Autonomous Vehicle Simulation (AVS) Laboratory

Basilisk Technical Memorandum

Document ID: Basilisk-ThrForceMapping

ALGORITHMS TO MAP DESIRED TORQUE VECTOR ONTO A SET OF THRUSTERS

Prepared by	H. Schaub

Status: Ready
Scope/Contents
This module takes a commanded attitude control torque vector and determines a set of desired thruster force values to implement this torque. It is assumed that the nominal thruster configuration is such that pure torque solutions are possible. The module supports both on- and off-pulsing solutions, including cases where the thruster solutions are saturated due to a large commanded attitude control torque.

Rev	Change Description	By	Date
1.0	Initial Documentation	H. Schaub	2019-02-09
1.1	Updated the thruster force evaluation to account for center of mass offsets	H. Schaub	

1.2	Updated the figure and the $[C]$ matrix notation	H. Schaub	
1.3	The thruster mapping logic has changed, and this documentation now reflects what the new algorithm does.	H. Schaub	
1.4	Discuss the new module feature of scaling the thruster force solution during periods where the thrusters are saturated.	H. Schaub	
2.0	Update document to adhere to new documentation format	J. Martin	2019-04
2.1	Update documentation after change to module to allow 3-axis control with DV thrusters	J. Hammerl	2024-01

Contents

---

1 Module Description

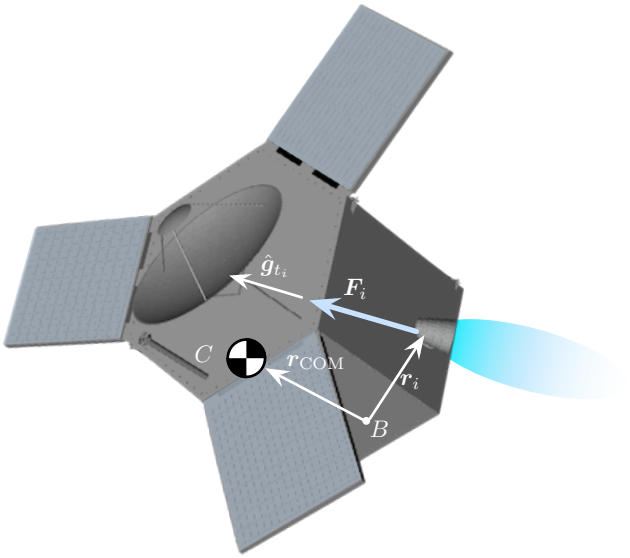

Fig. 1: Illustration of the Spacecraft Thruster Notation

1.1 Introduction

1.1.1 Torque Control Axes

This technical note describes a general algorithm that maps a desired ADCS external control torque $\boldsymbol{L}_r$ onto force commands for a cluster of thrusters. Let $\hat{\boldsymbol{c}}_j$ be the axis about which the thrusters are to produce the desired torque. The matrix of N_c thruster axes rows is then given by

$$[C] = \begin{bmatrix} \hat{\boldsymbol{c}}_1 \\ \vdots \\ \hat{\boldsymbol{c}}_{N_c} \end{bmatrix} \tag{1}$$

The module can accept up to 3 orthogonal control axis $\hat{\boldsymbol{c}}_j$. Let $\bar{\boldsymbol{L}}_r$ be the three-dimensional reduced set of $\boldsymbol{L}_r$ onto the set of control axes written in the body-frame, given by:

$${}^{\mathcal{B}}\bar{\boldsymbol{L}}_r = [C]^T[C]{}^{\mathcal{B}}\boldsymbol{L}_r \tag{2}$$

The goal of the thruster mapping strategy is to find a set of $\boldsymbol{F}$ thruster forces that yield $\bar{\boldsymbol{L}}_r$.

1.1.2 Thruster to Torque Mapping Matrix

The i^{th} thruster location relative to the spacecraft point B is given by $\boldsymbol{r}_i$ as illustrated in Figure 1. The unit direction vector of the thruster force is $\hat{\boldsymbol{g}}_{t_i}$, while the thruster force is given by

$$\boldsymbol{F}_i = F_i\hat{\boldsymbol{g}}_{t_i} \tag{3}$$

The toque vector produced by each thruster about the body fixed point C is thus

$$\boldsymbol{\tau}_i = (\boldsymbol{r}_i - \boldsymbol{r}_{\text{COM}}) \times F_i\hat{\boldsymbol{g}}_{t_i} \tag{4}$$

