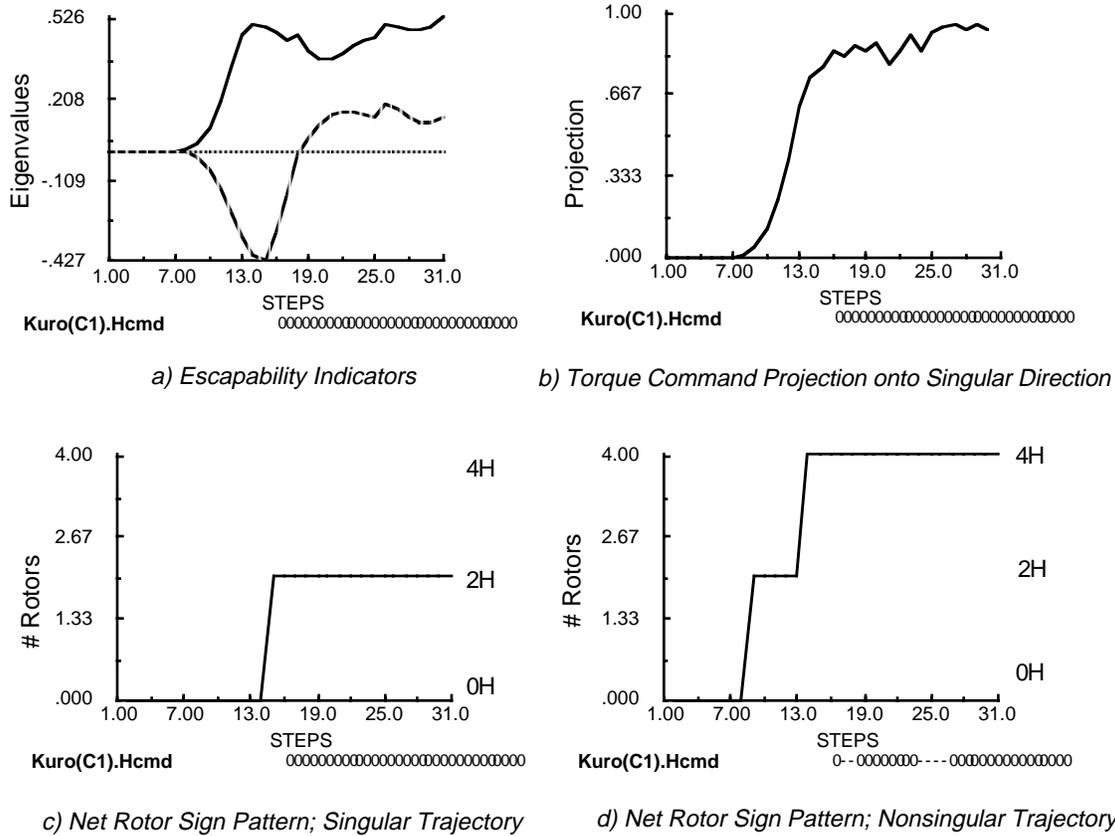
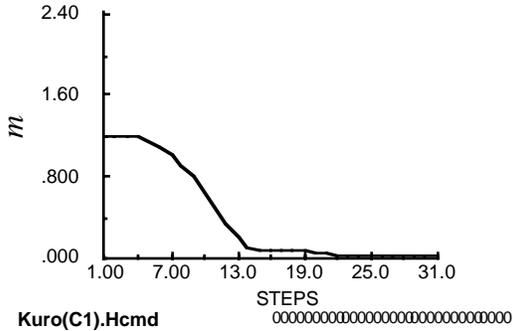


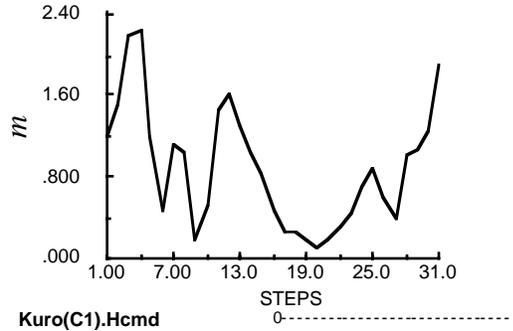
Figure 69: Singularity Analysis for "Kuro C"

In order to reach the elliptical state with this set of momentum commands, yet another set of initial gimbal angles were used with this example. As can be noted in Fig. 72, the starting angles for CMGs 1-4 were $[-120^\circ, -60^\circ, 60^\circ, 120^\circ]$.

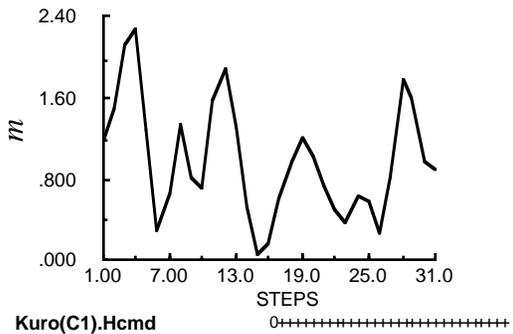
Figure 74 shows the summary for search operation in this example. Again, the gradient trial was seen to produce a passable trajectory, which was progressively improved upon by the search. Since the latter three trajectories stay mainly at high m , they exhibit little significant cost difference.



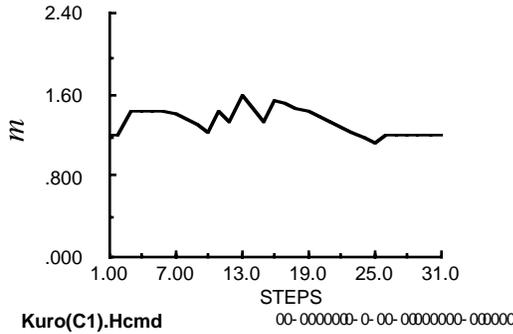
a) No Null Motion; 30 Nodes; $c = -20.6$



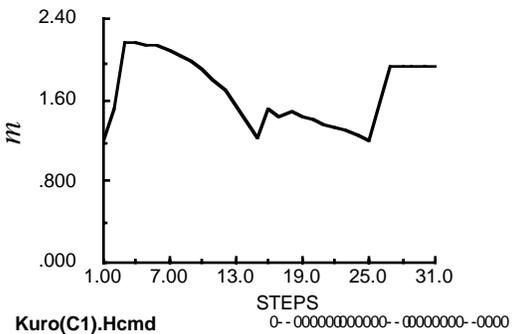
b) Negative Null Motion; 59 Nodes; $c = -5.53$



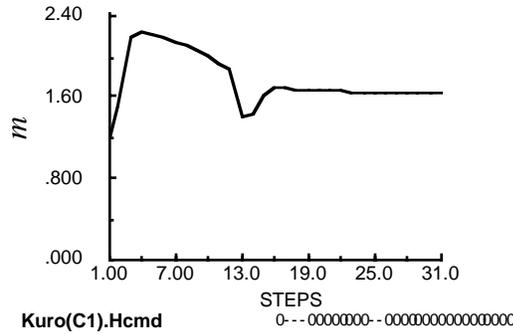
c) Positive Null Motion; 88 Nodes; $c = -4.13$



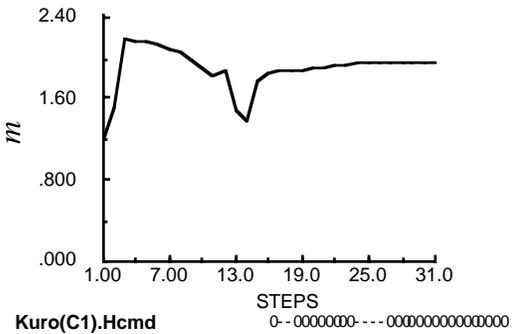
d) Gradient; 116 Nodes; $c = +21.2$



e) Search Result; 185 Nodes; $c = +23.5$

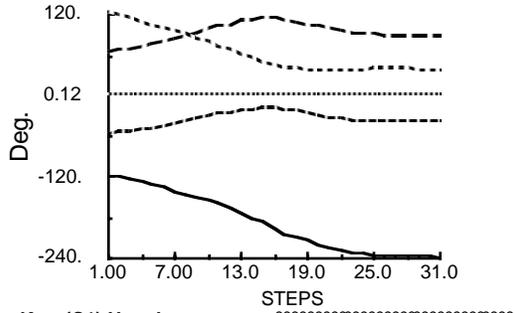


f) Search Result; 499 Nodes; $c = +23.8$

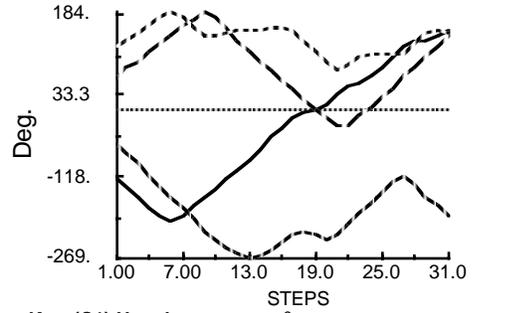


g) Search Result; 965 Nodes; $c = +24.0$

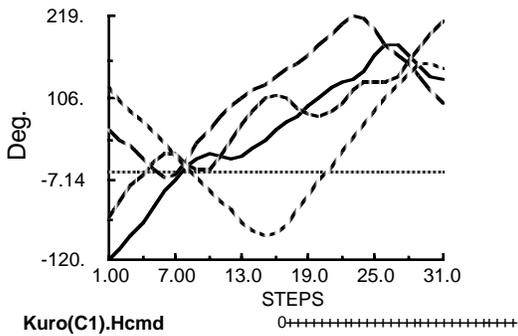
Figure 71: Evolution of CMG Gain for "Kuro C"



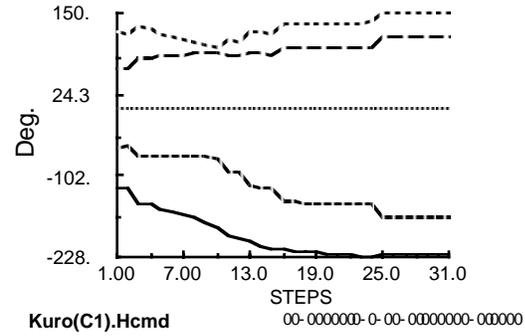
a) No Null Motion; 30 Nodes; $c = -20.6$



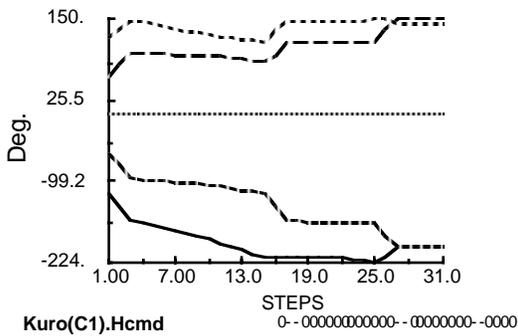
b) Negative Null Motion; 59 Nodes; $c = -5.53$



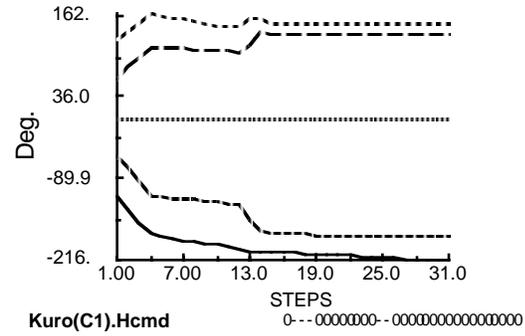
c) Positive Null Motion; 88 Nodes; $c = -4.13$



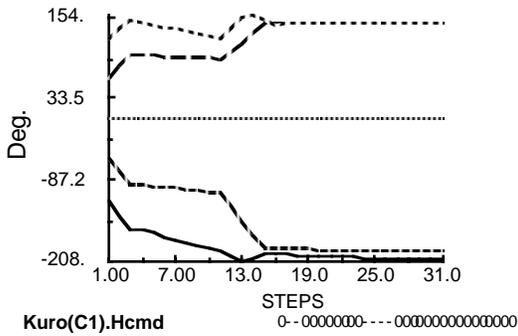
d) Gradient; 116 Nodes; $c = +21.2$



e) Search Result; 185 Nodes; $c = +23.5$

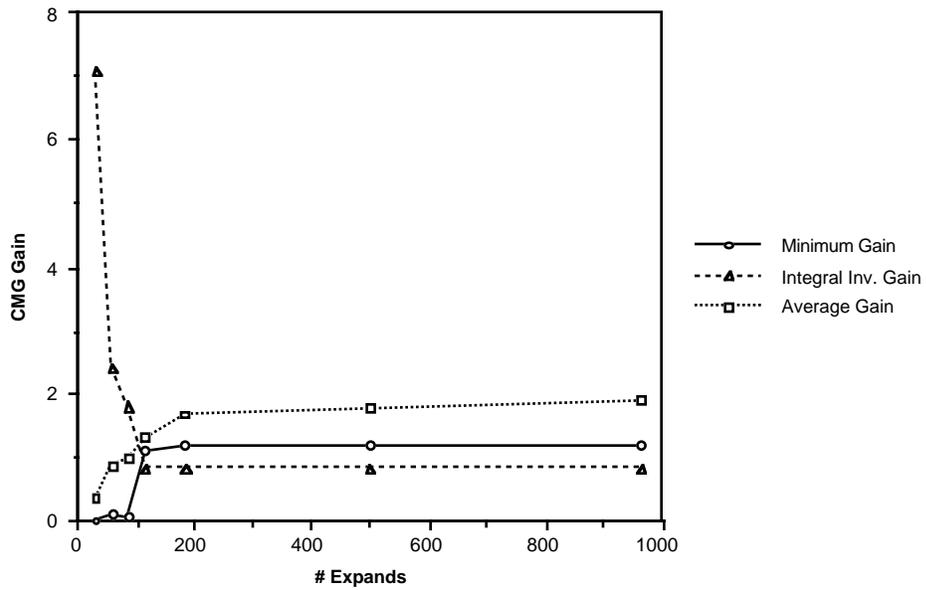


f) Search Result; 499 Nodes; $c = +23.8$

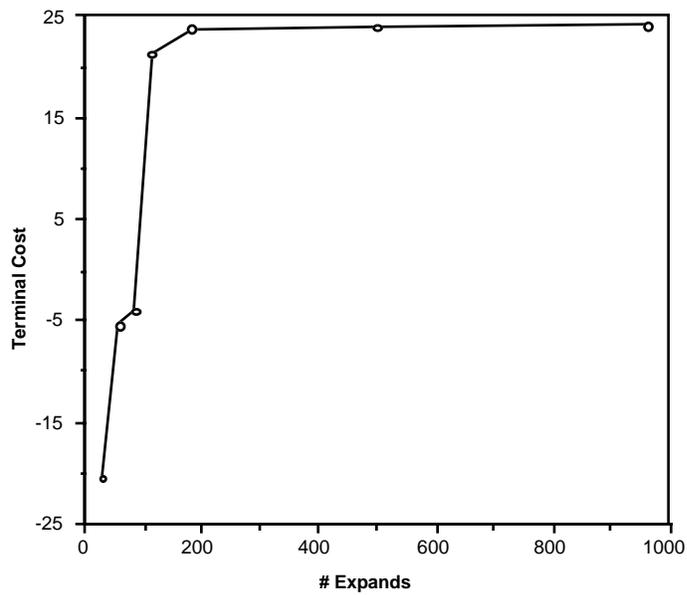


g) Search Result; 965 Nodes; $c = +24.0$

Figure 72: Evolution of CMG Gimbal Angles for "Kuro C"



a) CMG Gain Improvement



b) Trajectory Costs

Figure 74: Summary of Search Operation for "Kuro C"

start of the command sequence, but dips toward the singularity at the end. The best trial performance is from the "unkinking" attempt (d), adapted from Ref. [34]. This succeeds in "snapping" the rotors through the singular condition and into a better closure by applying an extended pulse of negative null motion at the start of the sequence, leading to much higher m later on. Despite the fact that this "unkinking" trajectory is much better than its predecessors, the quick passage through the singularity makes it prone to instability and sensitive to disturbances, as will be demonstrated in the next section.

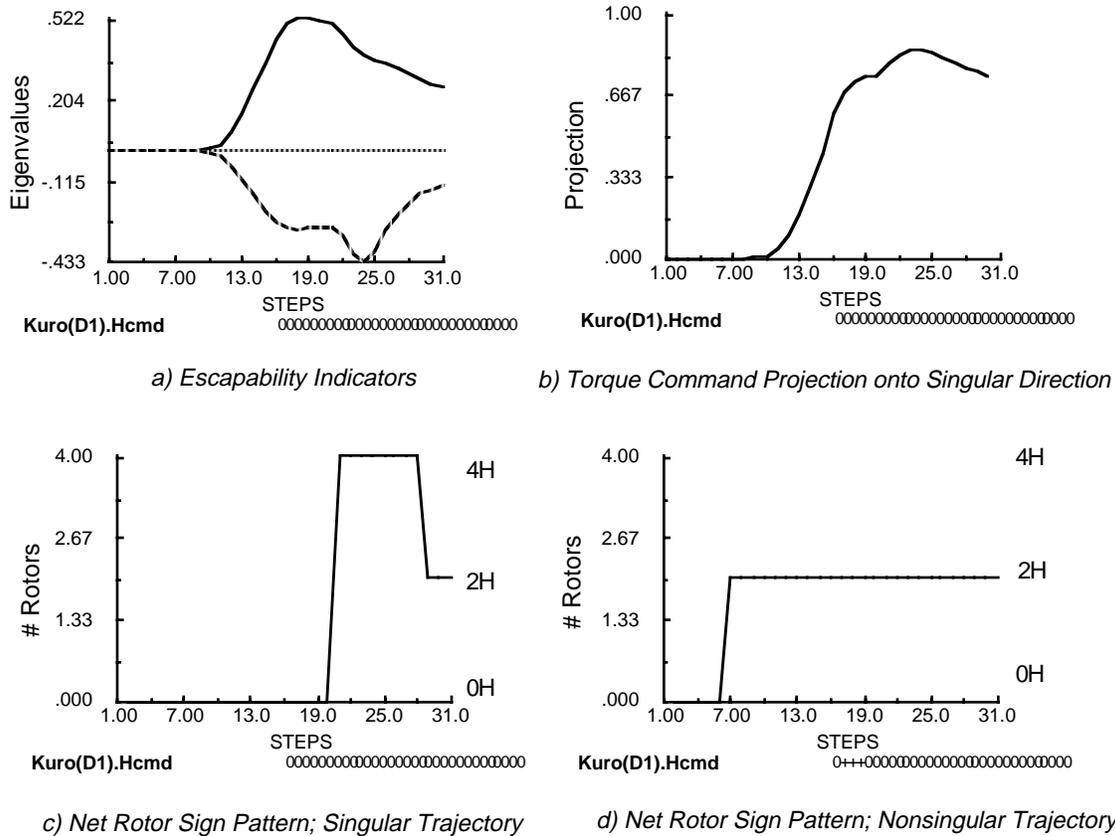
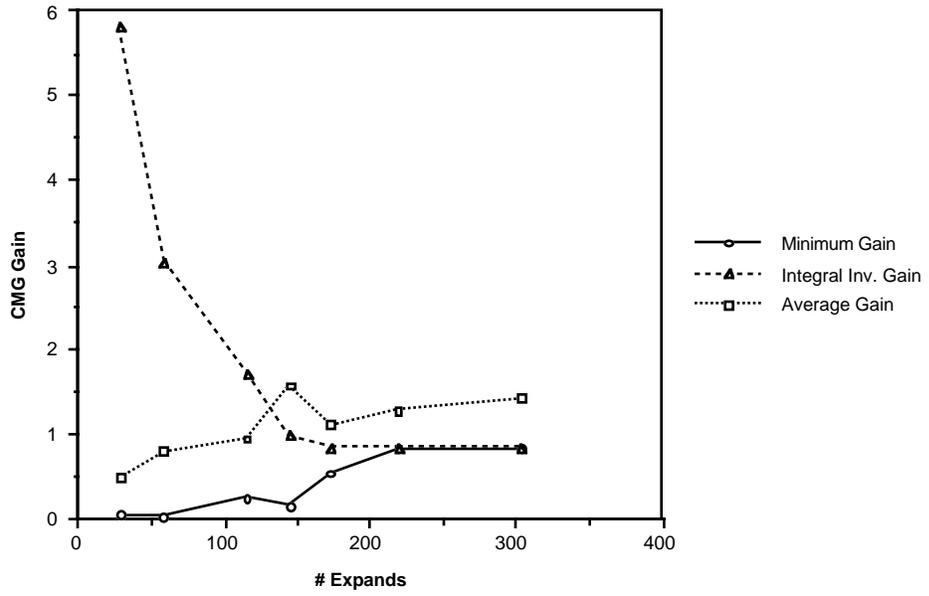
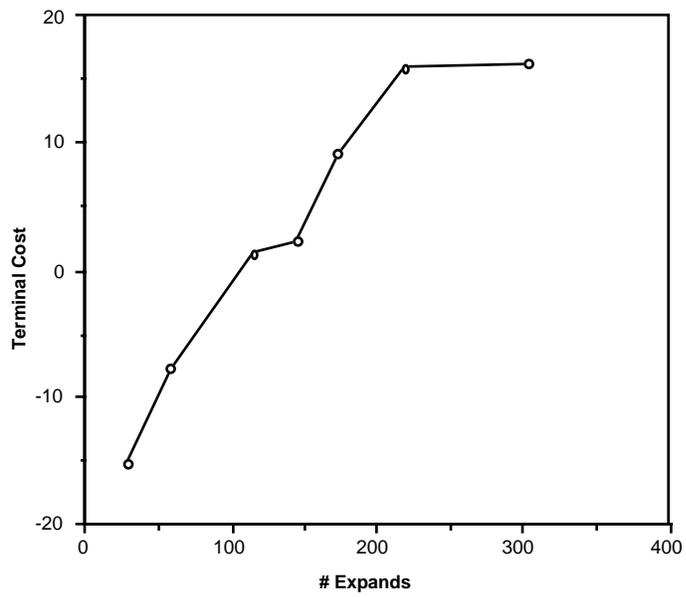


Figure 76: Singularity Analysis for "Kuro D"

The search quickly found another way around the singular encounter, however, as can be seen in the plots e-g. Looking at the final accepted trajectory (g), we see that an initial brief burst of positive null motion produced an high-gain CMG state, which dropped somewhat (yet remained comfortably nonsingular; m never fell much below 0.8, as seen in Fig. 78g) before rising again later in the sequence.



a) CMG Gain Improvement



b) Trajectory Costs

Figure 81: Summary of Search Operation for "Kuro D"

The search found a "threading the needle" strategy, similar to that discussed with Fig. 23 and seen in the example with a constant \hat{x} , $-\hat{y}$ torque (Figs. 40-46). The results here, however, were much better; the search was able to locate a path in this example that comfortably skirted the singularity, and did not pass directly through a singular state in order to abruptly switch closures and reconfigure for higher CMG gain. Section 4.4 will demonstrate the robustness of such trajectories when implemented with unmodeled disturbances.

The parametric summary of search operations is given in Fig. 81. Although the four initial accepted trials improved the trajectory cost significantly, the trajectories found by the search yielded a considerable objective increase, mainly due to their higher minimum gain. This example was run using the same set of initial gimbal angles as in the previous test.

4.3) Comparison with Local Steering Law

The previous section showed how the directed search was able to locate CMG gimbal trajectories that avoided singular states or worked to minimize the singular encounter. In this section, a performance comparison is made with a local "tangent-based" null motion algorithm. The tangent procedure applied here is based upon the "second inverse gain" method of Refs. [13,52], which was discussed in Sec. 2.4. This technique adds "undirected" null motion (i.e. not along any gradient of m) throughout the gimbal trajectory, in an attempt to keep the CMG system from being tracked into a singularity. This method was seen to work best with 4-SGCMG systems in Refs. [13,52], and generally performs better than other tangent methods (gradient and non-gradient) with the momentum commands used in this study (for the purposes of running such comparisons, the disturbance analysis package developed under this effort incorporates a version of the software used in Ref. [13]). The performance of the tangent algorithm used in these simulations can vary, depending upon the weights, thresholds, and integration intervals that are applied. This implementation of the algorithm may thus not be optimal, and these examples are not intended as a judgement on that technique; they are included to illustrate the difference between results obtained from a "good" steering law using local information only and a "global" steering algorithm that uses a search to optimize over entire gimbal trajectories.

