Nonredundant CMG Steering

Victor Hakim

For an audience at Millennium Space Systems

2020

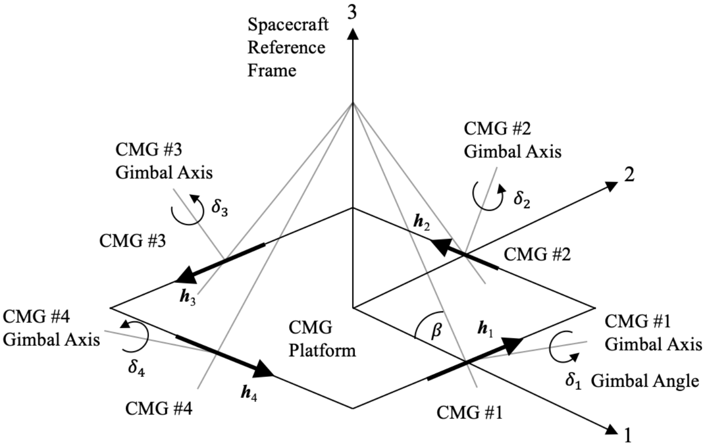
Overview

Most CMG configurations have four or more CMGs in order to be redundant. This redundancy helps to avoid or escape singularities.

If some CMG rotors have failed so that only three CMGs are operational, the configuration is **nonredundant**.

Recent IEEE conference paper: proposed using **Gauss Pseudospectral Method** to steer nonredundant CMG configurations, i.e. translate into a numerical optimal control problem, where gimbal angle trajectories can be computed by a nonlinear programming (NLP) solver.

4-CMG Pyramid Configuration



We took $\beta = 53.13^{\circ}$ ($\cos\beta = 0.6$) [1].

3/4-CMG Pyramid Configuration

Redundant configuration: $n \geq 4$ rotors (ideal)

Nonredundant configuration: $n = 3$ rotors (failure state)

We will consider the specific case of a four-CMG pyramid configuration where one CMG has failed and only three CMGs are operational; we'll call this the **3/4 pyramid configuration**. Without loss of generality, let CMG #4 be in failure.

Let $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3)^T$ be the vector of CMG gimbal angles.

System's Total Angular Momentum

Consider a rigid spacecraft with n CMG rotors. Let $\mathbf{H}$ be the system's total angular momentum. Then:

$$\mathbf{H} = J\boldsymbol{\omega} + \boldsymbol{h}$$

where J is the spacecraft tensor of inertia, which includes terms representing CMG rotors' inertia and center of mass. $\boldsymbol{h}$ is the total angular momentum of all CMG rotors *from the perspective of a bystander fixed on the spacecraft and rotating with the spacecraft.*

Given the external torque $\mathbf{T}_{\text{ext}}$ in the body frame, the evolution of $\boldsymbol{\omega}$ is given by:

$$\mathbf{T}_{\text{ext}} = \dot{\mathbf{H}} + \boldsymbol{\omega} \times \mathbf{H} = J\dot{\boldsymbol{\omega}} + \dot{\boldsymbol{h}} + \boldsymbol{\omega} \times \left(J\boldsymbol{\omega} + \boldsymbol{h}\right) \tag{1}$$

(The derivatives must be with respect to the spacecraft body-fixed frame.)

CMG Angular Momentum for the 3/4 Pyramid

$\boldsymbol{h}$ is given by the summation over the three functioning CMGs:

$$\boldsymbol{h} = \eta\left(\begin{bmatrix}-c\beta\, s_1\\ c_1\\ s\beta\, s_1\end{bmatrix} + \begin{bmatrix}-c_2\\ -c\beta\, s_2\\ s\beta\, s_2\end{bmatrix} + \begin{bmatrix}c\beta\, s_3\\ -c_3\\ s\beta\, s_3\end{bmatrix}\right) \tag{2}$$

where η is each CMG rotor's axial angular momentum magnitude, $c\beta \doteq \cos\beta$, $c_i \doteq \cos(\delta_i)$, and likewise for sine. Then $\dot{\boldsymbol{h}}$ is:

$$\dot{\boldsymbol{h}} = \eta A \dot{\boldsymbol{\delta}} \tag{3}$$

A is called the *Jacobian*, given by:

$$A = \begin{bmatrix}-c\beta\, c_1 & s_2 & c\beta\, c_3\\ -s_1 & -c\beta\, c_2 & s_3\\ s\beta\, c_1 & s\beta\, c_2 & s\beta\, c_3\end{bmatrix} \tag{4}$$

