Thruster calibration
for the SPHERES
spacecraft formation flight testbed

Pierre Blanc-Paques

October 2004 - June 2005

Advisor: Doctor Edmund Kong
Abstract

It is now established that distributed spacecraft architecture would allow to substantially reduce mission costs and, in the same time, improve performances for future space missions. Considering the possibilities that space interferometers, or satellites clusters should soon open, the autonomous formation flight thereby appears as an enabling technology for the future of the space adventure.

The SPHERES (Synchronized Position Hold, Engage and Reorient Experimental Satellites) testbed was thus designed to provide both the Air-Force and NASA with a long term, replenishable and upgradable testbed for the validation of high risk metrology, control and autonomy technologies.

The testbed consists of 5 small self-contained vehicles—the “spheres”—which can control their relative position and orientation. The vehicles are operable either on the Lab 2D frictionless platform, thanks to several air carriages, or in the pressurized micro-gravity environment that can be found in the NASA KC–135 or in the International Space Station (ISS). Tests that have yet been completed have demonstrated the use of SPHERES to design and study Formation Flight Algorithms.

The purpose of this report is to offer an overview of the satellites thruster management algorithm, that takes the state of the vehicle at time $t$ as input, and the forces and torques the vehicle is asked to provide in the absolute frame (to correct its state), and finally outputs the optimal thrusters firing times that enable the maneuver. We will see the problems raised by the development of such an algorithm: precision of the action performed, fuel consumption, time of computation, as well as some ways to overcome them.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of contents</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vii</td>
</tr>
<tr>
<td>Nomenclature</td>
<td>ix</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 A Spacecraft Formation Flight testbed</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Quick description of SPHERES</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1 General description</td>
<td>2</td>
</tr>
<tr>
<td>1.2.2 Actuation/Propulsion system</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Report outline</td>
<td>3</td>
</tr>
<tr>
<td><strong>2 SPHERES actuation mechanism</strong></td>
<td>5</td>
</tr>
<tr>
<td>2.1 Introduction to thrusters actuation</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Definition of the impulse</td>
<td>6</td>
</tr>
<tr>
<td>2.1.2 Control assuming minimal-constrained actuation</td>
<td>7</td>
</tr>
<tr>
<td>2.2 SPHERES constraints</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 Thruster provided forces</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2 Actuation period</td>
<td>9</td>
</tr>
<tr>
<td>2.2.3 Proof of SPHERES-constrained actuation</td>
<td>9</td>
</tr>
<tr>
<td>2.3 From the controller to the mixer</td>
<td>11</td>
</tr>
<tr>
<td>2.3.1 Description of the controller mechanism</td>
<td>11</td>
</tr>
<tr>
<td>2.3.2 Introduction to the mixer</td>
<td>12</td>
</tr>
<tr>
<td><strong>3 The existing mixer and its limitations</strong></td>
<td>15</td>
</tr>
<tr>
<td>3.1 Mechanism description</td>
<td>15</td>
</tr>
<tr>
<td>3.1.1 SPHERES physical properties assumption</td>
<td>15</td>
</tr>
</tbody>
</table>
CONTENTS

3.1.2 Mixer description .............................................. 17
3.1.3 Quantification of inherent errors ......................... 19
3.2 Actual physical properties .................................... 20
  3.2.1 Thrusters strengths ..................................... 20
  3.2.2 Thrusters misalignments ................................. 21
  3.2.3 Effects of multiple thrusters firing .................. 21
  3.2.4 Thrusters time of response ............................. 21
  3.2.5 Plan for designing a more accurate mixer ............ 21

4 Correction of thrusters induced errors ......................... 23
  4.1 Measurements of thruster properties ..................... 23
    4.1.1 Experiment description ................................. 23
    4.1.2 Results .................................................. 24
  4.2 Mixer and thrusters imperfections ....................... 26
    4.2.1 A 12×6 “mixing matrix” ...................................... 26
    4.2.2 Preliminary results ..................................... 30
    4.2.3 Tests .................................................... 30
  4.3 Considering only different thruster forces ................ 33
    4.3.1 Implementation ............................................. 33
    4.3.2 Tests .................................................... 34
  4.4 Outcome ....................................................... 36

5 Physical characteristics of thrusters ......................... 39
  5.1 Force drop due to multiple thrusters firing ............... 39
    5.1.1 Evaluation of the attenuation factor on the load cell... 40
    5.1.2 Implementation on SPHERES ............................. 41
    5.1.3 Tests of mixer 2 ......................................... 45
  5.2 Thruster delays, actuation and shut-off durations ......... 48
    5.2.1 Measuring the delays ..................................... 48
    5.2.2 Implementation on SPHERES ............................. 53
    5.2.3 As a conclusion on this subject .......................... 53