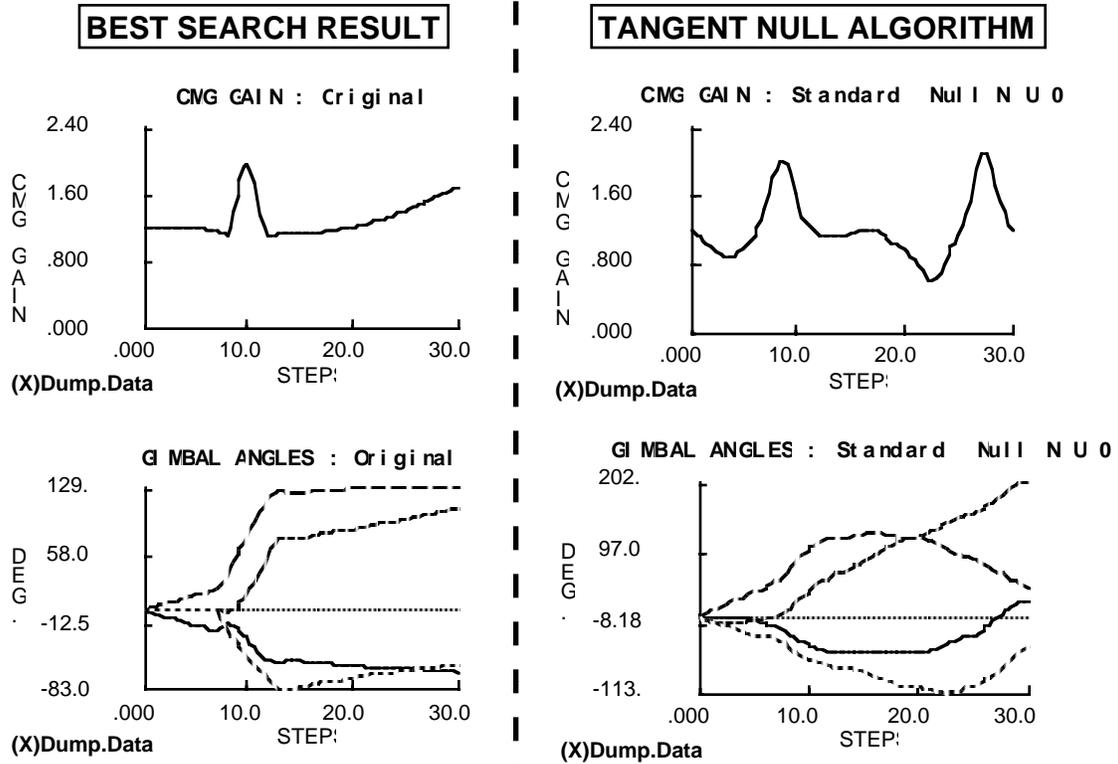


Figure 82: Search vs. Tangent Steering Law for Constant Torque Along \hat{x} Axis

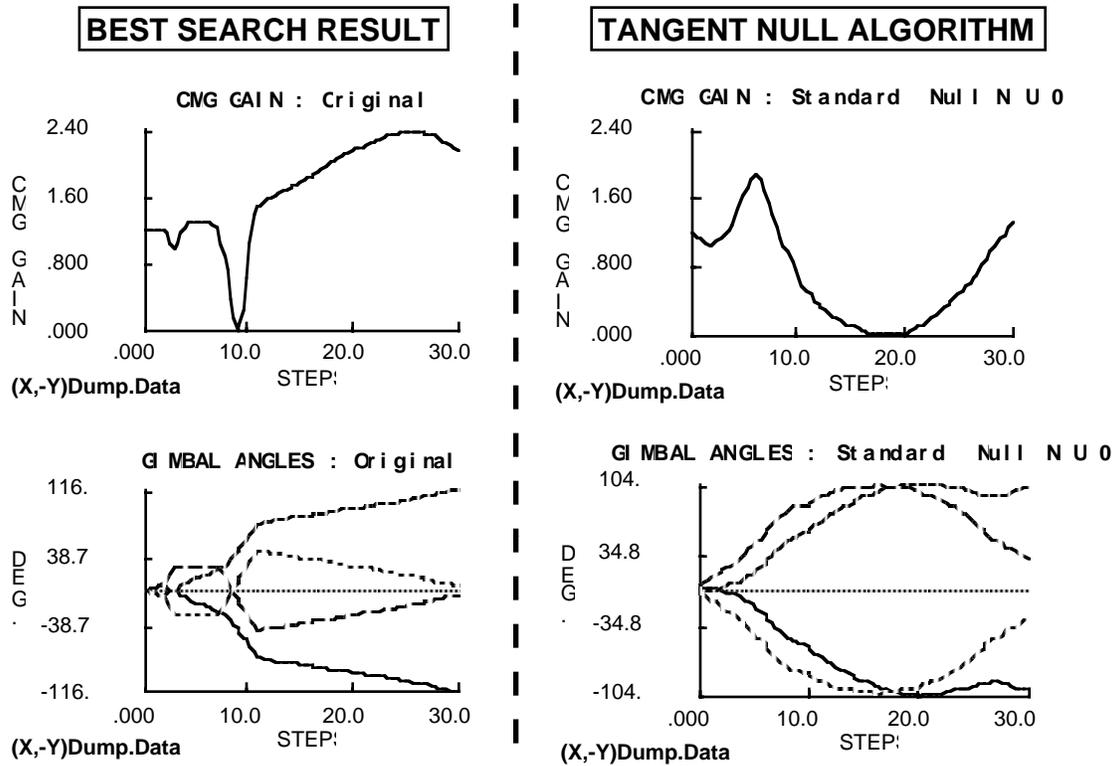


Figure 83: Search vs. Tangent Steering Law for Constant $\hat{x}, -\hat{y}$ Torque

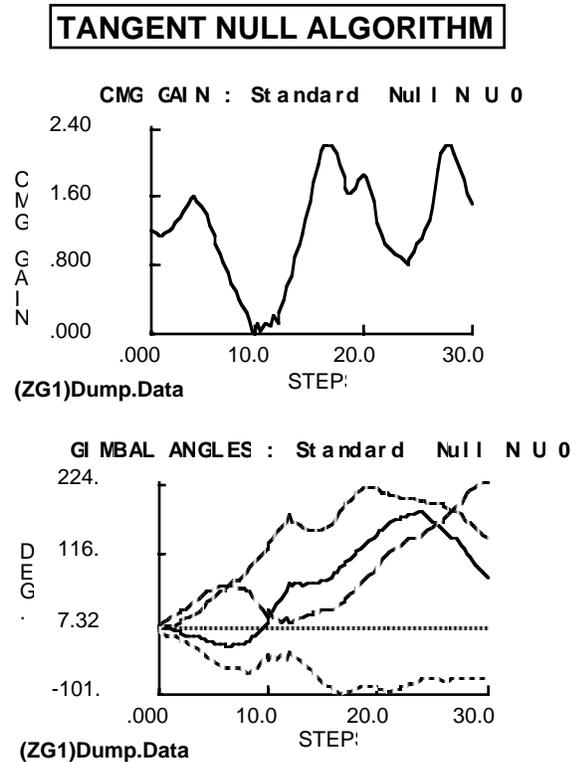
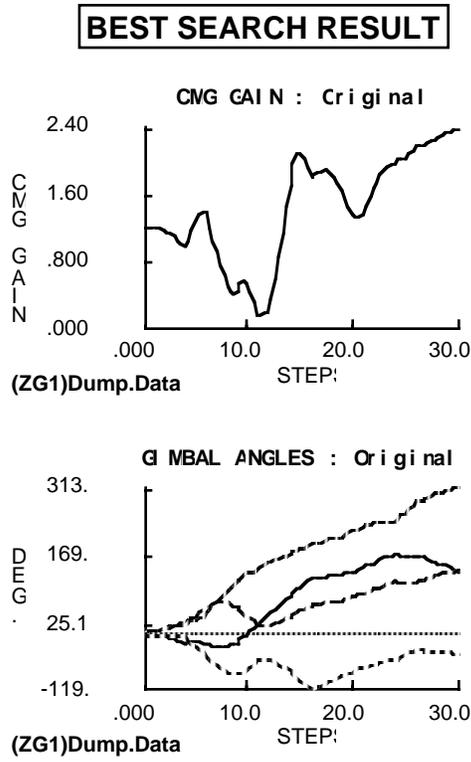


Figure 84: Search vs. Tangent Steering Law for Zig-Zag Sequence

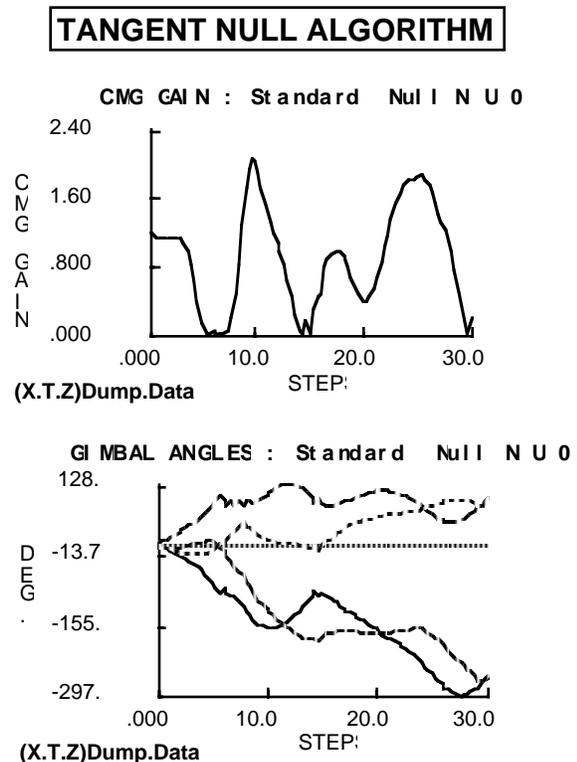
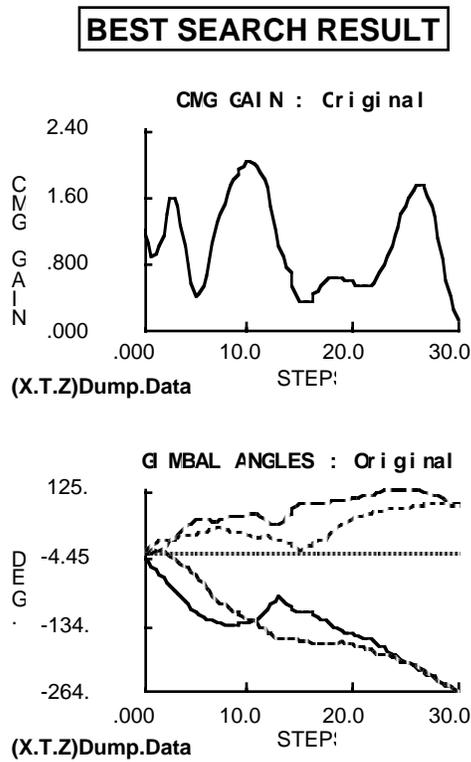


Figure 85: Search vs. Tangent Steering Law for " \hat{x} then \hat{z} "

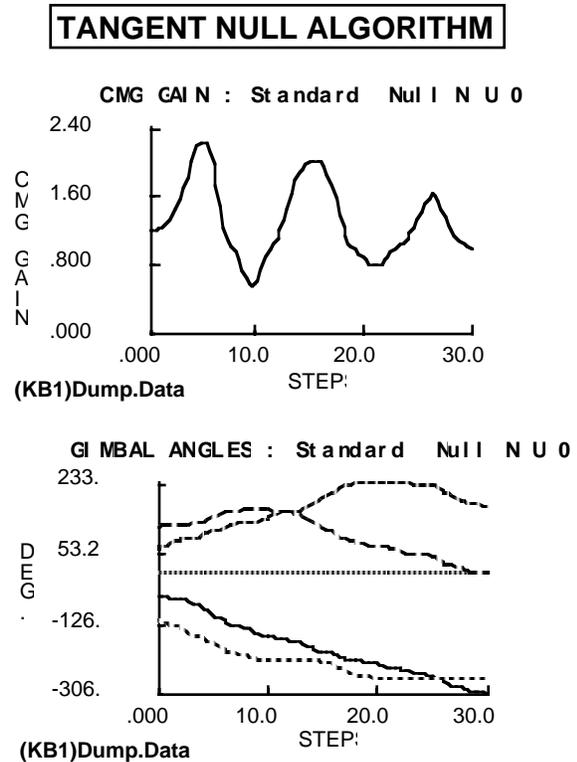
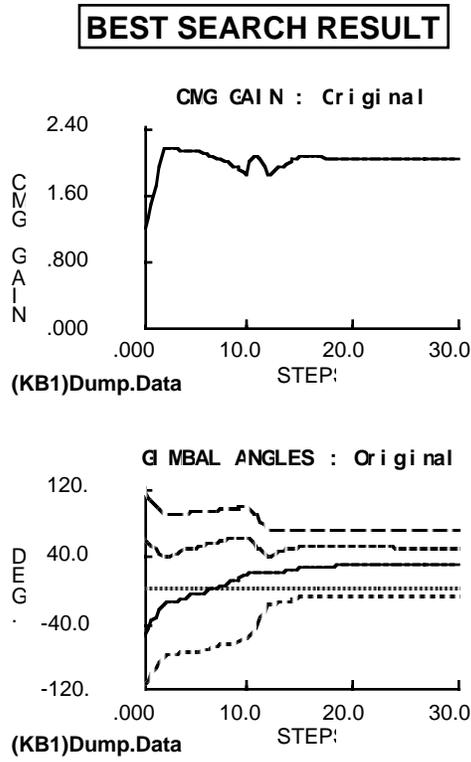


Figure 86: Search vs. Tangent Steering Law for "Kuro B"

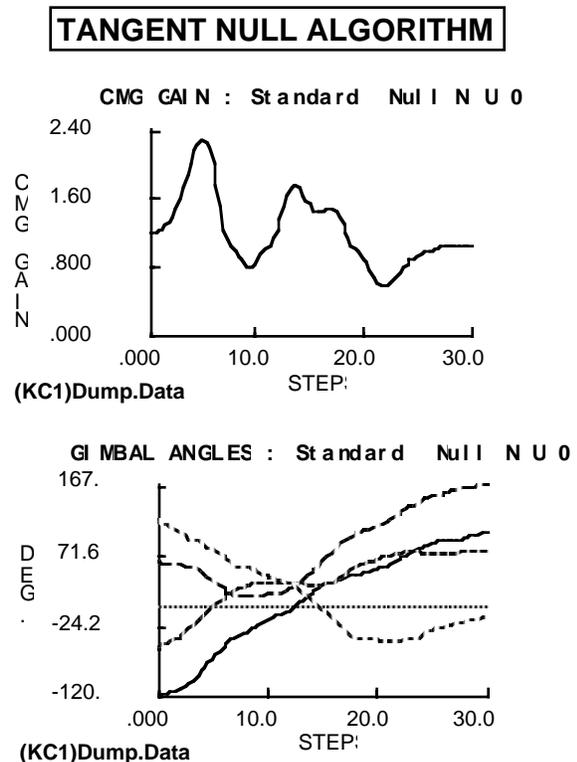
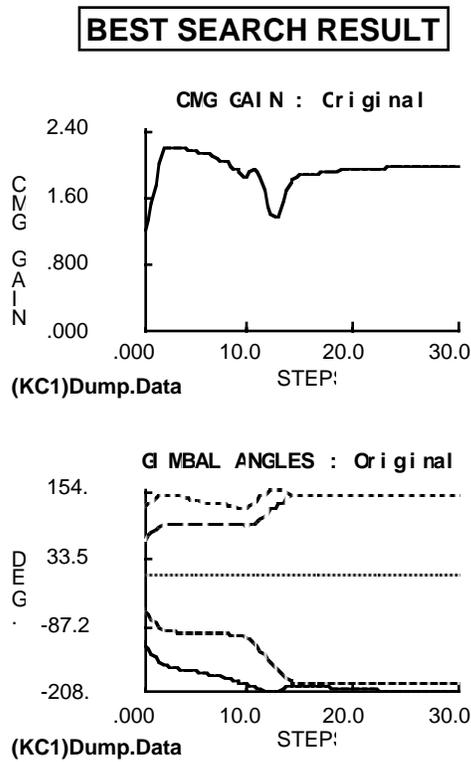


Figure 87: Search vs. Tangent Steering Law for "Kuro C"

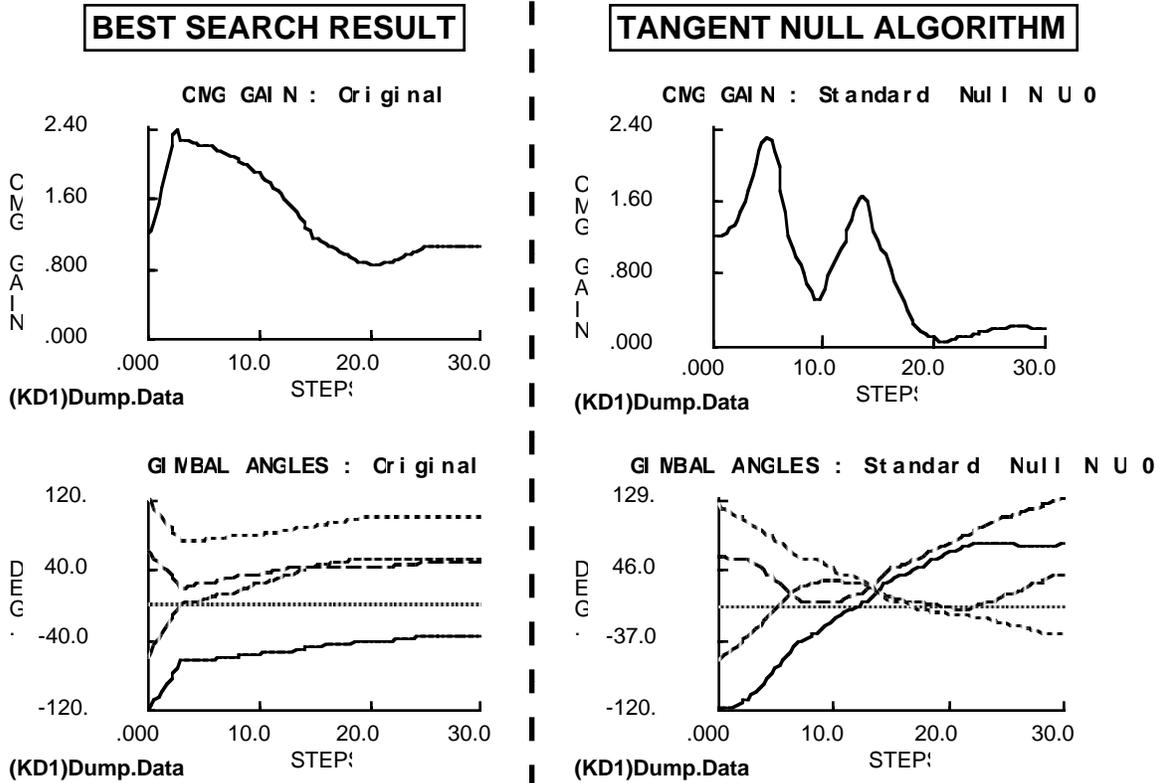


Figure 88: Search vs. Tangent Steering Law for "Kuro D"

Results of this comparison are shown in Figs. 82-88 for each of the momentum command sequences examined in Sec. 4.2. Because the tangent steering law used here does not direct its null vector along ∇m , it freely traverses regions of high and low gain, but does not generally become channeled directly into singular states (and locked-up), as frequently happens with gradient-based approaches.

In general, a significant difference can be noted between the two sets of results. The search trajectories (left columns) either avoid low m states, or minimize the duration of these encounters. The tangent law results (right column), however, generally reach lower m and remain in these less controllable states for a longer duration. Since the search uses global information to avoid these low m configurations, its superior performance is expected.

Looking at particular examples, Figs. 82,86,87,88 show a lower m to be attained by the tangent law. Fig. 83 shows that a singular configuration is encountered with both the tangent law and search trajectories, but the search has minimized the singular duration, passing through the singularity quickly in order to "snap" into an alternate closure. The results of Figs. 84 & 85 indicate some similarity between the search and tangent law; these momentum commands required fairly large gimbal rates to realize the commanded torques,

thereby leaving less margin to introduce null motion for avoiding singularities. Nonetheless, the search results for these examples still maintain higher m values, although trajectories such as these, that approach regions of small m , can be sensitive to disturbances, as will be demonstrated in the following section.

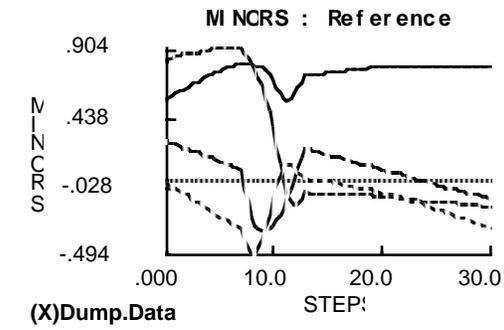
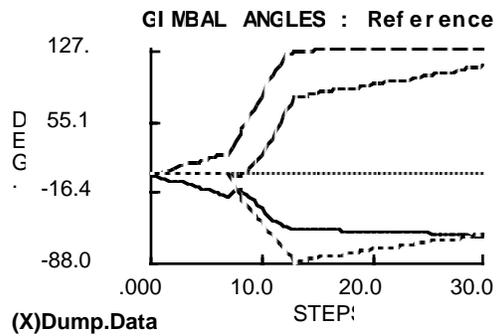
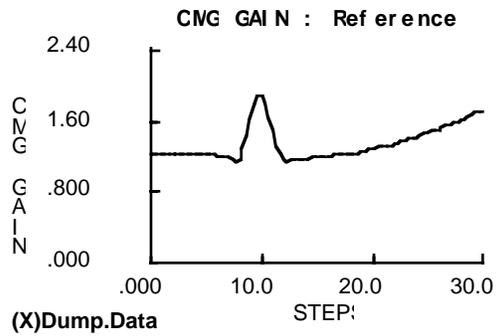
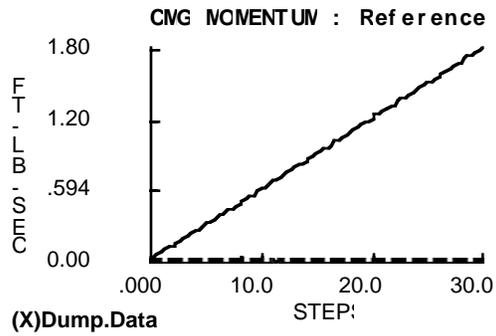
4.4) Sensitivity to Unmodeled Disturbances

The orbital environment can be difficult to accurately predict on an orbit-to-orbit basis. Dynamically changing aerodynamic torques (depending upon solar activity and upper atmosphere behavior) can affect the precision of expected momentum profiles used with momentum management routines and the search-based steering scheme. While certain applications (i.e. a fast maneuver on a well-determined spacecraft model) may be able to produce feedforward momentum trajectories with little error, any solution to the momentum management problem must deal with significant uncertainty.

A software package was developed that examines the sensitivity of feedforward gimbal trajectories to disturbance torques that were not included in the original momentum command sequence. This program essentially executes the implementation logic that was suggested in Fig. 30. A local steering law answers instantaneous torque requests (derived from the momentum command plus disturbance torque) through an SR-inverse, using a signed null amplitude \bar{k} that is output from the CMG search. The "Unmodeled Disturbance Torques" of Fig. 30 are injected as a constant secular torque that can be defined by the user. The steering law is iterated (i.e. integrated) three times per momentum step in these examples (vs. two times per step in the CMG search that produced the feedforward gimbal trajectories). The details of this program are given in the Appendix.

Fig. 89 shows a comparison of CMG gimbal trajectories followed without disturbance injected (the left column, with plots labeled "Reference") and with a secular torque injected along all three axes (right column, with plots labeled "Perturbed") for the momentum command sequence realizing a constant torque along the \hat{x} axis. The "Reference" plots (at left) are identical to those produced by the best search trajectory in Figs. 36-38 (the jagged character of the lines are aliasing artifacts introduced by the Macintosh graphics; the original curves are smooth).

