Bisection method for CMG steering logic in satellite attitude control

Max Demenkov * Evgeny Kryuchenkov **

* Department of Engineering, Faculty of Technology, De Montfort University, Queens bld. 2.14, The Gateway, Leicester LE1 9BH, UK (fax: +44(0)871 263 74 16, e-mail: demenkov@dmu.ac.uk)
** Faculty of Cybernetics, Moscow State Institute of Radioengineering, Electronics and Automation, pr. Vernadskogo 78, Moscow 119454, Russia (fax: +7(495)434 95 91, e-mail: evgencpp@mail.ru)

Abstract: Control moment gyros (CMG) are actuators for agile spacecraft attitude control. We connect the problem of moment distribution in CMG with the well-known control allocation problem in aircrafts. Using this analogy, we propose a novel moment distribution algorithm that is based on a multidimensional interval bisection technique presented before in a different context.

Keywords: satellite attitude control, control moment gyros, control allocation, actuator saturation

1. INTRODUCTION

The Control Moment Gyros (CMG) cluster has been studied for several decades as a basis for the space vehicles attitude and momentum control systems (Wie (2008)). Large space stations such as the Mir and the International Space Station utilized CMG as primary actuators. Recently, such a research led to the first European high-resolution imaging satellite being controlled by a pyramid cluster of four single-gimbaled CMG (Thieuw and Marcille (2007)). A renewed practical and theoretical interest arises in developing and analysis of CMG-based attitude control systems (see e.g. Lappas et al. (2005); Somov et al. (1999, 2003, 2007); MacKunis et al. (2008); Bhat and Tiwari (2006); Duozyev et al. (2005); Ignatov and Sazonov (2007)), especially for small agile satellites.

When we have multiple control effectors (such as more than three CMG, see Fig. 1) producing moments along different axes, the control generation problem may not be unique even if we choose a particular control law. The required moments to solve an upper-level control problem can be distributed between available control effectors in different ways.

The CMG steering logic, which generates the CMG gimbal rate commands for the commanded spacecraft control torques, is frequently based on the pseudo-inversion of the Jacobian matrix (Wie (2008)). Despite its computational simplicity, it suffers from the well-known singularity problem. The problem characterized by excessively large gimbal rates near a singular state, where rank of the Jacobian decreases (Wie (2008)). Various singularity-robust laws have been proposed to avoid this problem, including some modifications of the original pseudo-inverse equations as well as using variable-speed CMG (Wie (2008); Lappas et al. (2004); Ford and Hall (2000); Lee et al. (2007); Yoon and Tsiotras (2004); Pechev (2007)). Another drawback of the pseudo-inverse solution is that it may not utilize the whole attainable angular momentum rates set (MRS), induced by the gimbal rate constraints (see its definition below). Note that in practice it is hard to propose an alternative solution to the pseudo-inverse due to the simplicity of its on-board implementation (Thieuw and Marcille (2007)). Therefore, alternative steering algorithms having low algorithmic complexity and improved handling of control constraints may be of interest for CMG control engineers.

Despite the problem of CMG steering logic has been studied separately for many years, it has a lot of similarities with the more general problem of control allocation. Control allocation is quite useful for control of overactuated systems, and deals with distributing the total control demand among the individual actuators. Using control allocation, the actuator selection task is separated from the regulation task in the control design.

The idea of control allocation allows to deal with control constraints and actuator faults separately from the design of the main regulator, which uses virtual unconstrained control input. In case of fault, instead of reconfiguring the main control law, we change only the distribution of the virtual control input among physical actuators.

This constrained control allocation problem has been attracting much attention for more than 15 years since the first algorithm of this type - the so called direct allocation approach (Durham (1993, 1994)). The reason for a new approach to appear instead of the pseudoinverse solution (proposed initially for aircraft control as well as for CMG) was its inability to utilize the whole attainable set of solutions, which was proved in these first publications. For many years, control allocation has been studied almost solely within the aeronautical community, but recently the idea of control allocation was applied to attitude control of satellites (based on real-time optimization in Pulecchi and Lovera (2007)) as well as control of advanced cars (Tondel and Johansen (2005); Laine and Fredriksson (2008)) and redundant robotic manipulators (Altay (2006); Pechev (2008)).