Equations of Motion

In the absence of external torques:

$$\dot{\mathbf{q}}_v = -\frac{1}{2}\boldsymbol{\omega} \times \mathbf{q}_v + \frac{1}{2}q_4\boldsymbol{\omega} \tag{5a}$$

$$\dot{q}_4 = -\frac{1}{2}\boldsymbol{\omega} \cdot \mathbf{q}_v \tag{5b}$$

$$J\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \left(J\boldsymbol{\omega} + \boldsymbol{h}\right) = -\dot{\boldsymbol{h}} = -\eta A\dot{\boldsymbol{\delta}} \tag{6}$$

$$\boldsymbol{h} = \eta\left(\begin{bmatrix} -c\beta\, s_1 \\ c_1 \\ s\beta\, s_1 \end{bmatrix} + \begin{bmatrix} -c_2 \\ -c\beta\, s_2 \\ s\beta\, s_2 \end{bmatrix} + \begin{bmatrix} c\beta\, s_3 \\ -c_3 \\ s\beta\, s_3 \end{bmatrix}\right) \tag{7}$$

$$A = \begin{bmatrix} -c\beta\, c_1 & s_2 & c\beta\, c_3 \\ -s_1 & -c\beta\, c_2 & s_3 \\ s\beta\, c_1 & s\beta\, c_2 & s\beta\, c_3 \end{bmatrix} \tag{8}$$

Quaternion: $\mathbf{q} = (\mathbf{q}_v, q_4)^T$

Commanded Torque

The CMG configuration applies an effective torque onto the spacecraft body given by:

$$\boldsymbol{\tau}_{\text{eff}} = -\eta A \dot{\boldsymbol{\delta}}$$

Given a commanded torque $\boldsymbol{\tau}_c$, we can decide on the gimbal angle rates as:

$$\dot{\boldsymbol{\delta}} = -\frac{1}{\eta} A^{-1} \boldsymbol{\tau}_c$$

However, if the Jacobian A is singular, the inverse does not exist. When the CMG gimbal angles are in such a position, this is called a *singular* position. In a singular position, the Jacobian is rank-deficient:

$$\text{rank}(A) < 3$$

This means that torque cannot be provided in three dimensions, only two at the most. The objective is to avoid and/or escape from singularities.

Gauss Pseudospectral Method

We will summarize the **Gauss Pseudospectral Method**. First, the continuous Bolza problem with time $t \in [t_0, t_f]$ is transformed into the normalized time $\tau \in [-1, 1]$ with the transformation:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}$$

We wish to minimize the cost functional:

$$J = \Phi[\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f] + \frac{t_f - t_0}{2}\int_{-1}^{1} g[\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f]\, d\tau$$

Gauss Pseudospectral Method

We need to determine:

1. The state $\mathbf{x}(\tau) \in \mathbb{R}^n$
2. The control $\mathbf{u}(\tau) \in \mathbb{R}^m$
3. The initial and final times t_0, t_f

Subject to the constraints:

$$\frac{d\mathbf{x}}{d\tau} = \frac{t_f - t_0}{2} \mathbf{f}[\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f] \in \mathbb{R}^n \text{ (differential equation)}$$

$$\boldsymbol{\phi}\left[\mathbf{x}(-1), t_0, \mathbf{x}(1), t_f\right] = \mathbf{0} \in \mathbb{R}^q \text{ (boundary conditions)}$$

$$\mathbf{C}\left[\mathbf{x}(\tau), \mathbf{u}(\tau), \tau; t_0, t_f\right] \leq \mathbf{0} \in \mathbb{R}^c \text{ (path constraints)}$$

Gauss Pseudospectral Method

We collate at the Legendre-Gauss (LG) points, i.e. the roots of the Nth degree Legendre polynomial:

$$-1 < \tau_1 < \tau_2 < \cdots < \tau_N < 1$$

Let $\tau_0 = -1$. Let $L_i(\tau)$ $(i = 0, 1, 2, ..., N)$ be the Lagrange interpolating polynomials:

$$L_i(\tau) = \prod_{j=0, j \neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \;\; (i = 0, 1, 2, ..., N)$$

Note that $L_i(\tau_k) = \delta_{ik}$. We approximate the state as:

$$\mathbf{x}(\tau) \approx \mathbf{X}(\tau) = \sum_{i=0}^{N} \mathbf{X}(\tau_i) L_i(\tau) = \sum_{i=0}^{N} \mathbf{X}_i L_i(\tau)$$

Gauss Pseudospectral Method

Now let $L_i^*(\tau)$ $(i = 1, 2, ..., N)$ be defined as:

$$L_i^*(\tau) = \prod_{j=1, j\neq i}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j} \;\; (i = 1, 2, ..., N)$$

Note also that $L_i^*(\tau_k) = \delta_{ik}$. We approximate the control as:

$$\mathbf{u}(\tau) \approx \mathbf{U}(\tau) = \sum_{i=1}^{N} \mathbf{U}(\tau_i) L_i^*(\tau) = \sum_{i=1}^{N} \mathbf{U}_i L_i^*(\tau)$$

Gauss Pseudospectral Method

Differentiate $\mathbf{X}(\tau)$ to obtain an approximation for $\dot{\mathbf{x}}(\tau)$:

$$\dot{\mathbf{x}}(\tau) \approx \dot{\mathbf{X}}(\tau) = \sum_{i=0}^{N} \mathbf{X}_i \dot{L}_i(\tau)$$