Original Momentum Command



Perturbed Momentum Command

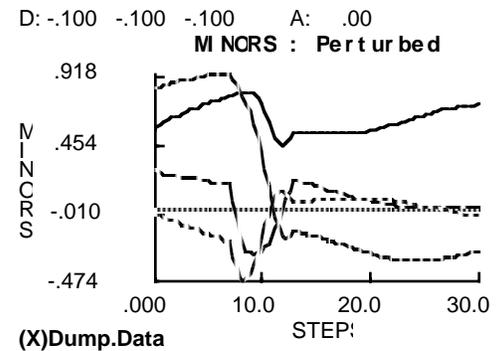
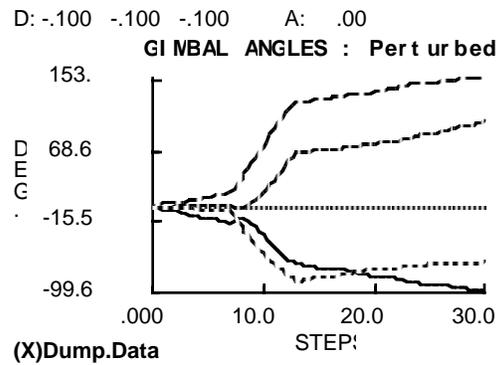
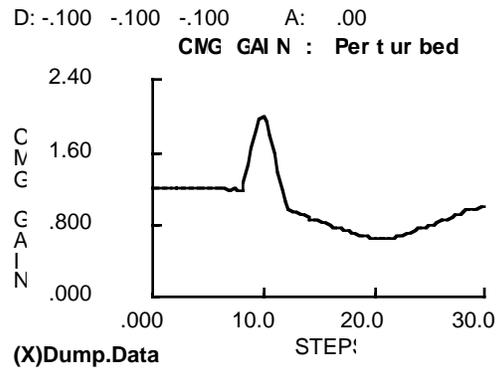
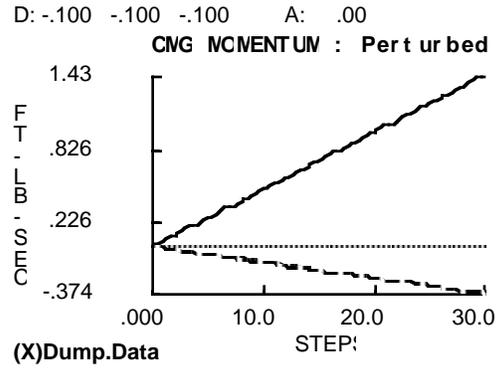


Figure 89: Ref. vs. Perturbed Performance for Constant Torque Along \hat{x} Axis

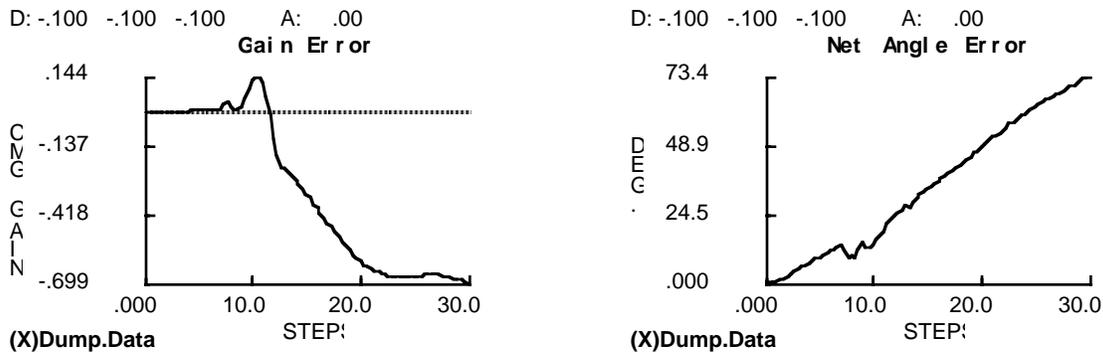


Figure 90: Trajectory Divergence for Constant Torque Along \hat{x} Axis

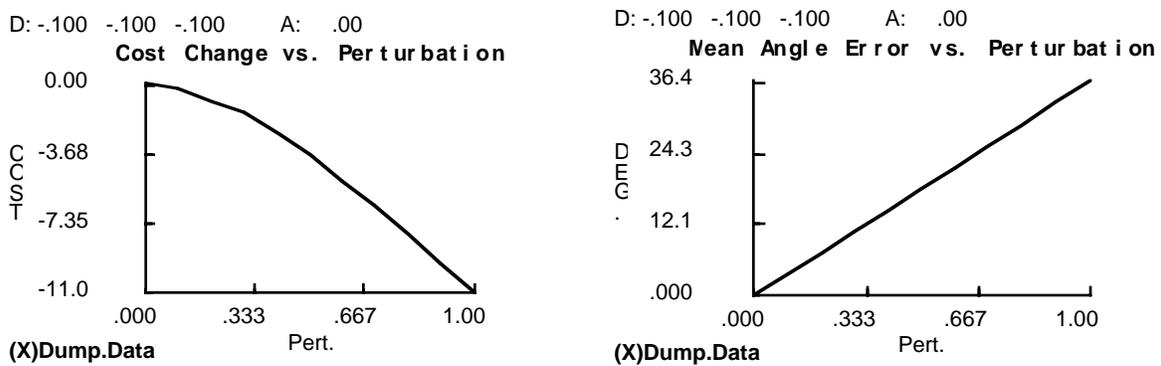


Figure 91: Perturbation Sensitivity for Constant Torque Along \hat{x} Axis

The perturbed momentum command was realized by the CMG array, as plotted in the upper right of Fig. 89. The components of the torque perturbation vector are listed atop each relevant plot as "D" (the adjacent "A" value refers to a shift in initial gimbal angles, which is zero in all of these examples). These are expressed as the fraction of the net CMG momentum that will be added to the original momentum command at the end of the command sequence. Since these tests assumed 4 CMGs of unit momentum each, the "0.1" entails a perturbation on the order of 0.4 momentum units per axis (actually, the code doesn't add a perturbation into the last two steps, thus the momentum shift in this case is $[0.4 * 29/31] = 0.374$, as plotted in the upper right of Fig. 89). This momentum error is distributed over the entire maneuver sequence (barring the last 2 steps) as a constant secular torque of magnitude $0.374/29$. Although the momentum units are labeled as being "ft-lb-sec" in Fig. 89, this scaling is actually in the units defined by the CMG search program, which assumed each CMG to have a momentum of 1. Adding the perturbed components of the final momentum vector in quadrature and scaling by the original

unperturbed value, the example of Fig. 89 is seen to introduce a momentum shift of approximately 35%.

The m values for the normal and perturbed gimbal trajectories are given as the "CMG Gain" in the second row of Fig. 89. The same general characteristics can be seen in each case; a gain peak around step 10 (brought on by a null motion pulse that was used to change closures), followed by a slowly changing m profile. The major difference is the way in which m begins to dip with the perturbed command after the null pulse at step 10, whereas it gradually increases in the original example (the null perturbation "D" was chosen here to drop the m value; if the signs on "D" are reversed, m will grow by a similar amount in this example).

The gimbal angles are given in the third row of Fig. 89. Little difference is seen in the general character of these plots; the gimbal angles seem to follow similar profiles. This is also reflected in the determinants of the four 3x3 minor matrices of the CMG Jacobian, which are plotted in the lower row of Fig. 89. The form of normal and perturbed plots are similar. No major shift in minor sign (indicating a closure change) is seen, excepting one value (the finely dashed line), which does not go negative after step 15 in the perturbed plot (this minor stays near zero and doesn't diverge in both cases, indicating that it plays a small role in determining the needed torque here).

Fig. 90 shows how the two trajectories diverge as a function of time. The change in m (perturbed-normal) is plotted at left. One can see that the major difference occurs after the null pulse at step 10, where a closure change occurred. This indicates that the closure achieved by the CMG array after step 10 will decrease m when subjected to the perturbed momentum commands. The change in gimbal angles (the sum of absolute differences between normal and perturbed angles) is plotted at right. Here, one sees a nearly linear divergence between normal and perturbed responses (barring some jitter during the null pulse, which is mainly due to the finer integration used here). This indicates the lack of any type of "crisis" point, where the perturbed system will diverge quickly from the reference; errors are accumulated in a linear fashion here.

The results of Fig. 91 take this point further. Here (at left), we plot the net cost change (the terminal cost of Eq. 16 without the null cost, residual, and gimbal over-rate contributions from Eq. 15) as a function of perturbation fraction. Results are plotted for eleven trajectories that are run for the specified perturbation vector ("D") having magnitude scaled between zero and one (as plotted on the horizontal axis). The left-hand plot shows a gradual, almost linear change in net cost as a function of the perturbation fraction. The plot shown at right gives the mean angle error (sum of angle differences at each scheduled momentum value, scaled by the number of points in the momentum profile) as a function of

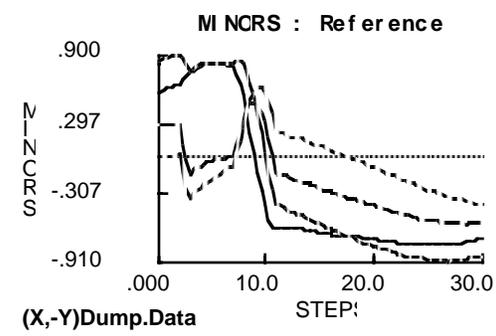
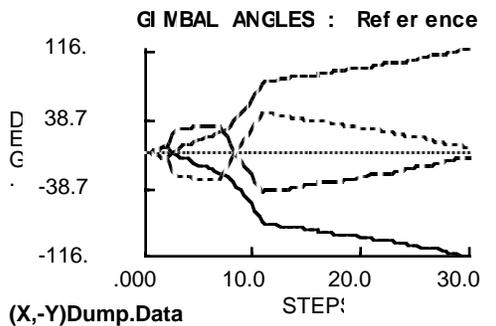
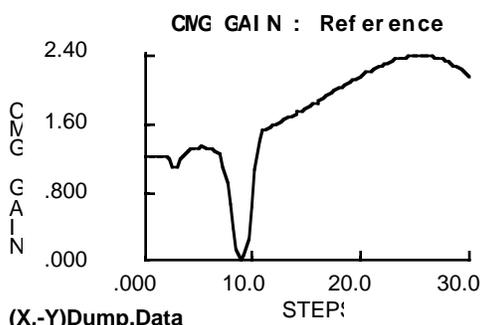
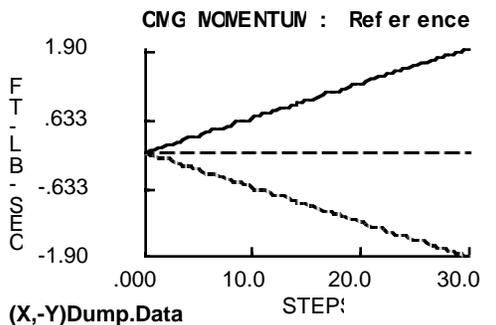
perturbation fraction. This certainly appears linear, indicating that the difference between normal and perturbed trajectories is proportional to the perturbation amplitude.

Figs. 89-91 seem to indicate that the gimbal trajectory found by the search for this momentum command (constant torque along the \hat{x} axis) is quite robust to secular disturbance torques (the "D" picked for this perturbation was along the worst-case direction). Differences in trajectories are accumulated gradually (nearly linearly) across the time history, and the amount of net difference is proportional to the magnitude of the perturbation. One of the distinguishing features of this gimbal trajectory, however, is its avoidance of singular regions. One may suspect that the situation is different for a trajectory that crosses a singularity. This is examined in the next example, which perturbs the search result for the case of a constant $\hat{x}, -\hat{y}$ torque.

The normal vs. perturbed comparison is given in Fig. 92. The reference results (left column) are identical to those presented earlier for this momentum command (Figs. 43-45). The perturbed results are given in the right column, and we indeed see a significant difference. The "D" vector for the momentum perturbation was directed along the $-\hat{x}$ axis, adding a net momentum disturbance (relative to the reference case) of roughly 15%. This is reflected in the CMG momentum plot (upper right), where it can be seen that the perturbed momentum command is able to be realized by the CMG system.

This is not easily accomplished, however, as can be noted in the plot of m given in the second row of Fig. 92. Here we see similar histories at the very beginning of the command sequence, but the situations diverge considerably after step 8, where a null pulse was introduced to change closures and quickly snap the CMGs through a singular state (as can be noted in the reference plot at left). In the perturbed case, however, the added null motion did not put the CMG system into the appropriate closure, and the system went (and for the most part, stayed) singular after the null motion was added. This difference is also reflected in the plots of gimbal angles (third row). After the reference trajectory zips through its singular encounter (step 8), the gimbal angle profiles evolve very differently, leading to rather large gimbal rates when the perturbed system goes singular (step #20). The mechanics of this change in behavior are apparent in the plots of the minors, given in the lower row. Although the minors appear quite similar before the singular encounter, they emerge very differently afterwards. In the original case, three out of the four minors switch sign (go negative) during the singular encounter at step 10. The perturbed situation, however, exhibits a different effect, where (after some gyration induced by the null pulse) all of the minors stay positive, eventually passing zero (as the system goes singular).

Original Momentum Command



Perturbed Momentum Command

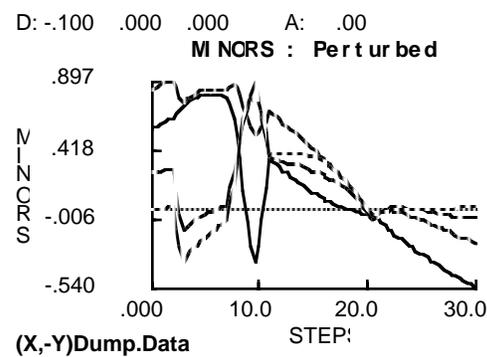
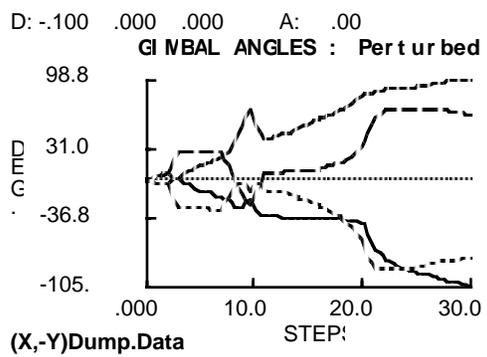
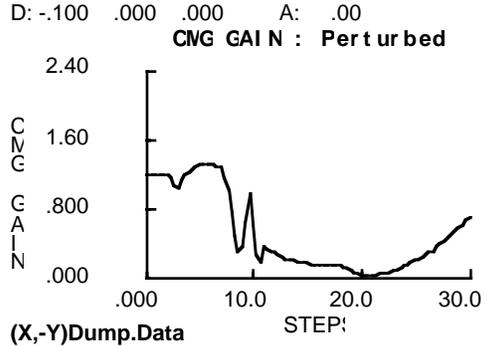
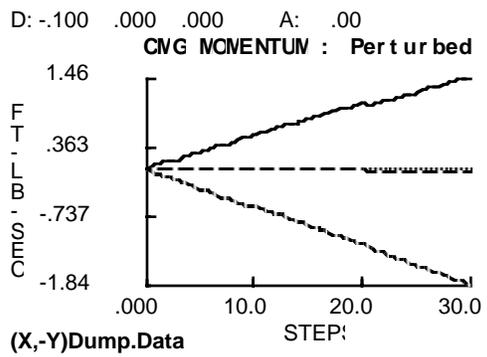


Figure 92: Reference vs. Perturbed Performance for Constant $\hat{x}, -\hat{y}$ Torque

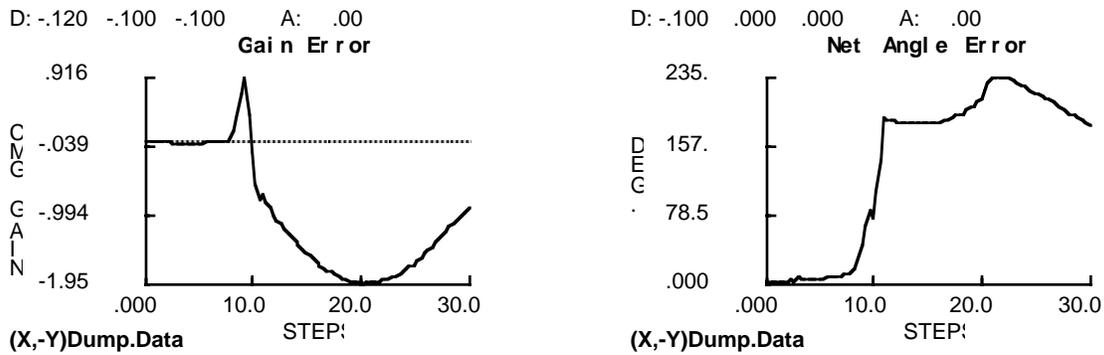


Figure 93: Trajectory Divergence for Constant \hat{x} , $-\hat{y}$ Torque

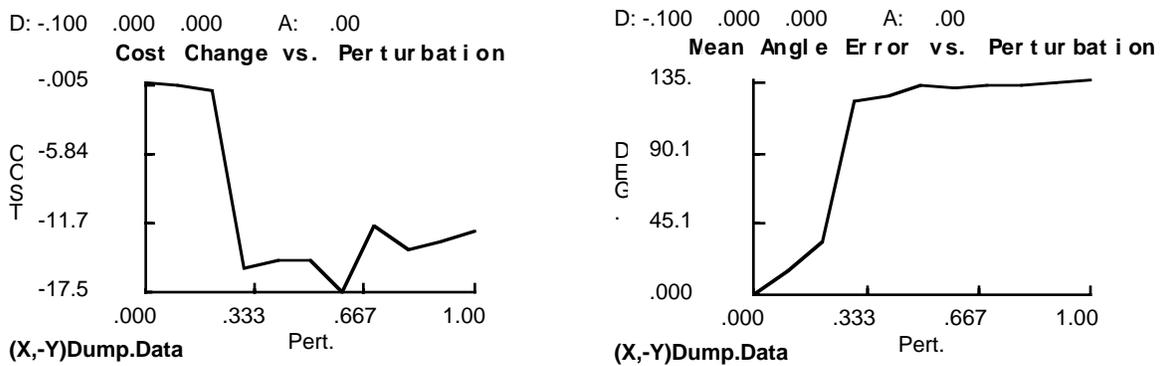


Figure 94: Perturbation Sensitivity for Constant \hat{x} , $-\hat{y}$ Torque

The divergence plots (Fig. 93) summarize the situation. Very little difference is noted between the two trajectories, until the reference hits the singular encounter near step 10. Thereafter, a sharp decrease in m and fast divergence in gimbal angles are seen to occur (contrast this to the previous example [Fig. 90], where the angle error increased almost linearly with time). This divergence is very sensitive to the amount of introduced perturbation, as seen in the sensitivity plots of Fig. 94. A small cost change and limited gimbal divergence is seen for small perturbation amplitudes (up to 0.25 or so in relative scaling), but the cost rapidly jumps (and gimbal divergence abruptly soars) after the introduced perturbation amplitude grows further.

This result indicates that it is very difficult to predict past a singular state; the null motion history \bar{k} prescribed by the search quickly loses relevance after the singular encounter. In the singular region, many CMG closures or gimbal trajectory possibilities patch together, thus a small change in the CMG condition (brought on by answering the unmodeled torques) can select an entirely different gimbal configuration, yielding a rapid

divergence, as characteristic of nonlinear dynamic (i.e. chaotic) systems[62]. This may be interpreted physically. In a singular state, the rotor axes are all aligned to project maximally or minimally along a common axis, hence small perturbations will cause the rotors to "scissor" one way or another, producing very different subsequent gimbal configurations (i.e. closures) that evolve in an entirely different fashion.

Before any additional conclusions are drawn on this matter, another example is examined for its disturbance sensitivity. This is the familiar "Kuro C" momentum sequence, and the normal/perturbed comparison is shown in Fig. 95. Again, the reference plots (left column) are identical to the earlier search results (Figs. 71-73). The "D" perturbation vector is chosen to produce a drop in m , and amounts to a roughly 40% change (relative to the reference) in the final net CMG momentum. As can be noted in the upper-right plot of Fig. 95, the CMG system is able to realize the perturbed momentum command.

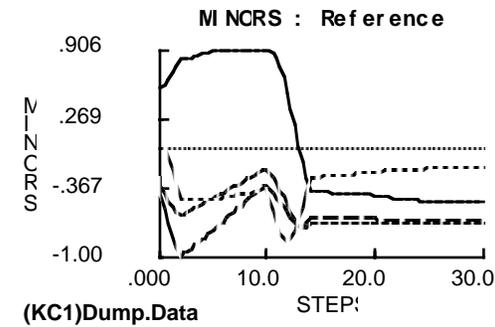
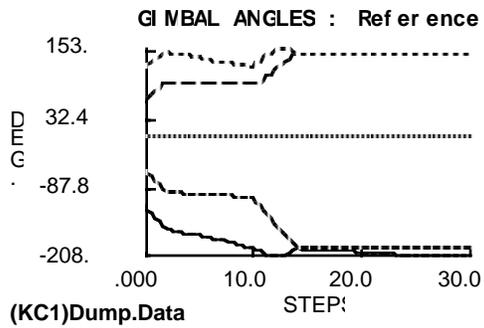
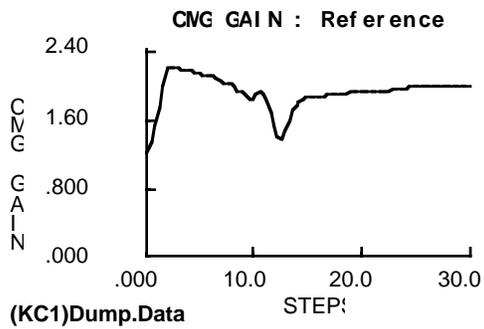
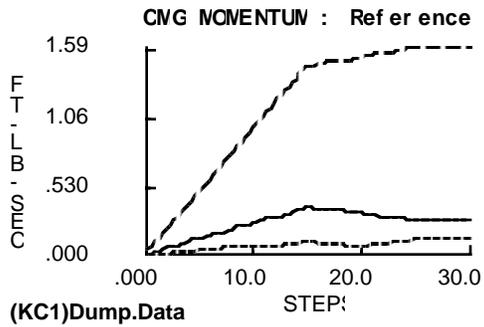
The m profiles have a similar character up to the point at which a null pulse is introduced to change closures (i.e. step 13); thereupon the perturbed case continues along at significantly lower m . This isn't nearly the major change seen in the previous example, as one can note in the gimbal angle profiles (third row), which both have a similar character. The minors (lower row) also reflect the similarities between reference and perturbed trajectories. Some difference can be noted after the null pulse at step 13, but the minors stay all negative in both cases and behave smoothly throughout the trajectory, implying that the same closure is maintained in normal and perturbed situations.

Divergence results are shown in Fig. 96 for this example. A sharp change in m , and break in the net gimbal angle error can be noted at the point where the null pulse was applied (and several minors switched through zero; see Fig. 95), but this alteration is nothing like the divergence that was seen in the previous example (Fig. 93).

The perturbation sensitivity is examined in Fig. 97. Here we see a slow (then linear) decrease in the configuration cost and a linear increase in trajectory divergence (net angle error) with increased perturbation. These results indicate a marked similarity to the first example investigated, the constant torque along the \hat{x} axis (Figs. 89-91), rather than the abruptly diverging behavior noted in the previous test (the constant $\hat{x}, -\hat{y}$ torque; Figs. 92-94). Again, the reference for both this trajectory and the trajectory of Figs. 89-91 maintained a large m throughout the command sequence, while the reference trajectory of Figs. 92-94 passed through a singular state.

A final example is given to push this emerging principle a bit further. The same reference trajectory is used ("Kuro C"), but now the perturbation vector "D" is doubled (but kept oriented in the same direction), leading to a huge perturbation, of order 80%.

