



## Autonomous Vehicle Simulation (AVS) Laboratory

### Basilisk Technical Memorandum

Document ID: Basilisk-ThrusterForces

#### ALGORITHMS TO MAP DESIRED TORQUE VECTOR ONTO A SET OF THRUSTERS

Prepared by	H. Schaub
-------------	-----------

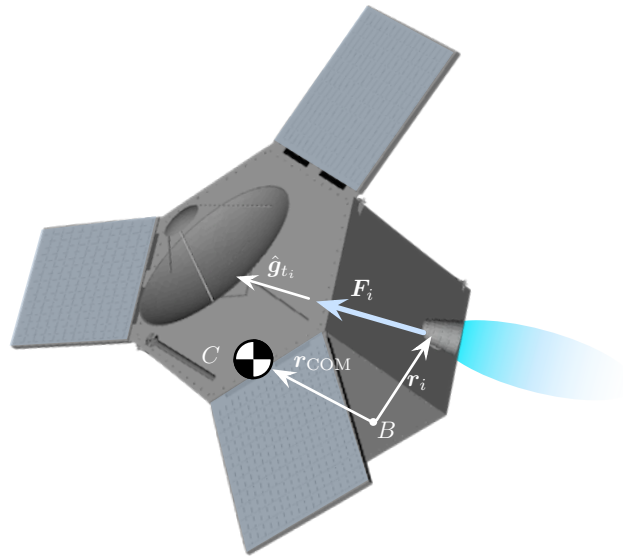
<b>Status:</b> Ready
<b>Scope/Contents</b>
Include a short summary of what this system engineering report is about. Should be 300 words or less.

Rev:	Change Description	By
v0.1	Updated the thruster force evaluation to account for center of mass offsets	H. Schaub
v0.2	Updated the figure and the $[C]$ matrix notation	H. Schaub
v0.3	The thruster mapping logic has changed, and this documentation now reflects what the new algorithm does.	H. Schaub

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Torque Control Axes . . . . .	2
1.2	Thruster to Torque Mapping Matrix . . . . .	2
<b>2</b>	<b>ACS Thruster Force Algorithm for a Thruster Configuration with Full Torque Controllability</b>	<b>2</b>
<b>3</b>	<b>2-Stage Minimum Norm ACS Thruster Mapping Algorithm</b>	<b>3</b>
<b>4</b>	<b>DV Thruster Firing Strategy</b>	<b>4</b>
<b>5</b>	<b>Performance Illustration</b>	<b>4</b>
5.1	ACS Thruster Configuration . . . . .	4
5.2	DV Thruster Configuration . . . . .	6
<b>6</b>	<b>Module Parameters</b>	<b>8</b>
6.1	$\epsilon$ Parameter . . . . .	8
6.2	$[C]$ matrix . . . . .	8
6.3	thrForceSign Parameter . . . . .	8

---



**Fig. 1:** Illustration of the Spacecraft Thruster Notation

# 1 Introduction

## 1.1 Torque Control Axes

This technical note describes a general algorithm that maps a desired ADCS external control torque  $\mathbf{L}_r$  onto force commands for a cluster of thrusters. Let  $\hat{\mathbf{c}}_j$  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

$$[C] = \begin{bmatrix} \hat{\mathbf{c}}_1 \\ \vdots \\ \hat{\mathbf{c}}_{N_c} \end{bmatrix} \quad (1)$$

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

$$\bar{\mathbf{L}}_r = [C]^T [C] \mathbf{L}_r \quad (2)$$

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

## 1.2 Thruster to Torque Mapping Matrix

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

$$\mathbf{F}_i = F_i \hat{\mathbf{g}}_{t_i} \quad (3)$$

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

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

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

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

where

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

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

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

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

## 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 torque 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  $\mathbf{F}$  such that the net force onto the spacecraft is

$$\boldsymbol{\tau} = \bar{\mathbf{L}}_r = [\mathbf{D}]\mathbf{F} \quad (8)$$

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

$$\mathbf{F} = [\mathbf{D}]^T([\mathbf{D}][\mathbf{D}]^T)^{-1}\bar{\mathbf{L}}_r \quad (9)$$

The  $3 \times 3$  matrix  $[\mathbf{D}][\mathbf{D}]^T$  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.

$$\mathbf{F} \leftarrow \mathbf{F} - \min(\mathbf{F}) \quad (10)$$

The resulting set of  $F_i$  forces will produce the desired control torque  $\bar{\mathbf{L}}_r$  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.