Define the **differential approximation matrix** $D \in \mathbb{R}^{N \times N+1}$ with elements:

$$D_{ki} = \dot{L}_i(\tau_k)$$

where $k = 1, 2, ..., N$ and $i = 0, 1, 2, ..., N$. The differential equation is then approximated as:

$$\sum_{i=0}^{N} D_{ki} \mathbf{X}_i = \frac{t_f - t_0}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \ (k = 1, 2, ..., N) \tag{9}$$

Gauss Pseudospectral Method

We approximate $\mathbf{x}(\tau = 1)$ with $\mathbf{X}_f$, defined by using a Gauss quadrature:

$$\mathbf{X}_f = \mathbf{X}_0 + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \, \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \tag{10}$$

where w_k are the Gauss weights. The cost J is approximated also using a Gauss quadrature:

$$J = \Phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) + \frac{t_f - t_0}{2} \sum_{k=1}^{N} w_k \, g(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \tag{11}$$

The boundary and path constraints are approximated as:

$$\phi(\mathbf{X}_0, t_0, \mathbf{X}_f, t_f) = \mathbf{0} \tag{12}$$

$$\mathbf{C}(\mathbf{X}_k, \mathbf{U}_k, \tau_k; t_0, t_f) \leq \mathbf{0} \ (k = 1, 2, ..., N) \tag{13}$$

The cost (11) must be minimized subject to the constraints (9), (10), (12), and (13). This is a nonlinear programming (NLP) problem, which can be solved with any number of established software packages, such as fmincon in MATLAB.

Optimization Problem Formulation

We take the state $\mathbf{x}$ to be:

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\omega} \\ \boldsymbol{\delta} \end{bmatrix} \in \mathbb{R}^{10} \tag{14}$$

The control will be taken as $\mathbf{u} = \dot{\boldsymbol{\delta}} \in \mathbb{R}^3$. The states and controls must satisfy the coupled differential equations:

$$\dot{\mathbf{q}}_v = -\tfrac{1}{2}\boldsymbol{\omega} \times \mathbf{q}_v + \tfrac{1}{2} q_4 \boldsymbol{\omega} \tag{15a}$$

$$\dot{q}_4 = -\tfrac{1}{2}\boldsymbol{\omega} \cdot \mathbf{q}_v \tag{15b}$$

$$\dot{\boldsymbol{\omega}} = -J^{-1}\Big[\boldsymbol{\omega} \times \big(J\boldsymbol{\omega} + \boldsymbol{h}\big) + \eta A \mathbf{u}\Big] \tag{16}$$

$$\dot{\boldsymbol{\delta}} = \mathbf{u} \tag{17}$$

The initial time will be fixed at $t_0 = 0$. The final time t_f will be free.

Optimization Problem Formulation

We proceed in two stages. First, we optimize with respect to the cost function:

$$C_1 = t_f$$

With the resulting t_f, we set an upper bound on the final time (say $1.35t_f$). Then in the second stage, we optimize with respect to:

$$C_2 = -\int_0^{t_f} \left| \det(A) \right| dt$$

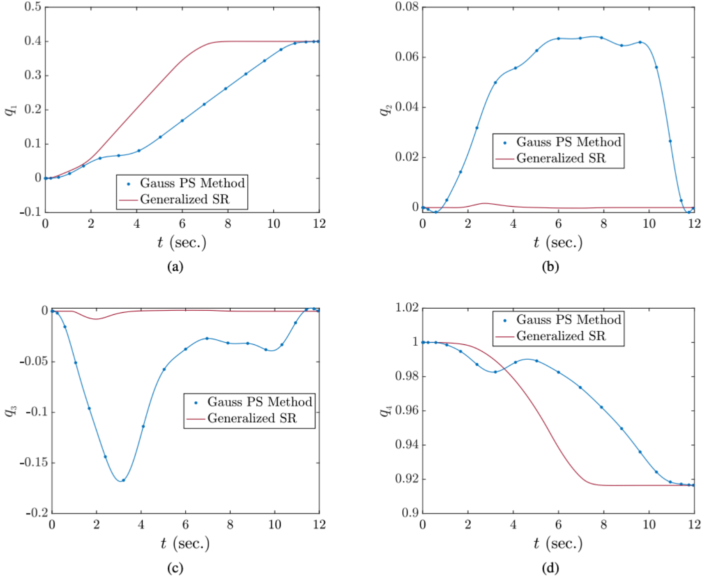
Optimization Problem Formulation

Skew angle, β	53.13°
CMG axial momentum, η	$1000\ \mathrm{N} \cdot \mathrm{m} \cdot \mathrm{s}$
Spacecraft inertia tensor, J	$\mathrm{diag}(21400, 20100, 5000)\ \mathrm{kg} \cdot \mathrm{m}^2$
Max gimbal angle rate	2 rad/s
Max spacecraft slew rate	10 deg/s
Initial angular velocity, $\boldsymbol{\omega}(0)$	$(0, 0, 0)^T$
Final angular velocity, $\boldsymbol{\omega}(t_f)$	$(0, 0, 0)^T$
Initial quaternion, $\mathbf{q}(0)$	$(0, 0, 0, 1)^T$
Final quaternion, $\mathbf{q}(t_f)$	$(0.4,\ 0,\ 0,\ \sqrt{0.84})^T$
Initial gimbal angles (Case 1)	$(60,\ 180,\ -60)^T$ deg.
Initial gimbal angles (Case 2)	$(90,\ 0,\ -90)^T$ deg.