The total torque onto the spacecraft, due to a cluster of N thrusters, is

$$\tau_j = \sum_{i=1}^{N} \boldsymbol{\tau}_i = \sum_{i=1}^{N}((\boldsymbol{r}_i - \boldsymbol{r}_{\text{COM}}) \times \hat{\boldsymbol{g}}_{t_i})F_i = \sum_{i=1}^{N} \boldsymbol{d}_i F_i \tag{5}$$

where

$$\boldsymbol{d}_i = (\boldsymbol{r}_i - \boldsymbol{r}_{\text{COM}}) \times \hat{\boldsymbol{g}}_{t_i} \tag{6}$$

In matrix form, the net spacecraft torque is written compactly as

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{d}_1 \cdots \boldsymbol{d}_N \end{bmatrix} \begin{bmatrix} F_1 \\ \vdots \\ F_N \end{bmatrix} = [D]\boldsymbol{F} \tag{7}$$

where $[D]$ is a $3 \times N$ matrix that maps the thruster forces F_i to the spacecraft torque $\boldsymbol{\tau}$.

1.2 ACS Thruster Force Algorithm for a Thruster Configuration with Full Torque Controllability

Here a thruster configuration is assumed that can produce pure torque-couples without exerting a net force onto the spacecraft. The thrusters force values F_i must be strictly non-negative (i.e. either 0 or positive). Note that in this configuration having all thrusters on will produce zero net force and torque onto the spacecraft. Thus the $F_i = F_j$ solution is in the nullspace of the mapping in Eq. (7).

The goal of the thruster force algorithm is to determine a set of thruster forces $\boldsymbol{F}$ such that the net torque $\boldsymbol{\tau}$ onto the spacecraft is

$$^{\mathcal{B}}\boldsymbol{\tau} = {}^{\mathcal{B}}\bar{\boldsymbol{L}}_r = [D]^{\mathcal{B}}\boldsymbol{F} \tag{8}$$

without bleeding torque onto the un-controlled axes. The first step is to perform a standard minimum norm inverse solution using

$$^{\mathcal{B}}\boldsymbol{F} = [D]^T([D][D]^T)^{-1\mathcal{B}}\bar{\boldsymbol{L}}_r \tag{9}$$

The 3×3 matrix $[D][D]^T$ is full rank and thus invertible with the assumption that this RCS configuration has a full 3D torque controllability. This set of thruster forces will contain F_i values that are both positive and negative. Next, to achieve strictly non-negative values, the minimum F_i value is determined and subtracted from all N force values.

$$\boldsymbol{F} \leftarrow \boldsymbol{F} - \min(\boldsymbol{F}) \tag{10}$$

The resulting set of F_i forces will produce the desired control torque $\bar{\boldsymbol{L}}_r$ and achieve a net zero force onto the spacecraft. The latter results is due to the assumption off an ACS thruster configuration that can produce pure moment couples.

1.2.1 ACS Thruster Force Algorithm for a Thruster Configuration with Partial Torque Controllability

In the case a control frame is not fully defined (i.e. only 1 or 2 control axes are explicitly specified), the math above needs to be slightly modified to compute the torques in the desired control space. Specifically, the mapping matrix $[D]$ and requested torque vector $\mathbf{L_r}$ must be projected into, and operated within, the control subspace. As such,

$$[C] = \begin{bmatrix} \hat{\boldsymbol{c}}_1 \\ \vdots \\ \hat{\mathbf{0}}_{N_c} \end{bmatrix} \tag{11}$$

$$^{\mathcal{C}}\bar{\boldsymbol{L}}_r = [C]^{\mathcal{B}}\boldsymbol{L}_r \tag{12}$$

$$[CD] = [C][D] \tag{13}$$

$$^{\mathcal{B}}\boldsymbol{F} = [CD]^T([CD][CD]^T)^{-1\mathcal{C}}\bar{\boldsymbol{L}}_r \tag{14}$$

Note: $([CD][CD]^T)^{-1}$ is not an invertable matrix if one or more of the control axes $\hat{\boldsymbol{c}}$ is a $\mathbf{0}$ vector. To circumvent this problem, two mathematical reconciliations must be made. First, when $\boldsymbol{L}_r$ is projected onto the control axes, $[C]^{\mathcal{B}}\boldsymbol{L}_r$, the component projected onto the $\mathbf{0}$ axis will be removed. I.e. if

$$[C] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$^{\mathcal{B}}\boldsymbol{L}_r = (1.0, -0.5, 0.7)^T$$

then

$$^{\mathcal{C}}\boldsymbol{L}_r = (1.0, 0.7, 0.0)^T$$

and

$$[CD] = \begin{pmatrix} CD_{11} & CD_{12} & \dots & CD_{1N} \\ CD_{21} & CD_{22} & \dots & CD_{2N} \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