Original Momentum Command



Perturbed Momentum Command

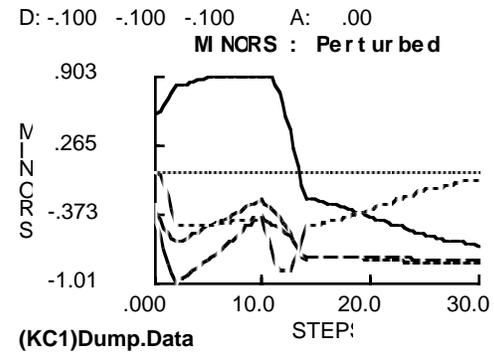
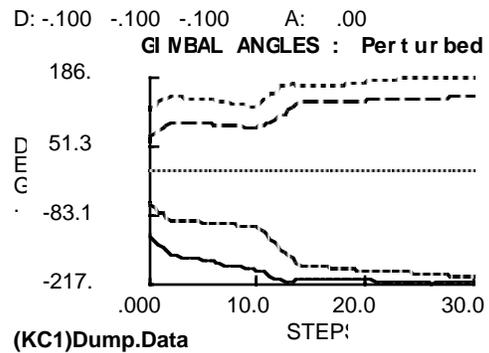
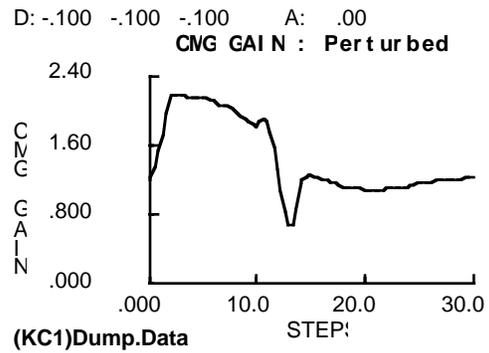
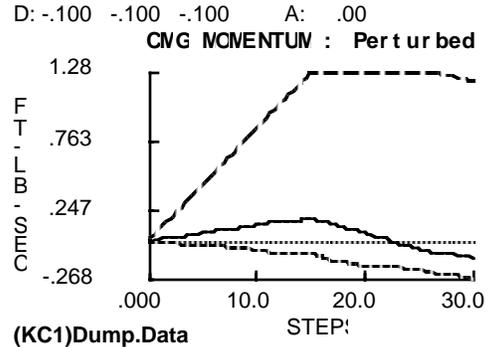


Figure 95: Reference vs. Perturbed Performance for "Kuro C"

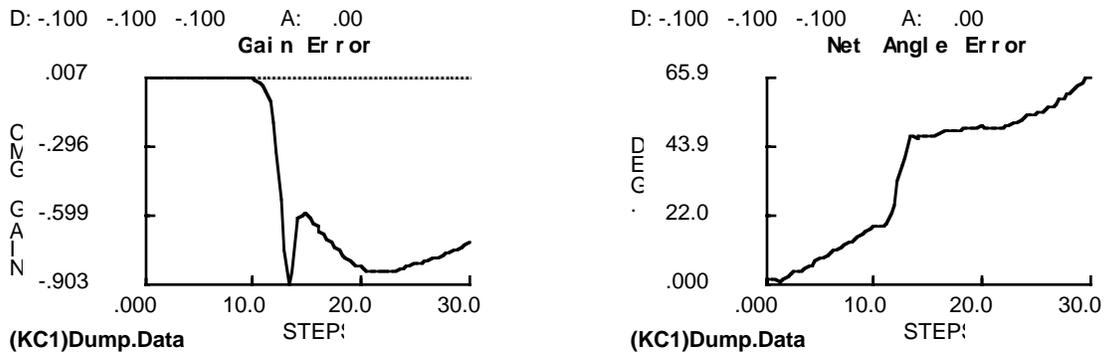


Figure 96: Trajectory Divergence for "Kuro C"

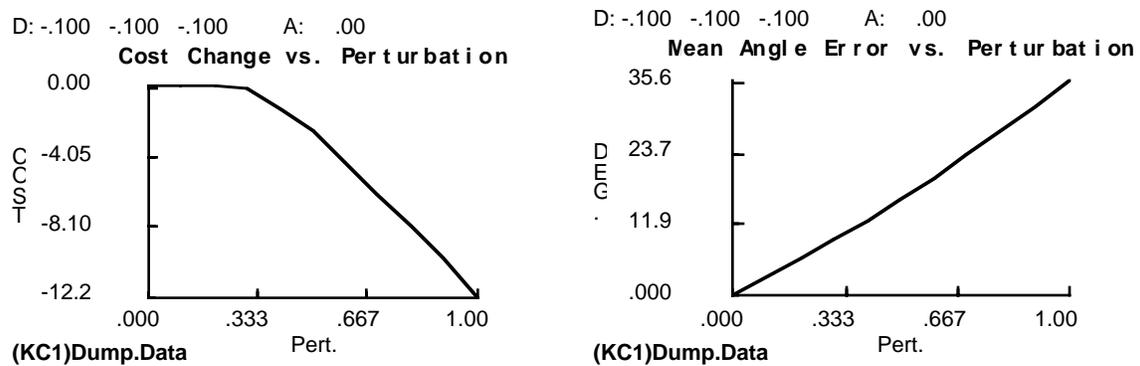
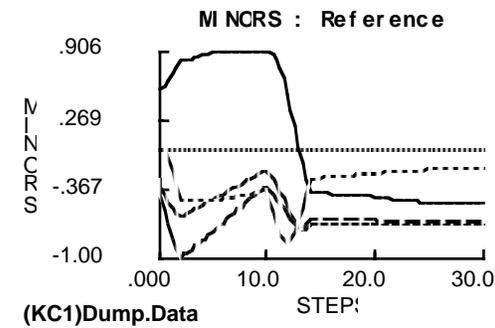
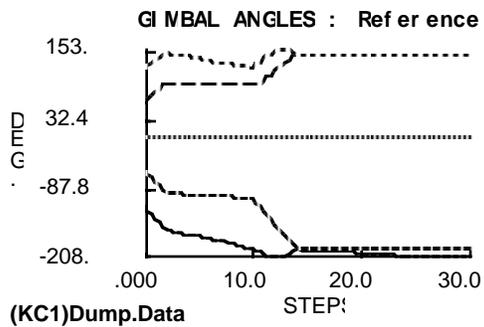
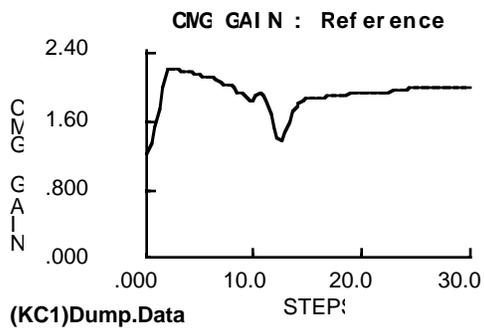
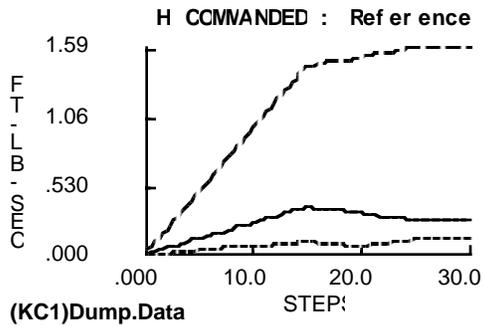


Figure 97: Perturbation Sensitivity for "Kuro C"

The reference-vs.-perturbed comparison is given in Fig. 98. The left-column reference plots are, of course, identical to those presented with Fig. 95. The right-column perturbed results, however, now differ substantially. The perturbed momentum is seen to be realized by the CMG system, although the m values (second row) now stay quite low after the null pulse is issued, and the two gimbals that start at $\pm 60^\circ$ remain at large angles (third row). Note that these two "middle" gimbals are now brought to smaller angles after the null pulse (i.e. $>$ step 15), in contrast to the reference and perturbed cases in the previous example (Fig. 95), where they drifted to larger angles. This suggests a closure shift, as is clearly seen in the minor plots (lower row); after the null pulse, three minors swapped sign in the perturbed case relative to the reference, indicating a different closure.

The trajectory divergence plots of Fig. 99 show a definite impulsive change after the closure switch at step 12; the magnitudes of this step difference are now much larger than seen with the smaller perturbation (Fig. 96).

Original Momentum Command



Perturbed Momentum Command

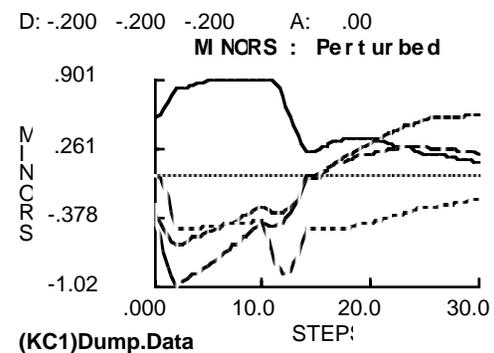
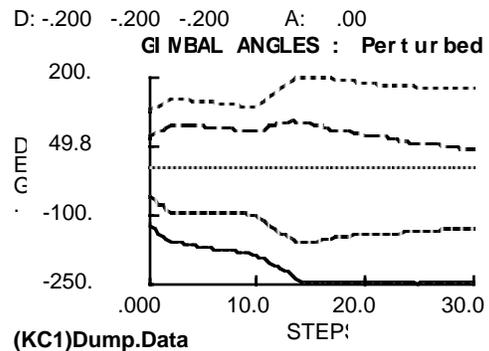
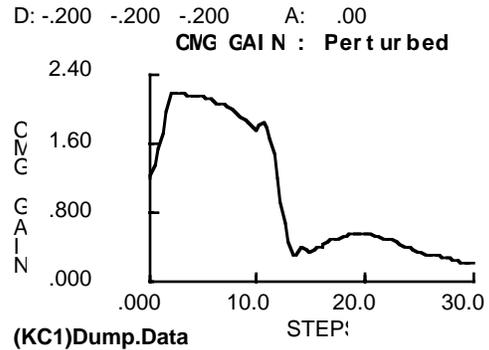
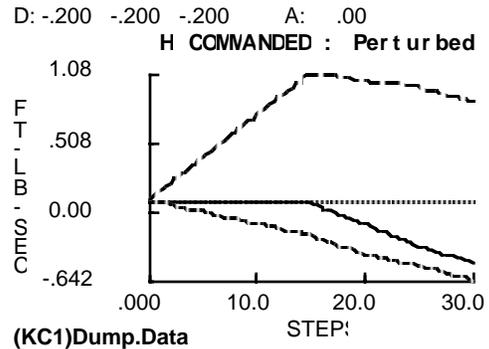


Figure 98: Ref. vs. Perturbed Performance for "Kuro C" (doubled disturbance)

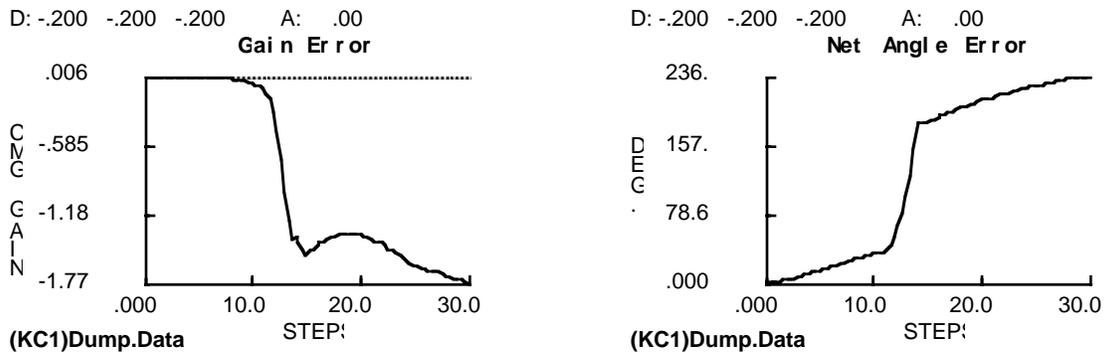


Figure 99: Trajectory Divergence for "Kuro C" (doubled disturbance)

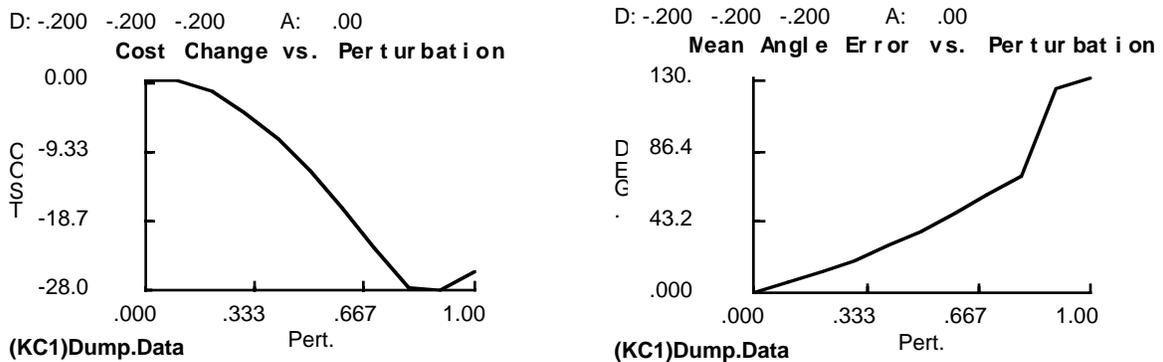


Figure 100: Perturbation Sensitivity for "Kuro C" (doubled disturbance)

The perturbation sensitivities (Fig. 100) show the occurrence of the closure change. A discontinuity can be noted at a disturbance scaling of 0.7, where the angular divergence and cost curves abruptly switch slope, signaling the achievement of a different closure.

These results again suggest that gimbals trajectories avoiding singular regions can be quite robust to the introduction of unmodeled momentum errors. In such situations, the net divergence of perturbed from reference trajectories generally appears to vary nearly linearly with the magnitude of the applied perturbation. After significant error is introduced (in the latter case examined, an 80% net disturbance), a closure change can be instigated and more rapid divergence experienced, but the region of relative stability around the reference trajectory seems to be substantial. In cases where the reference trajectory skirts a singular state, however, divergence can accumulate rapidly after the singular encounter, when a different closure is achieved. This behavior is very sensitive to the amount of injected perturbation; a large divergence in singularity-traversing gimbals trajectories can result from

a relatively small momentum error (i.e. a 3% net disturbance in the example examined here; Fig. 94).

Although the divergence in the neighborhood of a singular state can be problematic for gimbal solutions that attempt to transit a singularity, several possibilities exist for recovery, as were discussed in Sec. 3.4. One factor is a considerable aid; the locations of probable rapid divergence are known ahead of time to be at the gimbal states where the search had indicated proximity to a singularity. For situations where the plant and environment are well-modeled, it may be possible to implement the feedforward search trajectory without alteration (i.e. the local feedback in Fig. 30 is not needed). A divergence contingency is required, however, with uncertain plants & environments, where the logic of Fig. 30 is used to implement the feedforward search commands.

A major difficulty in implementing singularity-skirting trajectories is in managing the inverse-kinematic calculations across the singular region such that the desired closure (located by the search) is entered after the singular encounter. As a singularity is approached, the pseudoinverse (also to a lesser extent SR-inverse) and null algorithm calculations can lose accuracy. In addition, several "closures" can patch together through singularities, as mentioned earlier, hence small errors in calculations (coupled with unmodeled torque disturbances) can cause any of several closures to be entered after the singular crossing. The CMG system must thus be somehow be constrained in crossing the singular region. This constraint might be implemented in a variety of different fashions. One suggestion (for cases where the search trajectory dictates a brush with a singularity) may be to estimate the disturbance torque just before the singularity is hit, while implementing the feedforward gimbal solution. The gimbal configuration on the other side of the singularity (which is hopefully in the desired closure) can be modified by a pseudoinverse calculation that accounts for the estimated momentum disturbance at that point. This will yield a set of gimbal angles that will (hopefully!) be in the desired closure and provide a good approximation to the required momentum. The CMGs can thus be moved from the current pre-singular gimbal angles to this new set during the "singular" encounter.

This scenario is illustrated in Fig. 101 (which is derived from the example of Figs. 43 & 92). The transition may be accomplished in several manners, the simplest of which is to implement a linear interpolation between the two sets of gimbal angles, although other techniques may be possible that would reduce accumulated momentum errors (the search will make the singular encounter brief, hence the errors will hopefully be small, and can be compensated afterward). After attaining the new gimbal angles, the system is ideally in the preferred closure, and the results of the feedforward search will continue to provide some

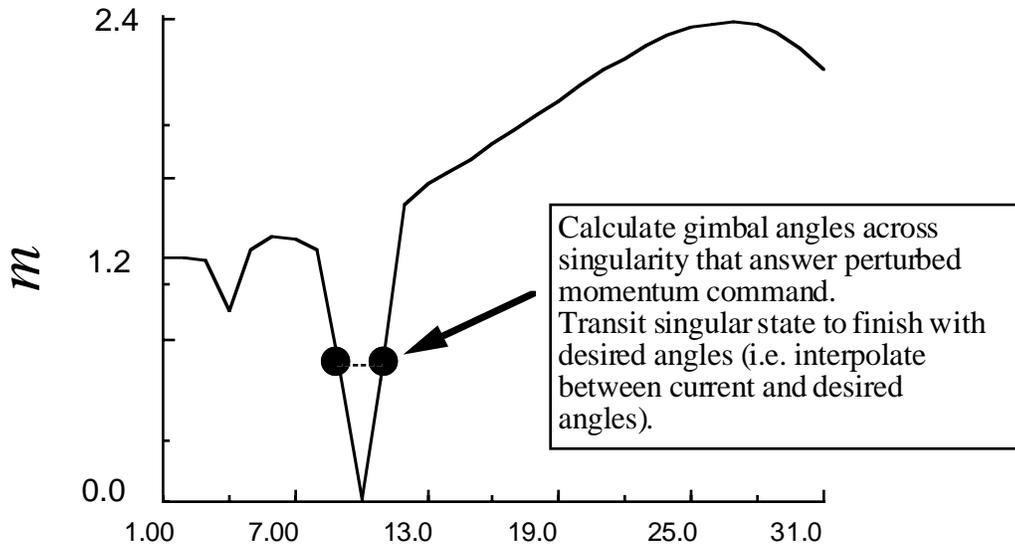


Figure 101: Stabilizing Feedforward CMG Trajectories Transiting a Singular Region

validity. Of course, there is no guarantee that the post-singular closure dictated by the search will admit the perturbed momentum value, in which case this technique will be of little use. In these situations, brute-force solutions may be possible; i.e. the gimbals can again be constrained to follow a particular path when traversing the singular state such that the gimbal state is pre-known both before and after the singularity is crossed, then the results of two feedforward searches can then be applied; one leading up to the singular state, and another from the opposite side of the singularity onward.

Much more effort clearly needs to be devoted to resolving the problem of implementing the trajectories determined by the search in the presence of momentum error. Other techniques are certainly possible, and significant benefit could be attained by a more analytical study of the nonlinear dynamics involved, progressing beyond the simple "show-and-tell" simulations discussed here.

This instability may not be a problem for many systems. In the case of a tight maneuver (or a manipulator system executing an end-effector command), the plant may be sufficiently well-known during the short maneuver duration, and the resulting errors may be small. The search trajectory could thus be implemented open-loop (or any feedback applied after unavoidable singular regions are crossed). In the slower momentum management case, there may be sufficient time available to re-run the search with an updated disturbance model upon encountering divergence, and thereby eliminate the problem.

These examples have investigated the effect of unmodeled disturbances as constant secular torques. Another class of possible disturbances are higher-bandwidth, zero-mean impulses (i.e. as produced by an astronaut bouncing off a bulkhead, onboard manipulators, machinery, or thrusters). The effects of these should be somewhat similar; i.e. small disturbances that don't cause a closure shift will produce a CMG response that commutes to some extent (causing a temporary departure from the current trajectory), and not cause significant divergence, whereas larger disturbances can change closures and create problems. Significant cumulative effects may be produced, however, hence this class of disturbance should be investigated in future implementation studies.

5) Conclusions

A global search has been shown to be effective in producing singularity-avoiding feedforward CMG gimbal trajectories in response to a command history forecast from a momentum management or maneuver scheduler. In cases where the singularity is unavoidable, the search acts to minimize the severity of the singular encounter (i.e. transits the singularity as rapidly as possible, with minimum induced momentum error). Gimbal trajectories that are removed from singular encounters have been noted to be appreciably robust to the introduction of unmodelled torque error. Trajectories that skirt a singular state, however, can show significant sensitivity and quickly diverge from the feedforward reference after the singular approach. Since the location of probable divergence is thus known *a priori*, one may apply constraints near the singular region that can maintain the anticipated system performance. Other techniques may exist that could provide a superior feedforward implementation of the search information; the stability of these feedforward gimbal trajectories is a promising topic for additional research.

The search implementation used in these examples allowed only three levels of null motion; i.e. negative, zero, or positive. Although gimbal chatter was minimized through the objective function, a smoother set of gimbal commands would be advantageous for spacecraft implementation. Greater dynamic range and smoother gimbal response may be achieved by increasing the number of nodes available at each step (producing a larger search space), or allowing the search to vary the change in null motion at each step (and keeping this change limited), rather than varying the net magnitude of null gimbal rates.

The CMG gimbal trajectory can be easily be coded as a "character" string, with each character representing the added null motion, and its position in the string specifying the corresponding momentum step (this parameterization is used for describing trajectories at the bottom of the plots given in Sec. 4.2). The optimality of these trajectory strings can be described by the cost of the trailing character (i.e. terminal node), as calculated through

Eq. 16. A genetic algorithm[63] is a powerful method of efficiently locating families of such strings (i.e. "populations") representing near-optimal solutions, hence may provide a superior means of globally specifying desirable CMG gimbal trajectories in future efforts.

This task has examined the utility of utilizing global information from a momentum management algorithm (or maneuver scheduler) to determine a satisfactory feed-forward command sequence for SGCMGs. As mentioned in the introduction to this document, there would be several advantages to solving both problems together; i.e. the momentum manager would deal with irregular CMG momentum envelopes (especially in failure cases, where mounting symmetry is lost), and a momentum command sequence (plus gimbal trajectory) would be determined to lead the CMG configuration away from singular states. Although adding the momentum (and attitude) degrees of freedom will significantly complicate a search process such as described here, the combined problem may prove tractable by running a constrained search.

Appendix: Software Implementation

A.1) Overview

An extensive software package was developed to facilitate the research performed under this task. Although the main purpose of these programs is to demonstrate the performance of the CMG search concept described in this text, they also provide a ready opportunity to interactively experiment with SGCMG systems, and obtain an understanding of their operation. A fairly complete Macintosh user interface makes this software easy to use. This Appendix gives a few quick details on this package, and presents an abbreviated "users manual" for those who are interested in applying it. This description assumes that readers are already acquainted with Macintosh operation.

The software runs on any Macintosh II type computer (the large screen is necessary, thus it won't work well on the smaller "classic" Macs). If run on any platform other than a Mac II, a warning message will appear, which is easily dispensed via a carriage return. The package has been run on several variants of the Macintosh II (i.e. the fx, ci, etc.), and it performs satisfactorily on all. A minimum of 4 megabytes of RAM is recommended for running the search routine (although only 120K would be needed for the data structures [Table 1] under on-line circumstances; the storage allocation used by this software is highly inefficient). If Multifinder is required, give these applications plenty of memory (particularly CMG Search). They are all written in FORTRAN, and compiled with Absoft FORTRAN 2.4 (Ref. [59]).

The relevant routines and files are kept on a disk, as shown in Fig. 102. Three applications are included. "Define H" provides an easy method to interactively specify angular momentum trajectories. "CMG Search" is a routine to execute and manage the global optimization of gimbal trajectories. "Disturbance" examines the robustness of feedforward search trajectories to a user-specified disturbance torque, and performs

comparisons with standard "tangent" steering laws. The applications are launched in the Macintosh fashion, by double-clicking on their icon. The three applications are described separately in the following sections. Once one is launched, the other applications can be directly activated via the common "Xfer" menu. The software developed to plot the momentum envelopes and singular states (as shown in Sec. 2.3) is not discussed in this document; this program is more complicated, and wasn't developed for general use.