### 3 2-Stage Minimum Norm ACS Thruster Mapping Algorithm

To increase the robustness of the sign-constrained minimum norm thruster force solution, as 2nd stage is included if the number of available ACS thrusters is not equal to the number of installed thrusters. This simulates scenarios where some thrusters are now offline. 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  $\mathbf{F}$  elements only contain forces that are either zero or values with the desired sign. Assume there are  $M$  force values in  $\mathbf{F}$  with a sign that matches  $s_F$ . The locations of these values is provided in the  $N$ -dimensional array  $\mathbf{t}_{\text{used}}$  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

$$\mathbf{t}_{\text{used}} = [1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1] \quad (11)$$

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

$$[\bar{\mathbf{D}}] = [\bar{\mathbf{d}}_1 \ \cdots \ \bar{\mathbf{d}}_M] \quad (12)$$

with

$$\bar{\mathbf{d}}_i = (\mathbf{r}_i - \mathbf{r}_{\text{COM}}) \times \hat{\mathbf{g}}_i \quad (13)$$

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

$$\bar{\boldsymbol{\tau}} = [\bar{\mathbf{D}}]\bar{\mathbf{F}} \quad (14)$$

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

$$\bar{\mathbf{F}} = [\bar{\mathbf{D}}]^T([\bar{\mathbf{D}}][\bar{\mathbf{D}}]^T)^{-1}\bar{\mathbf{L}}_r \quad (15)$$

The next step is to sum the individual  $\bar{\mathbf{F}}$  thruster solutions to the yield the net set of thruster forces required to produce  $\bar{\mathbf{L}}_r$ . This is done using the  $\mathbf{t}_{\text{used}}$  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.

## 4 DV Thruster Firing Strategy

With a DV thruster configuration the thruster force axes are parallel. As such, this configuration cannot produce a torque along the DV thrust axis. A slightly modified version of the above algorithm is used in this case. With the DV configuration the attitude along the axes orthogonal to the thrust vector are 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.

The modified algorithm still evaluates the first minimum norm inverse, but does not subtract out the  $\min(\mathbf{F})$  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  $\min(\mathbf{F})$  is not performed.

## 5 Performance Illustration

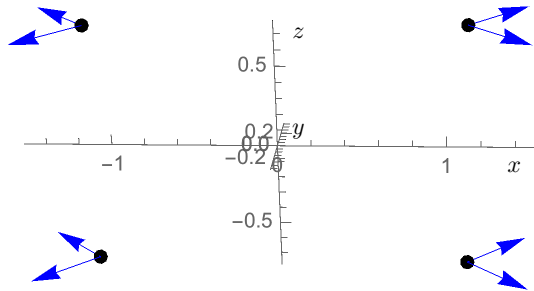
### 5.1 ACS Thruster Configuration

To illustrate the performance of this algorithm, the following simulation setup is used. Let the ACS system have a total of  $N = 8$  thrusters with the following body-fixed locations:

$$\begin{aligned} \mathbf{r}_1 &= [+1.125, 0.0, +0.75]^T \text{ m} & \mathbf{r}_2 &= [-1.125, 0.0, +0.75]^T \text{ m} \\ \mathbf{r}_3 &= [-1.125, 0.0, +0.75]^T \text{ m} & \mathbf{r}_4 &= [+1.125, 0.0, +0.75]^T \text{ m} \\ \mathbf{r}_5 &= [+1.125, 0.0, -0.75]^T \text{ m} & \mathbf{r}_6 &= [-1.125, 0.0, -0.75]^T \text{ m} \\ \mathbf{r}_7 &= [-1.125, 0.0, -0.75]^T \text{ m} & \mathbf{r}_8 &= [+1.125, 0.0, -0.75]^T \text{ m} \end{aligned}$$