Second, the $[M] = ([CD][CD]^T)$ matrix must be set to identity, and only the upper block entries in $[M]$ will be computed (2x2 or 1x1 depending on 2 or 1 control axes respectively) . So for the 2-axis control scheme,

$$[M] = \begin{pmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Allowing $[M]$ to be inverted. This assertion can be made because when mapping the control torques onto the control axes, the $[M]_{33} = 1$ component is multiplied by the zero in ${}^{\mathcal{C}}\boldsymbol{L}_r$. Thus:

$${}_B \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{pmatrix} = \begin{pmatrix} CD_{11} & CD_{21} & 0 \\ CD_{12} & CD_{22} & 0 \\ \vdots & \vdots & \vdots \\ CD_{1N} & CD_{2N} & 0 \end{pmatrix} \begin{pmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} L_1 \\ L_2 \\ 0 \end{pmatrix} \tag{15}$$

such that ${}^{\mathcal{C}}L_3$ will never be mapped onto the body frame force components ${}^{\mathcal{B}}F$.

1.3 2-Stage Minimum Norm ACS Thruster Mapping Algorithm

To increase the robustness of the sign-constrained minimum norm thruster force solution 2nd stage is included. This is helpful in configurations where the number of available ACS thrusters is not equal to the number of installed thrusters. This simulates scenarios where some thrusters are now offline. To engage this 2nd loop the module flag `use2ndLoop` must be set to 1. The default value is zero where the 2nd loop is not engaged. The minimum norm solution from the earlier solution is first evaluated, and then shifted by subtracting the minimum F_i value.

Each thruster can only produce a positive force. With off-pulsing, the nominal thrust force plus the negative correction must still yield a non-negative thrust force. The module parameter `thrForceSign` is either $+1$ or -1 to account for the desired force sign. The value of this parameter is represented through s_F. With the ACS thruster configuration this value would always be $+1$.

We assume that $\boldsymbol{F}$ elements only contain forces that are either zero or values with the desired sign. Assume there are M force values in $\boldsymbol{F}$ with a sign that matches s_F. The locations of these values is provided in the N-dimensional array $\boldsymbol{t}_{\text{used}}$ which contains either 0 or 1 values. For example, consider $N = 8$ and only thrusters 2 and 6 produce zero forces. In this case we find

$$\boldsymbol{t}_{\text{used}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{16}$$

This reduces the thruster force search to a subset of M thrusters. Let $\bar{\boldsymbol{F}}_j$ be a $M \times 1$ matrix of to be determined thruster forces. The corresponding $3 \times M$ mapping matrix $[\bar{D}]$ that projects $\bar{\boldsymbol{F}}$ onto a net body torque about point B is defined as:

$$[\bar{D}] = \begin{bmatrix} \bar{\boldsymbol{d}}_1 & \cdots & \bar{\boldsymbol{d}}_M \end{bmatrix} \tag{17}$$

with

$$\bar{\boldsymbol{d}}_i = (\boldsymbol{r}_i - \boldsymbol{r}_{\text{COM}}) \times \hat{\boldsymbol{g}}_i \tag{18}$$

The net torque due to $\bar{\boldsymbol{F}}$ is

$${}^{\mathcal{B}}\bar{\boldsymbol{\tau}} = [\bar{D}]^{\mathcal{B}}\bar{\boldsymbol{F}} \tag{19}$$

A modified set of thruster force solutions $\bar{\boldsymbol{F}}$ to generate the desired torque $\bar{\boldsymbol{L}}_r$ is found through a second minimum norm operation:

$$^{\mathcal{B}}\bar{\boldsymbol{F}} = [\bar{D}]^T([\bar{D}][\bar{D}]^T)^{-1\mathcal{B}}\bar{\boldsymbol{L}}_r \tag{20}$$

The next step is to sum the individual $\bar{\boldsymbol{F}}$ thruster solutions to the yield the net set of thruster forces required to produce $\bar{\boldsymbol{L}}_r$. This is done using the $\boldsymbol{t}_{\text{used}}$ matrix to determine which thrusters have non-zero contributions. The final step is to evaluate the minimum F_i force again and subtract this from all thruster force values.