It appears that control allocation principles can help to propose a new solution for CMG control with algorithmic complexity not exceeding the pseudo-inverse solution. Our solution (initially proposed for aircraft applications in Demenkov (2005, 2007)) can utilize the whole attainable momentum rates set, which may be very important for the agility of a satellite. The
From mathematical viewpoint, the problem of determining \( \ddot{\delta} \) for a given \( \dot{\delta} \) is the root-finding problem, and all allocation algorithms actually differ one from another by the root-finding method.

To produce correct gimbal rates in the vicinity of a pseudo-inverse singularity, the algorithm is applied to both 2D and 3D problems and the results are compared to detect the correct dimension. The normal to a 2D plane in the case of reduced dimension is obtained directly from the solution of the 3D problem. Nevertheless, a correct momentum rates vector should be supplied by the upper-level logic to pass through a singularity (if it is possible).

2. LINEAR MODEL

Let us suppose that 3-dimensional CMG angular momentum vector \( h \) can be obtained as

\[
h = H(\delta),
\]

where \( \delta \) is the 3-dimensional vector of gimbal angles.

Its time derivative

\[
\dot{h} = J(\delta) \dot{\delta},
\]

where \( J(\delta) = [J_1|J_2|...|J_m] \) is the Jacobian of \( H(\delta) \).

Gimbal rates \( \dot{\delta} \) are supposed to be limited by some maximal values:

\[
\dot{\delta} \in B, \ B = \{ \dot{\delta} \in R^m : |\dot{\delta}^{(i)}| \leq \dot{\delta}^{(i)}_{\text{max}}, i = 1,m \},
\]

here \( \dot{\delta}^{(i)} \) - \( i \)-th component of the vector.

The achievable momentum rates \( \dot{h} \) are then confined to some bounded momentum rates set (MRS) \( R \), which in general case is a 3-dimensional polytope:

\[
R(B) = \{ \dot{h} : \dot{h} = J(\delta) \dot{\delta}, \dot{\delta} \in B \}
\]

with constantly changing \( J(\delta) \).

From mathematical viewpoint, the problem of determining \( \dot{\delta} \) for a given \( \dot{h} \) is the root-finding problem, and all allocation algorithms actually differ one from another by the root-finding method.

The on-board implementation of a control allocation algorithm for satellites needs to be computationally effective and should time of computation may exceed the pseudo-inverse approach, but this time is known in advance. For any given nonzero accuracy, the Jacobian and a momentum rates vector, the algorithm yields the solution in a finite and known in advance number of iterations.

For any given accuracy, the Jacobian \( J(\delta) \) and a vector of gimbal rates \( \dot{\delta} \), the algorithm yields the solution \( \dot{h} \) in a finite number of iterations, while utilizing the whole attainable torque rates set \( R(B) \). The complexity of the algorithm is less than for optimization-based methods or direct allocation. Moreover, during the iterations the volume of the search space decreases exponentially and the number of required basic operations is proportional to the logarithm of the reciprocal of the accuracy. The control allocator based on this algorithm is therefore easily adaptable to any changes in \( J(\delta) \).

3. INTERVAL BISECTION

Let us recall the simple idea of the bisection method for a function of one variable. Over some interval the function is known to pass through zero because it changes sign. Evaluate the function at the interval’s midpoint and examine its sign. Use the midpoint to replace whichever limit has the same sign. Know the function at the interval’s midpoint and examine its sign. Evaluate the function at the interval’s limit's midpoint and examine its sign. Use the midpoint to replace whichever limit has the same sign.