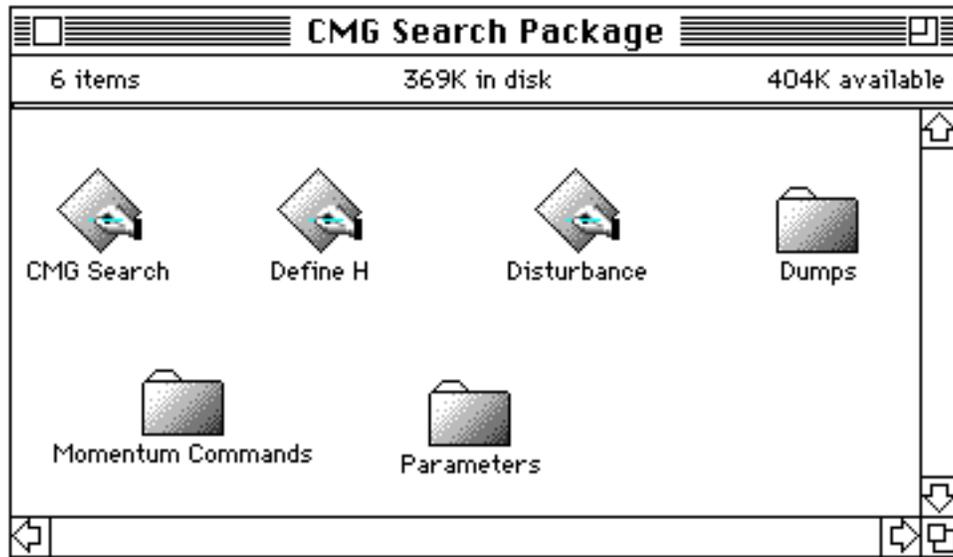


Figure 102: Disk Organization for CMG Search Package

The data files produced by these applications are "TEXT" format. The applications can not be launched by double clicking on these files (this will probably launch your text editor instead). The CMG Search routine requires the "Dumps" folder to be present at the same directory level as the program (such as in Fig. 102); otherwise the Search routine will not launch properly. This folder can initially be empty; CMG Search writes files into it as it progresses (see Sec. A.3), and creates them if they aren't present (it's unable to create the "Dumps" folder itself, however). The other folders aren't needed to run the programs, but are used to organize useful files. "Parameters" holds alternate configuration files used by CMG Search (see the Setup menu in Sec. A.3). "Momentum Commands" holds momentum command sequences produced by "Define H" and read by CMG Search.

Desk accessories are available for use in all applications. When a desk accessory (DA) window is in front of a program window, the DA has control. In this case, most of the application's menus are dimmed (unavailable), except the Edit menu, which is used by the DA. The application is re-activated by exiting the DA or clicking on one of the application's windows.

A.2) Define H

The "Define H" application enables the user to interactively specify and/or modify angular momentum command trajectories that are subsequently used by CMG Search. It was developed as a substitute to a momentum manager or maneuver scheduler routine, which would accomplish an analogous task onboard a spacecraft. The user graphically specifies a momentum command sequence by clicking the mouse in an "active" plot, thereby defining momentum values to be attained at a particular timestep. For situations that require greater precision, these momentum values may also be typed in manually. The resultant command sequence is then saved into a text file that can be opened by CMG Search. Existing momentum command files may also be read by Define H and modified as desired.

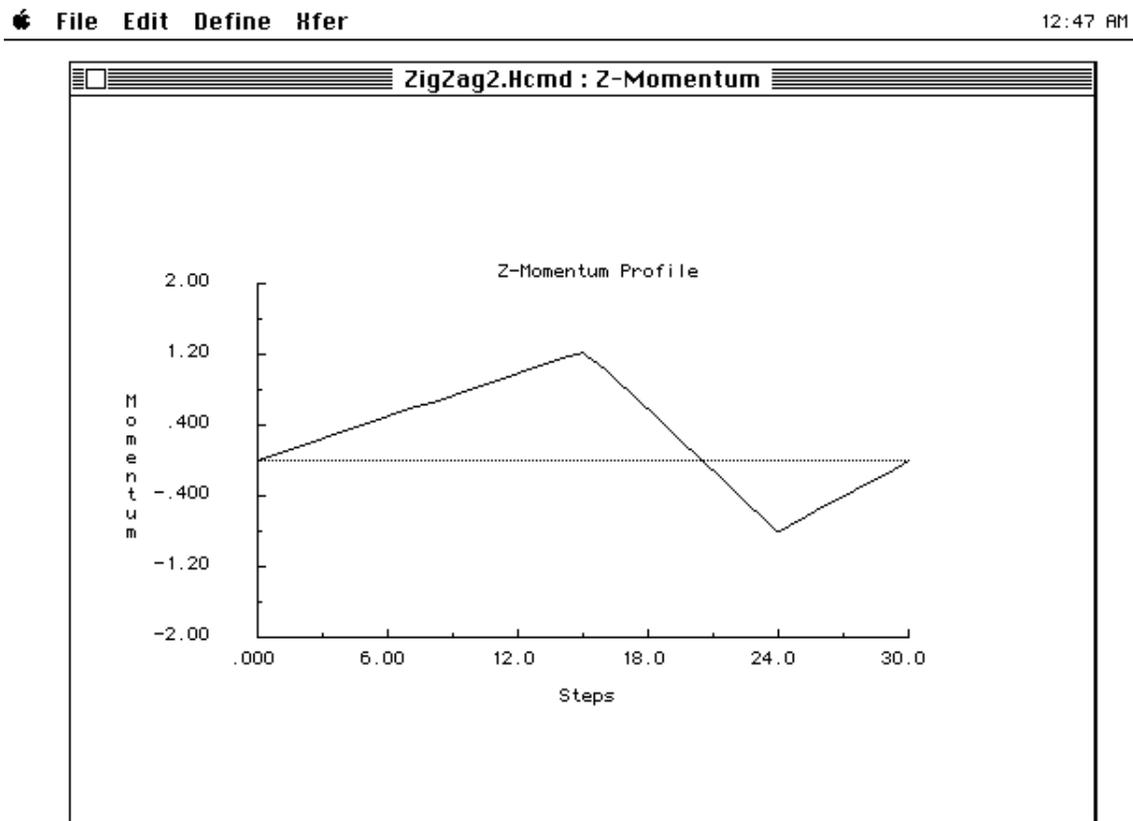


Figure 103: Layout of Macintosh Screen for "Define H"

Fig. 103 shows a typical view of the Macintosh screen while editing a momentum profile using Define H. As will be detailed below, the appropriate momentum component may be selected for graphical editing via the "Define" menu. A plot such as in Fig. 103 will

then appear (this figure shows the z-component). If no editing was previously performed on this momentum component (or a pre-existing file was not opened), this momentum value will be zero for all timesteps. One may then click the mouse in the graph to specify momentum values. A linear interpolation is performed from the previously specified point. In order to create the profile shown in Fig. 103, the mouse was clicked at 3 points; first at $(h=1.2, t=15)$, then at $(h=-0.7, t=24)$, and finally at $(h=0, t=30)$. The order of specification is very important; the program sets the data to the right of the specified point (i.e. at later times) to be constant at the selected value. If the mouse is moved off the labeled graph (but remains in the plot window), the program saturates the selected value to be at the maximum (or minimum) in that direction, both in time (horizontal) and momentum (vertical) coordinates.

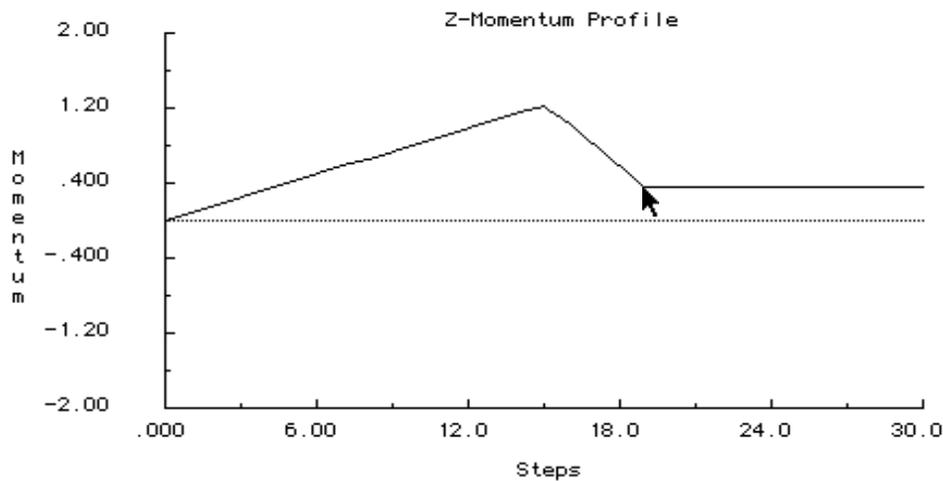


Figure 104: Interactive Editing of Momentum Profile

The effect of a mouse-click edit on the momentum profile plotted in Fig. 103 is illustrated in Fig. 104. Here, the mouse was clicked on this plot at the point where the cursor is visible. By repeating a succession of such clicks, a desired momentum command sequence can be iteratively "drawn".

The "delete" key will always zero out the momentum component being viewed, and the options under the "Edit" menu are very useful in copying and modifying existing profiles (see below). The default scaling on a new plot (with no previous specification) is always between ± 1 . The scale may then be changed (and any existing data re-normalized) by the "Magnitudes" option on the "Define" menu, as discussed below.

If a file has been opened or saved, its name will appear to the left of the colon in the window title (before the momentum component; see Fig. 103). A standard dialog box will appear (asking where to save the new momentum commands) if one has edited or specified

a momentum command, then attempts to exit the application or open a new input file without saving the previous work. One can then either specify a filename and hit the "Save" button, or hit the "Cancel" button, which will discard the current data.

If an existing momentum command file is opened, and the commands subsequently edited with the mouse (as outlined above), the program will act as if the mouse had been hit at every timestep. This is because Define H writes momentum values for each timestep (thus not only at each mouse click) upon saving the commands for subsequent reading by CMG Search.

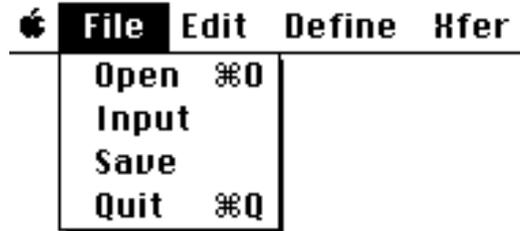


Figure 105: The "File" menu for "Define H"

The items under the "File" menu are shown in Fig. 105. Most are standard. The "Open" command will display a dialog box asking the user to identify an input file to edit. This can be any file previously saved by Define H. The "Save" command will display a dialog box asking where to save the current momentum commands, and "Quit" exits the application.

The "Input" command is somewhat different. This option will bring up a blank window, with a header asking the user to sequentially enter a set of data points. The specified format is then used to type the momentum values; i.e. the step number of the point, followed by the x,y,z momentum triplet. Upon hitting carriage return, prompts are issued for additional points, until the user indicates the end-of-list by entering a negative step number. The window will then disappear, and the data can be displayed and/or additionally mouse-edited as discussed earlier. If, during data input, the user specifies a step too large or out of sequence, the computer will issue a "beep" and ignore the bad point.

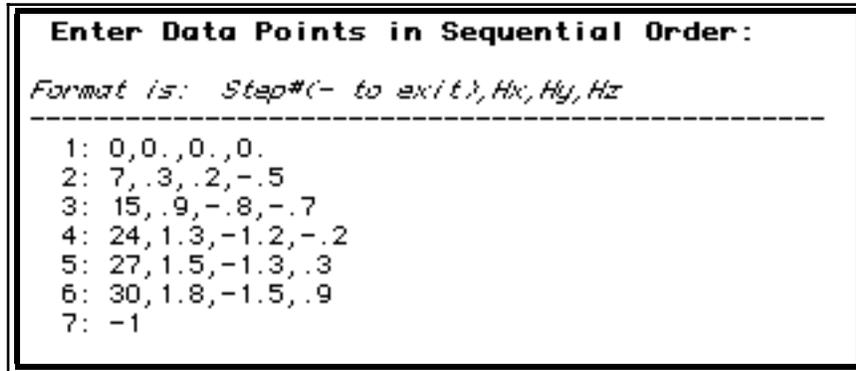


Figure 106: Sample Showing Manual Input of Momentum Points

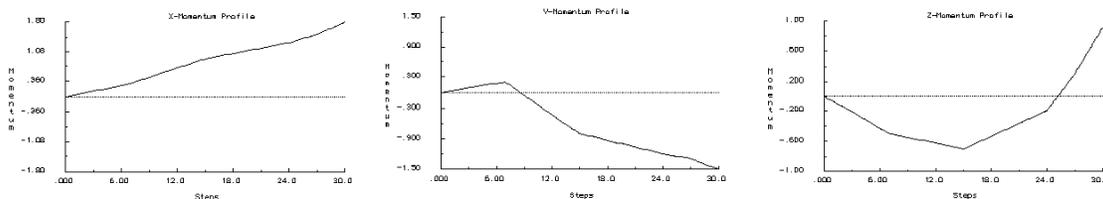


Figure 107: Momentum Command Profiles Generated from Above Data

Figure 106 shows an example of this procedure. A set of six data points were typed into the list. The resultant momentum command profiles were displayed (via the "Define" menu), and are shown in Fig. 107.

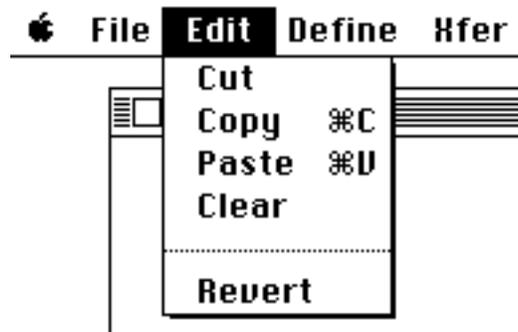


Figure 108: The "Edit" Menu for "Define H"

The "Edit" menu options are shown in Fig. 108. This menu, for the most part, has the usual appearance, and is quite useful in specifying trajectories. These options do not use the conventional Macintosh clipboard, however; data is cut, copied, and pasted to/from an internal array, and can not be ported to external applications in any form. Each of these menu commands refers to the currently displayed momentum component (they are dimmed if a plot is not shown). The menu items are discussed individually below.

The "Cut" item will store the currently displayed momentum component in an internal array, then zero out the displayed values. The "Copy" item will also store the currently displayed momentum component in the internal array, but will leave the displayed values intact. "Paste" will replace the currently displayed momentum component by the values stored in the internal array. "Cut", "Copy", and "Paste" thus provide an easy means of duplicating individual momentum components (i.e. use the same data for x and y), or transporting momentum components between files (i.e. use the x component in one data file for the z component in another).

The "Clear" item will zero out the currently displayed component (as will the "delete" key), while the "Revert" item will restore the currently displayed component to its last saved values.

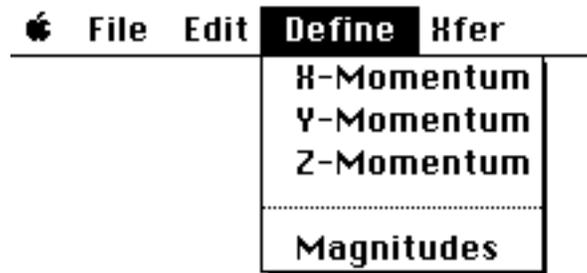


Figure 109: The "Define" Menu for "Define H"

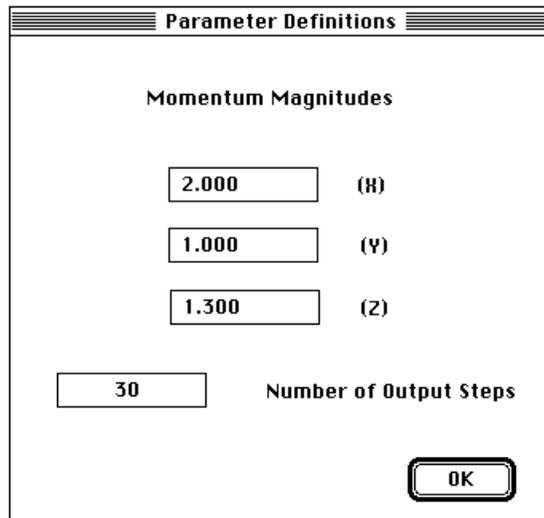


Figure 110: Dialog to Change Component Magnitudes & Discretization Density

Figure 109 shows the options available under the "Define" menu. A desired momentum component may be selected for plotting and editing (as shown in Figs. 102/103)

via the first 3 items. The "Magnitudes" item brings up the dialog box shown in Fig. 110, which allows one to independently set the scaling on x,y,z components (existing data is re-normalized accordingly), and specify the number of timesteps into which the data will be discretized when written into an output file. Although the program is able to specify up to 100 steps, CMG Search is currently wired to only look at up to 30, thus the default value listed in Fig. 110.



Figure 111 The "Xfer" Menu for "Define H"

Fig. 111 shows the options under the "Xfer" menu. This menu allows a quick transfer between other applications, without exiting first to the Finder. The companion routines in this package are specified via the first two items (it is assumed that they are in the same folder as Define H). The "Other..." item brings up a standard dialog box asking for the user to locate an application that is to be launched.

A.3) CMG Search

This application was developed to manage the gimbal search process detailed in Chapter 3. It reads the momentum command files produced by Define H, performs the directed search, and outputs gimbal trajectory files readable by "Disturbance" (Sec. A.4), along with other summary files that can be plotted with standard graphing or spreadsheet applications. CMG Search possesses a fairly complete interactive plotting capability of its own, however, and is a powerful tool for locating and analyzing CMG gimbal trajectories.



Figure 112: Initial Appearance of Menu Bar Upon Launching "CMG Search"

are printed at the right hand side of the trajectory summary. If it is accepted as the "current best" trajectory, a message is printed (as can be noted at the top of Fig. 115), a beep is sounded, and the other two windows are updated accordingly with this trajectory's parameters. The "TRY" item in Fig. 115 shows the current number of expansion steps (i.e. the total number of nodes that have been opened divided by the number of children per node). The following three parameters refer to the evaluation of objective components listed in Eq. 15 for the trailing trajectory node. "MG" refers to the minimum gain, "IC" is the integrated inverse gain, and "RS" is the accumulated momentum residual. The "CST" printed here is an evaluation of Eq. 15 for the last node on the trajectory.

These (and many other) parameters are printed for the accepted "best" trajectories in the lower right "summary" window (the cost listed here includes the average gain contribution of Eq. 16, which is lacking in the status window's "CST"). The summary window also rolls up dynamically as new "best" trajectories are accepted (this window, however, can be scrolled up and down through the "Operate" menu, allowing previous results to be reviewed). A sample summary window display can be noted in Fig. 113. The contents of the graphics window are updated to reflect the trajectory that is portrayed at the bottom of the summary window.

The trial trajectories run at the start of the search are appropriately labeled in the status window. During the execution of these trials, already opened nodes in the tree may be re-visited, and a message "ATTEMPT TO OPEN CLOSED NODE" will be printed (see Fig. 113). This message is provided only for informational purposes, and does not reflect an error.

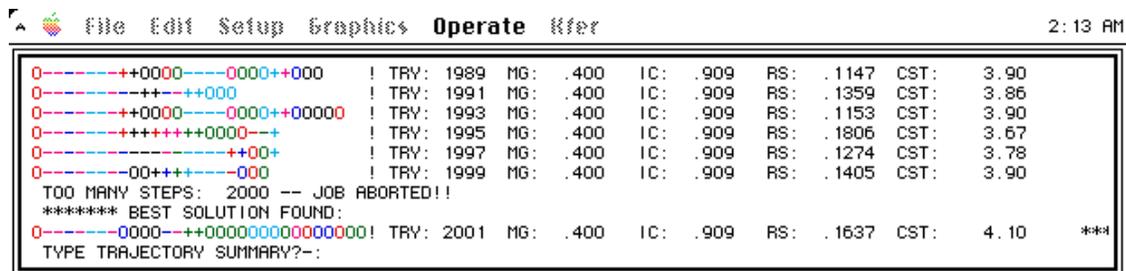


Figure 116: Appearance of the Status Window at the Search Conclusion

Fig. 116 shows the appearance of the status window when a search concludes. The "TOO MANY STEPS" message is the typical final remark; i.e. the search has experienced the maximum number of node-expand steps that was allowed. If this limit is set higher, another message might be shown, indicating that the program has run out of node storage. In either case, the search is halted, the best solution found is displayed, and the system waits

for a keyboard prompt. This can be a bit confusing, since mouse clicks are ignored here; a carriage return is necessary to continue program execution. If a "Y" is typed before hitting return, a rapid summary of the parameters for each node in the final "best" trajectory is typed into the status window. As this is a quickly rolling display, one is advised to employ ^S and ^Q to halt and restart the printout at interesting points.

```

**** NEW BRANCH SELECTED ****
LEVEL: 9 HIGHEST LEVEL: 20 *OPEN NODES: 480 *EXPANDED NODES: 1129
CURRENT NODE: 436 PARENT NODE: 431 GAIN: 1.490082 BEST COST:20.876602 AVG.GAIN: 1.404297
WORST GAIN: 1.183368 INTEGRAL CST: .84486 RATE COST: .00000 GRID CST: 2.701800
FRACT. NULL: .92177 MAX. TOE DSPL: 3.25205 SAT: .42564 NET CST: 20.9
GIMBAL ANGLES: -90.901 5.671 65.253 93.142
HNDW: .05266 .08555 .82070 MAG: .82683
TRQUE: .01364 .01336 .17511 MAG: .17615
00+00++000000000000+- ! TRY: 383 MG: .378 IC: .916 RS: .0134 CST: 4.14

```

Figure 117: Appearance of the Status Window During a Step Dump

An option can be selected under the "Setup" menu ("Type Progress"), that causes a step-by-step dump of parameters to be typed in the status window (at each node expansion). Such a display is shown in Fig. 117 (this is the dump for the last node of a trajectory, thus the "New Branch Selected" message). This display can be stepped at each node (thereby avoiding frantic text rolling) by selecting the "Step-by-Step" option; the system will then wait for a character to be typed before scrolling to the next expansion.

While it is executing, CMG Search updates a set of files in the mandatory "Dumps" folder, as shown in Fig. 118. The files "No Null.Data", "Grdient.Data", and "Cornick.Data" are summaries of the corresponding trial trajectories. The file "TrjDump.Data" is a summary of the best trajectory found by the search (this file is written at the end of the search execution, after the prompt in Fig. 116 has been answered). Other trajectories may be saved during search execution by the "Save Trj." and "Save Last" options under the "Operate" menu (these files don't have to reside in the "Dumps" folder).

These trajectory summary files can be opened directly by the "Disturbance" application for subsequent analysis. They can also be read by a spreadsheet or graphing program for off-line plotting, if desired. All data is tab delimited. The first 7 records of these files contain configuration and parameter information used to initialize "Disturbance". If one is plotting this data via another program, these records should be deleted. The remaining records give data for all nodes on the trajectory (one record per node), from the start node down through the terminal node. For the interest of the experimenter, I'll list the variables in these records. In order, they are: Node number, Timestep, # of children, Gimbal sign pattern for null motion (used to resolve sign ambiguities in "Disturbance"), Null index, Net null amplitude (\bar{K}), CMG Gain (m), Minimum CMG gain encountered

thus far, Integrated momentum residual, Integrated gimbal rate over maximum, Saturation index, Objective value (Eq. 15), Gimbal angles (Degs; one for each CMG [4 assumed]), Gimbal rates (Deg/sec; one for each CMG [4 assumed]), Components of net stored CMG momentum (3 values), Components of momentum command for current timestep (3 values), Components of torque command for current timestep (3 values).