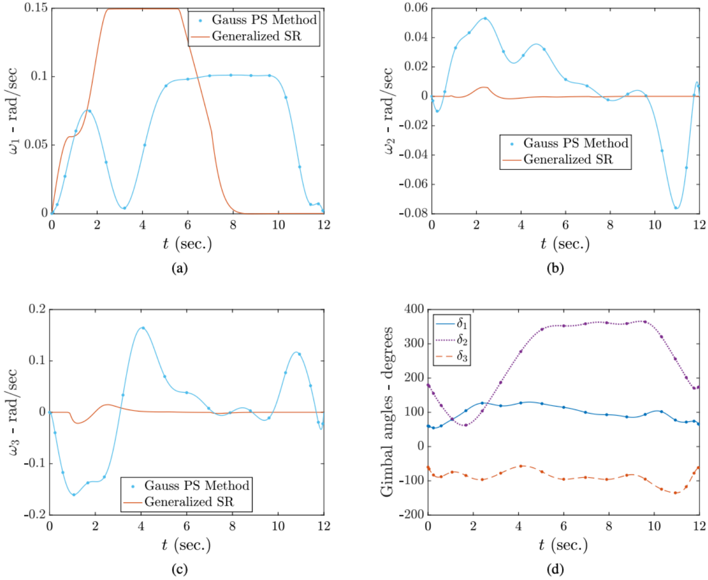
Table: **Simulation Parameters**

The maneuver trajectories were plotted in comparison with the trajectories computed using the generalized singularity robust (SR) steering [1] applied to the (non-failed) 4-CMG pyramid configuration.

Simulation Results (Case 1)



Simulation Results (Case 1)



(a) (b) (c) (d)

Simulation Results (Case 1)

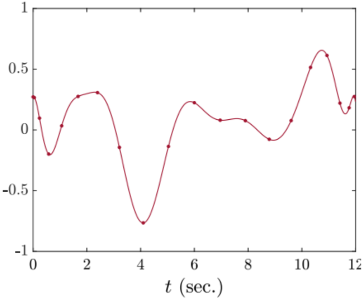


Figure: Jacobian determinant: $\mathrm{Det}(A)$

Advantages/Disadvantages

Advantages:

- Provides a steering method in the failure case where only three CMGs are operational.
- Cost function can be tailored to mission-specific requirements, whether more effort should be focused on minimum time or maximum singularity avoidance.

Disadvantages:

- This method is open-loop, not closed-loop—meant for trajectory computation before the maneuver, not during.
- Solution might not converge properly if the initial guess is not proper.

Optimal Steering of Nonredundant Single-Gimbal CMGs using Gauss Pseudospectral Method

Victor Hakim
Santa Clara University
500 El Camino Real
Santa Clara, CA 95053
vhakim@scu.edu

Mohammad Ayoubi
Santa Clara University
500 El Camino Real
Santa Clara, CA 95053
maayoubi@scu.edu

***Abstract*— This paper presents the problem of optimal steering with three single-gimbal control moment gyros (SGCMGs). Most SGCMG configurations use more than three rotors because the resulting Jacobian would always have a nontrivial nullspace. We consider the specific case of a four-CMG pyramid configuration where one CMG has failed and only three CMGs are operational. We use the Gauss pseudospectral method to find the optimal gimbal angle trajectories of the CMGs and the states of an agile spacecraft subject to mission-specific constraints while maximally avoiding the angular-momentum singularities. The proposed steering law is tested in scenarios where the gimbal angles initially start in both nonsingular and singular positions. Results are compared with the standard four-SGCMG maneuvering via the generalized singularity robust steering law in the literature.**

Table of Contents

1. Introduction 1
2. CMG Equations of Motion 2
3. The Gauss Pseudospectral Method 3
4. Optimal Control Formulation 4
5. Simulation Results 4
6. Conclusions 7
Appendix: Clarification on Spacecraft Tensor J 7
References 8
Biography 8

1. Introduction

For the past half-century, control moment gyroscopes (CMGs) have been studied extensively and used for the atti-

so faithful SGCMG implementations are becoming more common. Most SGCMG configurations would use $n \geq 4$ rotors in order to be redundant; that is, the resulting Jacobian would always have a nontrivial nullspace. With a nontrivial nullspace, the gimbal angles can be varied without producing an output torque on the spacecraft body, and this so-called "null motion" helps to steer the gimbal angles out of or away from singularities without excessive deviation from the commanded torque. The gimbal angles are in a singularity if the possible output torque, which should ideally span three dimensions, is reduced in dimensionality to two (or one) dimensions. Avoiding singularities is the main mathematical difficulty addressed in any SGCMG steering method. Since SGCMGs are the predominant focus of study, we will hereafter take "CMG" to mean "SGCMG."