In this paper we introduce a new method for CMG steering logic, which was initially developed for aircraft control allocation tasks in Demenkov (2005) — a version of generalized interval bisection. Our control allocation algorithm satisfies three criteria:

1. guarantee of convergence to a solution
2. a known upper bound for time to find a solution
3. the size of errors can be controlled

For any given accuracy, the Jacobian \( J(\delta) \) and a vector of gimbal rates \( \dot{\delta} \), the algorithm yields the solution \( \dot{h} \) in a finite number of iterations, while utilizing the whole attainable torque rates set \( R(B) \). The complexity of the algorithm is less than for optimization-based methods or direct allocation. Moreover, during the iterations the volume of the search space decreases exponentially and the number of required basic operations is proportional to the logarithm of the reciprocal of the accuracy. The control allocator based on this algorithm is therefore easily adaptable to any changes in \( J(\delta) \).
an interval of size $\varepsilon_i = b_i - a_i$ (see Fig. 6), then after the next iteration it will be bracketed within an interval of size

$$\varepsilon_{i+1} = \varepsilon_i / 2. \quad (5)$$

Thus, we know in advance the number of iterations $N$ required to achieve a given tolerance in the solution:

$$\Delta \approx \frac{\varepsilon_0}{2^N} \Rightarrow N \approx \log_2 \frac{\varepsilon_0}{\Delta}, \quad (6)$$

where $\varepsilon_0$ is the size of the initial interval, $\Delta$ is the desired ending tolerance.

![Fig. 3. Bisection method for one variable function (courtesy of Wikipedia)](image)

This classical bisection method can be generalized for $n$-dimensional problems, and has been extensively studied in the context of the so called interval analysis (Jaulin et al. (2001)). Nevertheless, in general it is impossible to construct its generalization in the same way as for the one-dimensional case, because it is hard to prove that the generalized interval in $n$ dimensions does not contain any solution. The number of intervals potentially containing a solution is then growing exponentially and this restricts the applicability of the approach.

In our case, however, it is possible, and the one-dimensional version of the algorithm can be generalized in the following way. Notice that (3) defines a box $B$ in the Euclidean space $R^n$ of all possible gimbal rate vectors $\delta$. In other words, it defines a subset of the space that is overall bounded by hyperplanes orthogonal to the axes of coordinates. Suppose that we have a method to determine if the given vector $h$ is inside the attainable momentum rates set $R(B)$ for the given box $B$. Then we cut the box $B$ into two boxes $B_1$ and $B_2$ by half-splitting it along the coordinate direction, in which $B$ is longest. We check each box for the ability to generate the given vector, replace $B$ by one of the two new boxes that has $h$ in its MRS, and repeat the procedure, constructing the diminishing sequence of boxes:

$$B \leftarrow \begin{cases} B_1, & \text{if } h \in R_1 = R(B_1); \\ B_2, & \text{if } h \in R_2 = R(B_2). \end{cases} \quad (7)$$

After $m$ steps of this procedure, we will have the longest facet of $B$ two times less than for the original box. So, if we specify in the same way some tolerance $\Delta$ for the longest facet of the box, we will obtain the solution in $N m$ bisection steps, where $N$ is given by (6) if we treat $\varepsilon_0$ as the length of the longest facet of the initial box.

![Fig. 4. The idea of our bisection algorithm](image)

$$\varepsilon_0 = \max_{i=1,m} \delta^{(i)}_{\text{max}}. \quad (8)$$

To guarantee the convergence to a solution, we must guarantee that a given vector $h_0$ belongs to the MRS of the initial box. For this, one can check the vector and replace it by some vector $h$ lying on the MRS boundary, if it violates the constraints. The easiest way to do so is to just scale the given $h_0$ preserving its direction, like in the direct allocation approach (see Fig. 4).

It is possible that both $B_1$ and $B_2$ contain the solution. In this case, one can apply some optimality criteria to decide which box will be deleted. For example, we can choose a box that has inside the previously generated vector $\delta$, to minimize the distance between two consequently generated rate vectors.