1.4 DV Thruster Firing Strategy

With the DV configuration the attitude is controlled via off-pulsing. As such, the thruster firing mapping must produce negative F_i values and the `thrForceSign` sign must be set to -1.

In the case where the thruster force axes of the DV Thruster configuration are parallel, no torque along the DV thrust axis can be produced. With such a DV configuration the only attitude along the axes orthogonal to the thrust vector is controlled via off-pulsing. In this case `thrForceSign` is -1 and the number of control axes is smaller than 3, and the second loop is always engaged.

The modified algorithm still evaluates the first minimum norm inverse, but does not subtract out the min($\boldsymbol{F}$) value. Rather, the 2nd stage is used to determine which DV thrusters produce the torque with a negative torque value to generate the corresponding $[\bar{D}]$ matrix. After performing the 2nd minimum norm inverse with $[\bar{D}]$ the subtraction of min($\boldsymbol{F}$) is not performed.

1.5 Saturating the Thruster Capability

The thruster force solution $\boldsymbol{F}$ is implemented using a pulse-width modulation where the thruster is fired only partially with the maximum thruster force F_{max} during the control period such that the net impulse achieved is equivalent as if a smaller thruster force F_i had been commanded. This section discusses the case where $F_i > F_{\text{max}}$ and the thruster is not able to achieve the desire control impulse due to be saturated.

Ideally the thrusters are not saturated, and the thruster force solution $\boldsymbol{F}$ yields a torque

$$[D]\boldsymbol{F} \rightarrow \boldsymbol{\tau}$$

that is equal to the desired reduced control torque vector $\bar{\boldsymbol{L}}_r$. However, if the thrusters are saturated, then the net torque $\boldsymbol{\tau}$ is not only different in the magnitude, but also the direction. Applying a control torque that has a significant direction error can lead to unstable behavior. Let $\theta_{\boldsymbol{\tau}/\bar{\boldsymbol{L}}_r}$ be the angle between the actual torque $\boldsymbol{\tau}$ and the desired torque $\bar{\boldsymbol{L}}_r$.

$$\theta_{\boldsymbol{\tau}/\bar{\boldsymbol{L}}_r} = \arccos\left(\frac{\boldsymbol{\tau}\cdot\bar{\boldsymbol{L}}_r}{|\boldsymbol{\tau}||\bar{\boldsymbol{L}}_r|}\right) \tag{21}$$

If a 12-thruster configuration is used where each thruster only produces a torque about a single axis, then this saturation issue might not be as concerning. However, if a reduced thruster solution is used, such as the common 8-thruster solution used in section **??**, then some thrusters must do double-duty and support multiple control axes. Being saturated can quickly yield an actual net thruster torque $\boldsymbol{\tau}$ that has a significantly different heading then $\bar{\boldsymbol{L}}_r$. In this case the thruster solution $\boldsymbol{F}$ is scaled such that no $|F_i|$ value is larger then F_{max}. This will reduce the $\boldsymbol{\tau}$ magnitude, but the direction will align with $\bar{\boldsymbol{L}}_r$. As a result the closed-loop performance tends to respond more slowly, but in a more stable manner during period of thruster saturation. If $|F_i|$ is used, the above algorithm works for both on- and off-pulsing solutions.

Then angular threshold beyond which this thruster force $\boldsymbol{F}$ scaling is implemented is set through the `thrForceMapping` module parameter `angErrThresh`. If

$$\theta_{\boldsymbol{\tau}/\bar{\boldsymbol{L}}_r} > \texttt{angErrThresh}$$

then the thruster force solution is scaled such that $|F_i| \leqslant F_{\text{max}}$.

The default value of `angErrThresh` is 0°. This means that this scaling during super-saturated thruster solutions is on by default. If a control torque miss-alignment of $10^\circ$is acceptable, then set $\texttt{angErrThresh} = 10^\circ$.

To turn off this thruster force scaling during super-saturated conditions, the parameter `angErrThresh` is set to a value larger than 180°. As the angle $\theta_{\boldsymbol{\tau}/\bar{\boldsymbol{L}}_r} \leqslant 180^\circ$, this ensures that the above threshold condition is never reached and the thruster forces are not scaled.

2 Module Functions

This module has the following functions:

- **Evaluate RW null projection matrix** $[\tau]$: When reset the module will pull in the current RW configuration data and create the null motion projection matrix. This matrix remains fixed unit the module is reset again.
- **Compute a RW deceleration torque**: With each update call the module computes a decelerating RW torque solution that lies in the null space of the RW array.
- **Output a net RW motor torque solution**: The module combined the feedback control torque and the null space torque to slow down the RW speeds and outputs a net solution solution.