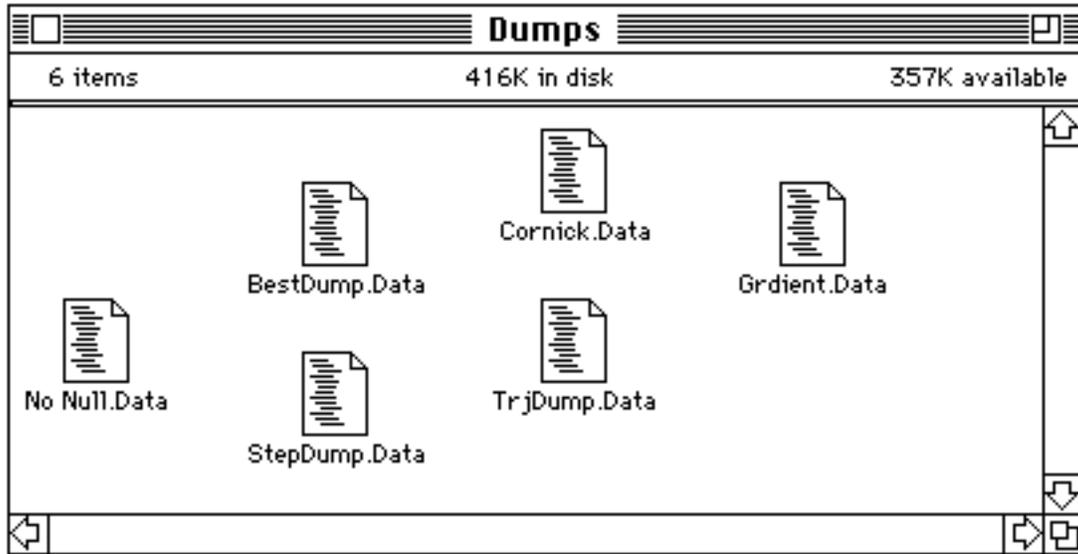


Figure 118: Standard Contents of the "Dumps" Folder

CMG Search also writes two additional files into the "Dumps" folder. These are "StepDump.Data" (which used to be a running step-by-step summary, but now is empty; an obsolete file) and "BestDump.Data", which holds a summary of the "Best" trajectory's evolution during the search execution. Each record summarizes the parameters of successive "Best" trajectories. This file is quite handy, and was used to generate summary plots such as in Fig. 39 (which plotted various parameters against the number of search expands). There are no annoying initialization records at the beginning of these files, thus there's no need to delete anything before plotting this data in a spreadsheet or graphing program. All data entries refer to values calculated across the gimbal trajectories (evaluated at the terminal node). The record format is somewhat different than the trajectory dumps listed previously, so I'll call it out separately here. Again, in order, the variables are: Trajectory index, Number of trajectory "grafts" thus far attempted, Number of nodes thus far created (minus 1), Number of search expands performed, Number of nodes thus far created, CPU ticks elapsed since last "best" trajectory, Sum of integer null motion across trajectory, Minimum CMG gain over trajectory, Integrated inverse CMG Gain, Integrated momentum residual, Objective value of terminal node (Eq. 16), Saturation index at terminal

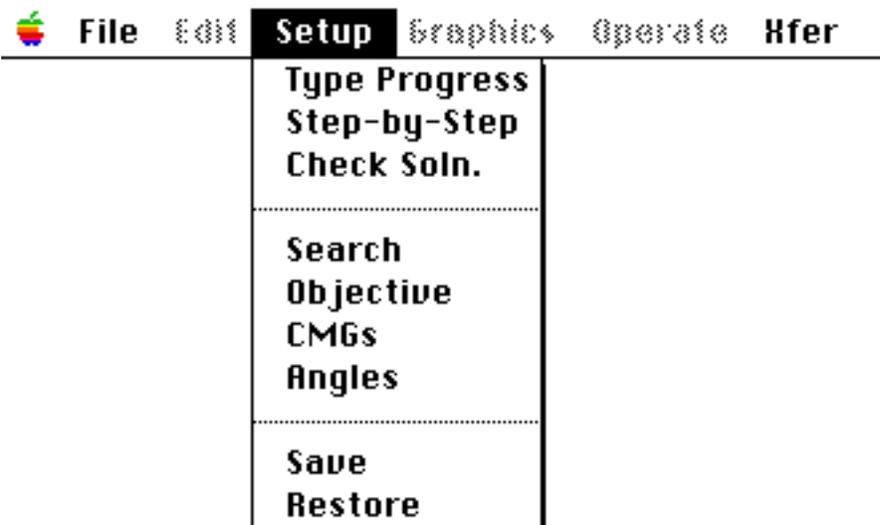


Figure 121: "Setup" menu for "CMG Search"

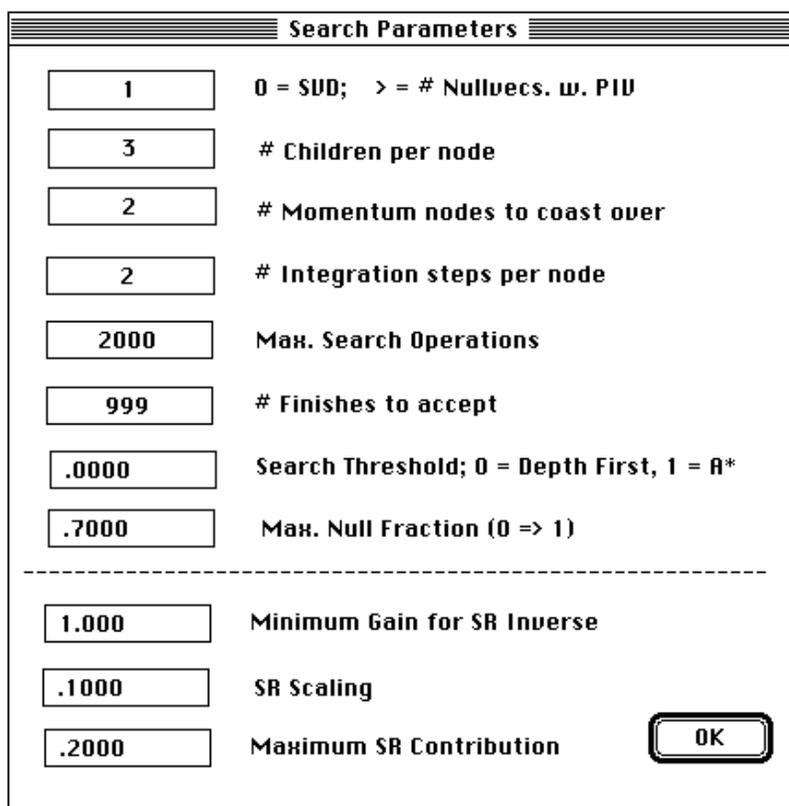


Figure 122: "Search" Dialog Box

The four items in the middle of the "Setup" menu bring up dialog boxes that allow the user to re-mode the search operation. They can only be selected before a search has begun. The dialog box associated with the "Search" item is shown in Fig. 122. The values shown here are the standard defaults used in the tests of Chapter 4.

The first item dictates how the kinematic calculations are performed. If this item is zero, a singular value decomposition is applied to the CMG Jacobian to form the torque and null solutions (the number of null vectors is determined automatically from the Jacobian rank). If this item is positive, an SR-Inverse is performed to calculate the torque solution, and null vectors are produced from the cross product scheme described in Eq. 10. In this case, the first item dictates the number of null vectors to be calculated. This is clipped by the rank of the full-up system (i.e. a 4-CMG ensemble has only 1 excess degree of freedom, so only one null vector will be calculated).

The second item specifies the number of child nodes per parent (see Fig. 28). The third item dictates the number of timesteps to "coast over" before making a search decision. The value of "2" indicates that every other node is eligible for consideration by the search. The fourth item specifies the number of kinematic integration steps (SR-Inverse and null motion calculations) per node. The fifth item is the maximum number of search steps (node expansions) that are attempted before the search quits, as depicted in Fig. 116. Note that this is not the maximum number of nodes allowed to be created (this is set by the parameter NNDES [currently = 10,000] in the "Search Header" file, thus the application must be re-compiled if it is changed). The sixth item is the number of candidate trajectories (propagated all the way to the terminal node) to encounter before quitting the search. The seventh item is a flag to toggle search strategies; a zero selects the depth-first operation discussed in Chapter 3, and used in this study. A one will produce a directed, breadth-first search in the style of A*. Intermediate values will select a hybrid strategy somewhere between (this variable is a multiplier on the global objective in the local-vs.-global cost comparison; when set to unity, local gradient expansions are abandoned if a better cost open node exists elsewhere on the tree, yielding a breadth-first search having the properties of A*). The eighth item is the fraction of the peak gimbal rate that the CMGs will be driven to under maximum null motion (η in Fig. 28).

The lower three items in Fig. 122 define the operation of the SR-Inverse. The first of these items defines the maximum CMG gain at which the SR contribution will become active. The middle item is the scale on the SR weighting ρ (see Eq. 13). The last item is the maximum allowed SR contribution. These factors were discussed in Sec. 4.1.

The "Objective" item under the "Setup" menu brings up the dialog box shown in Fig. 123, which allows one to set the various weightings on the objective function. Again,

default values are shown, as used in the simulations of Chapter 4. Referring to Eqs. 15 & 16, the first item is W_1 , the second item is W_2 , the third item is W_6 , the fifth item is W_3 , the sixth item is W_4 , and the seventh item is W_5 . The fourth term is a weighting on the instantaneous value of CMG Gain (thus does not integrate the effects of the preceding nodes on the trajectory). It thus introduces a bias in favor of the "instantaneous" cost gradient, where every step is taken to improve to the local CMG gimbal condition. Tests have indicated that such a local emphasis can appreciably degrade the search performance, thus this weighting is set to zero. The lower two items define the grid cost, as described in Eq. 17. Referring to this protocol, the upper item is the factor α , and the lower item is the time constant γ .

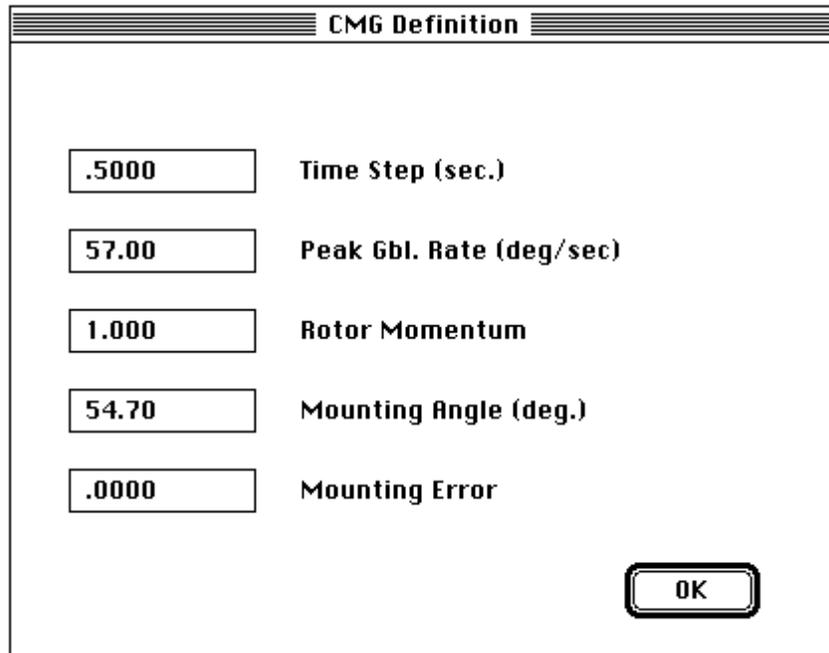
Objective Weights	
20.00	Minimum Gain
3.000	Integral Gain
1.800	Average Gain
.0000	Instantaneous Gain
2.000	Residual
100.0	Gimbal Over-rate
5.0000E-02	Null Penalty

5.000	Grid Cost
.9500	Grid cost attenuation factor
OK	

Figure 123: "Objective" Dialog Box

Fig. 124 shows the dialog box that appears in response to the "CMGs" menu item. The CMG configuration can be adjusted and specified here. The values shown are defaults used in the calculations of Chapter 4. The first item is an assumed time interval corresponding to a timestep (as discretized in the input momentum command files). This quantity plays little role in this software, except as an optional scaling on plots (which wasn't used in those shown in Chapter 4), and as a normalization on the peak gimbal rate to determine the maximum allowed CMG gimbal deflection per timestep. The second item is

the maximum allowed gimbal rate, the third item is the magnitude of the CMG rotor momentum, and the fourth item is the skew angle of the mounting pyramid (see Fig. 9).

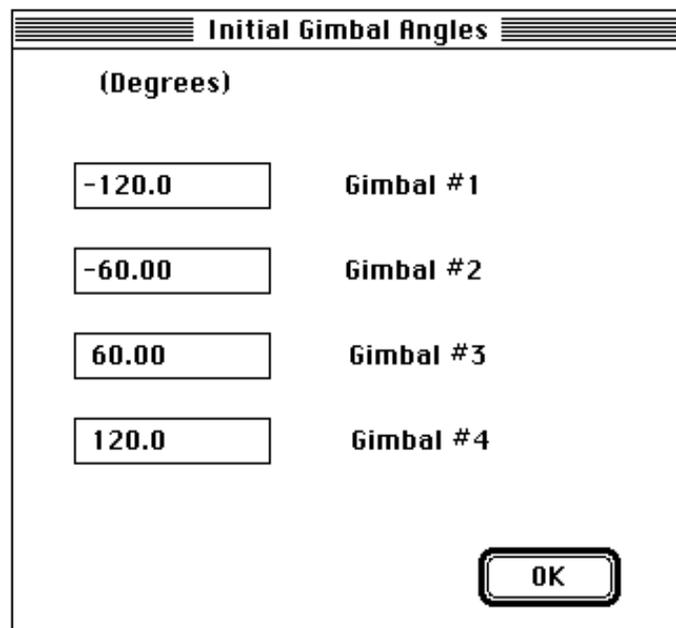


The dialog box titled "CMG Definition" contains five input fields with corresponding labels and an "OK" button. The input fields contain the values: .5000, 57.00, 1.000, 54.70, and .0000.

Input Field	Label
.5000	Time Step (sec.)
57.00	Peak Gbl. Rate (deg/sec)
1.000	Rotor Momentum
54.70	Mounting Angle (deg.)
.0000	Mounting Error

OK

Figure 124: "CMGs" Dialog Box



The dialog box titled "Initial Gimbal Angles" contains four input fields with corresponding labels and an "OK" button. The input fields contain the values: -120.0, -60.00, 60.00, and 120.0. The label "(Degrees)" is positioned above the input fields.

Input Field	Label
-120.0	Gimbal #1
-60.00	Gimbal #2
60.00	Gimbal #3
120.0	Gimbal #4

(Degrees)

OK

Figure 125: "Angles" Dialog Box

Although the CMGs are always assumed to be mounted in the pyramid fashion, the mounting symmetry can be broken by specifying a nonzero value for the fifth item. All CMGs in the array are assumed to be rotated by an equal angle about the central (z) axis (see Fig. 9, where this interCMG angle is $90^\circ = 360^\circ/4$ for the 4-CMG system). This parameter is an additional angle offset that is added in the midst of the CMG array. For example, with the 4-CMG array, CMG's 1 and 2 would be 90° apart, as would CMG's 3 and 4. The angle between CMG's 2 and 3, however, would be 90° plus this offset value. This effectively "splits" the mounting symmetry around the z-axis, and significantly skews the momentum envelopes and singular states. Although some interesting tests have been run using this quantity, it is not used in the examples of Chapter 4 (for the sake of conventionality), thus was kept zero.

Fig. 125 shows the dialog box that is achieved by selecting the "Angles" menu item. This allows one to set the desired initial gimbal angles (in degrees) for the CMG system. The default values are all zero (which is a zero momentum state for this mounting protocol). The values shown in Fig. 125 are an alternate zero momentum state, proposed in Ref. [55] and used in the examples "Kuro C" and "Kuro D" of Chapter 4. After setting these parameters (or changing any of the other dialog boxes), the initialization message appears in the status window (as shown in Fig. 114), displaying the net angular momentum of the CMG system at the specified gimbal orientation.

Fig. 126 shows the items residing under the "Graphics" menu. This selection allows one to specify the information displayed/plotted in the graphics window, and specify the mode of presentation. Like all other menus, these options can only be specified when the search is paused. Before the search is begun, only one of these items will display a plot; this is the "H Command" selection, which will show the commanded momentum profile. The other plots will appear when a complete gimbal trajectory has been created (i.e. the non-null trial is the first). The plotted data reflects the gimbal trajectory referenced in the lower position of the summary window. The plot is updated whenever a better trajectory is found or the summary window is scrolled. An exception may be found in the first two options; "Tree" and "Plane". These are dynamically updated as the search expands each new node. A sample "Tree" plot can be seen in Fig. 127. This is a representation of the actual search tree, with the start node at the bottom, terminal node at top, and null displacement to the right and left. The color of the plotted nodes indicates their CMG gain (hence optimality), as mentioned earlier. Due to the exponentially increasing node density, this picture is rapidly becomes too thick to discern independent branches and nodes at later search levels, thus the right-left separation is determined very early in the search. Although the information content is limited, it has aesthetic value. The other dynamically updated plot is produced by

the "Plane" selection. This is shown in Fig. 128 for the same example as the tree in Fig. 127. Here, the start node is at left, terminal state at right, and null motion extends up and down. Nodes are drawn as squares, again color coded as before. All null displacements are equal here (as opposed to the tree, where they decrease logarithmically with branching index), thus later trajectories will be drawn on top of earlier paths. Some information can be gleaned regarding regions frequented by the search process, however, and the plot does have some aesthetic value, particularly while it is being drawn; the animation yields insight into the search dynamics.

The plot labeled "Stability" is the startup default selection, and it produces the topographic maps used in Sec. 4.2 and defined in Fig. 31. The items in the second batch of selections are all standard plots, and for the most part self-explanatory (most are used in the various figures of Chapter 4).

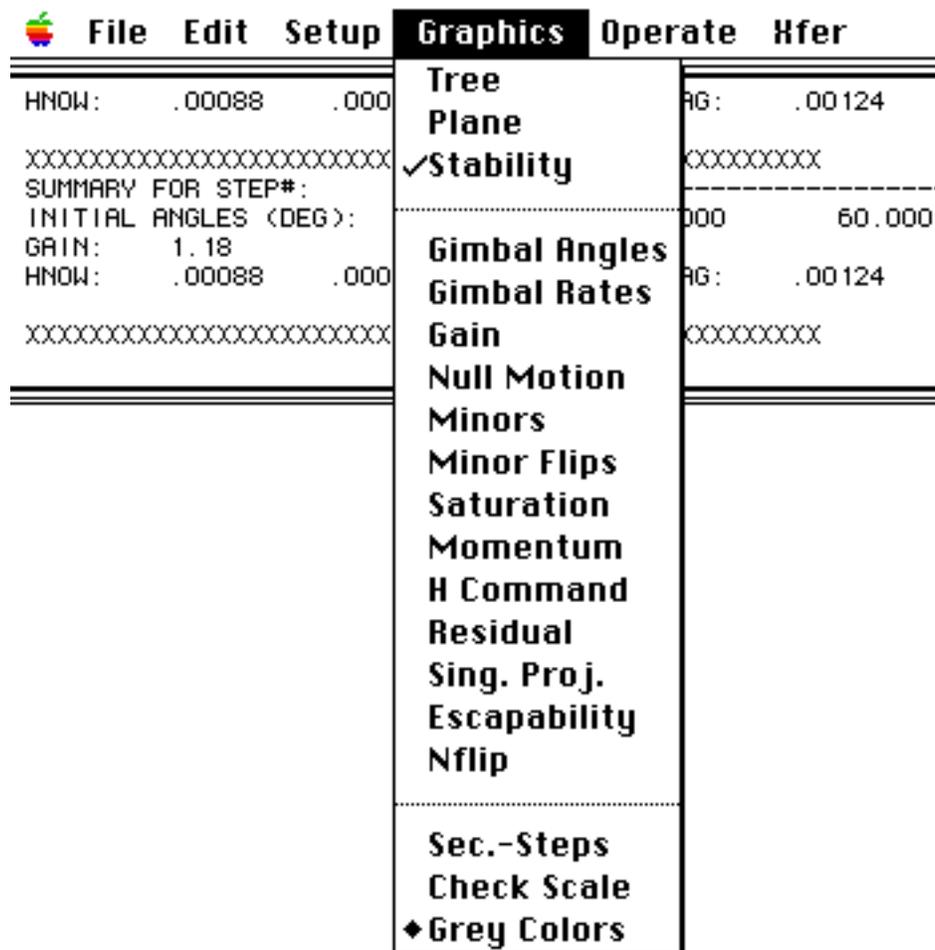


Figure 126: "Graphics" Menu for "CMG Search"

The lower items change the format of plotted data. The "Sec.-Steps" item toggles the plotted x-axis units & labels between seconds and timestep counts (the latter is used with all plots presented here). "Check Scale" will produce a dialog box as shown in Fig. 129 every time a new plot is produced. This allows the user to fix the axis scalings as desired, rather than being forced to adopt the protocol accompanying each plot. "Grey Colors" changes the color map used to denote the CMG gain values. When this is unchecked, the colors are used as discussed earlier; i.e. black/blue = high gain (good), red/pink = low gain (bad). If "Grey Colors" is selected, a checkmark will appear, and the colors will be coded such that Claris MacDraw II will export an appropriate grey scale to the clipboard (when the "Color Clipboard" option is deselected in MacDraw); this option was used to generate the shaded plots presented in Chapter 4. If "Grey Colors" is again selected, the checkmark will turn to a diamond (as in Fig. 126). In this case, the greyscale will be set so that a POSTSCRIPT printer or monochrome monitor will deliver appropriate shadings (dark = good, light = bad).

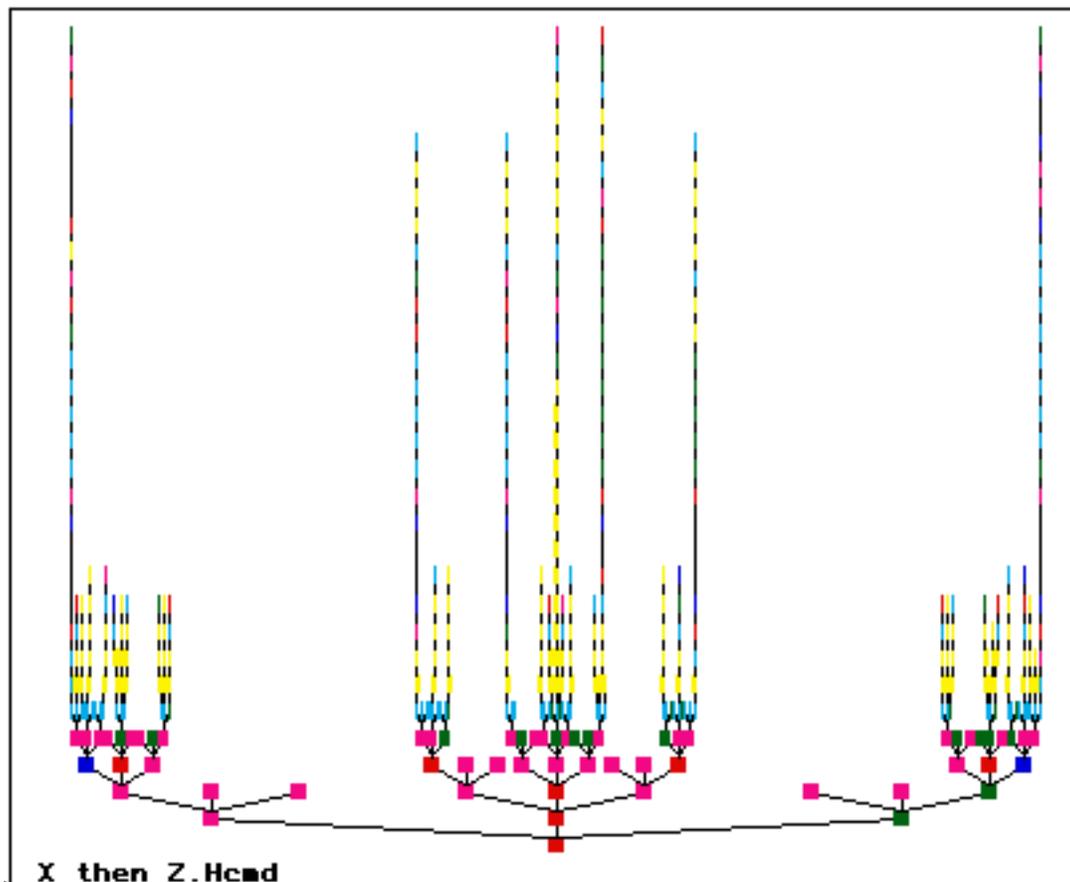


Figure 127: A Typical "Tree" Diagram

Figure 130 shows the items under the "Operate" menu (this is the way the menu looks after search has been started). The "Run" and "Pause" items toggle the operation of the search process on and off. Most menu items are unavailable while a search is running, thus the search must be paused before anything can be selected or changed. "Restart" will re-initialize the search logic. This item must be selected to scrap a search before it has completed and change search or configuration parameters (via the "Setup" menu), or open a new momentum command file.