Most of the literature [1], [4], [5], [6], [7], [8], [11], [12], [13], [14], [17], [18], [19] focuses on analyzing the four-CMG pyramid configuration, depicted in Figure 1. Clearly, an $n = 4$ configuration is advantageous because it is the most mass-efficient while still being redundant, as long as the steering algorithm can effectively deal with singular states. However, if one CMG rotor fails, then the configuration becomes nonredundant, and singularity avoidance is much more difficult. It is still possible to provide three-axis attitude control on the spacecraft as long as there are at least three functioning CMG rotors with independent gimbal axes. Not much literature is devoted to gimbal steering in the case of three-CMG configurations. Ref. [19] proposes a method by constraining the momentum envelope, but the method requires the skew angle β to be varied, which isn't usually possible in SGCMG designs. Ref. [6] provides "preferred" gimbal angles for three-CMG configurations, a useful construct in developing a steering law. This paper deals with the specific case of a four-CMG pyramid configuration where one

Accepted at 2021 IEEE Aerospace Conference in March.

`https://github.com/hakimv2322/IEEE_CMG_paper`

Subsequent Investigations...

- Can we find a closed-loop steering law for the nonredundant CMG configuration?
- What other results can we find?

Adjugate instead of inverse?

The Jacobian's inverse, if it exists, is given by:

$$A^{-1} = \frac{\text{adj}(A)}{\det(A)}$$

Near singularity, the determinant goes to zero. So what if we just use the adjugate near singularity?

Proposal: Near singularity, use $\text{adj}(A)$ instead of A^{-1}.

$$\dot{\boldsymbol{\delta}} = -\frac{1}{\eta}\text{adj}(A)\boldsymbol{\tau}_c$$

This doesn't work...

Adjugate instead of inverse?

The CMG configuration applies an effective torque onto the spacecraft body given by:

$$\boldsymbol{\tau}_{\text{eff}} = -\eta A\dot{\boldsymbol{\delta}}$$

Notice:

$$\lim_{\det(A)\to 0} A\,\mathrm{adj}(A) = \lim_{\det(A)\to 0} \det(A)\left(AA^{-1}\right) = 0_{n\times n}$$

When near singularity, we're basically doing nothing!

Lemma: $\text{rank}(A) \geq 2$

Lemma: *In the 3/4 pyramid configuration, the rank of the Jacobian is always at least two:*

$$\text{rank}(A) \geq 2$$

Thus either the Jacobian is invertible and has rank 3, or the Jacobian is singular and has rank no less than 2.

Note that this lemma is not true for other CMG configurations. For example, in any roof-type configuration, if we point all CMG angular momentum vectors toward the “crease” of the roof, the Jacobian will be rank 1 in this instance.

Lemma: $\text{rank}(A) \geq 2$

Proof of lemma: Consider the ith CMG rotor. The rotor is gimbaled about the gimbal axis unit vector $\hat{\mathbf{g}}_i$ (which is constant in the body frame). That is, the rotor's angular momentum $\boldsymbol{h}_i$ can be rotated to point in any direction perpendicular to $\hat{\mathbf{g}}_i$. Define $\hat{\mathbf{u}}_i = \dot{\boldsymbol{h}}_i / \|\dot{\boldsymbol{h}}_i\|_2$ to be the unit vector in the direction of $\dot{\boldsymbol{h}}_i$. Note that $-\hat{\mathbf{u}}_i$ represents the direction of torque applied onto the spacecraft body by the ith CMG.

The Jacobian A is rank 1 if and only if all the $\hat{\mathbf{u}}_i$ ($i = 1, 2, 3$) are parallel or anti-parallel. For the ith CMG, the set of all possible $\hat{\mathbf{u}}_i$ forms a unit circle in $\mathbb{R}^3$. This unit circle is a *great circle* on the unit sphere in $\mathbb{R}^3$; call this great circle $\text{gc}(\hat{\mathbf{u}}_i)$. $\hat{\mathbf{u}}_i$ is parallel or anti-parallel to $\hat{\mathbf{u}}_j$ ($i \neq j$) if and only if both $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{u}}_j$ are on an intersection point of $\text{gc}(\hat{\mathbf{u}}_i)$ and $\text{gc}(\hat{\mathbf{u}}_j)$. It is well known that two distinct great circles on a sphere will have exactly two intersection points, and those two intersection points will be antipodal on the sphere.

Lemma: $\text{rank}(A) \geq 2$

Proof, continued. In order for all the $\hat{\mathbf{u}}_i$ to be parallel or anti-parallel, there must exist a mutual intersection point of all $\text{gc}(\hat{\mathbf{u}}_i)$ $(i = 1, 2, 3)$. For the 3/4 pyramid configuration, no such intersection point exists. We can see that $\text{gc}(\hat{\mathbf{u}}_1)$ and $\text{gc}(\hat{\mathbf{u}}_3)$ intersect at $\hat{\mathbf{e}}_2$ and $-\hat{\mathbf{e}}_2$ in $\mathbb{R}^3$. However, $\text{gc}(\hat{\mathbf{u}}_2)$ passes through neither $\hat{\mathbf{e}}_2$ nor $-\hat{\mathbf{e}}_2$. Thus it is impossible for all the $\hat{\mathbf{u}}_i$ to be parallel or anti-parallel, and so the Jacobian can never be rank 1. Its rank must be 2 or 3. $\square$