### 4. NUMERICAL ALGORITHM

The following result was first used for control allocation purposes in Petersen and Bodson (2000):

**Theorem 1.** Any normal vector $d$ of a facet of the polytope $R$ is a scaled cross product of some two columns $J_i$ and $J_k$ taken from the matrix $J(\delta)$:

$$d^{(1)} = J_i^{(2)} J_k^{(3)} - J_i^{(3)} J_k^{(2)},$$

$$d^{(2)} = J_i^{(3)} J_k^{(1)} - J_i^{(1)} J_k^{(3)},$$

$$d^{(3)} = J_i^{(1)} J_k^{(2)} - J_i^{(2)} J_k^{(1)}. \quad (8)$$

Note that for any facet normal vector $d$ there exists its opposite in sign vector $-d$, which is defined as the normal vector of the opposite facet. Because of this, our polytope is a symmetric one.

Suppose that we have all columns of $J(\delta)$ in a list and determine all pairs of one column $J_i$ and any other column from the list; the number of such pairs is $m - 1$ and any pair gives us two valid facets of the polytope. In the next step, we have to remove this column ($J_i$) from the list and repeat the procedure (now the number of pairs is $m - 2$). Proceed the same way until we have at least two columns. The maximum number of facets $N_f$ and therefore the complexity of the facet determination procedure is given by the next equation:
It is clear that $N_f < 2m^2$ and the complexity is at least polynomial. Generally, in the case of several identical (or scaled by a factor) columns in the matrix $J(\delta)$ the number of facets is less than $N_f$ because different couples produce normal vectors in the same direction in space.

If we compute vectors $d_k$ for all possible non-degenerate combinations of two columns of matrix $J(\delta)$, we can be sure that we have caught all directions perpendicular to MRS facets. The particular magnitude of these vectors is not important for our procedure (i.e. we do not need to normalize them first).

It is possible that our Jacobian leads not to 3-dimensional, but 2-dimensional MRS. In this case, it is still possible to find directions $d_k$ in the plane (see Demenkov (2008) for the particular details).

Assume that we want to maximize a linear function $d_k^T \dot{h}$ over the whole MRS induced by the given box of gimbals constraints $B$:

$$\text{dist}(B, d_k) = \max_{h \in \mathbb{R}^d} d_k^T \dot{h}.$$  \hspace{1cm} (9)

The maximization over vectors $\dot{h}$ can be easily replaced by the maximization over gimbals rates:

$$\text{dist}(B, d_k) = \max_{\delta \in B} d_k^T J(\delta) \dot{\delta} = \max_{\delta \in B} \sum_{i=1}^{m} d_k^T J_i \dot{\delta}^{(i)},$$  \hspace{1cm} (10)

and we can maximize this sum by maximizing each of the summands separately:

$$\text{dist}(B, d_k) = \sum_{i=1}^{m} d_k^T J_i \text{sign}(d_k^T J_i) \dot{\delta}^{(i)}_{\max}.$$  \hspace{1cm} (11)

Let us formally construct the indicator function $I_B(\dot{h})$, which is TRUE if vector $\dot{h}$ belongs to the MRS for the given box $B$ or FALSE otherwise. Then $I_B(\dot{h}) \equiv \text{TRUE}$ if and only if $\dot{h}$ satisfies the following system of linear inequalities:

$$d_k^T \dot{h} \leq \text{dist}(B, d_k), \quad k = 1, M$$

$$d_k^T \dot{h} \leq \text{dist}(B, -d_k), \quad k = 1, M$$  \hspace{1cm} (12)

It can be seen in Fig. 5 how this system defines the MRS. Here $d_i$ is a true normal vector to a facet, while $d_i$ is redundant for the representation (but its presence does not affect it).

Note that during bisection process the box $B$ become non-symmetric, nevertheless we can always compute its centre and apply the same procedure to the centered box.

To avoid the singularity problem (when it is possible), one can compute the solution for both 2D and 3D case and then choose one that gives the closest match with the supplied momentum rates vector. In the case of near-singular solution, all columns of the Jacobian matrix span a 2D plane. The normal $n$ to this plane can be obtained from the cross product of any two columns of $J(\delta)$. The required output $\dot{h}$ as well as all columns can be projected onto this plane using the pseudoinverse solution and the problem can be solved by running similar algorithm for 2D case. If $\dot{h}$ lies outside the plane, the solution gives us the closest vector in the least-squares sense.