6 Summary .................................................................. 55
  6.1 The choice of a mixer ......................................... 55
  6.2 Implementation on SPHERES .................................. 57
    6.2.1 Coding ...................................................... 57
    6.2.2 Testing ...................................................... 58
  6.3 Conclusion related to the mixer design ..................... 58
List of Figures

1.1 A SPHERES satellite ........................................ 3
1.2 CAD model of a SPHERES vehicle .......................... 4

2.1 Actuation period for a single thruster ...................... 9
2.2 Difference between commanded and performed impulse ..... 13

3.1 Thruster geometry ............................................ 16
3.2 Plan for the design of a new mixer .......................... 22

4.1 Load cell experiment setup .................................... 24
4.2 Secondary axis errors for old mixer and mixer A .......... 32
4.3 Secondary axis errors for old mixer and mixer 1 .......... 35
4.4 Primary axis precision for old mixer and mixer 1 .......... 37

5.1 Firing schedule for two thrusters ............................ 42
5.2 Firing schedule for three thrusters .......................... 43
5.3 Explanation of the described mixers ........................ 45
5.4 Comparison between old mixer and mixer 2 ............... 47
5.5 Timing sequence for measurement of ignition delay ........ 49
5.6 Thruster rise time ............................................. 50
5.7 Thruster shut-off time ........................................ 52

6.1 Mechanism of the finally designed mixer ................... 57

B.1 Force measurement raw data ................................. 67
B.2 Comparison between old mixer and mixer 1 ................ 68
Nomenclature

DOF  Degrees of freedom
SPHERES  Synchronized Position Hold, Engage, Reorient Experimental Satellites
SSL  Space Systems Laboratory

\( F \)  Nominal thruster force (N)
\( F_x, F_y, F_z \)  Nominal forces in primary directions (N)
\( F_i \)  Measured force of thruster \( i \) (N)
\( r \)  Nominal thruster moment arm (m)
\( T \)  Control period (s)
\( T_a \)  Actuation period (s)
\( \eta \)  Duty cycle (%)
\( \bar{\mu} \)  Force impulse vector (N.s)
Chapter 1

Introduction

1.1 A Spacecraft Formation Flight testbed

Both NASA’s and ESA’s future space projects will widely call for the use of Separated Spacecraft Formation Flight Systems. In the space observation field for instance, upcoming missions (Terrestrial Planet Finder, Darwin) will depend on space interferometry to increase angular resolution of space-based telescopes. Thus, by using small separated optics to replace large mirrors that cannot easily be deployed in space, separated spacecraft architecture not only solve the size limitation problems due to launch fairings, but also improve overall space system performance by providing a high degree of redundancy and splitting up mission subfunctions.

However, interferometry requirements for observation at visible wavelength are extremely high. Position-wise, it demands on spacecraft being able to control their relative distance at nanometer level, over a separation of several meters.

In order to achieve such level of accuracy, and to as well master separated formation flight methods and technics, it is thus necessary to design and test new formation flight control, autonomy and metrology algorithms. This is the intent of the SPHERES testbed, which provide a low-cost, low-risk upgradable environment to design algorithms that will enable space formation flight.

As it has been pointed out by Alvar Saenz Otero [7], the SPHERES testbed provides the following advantages over the immediate deployment of a mission:

1. No research of new technologies is necessary for the development of the
testbed.

2. Minimal resources, both monetary and human, are required for deployment, operation, and maintenance of the units.

3. High risk controllers can be developed on a non-critical, low-risk environment.

4. Replenishment of propellant is quick and easy.

5. The testbed can be used for the development of thrust-balancing algorithms that optimize fuel consumption between units, and thus requires the depletion of propellant.

6. It can also be used for several future missions via software changes.

7. It provides experimental validation of theoretical models, and an understanding of physical events that may not be present in these models.

8. The testbed is upgradable to closer simulate specific systems.

1.2 Quick description of SPHERES

This report essentially focuses on the actuation/propulsion system. Therefore, a brief description of it is presented here.

1.2.1 General description

The SPHERES (Synchronized Position Hold, Engage and Reorient Experimental Satellites) testbed consists of 5 identical, independent and autonomous vehicles: the “spheres”, associated with a laptop computer standing for the ground-station. A view of a vehicle is shown on Figure 1.1. Each of the units contains propulsion, communication, processing and metrology systems, and is powered by AA batteries.