Planar Range Approximation

Given a commanded torque $\boldsymbol{\tau}_c$, we can decide on the gimbal angle rates as:

$$\dot{\boldsymbol{\delta}} = -\frac{1}{\eta} A^{-1} \boldsymbol{\tau}_c$$

As the 3/4 pyramid configuration approaches singularity, $\dot{\boldsymbol{\delta}}$ goes to infinity. The range space of A, which should ideally be $\mathbb{R}^3$, becomes planar at singularity. (By the Lemma, we know the range space can never be one-dimensional.)

When the CMG configuration is *near* singularity, but not exactly at singularity, the range space is still the full $\mathbb{R}^3$. The range space is only planar at the instant when $\det(A) = 0$. In deciding how to steer the gimbal angles when nearing singularity, it would be helpful to find an approximation for the planar range of the approaching singularity.

Planar Range Approximation

Let $\mathbf{c}_i = (x_i, y_i, z_i)^T$ for $i = 1, 2, 3$ be the three column vectors of A. The span of $\{\mathbf{c}_i\}$ is the range space of A. A is singular exactly when its three column vectors lie in the same plane. When A is nearing singularity, the column vectors are "almost" in the same plane.

Thus we may take our singular planar range approximation to be the plane which "fits best" among the three nearly co-planar column vectors. This is exactly the plane $\mathbb{P}$ which goes through the origin and is the plane of best fit between the three points $\mathbf{c}_i$.

Planar Range Approximation

We'll write our plane $\mathbb{P}$ in the form:

$$z = ax + by$$

where a, b are two real constants to be determined. Once a, b are determined, we can write for each point $\mathbf{c}_i$:

$$z_i = ax_i + by_i + \epsilon_i$$

where ϵ_i is the corresponding error from the plane. We can write the equations for $i = 1, 2, 3$ as:

$$\mathbf{z} = X\mathbf{w} + \boldsymbol{\epsilon}$$

where $\mathbf{z} = (z_1, z_2, z_3)^T$, $\mathbf{w} = (a, b)^T$, $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \epsilon_3)^T$ and

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}$$

Planar Range Approximation

We wish to choose the weights a, b so that the sum of the squares of the errors is minimized. That is, we wish to minimize:

$$\boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = \left(\mathbf{z} - X\mathbf{w}\right)^T \left(\mathbf{z} - X\mathbf{w}\right)$$

Taking a gradient with respect to **w** and setting equal to zero, we get:

$$\mathbf{w} = \left(X^T X\right)^{-1} X^T \mathbf{z}$$

$\left(X^T X\right)^{-1} X^T$ is the expression for the Moore-Penrose pseudoinverse when X has more rows than columns.

Planar Range Approximation

Once we have our planar equation $z = ax + by$, we need to find an orthonormal basis $\mathcal{B}_{\mathbb{P}} = \left\{\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2\right\}$ for our plane $\mathbb{P}$. There are many ways to do this. For example, let

$$\mathbf{b}_1 = \begin{bmatrix} 1 \\ 1 \\ a+b \end{bmatrix}, \quad \hat{\mathbf{b}}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|_2}$$

$$\mathbf{b}_2 = \begin{bmatrix} a \\ b \\ -1 \end{bmatrix} \times \hat{\mathbf{b}}_1, \quad \hat{\mathbf{b}}_2 = \frac{\mathbf{b}_2}{\|\mathbf{b}_2\|_2}$$

Modified Torque

When nearing singularity, the plane $\mathbb{P}$ approximates the range of possible torques at singularity. We can decompose the commanded torque $\boldsymbol{\tau}_c$ into components parallel and perpendicular to $\mathbb{P}$:

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_{\parallel} + \boldsymbol{\tau}_{\perp}$$

where:

$$\boldsymbol{\tau}_{\parallel} = \left(\boldsymbol{\tau}_c \cdot \hat{\mathbf{b}}_1\right)\hat{\mathbf{b}}_1 + \left(\boldsymbol{\tau}_c \cdot \hat{\mathbf{b}}_2\right)\hat{\mathbf{b}}_2$$

$$\boldsymbol{\tau}_{\perp} = \boldsymbol{\tau}_c - \boldsymbol{\tau}_{\parallel}$$

The perpendicular component is the component which requires gimbal rates to approach infinity. At singularity, no perpendicular torque is possible.