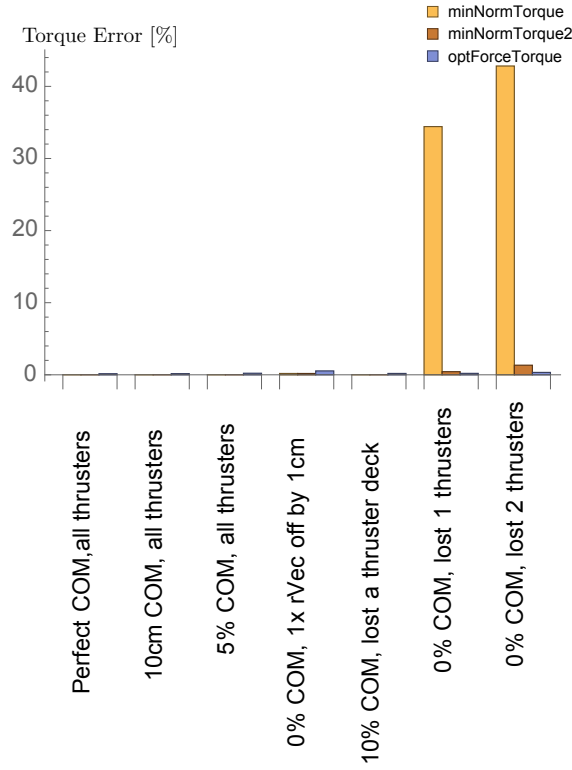
with the force unit direction vectors:

$$\begin{aligned} \mathbf{g}_{t_1} &= [+0.707107, +0.707107, 0]^T & \mathbf{g}_{t_2} &= [-0.707107, +0.707107, 0]^T \\ \mathbf{g}_{t_3} &= [-0.707107, -0.707107, 0]^T & \mathbf{g}_{t_4} &= [+0.707107, -0.707107, 0]^T \\ \mathbf{g}_{t_5} &= [+0.707107, +0.707107, 0]^T & \mathbf{g}_{t_6} &= [-0.707107, +0.707107, 0]^T \\ \mathbf{g}_{t_7} &= [-0.707107, -0.707107, 0]^T & \mathbf{g}_{t_8} &= [+0.707107, -0.707107, 0]^T \end{aligned}$$