1.2.2 Actuation/Propulsion system

Most modern control design methodologies use of linear dynamics and linear, continuous, unbounded actuation. It is true that linear, continuous
momentum-transfer actuators such as reaction wheels are widely used for control of spacecraft attitude, but there are no effective linear, continuous mass-transfer actuators available for management of spacecraft position. Nonlinear, discontinuous actuators such as on-off thrusters are therefore necessary both for translation control and periodic desaturation of the reaction wheels. In this perspective, the SPHERES satellites only rely on on-off thrusters for management of both position and attitude.

The hardware thereby consists of twelve cold-gas thrusters—one positive and one negative per DOF, a tank containing CO$_2$ propellant, and regulators, valves and pipes required to connect the tank to the thrusters. It is associated with a dedicated flight software running at a high frequency timed interrupt process.

1.3 Report outline

This report describes the work that has been performed concerning the SPHERES thruster management algorithm. The purpose of this brief introduction is then to allow the reader to get familiar with the main concepts and systems that will be further discussed.

Developing a correct thruster management algorithm—mixer—should improve the actuation accuracy. Such an algorithm thus takes as input the state of the vehicle at time $t$ and the forces and torques the vehicle is asked to provide in the absolute frame (to correct its state), and outputs the optimal thrusters firing times that enable the maneuver between $t$ and $t + \delta t$.

To go through the steps of developing such a mixer, we are first going to
Introduction

Figure 1.2: CAD model of a SPHERES vehicle, showing important external features. The external diameter of a sphere is approximately 21 cm, and the wet mass 4.82 kg.

fully describe the mechanism of the SPHERES actuation, and introduce the existing mixer. Then, we will point out its limitation, and suggest some ways to overcome them. As a conclusion, we will then put forward a mixer design that addresses the limitations of the current mixer.
Chapter 2

SPHERES actuation mechanism

Since the SPHERES vehicles only rely on on-off thrusters for management of both position and attitude, an accurate control of the thrusters is critical for the needs of the testbed: it represents the only way to act on the spheres dynamics. In this particular case though, controlling the thrusters only implies to properly set their firing times, since the thrust they provide cannot be modified.

It is not particularly obvious though, that this actuation mechanism is dependable. The goal of this chapter is to prove its concept according to the physical constraints imposed by the vehicles design. This way, we are going to have an indication of the control accuracy we can aim for considering that the vehicles are able to perfectly execute the actuation command, which is a good starting point for evaluating the thruster management algorithm performances.

For this chapter, we are first going to introduce some basic physical laws, and then interpret them according to the SPHERES parameters to point out how to reliably control the vehicles. What we are are going to show thus is, first, how thruster actuation is possible considering the SPHERES vehicles, and second, how it is actually realized considering the constraints linked to the testbed.

2.1 Introduction to thrusters actuation

We define here the actuation of a spacecraft as:

The act of propelling a spacecraft by applying to it forces and
torques that allow to update its state according to the control command.

The actuation on SPHERES is based on thrusters actuation, that we are going to introduce now.

### 2.1.1 Definition of the impulse

Considering a spacecraft defined by its mass $m(t)$, its constant inertia matrix $I$, and its initial state $S_0$ (position vector $\overrightarrow{P}_0$, velocity $\overrightarrow{v}_0$, attitude quaternion $\overrightarrow{q}_0$, rotation vector $\overrightarrow{\Omega}_0$), let’s analyze its state evolution when it gets applied a propulsive force $\overrightarrow{F}(t)$ and torque $\overrightarrow{M}(t)$.

Calling $\overrightarrow{P}(t)$ and $\overrightarrow{v}(t)$ respectively the position and velocity of the spacecraft center of mass at instant $t$, it comes:

$$\overrightarrow{P}(t) = \overrightarrow{P}_0 + \int_{t_0}^{t} \overrightarrow{v}(\tau) \, d\tau$$  \hspace{1cm} (2.1)

We also get the same kind of relation for the velocity:

$$\overrightarrow{v}(t) = \overrightarrow{v}_0 + \int_{t_0}^{t} \frac{d\overrightarrow{v}}{dt}(\tau) \, d\tau$$  \hspace{1cm} (2.2)

$\overrightarrow{F}(t)$ being a propulsive force, we know that this equation can also be written:

$$\overrightarrow{v}(t) = \overrightarrow{v}_0 + \int_{t_0}^{t} \frac{\overrightarrow{F}(\tau)}{m(\tau)} \, d\tau$$  \hspace{1cm} (2.3)