Modified Torque

Near singularity, we will modify the commanded torque so that no component is perpendicular to the plane $\mathbb{P}$. Instead of trying to provide perpendicular torque, we will replace $\boldsymbol{\tau}_{\perp}$ with equal effort put into "pushing" the plane $\mathbb{P}$ toward the commanded torque $\boldsymbol{\tau}_c$. This would help the spacecraft maneuver through singular positions. We choose our modified torque to be:

$$\begin{aligned}
\boldsymbol{\tau}_{\text{mod}} &= \boldsymbol{\tau}_{\parallel} + \|\boldsymbol{\tau}_{\perp}\|_2 \frac{\boldsymbol{\tau}_{\parallel} \times \boldsymbol{\tau}_{\perp}}{\|\boldsymbol{\tau}_{\parallel} \times \boldsymbol{\tau}_{\perp}\|_2} \\
&= \boldsymbol{\tau}_{\parallel} + \frac{\boldsymbol{\tau}_{\parallel} \times \boldsymbol{\tau}_{\perp}}{\|\boldsymbol{\tau}_{\parallel}\|_2}
\end{aligned}$$

Calculating Gimbal Rates Near Singularity

When the CMG configuration is near singularity, but not at singularity, row-reducing the Jacobian A simply gives the 3×3 identity matrix. We want to approximate A with the nearby singular matrix whose range space is $\mathbb{P}$. Thus we'll replace each column $\mathbf{c}_i$ with its projection onto the plane $\mathbb{P}$:

$$\overrightarrow{\text{proj}}_{\mathbb{P}}\, \mathbf{c}_i = \left(\mathbf{c}_i \cdot \hat{\mathbf{b}}_1\right)\hat{\mathbf{b}}_1 + \left(\mathbf{c}_i \cdot \hat{\mathbf{b}}_2\right)\hat{\mathbf{b}}_2$$

Define A_{proj} to be the resulting matrix when each column of A is replaced with its corresponding projection onto $\mathbb{P}$. Note that A_{proj} is singular with rank 2. The range of A_{proj} is exactly $\mathbb{P}$.

Calculating Gimbal Rates Near Singularity

We constructed $\boldsymbol{\tau}_{\text{mod}}$ to be in the plane $\mathbb{P}$. Therefore, there exist gimbal rates $\dot{\boldsymbol{\delta}}$ such that:

$$A_{\text{proj}}\dot{\boldsymbol{\delta}} = -\frac{1}{\eta}\boldsymbol{\tau}_{\text{mod}} \tag{18}$$

Because A_{proj} is rank 2, the reduced row-echelon form of A_{proj} has a row of zeros. We can fully row-reduce equation 18 to become:

$$\widetilde{A}\,\dot{\boldsymbol{\delta}} = \boldsymbol{\gamma} \tag{19}$$

where $\widetilde{A}$ is a 2×3 matrix and $\boldsymbol{\gamma}$ is a column vector of length 2.

Calculating Gimbal Rates Near Singularity

Because equation 19 is only two equations, we have an extra degree of freedom in choosing $\dot{\delta}_1, \dot{\delta}_2, \dot{\delta}_3$. Thus we may choose the gimbal rates so as to minimize the cost:

$$C = \dot{\delta_1}^2 + \dot{\delta_2}^2 + \dot{\delta_3}^2$$

This is a constrained optimization problem with two constraints given by equation 19. Define the Lagrangian $\mathcal{L}$ as:

$$\mathcal{L} = \dot{\delta_1}^2 + \dot{\delta_2}^2 + \dot{\delta_3}^2 + \boldsymbol{\lambda}^T \left(\gamma - \widetilde{A}\,\dot{\boldsymbol{\delta}}\right)$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)^T$ are the two Lagrange multipliers associated with each constraint.

Calculating Gimbal Rates Near Singularity

Let $\tilde{\mathbf{c}}_i$ $(i = 1, 2, 3)$ be the column vectors of $\widetilde{A}$. In solving our constrained optimization problem, we set the gradients of the Lagrangian to zero:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \dot{\delta}_i} &= 2\dot{\delta}_i - \boldsymbol{\lambda}^T \tilde{\mathbf{c}}_i = 0 \\
\dot{\delta}_i &= \frac{1}{2}\boldsymbol{\lambda}^T \tilde{\mathbf{c}}_i \\
\Rightarrow \dot{\boldsymbol{\delta}}^T &= \frac{1}{2}\boldsymbol{\lambda}^T \widetilde{A} \\
\dot{\boldsymbol{\delta}} &= \frac{1}{2}\widetilde{A}^T \boldsymbol{\lambda} \qquad (20)
\end{aligned}
$$

Calculating Gimbal Rates Near Singularity

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = \boldsymbol{\gamma} - \widetilde{A}\dot{\boldsymbol{\delta}} = 0$$
$$\boldsymbol{\gamma} = \widetilde{A}\dot{\boldsymbol{\delta}}$$

Plugging in equation 20:

$$\boldsymbol{\gamma} = \frac{1}{2}\widetilde{A}\widetilde{A}^T\boldsymbol{\lambda}$$
$$\boldsymbol{\lambda} = 2\left(\widetilde{A}\widetilde{A}^T\right)^{-1}\boldsymbol{\gamma}$$

Plugging this result back into equation 20, we get:

$$\dot{\boldsymbol{\delta}} = \widetilde{A}^T\left(\widetilde{A}\widetilde{A}^T\right)^{-1}\boldsymbol{\gamma}$$

$\widetilde{A}^T\left(\widetilde{A}\widetilde{A}^T\right)^{-1}$ is the expression for the Moore-Penrose pseudoinverse when $\widetilde{A}$ has more columns than rows. Note that $\widetilde{A}\widetilde{A}^T$ is always full rank because $\widetilde{A}$ always has rank 2. Therefore, the inverse of $\widetilde{A}\widetilde{A}^T$ always exists.