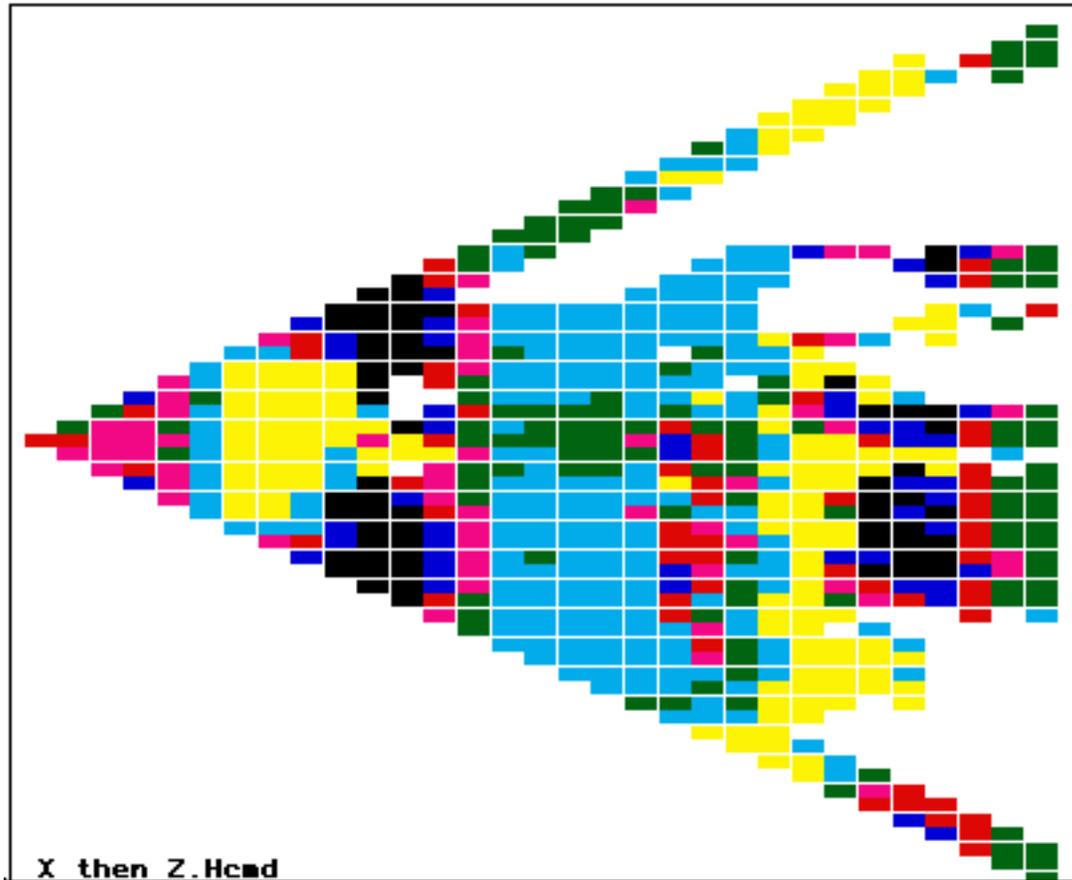


Figure 128: A Typical "Plane" Diagram

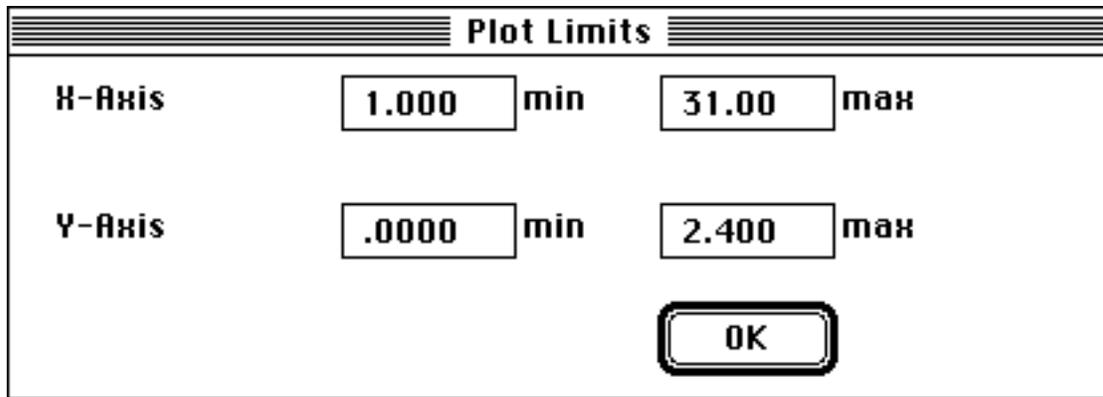


Figure 129: Dialog Box Enabled by "Set Scale"

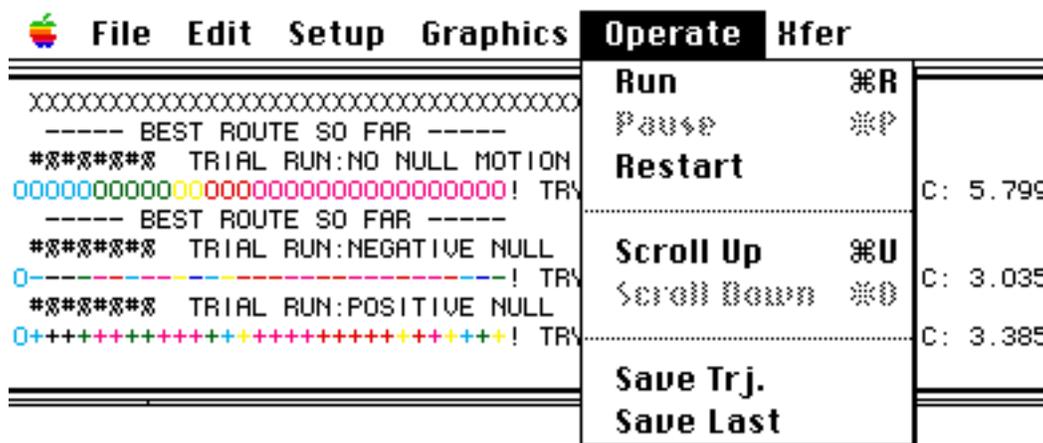


Figure 130: "Operate" Menu for "CMG Search"

The middle block of items in the "Operate" menu control the scrolling of the summary window. Normally, the summary window is scrolled down after every new "best" trajectory is accepted, such that the latest trajectory is at the bottom of the window. When the search is paused, however, the summary window may be scrolled up and down trajectory-by-trajectory via these menu items. The plots in the graphics window are also changed to reflect the bottom trajectory that is displayed in the summary window. By using these scroll options, one may examine previous trajectories for comparison purposes. If the summary window contents point to the first or last accepted trajectory, the "Scroll Up" or "Scroll Down" items are (respectively) made unselectable (as seen in Fig. 130). When the search is continued (via the "Run" item), the summary window is scrolled to the last accepted trajectory, and the graphics window accordingly updated.

The bottom items in the "Operate" menu allow the user to save particular trajectory dumps (in the format described previously) for subsequent analysis by "Disturbance" or a

plotting application. "Save Trj." writes a file containing the trajectory displayed at the bottom of the summary window (other trajectories can be selected by scrolling, as mentioned above). "Save Last" writes a file containing the last trajectory completed by the search (i.e. the last trajectory that was accompanied by a trio of asterisks "***" in the summary window; see Fig. 115), thereby allowing analysis of trajectories not necessarily considered by be "current best". A standard file dialog box appears when these items are selected, asking for a filename and destination.

The "Xfer" menu is shown in Fig. 131. This allows the user to execute another routine in the CMG package (or transfer to a user-defined application) without exiting first to the Finder.

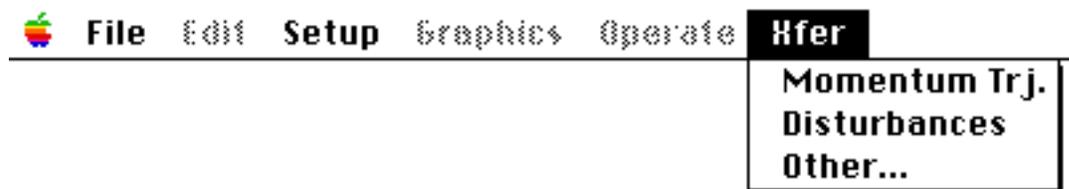


Figure 131: "Xfer" Menu for "CMG Search"

CMG Search is currently set up to handle a 4-SGCMG array, and this restriction can not be altered during runtime. It may be changed, however, by re-compiling the program with the parameter "NGBLS" in the "Search Header" file (where all the global variables and constants are defined) set to the number of CMGs desired. This has not been attempted for a while, thus might be a risky proposition (although the appropriate hooks should reside in the code). In a similar vein, CMG Search is also hardwired to accept momentum command files discretized into at most 30 timesteps. This restriction should be able to be extended by re-compiling with the "Search Header" parameter "NSTTOP" set to the maximum number of timesteps desired, and the parameter "NSTTP1" set to this value plus one. Again this has never been actually attempted, so good luck!

A.4) Disturbance

This application reads the trajectory files output from CMG Search, and essentially applies the implementation logic of Fig. 30. A "local" steering law calculates the SR-Inverse and null motion at the instantaneous gimbal location. The signed amplitude of null motion is applied as forecast from the CMG search. "Disturbance" allows the user to specify a constant secular torque, however, that was not accounted for in the momentum commands that were answered by the search. "Disturbance" also implements standard

tangent CMG steering laws for performance comparisons. Several graphical tools are implemented to analyze differences between search-reference, torque-perturbed, and tangent-null solutions. This application was employed to generate the results shown in Sections 4.3 & 4.4.

The standard appearance of the Macintosh screen while running "Disturbance" is given in Fig. 132. Four plots can be seen (in this case, the CMG gain is graphed). The upper left plot is a "Reference", which arises from the implementation of the feedforward search solution without adding disturbance torques. A finer integration is generally performed with the reference than was used in the search (the integration step may be arbitrarily adjusted, as will be detailed below); this usually has little effect, except in some cases that closely skirt a singular region. The "Perturbed" results (upper right) arise from the feedforward null vector applied with the CMGs answering the original command sequence plus disturbance torque. The numbers plotted above perturbed results give the disturbance vector ("D:") and initial gimbal angle error vector (A:) used in calculating the data. The "Original" results (lower left) are calculated by the CMG search (they are taken directly from the data file). The "Standard" results (lower right) show the response of the selected tangent null algorithm on this momentum command. The character codes printed beside the plot title refer to the null algorithm used (see below).

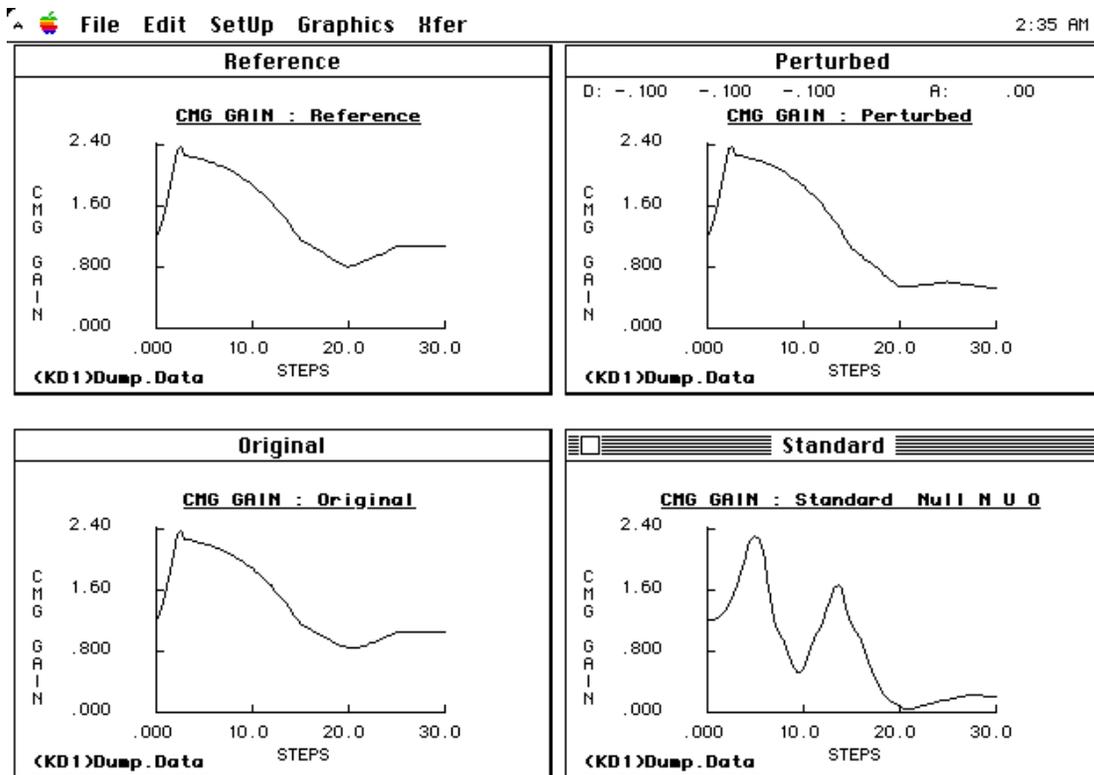


Figure 132: Layout of Macintosh Screen for "Disturbance"

Certain types of plots (i.e. the error plots or perturbation results) show comparisons between the reference and perturbed trajectories, thus only one or two plot windows may be produced, rather than the four shown in Fig. 132.

When the application is first launched, all menu items are unselectable, except for those under "File". Once a trajectory file is opened, the application pauses to integrate the solutions forward (a few seconds or so), then the selected plots are produced (initially the angle errors), and all menu items are made available.

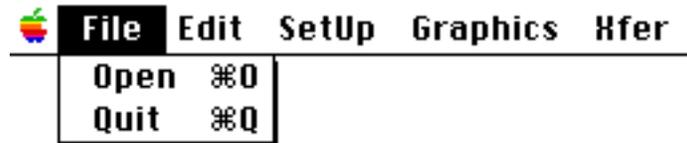


Figure 133: "File" Menu for "Disturbance"

Fig. 133 shows the items under the "File" menu. "Open" opens a gimbal trajectory dump as produced by CMG Search, and "Quit" exits to the Finder.

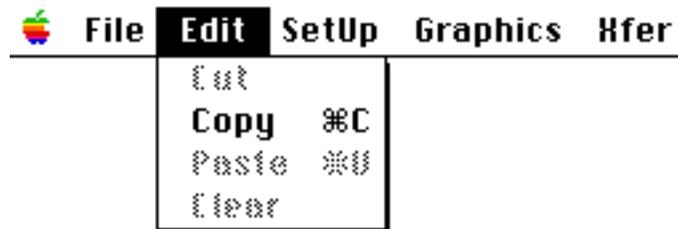


Figure 134: "Edit" Menu for "Disturbance"

Fig. 134 shows the items under the "Edit" menu. The only relevant option here is "Copy", which transfers the graphic contents of the selected window to the clipboard as a collection of PICT objects. These can later be edited in a drawing program, and used for presentation, as seen in Secs. 4.3 & 4.4. A window may be selected in the standard fashion, by clicking somewhere in its area. Occasionally, when different windows obscure one another, and a re-draw is queued by a plot window being brought forward, the plotting routine can loose track of scale, and the graph can be distorted. In this case, merely close the troublesome window, then re-select the desired plot from the "Graphics" menu. The scale difficulty should now be corrected.

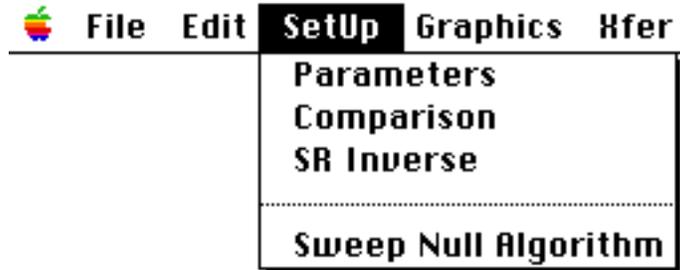


Figure 135: "SetUp" Menu for "Disturbance"

Fig. 135 shows the items under the "SetUp" menu. These mode the operation of the Disturbance application. The first three items bring up dialog boxes. Whenever any of these options are chosen, the application pauses to update its trajectory calculations.

The dialog box that appears in response to the "Parameters" item is shown in Fig. 136. The operation of the forward trajectory integration may be set here. The values shown in Fig. 136 are default settings, except for the disturbance torque, which is initially zero in all three coordinates.

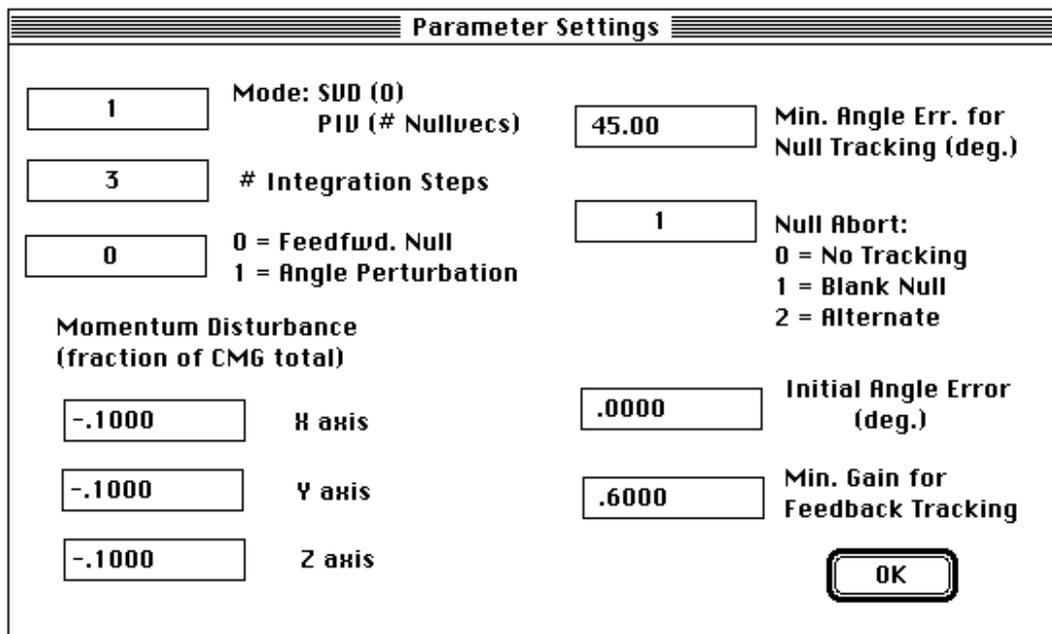


Figure 136: "Parameters" Dialog Box

The "Mode" variable is analogous to the like-named parameter in the setup dialog for CMG Search (Fig. 122). Torque and null solutions can be calculated via a singular value decomposition or SR-Inverse (see earlier discussion for details i.e. Fig. 122). This doesn't have to track the selection made in CMG Search; a different "mode" of calculation can be performed here if desired. The "Integration Steps" parameters specifies the number

of integrations to be performed per timestep. The third parameter in this list specifies how the feedforward trajectory will be implemented. The "0" entry specifies the method of Fig. 30, where the torque and null solutions are calculated locally, and the null amplitude (\bar{K}) is fed forward from the search results. This is the conventional approach, and is used in the results of Sec. 4.4. If this parameter is set to 1, however, a different technique is applied, where the gimbal angles predicted from the search (interpolated to the higher integration rate) are first adopted, then modified by an iterated torque solution (SR-inverse or SVD) to answer the unmodelled disturbance. Although this technique did not perform as well in tests, it might be adapted to stabilize the feedforward trajectories around singular regions, as suggested at the end of Sec. 4.4 and Fig. 101. A topic for future endeavor....

The next three parameters represent the components of the unmodelled error. They are in fractional units of total CMG momentum, and represent the error in the momentum vector at the terminal state (the momentum error is distributed as a constant torque along the entire command sequence). This was discussed in detail at the beginning of Sec. 4.4.

The upper right parameter in Fig. 136 is the largest angular divergence (sum of absolute gimbal angle differences between original and perturbed trajectories) that is allowed before the null tracking (i.e. adoption of feedforward null amplitude \bar{K}) is disabled. When this limit is exceeded, it is assumed that the perturbed trajectory has drifted so far from the original that the search's predictions have become meaningless. When this occurs, the strategy selected in the item below is pursued. Three possibilities exist. The first (coded by "0") abandons the use of null motion, and applies only the torque solution for the remainder of the trajectory. The second (coded by "1") applies the feedforward null vector, but doesn't try to match the signs of null gimbal displacements (see below), which can cause appreciable gimbal jitter when the divergence is high. The third possibility (coded by "2") will abandon the feedforward search results, and manage null motion through the tangent CMG steering law that has been selected in the "Comparison" dialog box. In the examples presented here, the "1" option was always used; this seemed to exhibit the best performance in evading singular conditions after trouble arose. The other options have interesting performance as well, but space and time constraints eliminated their description here. Clearly, this study will benefit from further research into implementation techniques.

Sign matching is an important concept in following the feedforward trajectories, thus I'll take a moment to describe it here. When exporting a trajectory file, CMG Search calculates (and writes) a variable at each timestep that describes the sign pattern of gimbal displacements under "positive" null motion. When "Disturbance" implements the feedforward null amplitude, an ambiguity can exist between what is defined as "positive" null motion, especially if the null motion is calculated differently in "Disturbance" than in

CMG Search (i.e. SVD vs. cross product), or if the gimbals have diverged somewhat from their reference trajectory. To resolve this, the "positive" sense of null motion is defined at each step to be the direction in which the sign pattern of positive null gimbal displacements agrees best with that calculated by the search. This technique has been seen to work quite well in locking the feedforward null motion to that calculated locally.

The second-last item in the right column of Fig. 136 specifies the initial gimbal angle error in the "perturbed" implementation. If this is non-zero, the gimbals are moved from their initial states (using null motion to preserve the net CMG momentum) until they differ from their starting values by the specified net angle. This option was not used in the tests of Chapter 4 (it was always zero), but interesting results have been derived by playing with it; one can exploit this parameter to explore the boundaries of various gimbal closures.

The angular momentum attained at each timestep is subtracted from the desired value to form the torque command for the following timestep. The momentum residual is thus feedback compensated. This can be disabled in situations where a singularity is approached (as mentioned at the end of Chapter 3, feedback compensation of the momentum errors can lock the CMG system into singular configurations). The last item in Fig. 136 sets the minimum value of the CMG Gain (m) that is allowed before feedback tracking is disabled in generating the torque command. The system thus answers input commands in an open-loop fashion for CMG configurations exhibiting lower values of CMG gain.

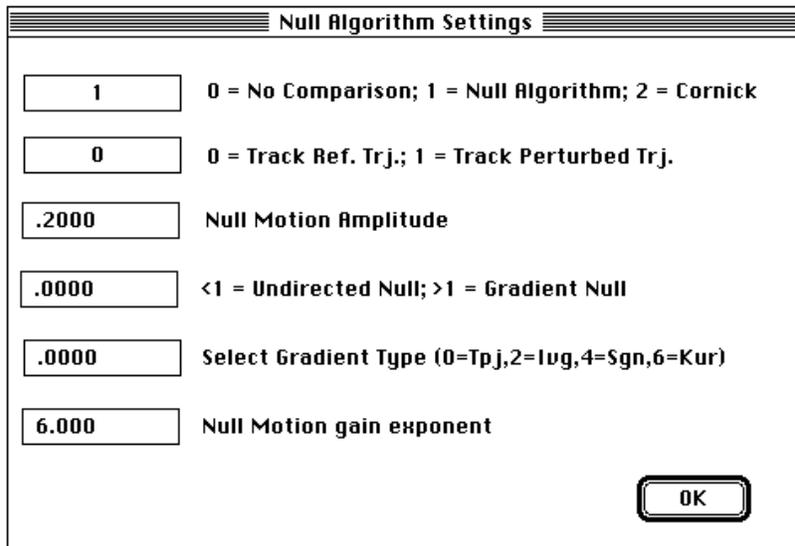


Figure 137: "Comparison" Dialog Box

Fig. 137 shows that dialog box that appears in response to selection of the "Comparison" item of the "SetUp" menu. These entries control the properties of the tangent CMG steering law that is implemented. The values shown are defaults, and were used for the results given in Sec. 4.3. All tangent law calculations are based on code from Ref. [13]; consult this source for details. The implementation of this code was quickly accomplished, thus some inaccuracies may exist; beware...

The top item denotes the master strategy chosen. If a "0" is chosen here, no tangent law is used, and only three plots are shown (the "Standard" plot seen in Fig. 132 is omitted). This is actually the startup default; this item must be set non-zero for tangent CMG steering results to be produced. If a "1" is chosen, a null algorithm is pursued, as configured by the other parameters. If a "2", is picked, a Cornick "unkinking" law is used, as dictated in Ref. [34], and discussed in Sec. 4.1 (according to the source of Ref. [13], however, the tangent null implementation of this logic may have difficulty; beware).