Let us imagine that the normal vector $p$ is computed and we have chosen one column $d_k$. Then, to built orthogonal coordinate system in the plane, we can compute cross product $c$ of $p$ and $d_k$. Let us form the matrix $A = [c \ d_k]$. Now, the projection operator is given by the well-known Moore-Penrose formula:

$$P = (A^T A)^{-1} A^T,$$

and the projected output and Jacobian matrix in the plane coordinate system are given by $P \dot{h}$ and $P J(\delta)$.

5. NUMERICAL EXAMPLE

Consider a pyramid type CMG where four actuators are constrained to the gimbal on the faces of a pyramid (see Fig. 1). The Jacobian in this case become (Wie (2008)) as follows:

$$J(\delta) = \begin{bmatrix}
-\cos(\beta)\cos(\delta_1) & \sin(\delta_2) & \cos(\beta)\cos(\delta_3) & \ldots \\
-\sin(\delta_1) & -\cos(\beta)\cos(\delta_2) & \sin(\delta_3) & \ldots \\
\sin(\beta)\cos(\delta_1) & \sin(\beta)\cos(\delta_2) & \sin(\beta)\cos(\delta_3) & \ldots \\
-\sin(\delta_3) & \cos(\beta)\cos(\delta_4) & \sin(\beta)\cos(\delta_4) \\
\sin(\beta)\cos(\delta_4) & \end{bmatrix},$$

where $\beta$ is the skew angle.

An example with skew angle of 53.13 deg and constant unit momentum magnitude for each CMG are taken from Lee et al. (2007). Initial gimbal angles are given by $\delta = [90^\circ \ 0 \ -90^\circ \ 0]^T$. In this case, the rank of the Jacobian matrix is two and the singularity problem arises. The Jacobian matrix is as follows:

$$J(\delta) = \begin{bmatrix}
0 & 0 & 0 & 0 \\
-1 & -0.6 & -1 & 0.6 \\
0 & 0.8 & 0 & 0.8 \\
\end{bmatrix}.$$  

The singular layout of CMG cannot produce any momentum along the X-axis direction. The required output is assumed as $\dot{h} = [0 \ 1 \ 0 \ 0]^T$ and $|\dot{\delta}^{(i)}_{\max}| \leq 1$ for all $i$.

Due to the singularity, the solution computed by 3D algorithm gives us quite different output from the required one. The solution $\dot{\delta}$ for 2D problem $P \dot{h} = P J(\delta) \dot{\delta}$ is defined in the plane with the normal vector $p = [1 \ 0 \ 0]^T$ and

$$P \dot{h} = [1 \ 0], \quad P J(\delta) = \begin{bmatrix}
-1 & -0.6 & -1 & 0.6 \\
0 & 0.8 & 0 & 0.8 \\
\end{bmatrix}.$$  

After 32 bisections of the initial control box, we have obtained the following enclosing box for the final solution:
which is consistent with the result obtained in Lee et al. (2007) by a different method.

In Fig. 6 the momentum rates sets for $B_1$ and $B_2$ boxes (see Fig. 4) are shown for the beginning of bisection iterations. By solid lines we depict a projections of the box that has been chosen by the algorithm at this iteration, while dashed lines represent the one for the box that has been deleted. The cross inside the circle represents the commanded vector $P_\delta$. One can notice that two momentum rates sets overlap each other, representing projections of two adjacent boxes.

![Fig. 6. Bisection iterations](image)

6. CONCLUSION

A novel control allocation algorithm is presented for CMG moment distribution. It has a guarantee of obtaining the solution for every given momentum rates vector, Jacobian matrix and the set of control constraints in a finite and known in advance number of iterations with required numerical accuracy. The proposed method possess the property of utilizing the whole attainable momentum rates set (which can increase the agility) and has relatively low algorithmic complexity. These properties may allow to consider the algorithm for implementation in modern small agile satellites.

7. ACKNOWLEDGMENT

First author expresses his sincerest gratitude to Dr Christophe Louembet, who noticed the applicability of the author’s research to CMG control problem during 17th IFAC Symposium on Automatic Control in Aerospace held in Toulouse.

REFERENCES