Saturation Functions

Given a scalar x and a positive constant L, define the *scalar saturation function* as:

$$\underset{L}{\text{sat}}(x) = \begin{cases} L & \text{if} \quad x \geq L \\ x & \text{if} \quad |x| \leq L \\ -L & \text{if} \quad x \leq -L \end{cases}$$

We can apply the scalar saturation function component-wise to a vector $\mathbf{x} = (x_1, x_2, ..., x_n)^T$:

$$\underset{L_i}{\text{sat}}(\mathbf{x}) = \begin{bmatrix} \text{sat}_{L_1}(x_1) \\ \text{sat}_{L_2}(x_2) \\ \vdots \\ \text{sat}_{L_n}(x_n) \end{bmatrix}$$

given the positive constants $L_1, L_2, ..., L_n$.

Saturation Functions

Given a vector $\mathbf{x}$ and a positive constant U, define the *vector saturation function* as:

$$\underset{U}{\overrightarrow{\text{sat}}}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if} \quad \mu(\mathbf{x}) < U \\ \dfrac{U\mathbf{x}}{\mu(\mathbf{x})} & \text{if} \quad \mu(\mathbf{x}) \geq U \end{cases}$$

Here, $\mu(\mathbf{x})$ is a positive scalar norm of $\mathbf{x}$ which is chosen in context. For example, we may choose the L-2 norm $\mu(\mathbf{x}) = \|\mathbf{x}\|_2 = \sqrt{\mathbf{x} \cdot \mathbf{x}}$ or the L-∞ norm $\mu(\mathbf{x}) = \|\mathbf{x}\|_\infty = \max\{|x_1|, ..., |x_n|\}$.

From here on, we'll use $\mu(\mathbf{x}) = \|\mathbf{x}\|_\infty = \max\{|x_1|, ..., |x_n|\}$.

Proposed Closed-Loop Steering Logic

Following the quaternion feedback logic developed in [1, **?**], we propose a closed-loop steering logic for the 3/4 pyramid configuration as follows:

$$\boldsymbol{\tau}_c = -J\left[2k \operatorname*{sat}_{L_i}\left(\boldsymbol{e} + \frac{1}{T}\int \boldsymbol{e}\right) + c\boldsymbol{\omega}\right] + \boldsymbol{\omega} \times \boldsymbol{h} \tag{21}$$

$$L_i = \frac{c}{2k}\ \min\left\{\sqrt{4a_i|e_i|},\ |\omega_i|_{\max}\right\} \tag{22}$$

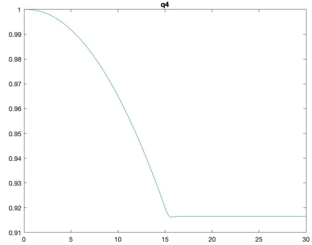
If $|\det(A)| > 0.1$, then set the commanded gimbal rates as:

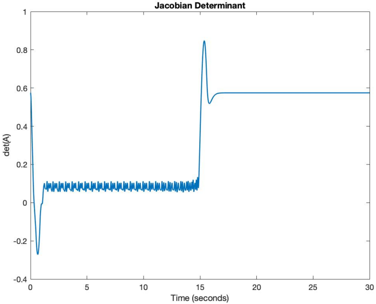
$$\dot{\boldsymbol{\delta}}_c = \operatorname*{\overrightarrow{sat}}_{\dot{\boldsymbol{\delta}}_{\max}}\left(-\frac{1}{\eta}A^{-1}\boldsymbol{\tau}_c\right) \tag{23}$$

If $|\det(A)| \leq 0.1$, solve for $\mathbb{P}$, $\boldsymbol{\tau}_{\text{mod}}$, A_{proj}, $\widetilde{A}$, $\boldsymbol{\gamma}$, and set:

$$\dot{\boldsymbol{\delta}}_c = \operatorname*{\overrightarrow{sat}}_{\dot{\boldsymbol{\delta}}_{\max}}\left(\widetilde{A}^T\left(\widetilde{A}\widetilde{A}^T\right)^{-1}\boldsymbol{\gamma}\right) \tag{24}$$

This also doesn't work...





Possible Nonredundant Closed-Loop Steering?

I've found a recent paper from 2018 which proposes a closed-loop steering method for the 3/4 pyramid configuration [2].

Next steps: investigate this steering and try to reproduce their results.

The End

Thank you very much!!

References

[1] B. Wie, et al.
"Singularity Robust Steering Logic for Redundant Single-Gimbal Control Moment Gyros,"
Journal of Guidance, Control, and Dynamics. Vol. 24, No. 5, 2001.

[2] A. Meldrum, et al.
"Attitude Control Using Three Control Moment Gyros,"
Japan Society for Aeronautical and Space Sciences. Vol. 16, No. 5, 2018.