The next item selects the source of momentum commands for the tangent steering law. If it is "0", the momentum commands come exactly as output from CMG Search (and as used in the "Reference" plot). If it is "1", the perturbed momentum commands are adopted.

The third item specifies a scaling imposed on the calculated null amplitude k .

The fourth item selects whether the sign of null motion is gradient guided. If this item is "0", no gradient is used in determining the sign. If it is larger than unity, the gradient calculation specified in the fifth item is applied (these cryptic abbreviations arise from Ref. [13]).

The sixth item is the power that the CMG gain (m) will be raised to before scaling the null motion amplitude (see discussion in Sec. 4.3).

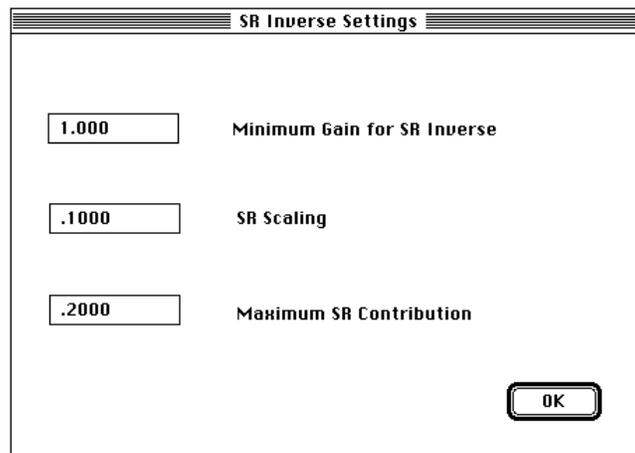


Figure 138: "SR Inverse" Dialog Box

Fig. 138 shows the dialog box that materializes in response to the "SR Inverse" item on the "SetUp" menu. This allows one to mode the characteristics of the SR Inverse. These parameters correspond exactly to those at the bottom of Fig. 122, hence were discussed earlier. They should be set exactly as used in CMG Search to insure proper trajectory tracking.

The lower parameter in the "SetUp" menu of Fig. 135, "Sweep Null Algorithm", enables additional analysis to be performed on the selected tangent steering law. When this option is chosen, both the "Reference" and "Perturbed" results plotted on the Macintosh screen will be derived from the tangent steering law, and not the feedforward null motion from CMG Search. The major utility of this option is associated with the perturbation sensitivity plots that can be summoned up under the "Graphics" menu; these plots will now summarize the behavior of the tangent steering law when subject to the selected disturbance torques. After "Sweep Null Algorithm" has been chosen, the "SetUp" menu changes, as shown in Fig. 139. The item "Sweep FeedFwd. Trj." now appears at the bottom of the menu list, and must be selected to restore the analysis of the feedforward trajectories from CMG Search.

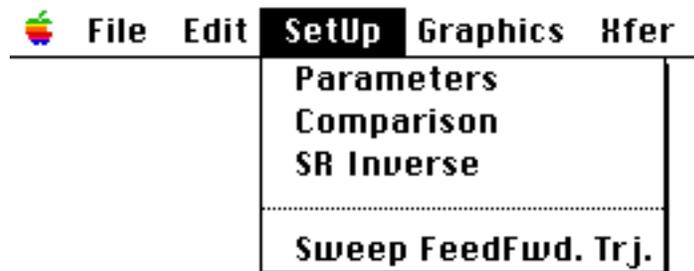


Figure 139: Appearance of "SetUp" Menu after "Sweep Null Algorithm" Selected

Fig. 140 shows the multitude of items available under the "Graphics" menu. The top three groups of options specify data that can be plotted. The third group offers the largest selection, and the plotted quantities are analogous to those that can be graphed in CMG Search. These selections will produce a group of plots as in Fig. 132, where the selected quantity is plotted in a separate window for each situation.

The top pair of items produce the "Error" plots, as given in Sec. 4.4. These plot differences in the selected quantity between reference and perturbed trajectories at each timestep, and show how the divergence develops with time. The first item, "Angle Errors", produces two plots. One shows the absolute difference ($|\text{reference} - \text{perturbed}|$) in gimbal angles for each CMG. The other plots the sum of these absolute differences over all CMGs

(such plots were included in the results of Sec. 4.4). The second item in this list produces a single plot that shows the difference in CMG gain (perturbed - reference) for each timestep. Even when run with zero disturbance, these plots can show some small transient divergences (generally during applied null motion) due to the finer integration performed in "Disturbance" (vs. CMG Search).

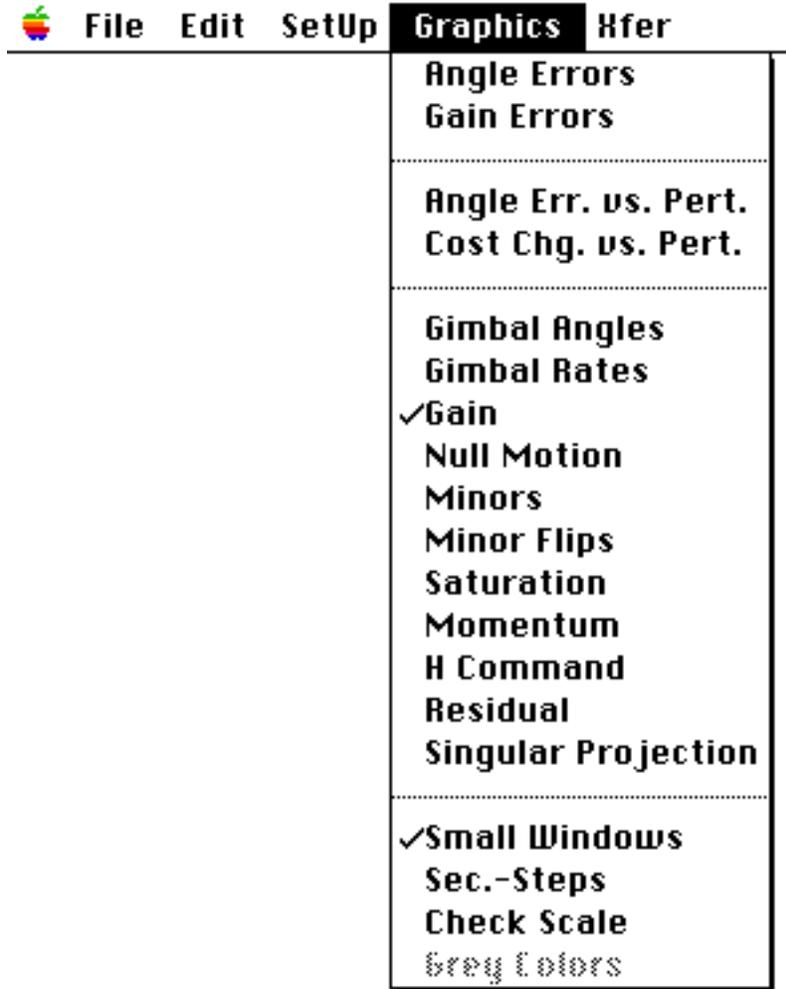


Figure 140: "Graphics" Menu for "Disturbance"

The second pair of items produce plots that show the sensitivity of the gimbal trajectories to the magnitude of the applied perturbation. The top item in this list will plot the average net angle error (sum of absolute angle differences averaged over all timesteps) experienced by a gimbal trajectory as a function of the relative magnitude of the applied perturbation. The lower item in this list plots the objective change (perturbed - reference) [Eq. 16, but without residual, overrate, and null terms, from Eq. 15] as a function of the relative magnitude of the applied perturbation. When either these items is selected, 11

complete trajectories will be calculated, with torque perturbations in the direction selected via Fig. 136, but with magnitudes ranging from zero to full (as specified in Fig. 136). The scale on these plots thus ranges from 0 to 1, and defines the fraction of the full-scale disturbance that was applied during the trajectory. These plots are very useful indicators of trajectory sensitivity to applied disturbances, and are presented with the results of Sec. 4.4.

In addition to showing the sensitivity to applied torque shifts, these plots can also show the sensitivity to changes in initial CMG angles. If a nonzero initial angle difference is entered into the dialog of Fig. 136, the angle error will be varied (from zero to full-scale) with the applied perturbation.

When a sensitivity plot is selected for the first time, the system will pause while calculating the associated ten trajectories. This will cause a brief delay. The calculation progress can be tracked, however, since the cursor will briefly flash after each trajectory is finished.

The bottom set of items in Fig. 140 determine the plot appearance. The "Small Plots" option is selected as the default, and produces small plot windows (as in Fig. 132) that can be tiled easily, 4 to a Mac II screen. If "Small Plots" is deselected, larger plots are produced, with overlapping windows. The larger plots may have some benefit in size and resolution when copied onto the clipboard and pasted into a drawing program. "Sec.-Steps" and "Check Scale" have the same effect as discussed in Sec. A.3. "Sec.-Steps" will toggle the x-axis scaling between "seconds" and "timesteps", using the normalizations specified in CMG Search. "Check Scale" will display the dialog box of Fig. 129, allowing the user to specify the axis scaling before each plot is drawn. "Grey Colors" has no meaning in this application, since all graphics are monochrome. It is therefore always unselectable.

Fig. 141 shows the "Xfer" menu. It provides a convenient means of transferring to other applications without exiting first to the Finder.

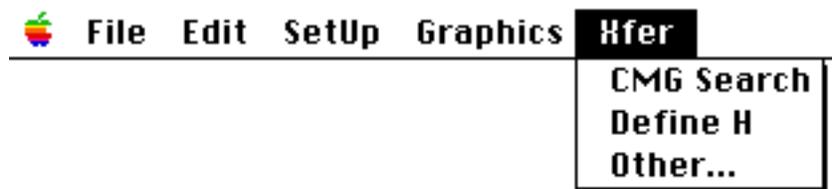


Figure 141: "Xfer" Menu for "Disturbance"

Acknowledgements

John Sunkel and Lynda Bishop of NASA/JSC are earnestly thanked for their interest and support in sponsoring this effort. The information on search techniques provided by Owen Deutsch and Bob Beaton of CSDL is greatly appreciated. The many spirited CMG discussions held with Nazareth Bedrossian (CSDL/MIT) and Haruhisa Kurokawa (MEL, Japan) were invaluable. Owen Deutsch and Paul Zarchan (CSDL) provided much useful advice on the the toolbox implementations of Macintosh FORTRAN, in which the search and analysis software was written. Eli Gai and Milt Adams (CSDL) are thanked for setting up the initial meetings with their trajectory optimization staff, which led to the realization of this effort. Janet Lepanto (CSDL) is appreciated for arranging the early discussions at NASA/JSC that started this RTOP.

Much of the software development and document preparation involved with this task was accomplished in my basement on a Macintosh II computer (home computing for aerospace applications!). During the last year of this effort, my schedule became so burdened that delays were inevitable; I sincerely apologize to those who were long looking forward to the issue of this document (particularly John Dowdle and Paul Motyka of CSDL). Apologies are also due to many of my friends (especially Leonore), who saw little of me on many a beautiful summer weekend, while I was holed up in my basement, hectically hacking this opus together. Lyle Mays is thanked for the loans of his Macintosh II computer, which was used for much of the early software development and analysis. Leonore Eforo's assistance with the nuances of Microsoft Word were much appreciated.

I'd like to again thank Nazz Bedrossian and Haruhisa Kurokawa for allowing me to use some of their figures, which saved considerable time in preparing this manuscript. In particular, Figs. 4,6,7,8,10,13 were adapted from Nazz's thesis (Ref. [13]), and Fig. 5 came from a Japanese report of Haruhisa's (Ref. [64]).

The image presented on the cover and front page of this document is a computer-generated projection of the 2H singular surfaces for a 4-SGCMG parallel mounted array. It is essentially derived from the 3D database of Fig. 15; the related text should be consulted for details. Yes, the study of CMG systems does have its aesthetic aspects!

Most of this work was supported under NASA contract NAS9-18147. This paper is published solely for the exchange and stimulation of ideas, and does not constitute approval by NASA of the findings or conclusions contained herein.

List of References

- 1) Pioneering the Space Frontier, Report of the National Commission on Space, Bantam Books, Inc., May 1986.
- 2) Sharir, M., "Algorithmic Motion Planning in Robotics," *IEEE Computer*, March 1989, pp. 9-19.
- 3) Pearl, J., Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley Publishing Co., 1984.
- 4) Deutsch, O.L., Desai, M., McGee, L.A., "Far-Field Mission Planning for Nap-Of-The-Earth Flight," *Proc. of the National Specialists' Meeting of the American Helicopter Society, Motorcraft Flight Controls and Avionics Session, Cherry Hill, NJ*, Oct. 13-15, 1987.
- 5) Miller, D., et. al., "Autonomous Navigation and Mobility for a Planetary Rover", *Proc. of the AIAA 27'th Aerospace Sciences Meeting*, January 1989.
- 6) Wang, T.C. et. al., "Applications of Optimization Techniques to Spacecraft Fuel Usage Minimization in Deep Space Navigation," *Mathematical Engineering in Industry*, Vol. 1, No. 2, 1987, pp. 97-122.
- 7) Niiya, C.K., "An Application of the A* Search Technique to Trajectory Optimization," MS Thesis, MIT Dept. of Aeronautics and Astronautics, CSDL-T-1952, June 1990.

- 8) Chamitoff, G.E., "Robust Intelligent Flight Control for Hypersonic Vehicles", PhD Thesis, MIT Dept. of Aeronautics and Astronautics, Fall 1991.
- 9) Kelly, J., Wilson, D., "CMG Trade Study Preliminary Results", Presentation at NASA/JSC, April 27, 1988.
- 10) Redding, D.C., Adams, N.J., "Optimized Rotation-Axis Attitude Maneuver Controller for the Space Shuttle Orbiter", *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 1, Jan.-Feb. 1987, p. 4.
- 11) Bauer, S.R., "Single Gimbal CMG Steering Laws", C.S. Draper Lab. Report Space Guidance & Navigation Memo 10E-87-06, May 1987.
- 12) Kurokawa, H., Nobuyuki, Y., "A Study of CMG Systems - for Selection and Evaluation", *Proc. of the 16'th International Symposium on Space Technology and Science, Sapporo*, 1988, pp. 1243-1249.
- 13) Bedrossian, N., "Steering Law Design for Redundant SGCMG Systems", MS Thesis, MIT Dept. of Mechanical Engineering, CSDL-T-965, August, 1987.
- 14) Glavina, B., "Solving Findpath by Combination of Goal-Directed and Randomized Search", *Proc. of the IEEE Int. Conference on Robotics and Automation, Cincinnati, OH*, May 13-18, 1990, pp. 1718-1723.
- 15) Parker, J.K., Khoogar, A.R., and Goldberg, D.E., "Inverse Kinematics of Redundant Robots using Neural Networks", *Proc. of the 1989 IEEE Int. Conference on Robotics and Automation*, pp. 271-276.
- 16) Guez, A. and Ahmad, Z., "Solution to the Inverse Kinematics Problem in Robotics by Neural Networks", *Proc. of the IEEE Int. Conf. on Neural Networks*, Vol. II, July 1988, pp. 617-624.
- 17) Paradiso, J., "A Highly Adaptable Steering/Selection Procedure for Combined CMG/RCS Spacecraft Control; Detailed Report", CSDL-R-1835, March, 1986.

- 18) Seltzer, S.M., "CMG-Induced LST Dynamics", NASA/MSFC Report TM-X-64833, Feb. 21, 1974.
- 19) "Control Moment Gyros", Brochure from Sperry Flight Systems, Space Systems Marketing, Phoenix, AZ, May, 1982.
- 20) "Astronaut Maneuvering Equipment (Experiment 509) Attitude Control System Flight Test Analysis Plan", Project Technical Report (TRW) Task E-38D, NAS 9-8166, Nov. 9, 1970.
- 21) Kennel, H.F., "Steering Law for Parallel Mounted Double-Gimballed Control Moment Gyros - Revision A," NASA TM-82390, Jan. 1981.
- 22) Yoshikawa, T., "A Steering Law for Three Double-Gimballed Control Moment Gyro Systems," NASA TM X-64926, March 1975.
- 23) Paradiso, J.A., "A Highly Adaptable Steering/Selection Procedure for Combined CMG/RCS Spacecraft Control," *1986 AAS Guidance and Control Conf., Keystone, CO.*, AAS 86-036, *Adv. Ast. Sci.* Vol. 61, pp. 263-280, Feb. 1986.
- 24) Dzielski, J., Bergmann, E., Paradiso, J., "A Computational Algorithm for Spacecraft Control and Momentum Management," *Proc. of the 1990 American Control Conference, San Diego, CA.* (Vol. 2), pp. 1320-1325, May 23-25 (1990).
- 25) Wie, B., et. al., "A New Approach to Attitude/Momentum Control of the Space Station", *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 5, 1989, pp. 714-722.
- 26) Hattis, P., "Predictive Momentum Management for The Space Station", *Journal of Guidance, Control, and Dynamics*, Vol. 11, No. 1, Jan.-Feb. 1988, pp. 19-25.
- 27) Applegate, J.F., McMillion, J.M., Smith, R.E., "Design and Operation of the Skylab Attitude and Pointing Control System", *Proc. of the AAS Meeting at UCLA*, Aug. 20-22, 1974, AAS74-126, *Adv. Ast. Sci.*, p. 467.

- 28) "Space Station Advanced Development Control Moment Gyro (CMG) Data", NASA MSFC memo ED15-85-43, July 1985.
- 29) Branets, V.N. et. al., "Development Experience of the Attitude Control System Using Single-Axis Control Moment Gyros for Long-Term Orbiting Space Stations," *Proc. of the 38'th Congress of the International Astronautical Federation, Brighton, UK*, Oct. 1987.
- 30) Meffe, M., "Control Moment Gyroscope Configurations for the Space Station," *1988 AAS Guidance and Control Conf., Keystone, CO.*, AAS 88-040, Jan. 1988.
- 31) Barry, T.D., "Survey of Momentum Exchange Devices", NASA/JSC Avionics Systems Division Note EH-82-05, JSC-18321, May 1982.
- 32) Weinberg, M.S., "Momentum Exchange Effectors for Space Applications", CSDL-P-1532, May 1982.
- 33) Margulies, G., Aubrun, J.N., "Geometric Theory of Single-Gimbal Control Moment Gyro Systems," *Journ. Ast. Sci.*, Vol. 26, No. 2, April/June 1978, p. 159.
- 34) Cornick, D.E., "Singularity Avoidance Control Laws for Single Gimbal Control Moment Gyros," *1979 AIAA Guidance and Control Conf., Boulder, CO.*, pp. 20-23, Aug. 1979, AIAA 79-1968.
- 35) Kurokawa, H., Yajima, N., Usui, S., "A New Steering Law of a Single-Gimbal CMG System of Pyramid Configuration," *Proc. of the X'th IFAC Symposium on Automatic Control in Space, Toulouse, France, 25-28 June, 1985*, p. 249.
- 36) Kurokawa, H., Nobuyuki, Y., Usui, S., "A Study of Single Gimbal CMG Systems", *Proc. of the 15'th International Symposium on Space Technology and Science, Tokyo, 1986*, pp. 1219-1224.
- 37) Yoshikawa, T., "Steering Law for Roof Type Configuration Control Moment Gyro System", *Automatica*, Vol. 13, 1977, pp. 359-368.

- 38) Meffe, M., Stocking, G., "Momentum Envelope Topology of Single-Gimbal CMG Arrays for Space Vehicle Control", *Proc. of the 10'th Annual AAS Guidance and Control Conference, Keystone, CO.*, Jan. 31 - Feb. 4, 1987, AAS 87-002.
- 39) Foley, T.A., et. al., "Visualizing Functions Over a Sphere", *IEEE Computer Graphics and Applications*, Jan., 1990.
- 40) Aldus Super 3D, Version 2.5 for the Apple Macintosh Computer, Silicon Beach Software, Inc., San Diego, CA., 1991.
- 41) MacRenderMan, Version 1.1, Pixar Corp., Richmond, CA., April 1991.
- 42) Powell, B., et. al., "Synthesis of Double Gimbal Control Moment Gyro Systems for Spacecraft Attitude Control", *Proc. of the AIAA Guidance, Control, and Flight Mechanics Conference, Hempstead, NY.*, AIAA 71-937, Aug. 16-18, 1971.
- 43) Paradiso, J., "Characteristics of Candidate Reaction Wheel Configurations for Satellite Attitude Control", CSDL Dept. 10C memo NRL-87-1, Oct. 1, 1987.
- 44) Daleh, M.A., Richards, D.M., " ℓ^1 Optimal Attitude Control of a Space Station", MIT Industrial Liason Program Report 10-31-90, LIDS-P-1984, April 30, 1990.
- 45) Matusky, M.P., "Systems of Single-Gimbal Control Moment Gyros", CSDL Space Guidance & Navigation Memo No. 10E-87-07.
- 46) Kurokawa, H., "Topological Study of Single Gimbal Control Moment Gyro Systems," to be published.
- 47) Tokar, E.N., "Control of Redundant Gyroforce Systems", *Cosmic Research*, Vol. 18, No. 2, 1980, pp. 115-123.
- 48) Tokar, E.N. and Platanov, V.N., "Singular Surfaces in Unsupported Gyrodine Systems," *Cosmic Research*, Vol. 16, No. 5, 1978, pp. 547-555.

- 49) Bedrossian, N., Paradiso, J., Bergmann, E., Rowell, D. "Redundant Single Gimbal CMG Singularity Analysis," *AIAA Journ. of Guidance & Control*, Vol. 13, No. 6, pp. 1096-1101, Nov.-Dec., 1991.
- 50) Bedrossian, N., "Classification of Singular Configurations for Redundant Manipulators", *Proc. of the IEEE Int. Conference on Robotics and Automation, Cincinnati, OH*, May 13-18, 1990, pp. 818-823.
- 51) Press, W.H., et. al., Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, 1988.
- 52) Bedrossian, N., Paradiso, J., Bergmann, E., Rowell D., "Steering Law designs for Redundant SGCMG Systems," *AIAA Journ. of Guidance & Control*, Vol. 13, No. 6, pp. 1083-1089, Nov.-Dec., 1991.
- 53) Nakamura, Y., Hanafusa, H., "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 108, Sept. 1986, pp. 163-171.
- 54) Crenshaw, J.W., "2-SPEED, a Single-Gimbal Control Moment Gyro Attitude Control System", *AIAA Guidance and Control Conference, Key Biscane, FL.*, Aug. 20-22, 1973, AIAA Paper 73-895.
- 55) Vadali, S.R., Oh, H-S, Walker, S.R., "Preferred Gimbal Angles for Single Gimbal Control Moment Gyros," *AIAA Journ. of Guidance & Control*, Vol. 13, No. 6, pp. 1090-1095, Nov.-Dec., 1991.
- 56) Schiehlen, "Two Different Approaches for a Control Law of Single Gimbal Control Moment Gyro Systems", NASA TM X-64693, Aug. 2, 1972.
- 57) Deutsch, O.L., Desai, M., "Development and Demonstration of an On-Board Mission Planner for Helicopters", CSDL-R-2056, April, 1988.
- 58) Fijany, A., Bejczy, A., "Parallel Architecture for Robotics Computation," NASA/JPL Invention Report NPO-17629/7126, *NASA Tech Briefs*, Vol. 14, No. 6, Item #21, June 1990.

- 59) MacFORTRAN Version 2.4, Absoft Corp., Rochester Hills, MI., 1990.
- 60) Paradiso, J., "Application of a Directed Search to Global Steering of Single Gimballed CMGs", CSDL-P-3014, *Proc. of the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA.*, Aug. 12-15, 1991, AIAA Paper 91-2718.
- 61) Kurokawa, H., Personal Communication, Winter 1990-91.
- 62) Lokshin, A., Zak, M.A., "Chaotic Motion of a Two-Link Planar Mechanism," NASA/JPL Invention Report NPO-17387/6898, *NASA Tech Briefs*, Vol. 13, No. 8, Item #39, Aug. 1989.
- 63) Goldberg, D.E., Genetic Algorithms in Search, Optimization, & Machine Learning, Addison-Wesley, 1989.
- 64) Kurokawa, H., "A Study on Control Moment Gyro Systems: Singularity and Steering Laws of Single Gimbal CMG Systems", Report of the Mechanical Engineering Laboratory No. 140, MEL, Japan, 1987.