3 Module Assumptions and Limitations

The module assumes all RW devices are operating and available. It also assumes the RW spin axes don't change during the regular update cycles.

4 User Guide

1. ϵ **Parameter**: The minimum norm inverse requires a non-zero determinant value of $[D][D]^T$. For this setup, this matrix is a scalar value

$$D_2 = \det([D][D]^T) \tag{22}$$

If this D_2 value is near zero, then the full 3D $\bar{\boldsymbol{L}}_r$ vector cannot be achieved. A common example of such a scenario is with the DV thruster configuration where all $\hat{\boldsymbol{g}}_{t_i}$ axes are collinear. Torques about these thrust axes cannot be produced. In this case, the minimum norm solution is adjusted to only match the torques along the control matrix $[C]$ sub-space. Torques being applied outside of $\hat{\boldsymbol{c}}_j$ is not possible as D_2 is essentially zero, indicating the other control axis cannot be controlled with this thruster configuration.

The minimum norm torque solution is now modified to use

$$\bar{\boldsymbol{F}} = ([C][\bar{D}])^T([C][\bar{D}][\bar{D}]^T[C]^T)^{-1}[C]\boldsymbol{L}_r \tag{23}$$

As the thruster configuration cannot produce a general 3D torque, here the $[C]$ matrix must have either 1 or 2 control axes that are achievable with the given thruster configuration.

To set this epsilon parameter, not the definition of the $[D]$ matrix components $\boldsymbol{d}_i = (\boldsymbol{r}_i \times \hat{\boldsymbol{g}}_{t_i})$. Note that $\boldsymbol{r}_i \times \hat{\boldsymbol{g}}_{t_i}$ is a scaled axis along which the i^{th} thruster can produce a torque. The value $\boldsymbol{d}_i$ will be near zero if the dot product of this axis with the current control axis $\hat{\boldsymbol{c}}_j$ is small.

To determine an appropriate ϵ value, let α be the minimum desired angle to avoid the control axis $\hat{\boldsymbol{c}}_j$ and the scaled thruster torque axis $\boldsymbol{r}_i \times \hat{\boldsymbol{g}}_{t_i}$ being orthogonal. If $\bar{r}$ is a mean distance of the thrusters to the spacecraft center of mass, then the d_i values must satisfy

$$\frac{d_i}{\bar{r}} > \cos(90^\circ - \alpha) = \sin\alpha \tag{24}$$

Thus, to estimate a good value of ϵ, the following formula can be used

$$\epsilon \approx d_i^2 = \sin^2\alpha \; \bar{r}^2 \tag{25}$$

For example, if $\bar{r} = 1.3$ meters, and we want α to be at least 1°, then we would set $\epsilon = 0.000515$.

2. $[C]$ **matrix**: The module requires control control axis matrix $[C]$ to be defined. Up to 3 orthogonal control axes can be selected. Let N_c be the number of control axes. The $N_c \times 3$ $[C]$ matrix is then defined as

$$[C] = \begin{bmatrix} \hat{\boldsymbol{c}}_1 \\ \vdots \end{bmatrix} \tag{26}$$

Not that in python the matrix is given in a 1D form by defining `controlAxes_B`. Thus, the $\hat{\boldsymbol{c}}_j$ axes are concatenated to produce the input matrix $[C]$.

3. `thrForceSign Parameter`: Before this module can be run, the parameter `thrForceSign` must be set to either $+1$ (on-pulsing with the ACS configuration) or -1 (off-pulsing with the DV configuration).

4. `use2ndLoop` Flag: If the `thrForceSign` flag is set to $+1$ then an on-pulsing configuration is setup. By default the optional flag `use2ndLoop` is 0 and the algorithm only uses the least-squares fitting loop once. By setting the `use2ndLoop` to $+1$ then the 2nd least squares fitting loop is used during this on-pulsing configuration,.

5. `angErrThresh` **Parameter**: The default value of `angErrThresh` is 0°. This means that during periods of thruster saturation the thruster force solution $\boldsymbol{F}$ is scaled such that $|F_i| \leqslant F_{\text{max}}$. If this scaling should only be done if $\boldsymbol{\tau}$ and $\bar{\boldsymbol{L}}_r$ differ by an angle α, then set `angErrThresh` equal to α. To turn off this force scaling during thruster saturation the parameter `angErrThresh` should be set to a value larger than 180°.