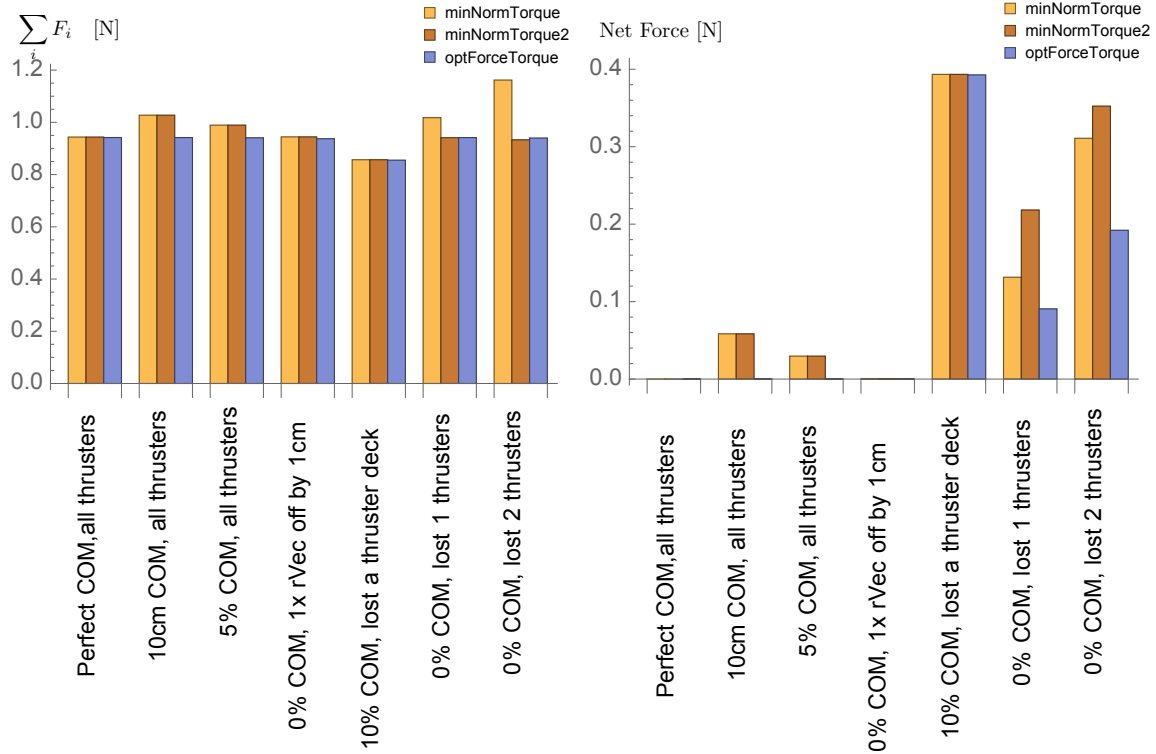


**Fig. 2:** Illustration of an 8-thruster ACS configuration

The resulting configuration is illustrated in Figure 2. In this setup the center of mass position vector is set to  $\mathbf{r}_{\text{COM}} = (0, 0, 0)^T \text{ m}$ . The following plots show the thruster firing performance by considering a rang of scenarios. The first case assumes all thrusters are available, and the  $\mathbf{r}_{\text{COM}}$  vector is known perfectly. The second and third case also have all thrusters available, but the  $\mathbf{r}_{\text{COM}}$  vector knowledge is



(a) Percent Torque Error



(b) Thruster Control Implementation Effort

(c) Net Thruster Disturbance Force

**Fig. 3:** ACS Thruster Mapping Performance Mapping Illustration Comparing the 1-stage and 2-stage minimum norm solution to the thruster mapping of a nonlinear optimization solution.

off by 10cm or 5cm in the  $z$  direction. The 4th case has the perfect  $r_{\text{COM}}$  vector, but the first thruster location is off by 1cm in the  $y$  direction. The 5th case again assumes a 10cm COM offset, but also that only 4/8 thrusters are available. In essence, all the lower or upper thruster have become un-available. The 6th and 7th case assumes perfect COM knowledge, but either the last one or last two thrusters are un-available.

The resulting performance is illustrated in Figure 3. Here 20 random  $L_r$  torques are generated, and the mean results are shown. The  $[C]$  matrix is set to the identity matrix to control all three axes. The 1-stage minimum norm algorithm is compared to an optimal thruster firing solution which minimizes the net thruster forces used, the net disturbance torque onto the craft, all subject to producing the desired control torque. Regarding generating the desired control torque, only when 1 or 2 thrusters were lost did the algorithm have issues generating the required torque. In all other cases the desired torque was always produced as shown in Figure 3(a). In comparison, the 2-stage minimum norm algorithm performs very well in these latter cases where 1-2 thrusters are lost, yielding only very small torque errors on average.

The control effort required to achieve this torque is shown in Figure 3(b). Here the  $F_i$  thruster force values are simply summed to have a sense of how much on-time or fuel is required for a given scenario. If all ACS thrusters are operating and the COM is perfectly modeled is the control effort equivalent to the optimal solution. In all other case the effort is about 10-20% larger. in contrast, the 2-stage minimum norm algorithm control efforts in the cases with lost thrusters is very similar to those of the optimal answers. This illustrates the robustness achieved by adding this 2nd stage if some thrusters are lost.

Finally, to see how well this algorithm is able to avoid net disturbance forces onto the spacecraft, the results in Figure 3(c) are shown. Overall the 1-stage algorithm did well, producing 0 net force in the ideal case and the case where the 1st thruster location is different, and matches the optimal answer for the lost thruster deck and lost single thruster case. In the other cases there is some net disturbance force. The 2-stage algorithm only differs here when thrusters are lost. Here the net disturbance torque is higher than the 1-stage solution. However, this is an expected result as the 1-stage solution produces huge torque errors, while the 2-stage solution produces a very different thruster solution which yields very small torque errors.

Overall the 2-stage thruster firing performs very well in comparison to the the optimal thruster firing solutions. The optimal solutions are computationally very expensive and not suitable to realtime flight-software implementations.

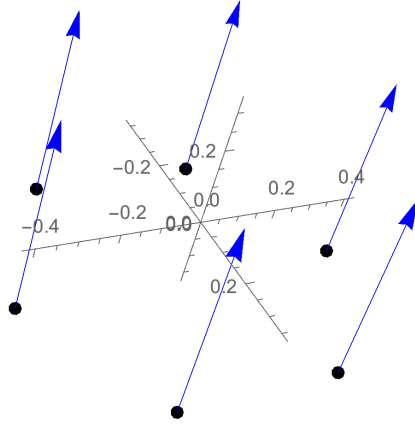
## 5.2 DV Thruster Configuration

Next consider a DV thruster configuration where the off-thruster-axis control torques are produces through off-pulsing. Let the DV system have a total of  $N = 6$  thrusters with the following body-fixed locations:

$$\begin{aligned} r_1 &= [0, 0.413, -0.1671]^T \text{ m} & r_2 &= [0.357668, 0.2065, -0.1671]^T \text{ m} \\ r_3 &= [0.357668, -0.2065, -0.1671]^T \text{ m} & r_4 &= [0, -0.413, -0.1671]^T \text{ m} \\ r_5 &= [-0.357668, -0.2065, -0.1671]^T \text{ m} & r_6 &= [-0.357668, 0.2065, -0.1671]^T \text{ m} \end{aligned}$$

with the force unit direction vectors given by  $\hat{g}_{t_i} = (0, 0, 1)^T$  as illustrated in Figure 4.

Figure 5 illustrates the DV off-pulsing performance for 3 scenarios. The 1st one has all thrusters available and no COM error. The 2nd case adds 2cm COM offsets to the  $x$  and  $y$  axes, while the 3rd case has no COM error but lost an opposing set of thrusters. Again 20 random  $L_r$  vectors are generated and the mean performance evaluated. As only torques about the  $x$  and  $y$  axis can be controlled with



**Fig. 4:** Illustration of an 6-thruster DV configuration

this DV thruster configuration, the control axis matrix is set to

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

The torque implementation errors are compared in Figure 5(a). The 2-stage process outlined above is able to produce the required torques in all cases, illustrating good robustness to COM offsets and losing 2/6 thrusters. Figure 5(b) illustrates the net off-pulsing effort required to achieve these random control torque vectors. The 2-stage algorithm requires slightly more off-pulsing than the optimal solution. However, the difference isn't very large, especially in comparison to the drastically faster computational evaluation time.

## 6 Module Parameters

### 6.1 $\epsilon$ Parameter

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

$$D_2 = \det([D][D]^T) \quad (16)$$

If this  $D_2$  value is near zero, then the full 3D  $\bar{\mathbf{L}}_r$  vector cannot be achieved. A common example of such a scenario is with the DV thruster configuration where all  $\hat{\mathbf{g}}_{t_i}$  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  $\hat{\mathbf{c}}_j$  is not possible as  $D_2$  is essentially zero, indicating the other control axis cannot be controlled with this thruster configuration.

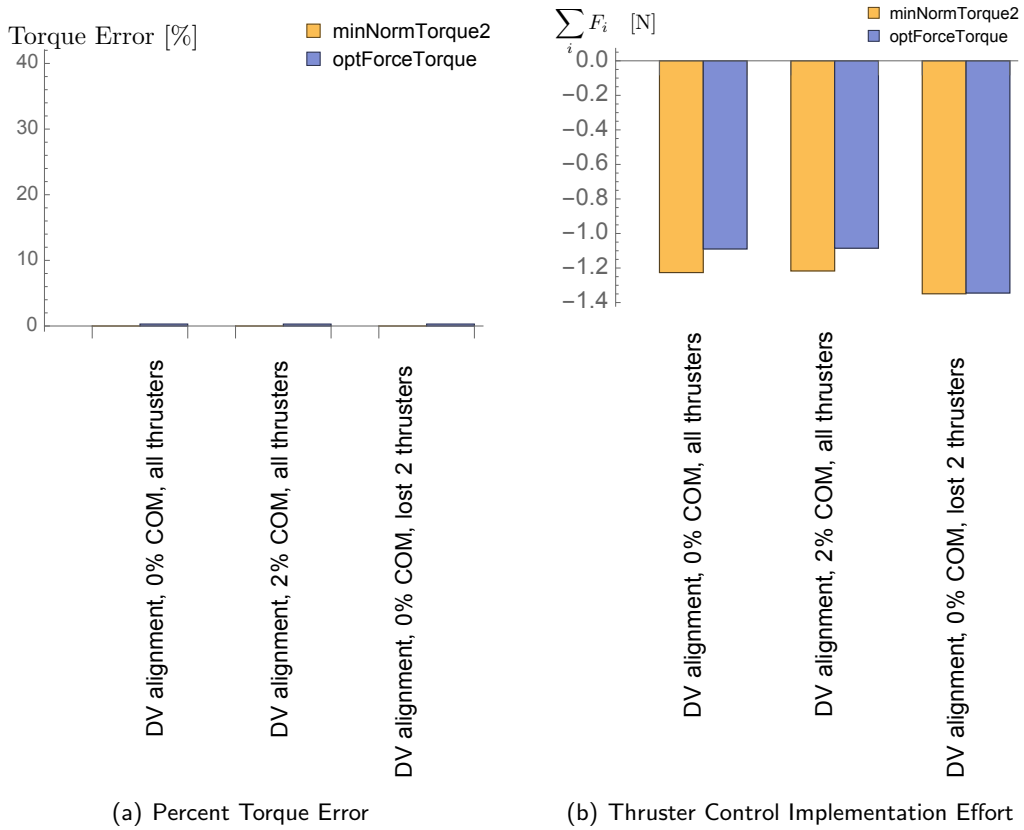
The minimum norm torque solution is now modified to use

$$\bar{\mathbf{F}} = ([C][\bar{D}])^T ([C][\bar{D}][\bar{D}]^T [C]^T)^{-1} [C] \mathbf{L}_r \quad (17)$$

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  $\mathbf{d}_i = (\mathbf{r}_i \times \hat{\mathbf{g}}_{t_i})$ . Note that  $\mathbf{r}_i \times \hat{\mathbf{g}}_{t_i}$  is a scaled axis along which the  $i^{\text{th}}$  thruster can produce a torque. The value  $\mathbf{d}_i$  will be near zero if the dot product of this axis with the current control axis  $\hat{\mathbf{c}}_j$  is small.





**Fig. 5:** DV Thruster Mapping Performance Mapping Illustration Comparing the 1-stage and 2-stage minimum norm solution to the thruster mapping of a nonlinear optimization solution.

To determine an appropriate  $\epsilon$  value, let  $\alpha$  be the minimum desired angle to avoid the control axis  $\hat{c}_j$  and the scaled thruster torque axis  $\mathbf{r}_i \times \hat{\mathbf{g}}_{t_i}$  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

$$\frac{d_i}{\bar{r}} > \cos(90^\circ - \alpha) = \sin \alpha \quad (18)$$

Thus, to estimate a good value of  $\epsilon$ , the following formula can be used

$$\epsilon \approx d_i^2 = \sin^2 \alpha \bar{r}^2 \quad (19)$$

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

## 6.2 $[C]$ matrix

The module requires 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

$$[C] = \begin{bmatrix} \hat{c}_1 \\ \vdots \end{bmatrix} \quad (20)$$

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

### 6.3 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).

## REFERENCES

- [1] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems*. AIAA Education Series, Reston, VA, 3rd edition, 2014.