The demonstration is shown in appendix A.1 that proves that considering the SPHERES physical parameters, and particularly the actuation period width $t - t_0$, it can be assumed with an excellent accuracy that the mass of the spacecraft is constant between $t_0$ and $t$. Then, calling impulse and writing $\overrightarrow{\mu}$ the quantity $\overrightarrow{\mu}(t) = \int_{t_0}^{t} \overrightarrow{F}(\tau) \, d\tau$, and replacing in equation 2.1 it comes:

$$\begin{cases} 
\overrightarrow{v}(t) = \overrightarrow{v}_0 + \frac{\overrightarrow{\mu}(t)}{m} \\
\overrightarrow{P}(t) = \overrightarrow{P}_0 + (t - t_0)\overrightarrow{v}_0 + \frac{1}{m} \int_{t_0}^{t} \overrightarrow{\mu}(\tau) \, d\tau
\end{cases}$$  \hspace{1cm} (2.4)

\footnote{The geometry of the SPHERES vehicles makes that the inertia matrix can be assumed constant with an excellent accuracy.}
2.1 Introduction to thrusters actuation

Proving that for short actuation period, the velocity is only a function of the impulse $\vec{\mu}(t)$, while the position is a function of time and the temporal integral of the impulse. A similar development would show that, in the same way, the rotation vector only depends of $\vec{\mu}$, and the attitude quaternion only depends on time $t$ and the temporal integral of $\vec{\mu}$.

We are thus going to prove the concept of actuating the spheres vehicles by modifying the impulse function $\vec{\mu}$ due to the thrusters. Since the velocity/position and angular rate/attitude equations are similar, we are only going to develop here the control of velocity and position.

2.1.2 Control assuming minimal-constrained actuation

In this section, we are going to prove the possibility of controlling spheres-like spacecrafts using minimal constrained actuation.

Considering real thrusters, the force they provide basically is temporally piecewise constant since it only depends on the mass flow of expelled gas and its relative expelling speed, which are usually constant for a particular motor working at constant external pressure. Assuming additionally that we dispose of multiple thrusters, the value of the total force can be adjusted according to a specified time-function. We can then defined minimal-constrained actuation as:

Actuating a spacecraft by applying to it temporally piecewise constant forces and torques, meaning that (for $\vec{F}$):

$\forall t \in \mathbb{R}, \exists \alpha > 0 \mid \vec{F}$ can be freely set constant on $[t - \alpha, t + \alpha]$

Considering a vehicle in its initial known state $S_0$ at instant $t_0 = 0$, the desired state $S_1$ at instant $t_1 > t_0$, and using the constant mass approximation, let’s determine the time-dependant force function $\vec{F}$ that enables to shift from $S_0$ to $S_1$.

Calling $\vec{P}_i$ and $\vec{v}_i$ respectively the position and the velocity of the vehicle at instant $t_i$, it comes according to system 2.4:

$$\begin{cases}
\vec{\mu}(t_1) = m(\vec{v}_1 - \vec{v}_0) \\
\int_0^{t_1} \vec{\mu}(\tau) \, d\tau = m(\vec{P}_1 - \vec{P}_0 - t_1 \vec{v}_0) 
\end{cases}$$

The impulse function required to shift from $S_0$ to $S_1$ is then solution of a
system of the following kind:
\[
\begin{align*}
\vec{\mu}(0) &= 0 \\
\vec{\mu}(t_1) &= \vec{\mu}_f \\
\int_0^{t_1} \vec{\mu}(t) \, dt &= \vec{\xi}
\end{align*}
\] (2.6)

The demonstration is given on appendix A.2 that the particular piecewise constant function \( \vec{F} \) defined as follows creates an impulse function \( \vec{\mu} \) solution of this problem.

\[
\vec{F} : \begin{cases} 
[0, t] &\rightarrow \mathbb{R}^3 \\
\tau &\rightarrow \vec{F}(\tau) = \begin{cases} 
\frac{1}{t} \left( \frac{4}{t} - \vec{\mu}_f \right) &\text{if } \tau \leq \frac{t}{2} \\
\frac{1}{t} \left( 3\vec{\mu}_f - \frac{4}{t} \right) &\text{if } \tau > \frac{t}{2}
\end{cases}
\end{cases}
\]

We thus have proven the concept of thrusters actuation for a perfect thrusters combination that can produce whatever force is required for an arbitrary duration. Now, let’s see how to adapt this to SPHERES.

2.2 SPHERES constraints

2.2.1 Thruster provided forces

All of the thrusters used on the SPHERES vehicles consist of a solenoid valve and a nozzle, involving non-linear, discontinuous, bounded behavior.

When a thruster is commanded on, a voltage spike and hold circuit activates and holds open the solenoid valve. The thruster output force then increases from zero to the steady-state thrust, and returns to zero when the thruster is commanded off and the solenoid valve closes. For now, we are going to assume that there is no delay between the on-command and the valve opening, and that the force rise is instantaneous. The force provided by each thruster thus equals the nominal thrust when the thruster is commanded open, and zero otherwise.

Therefore, and since the spheres possess one negative and one positive thruster per DOF, the vehicles are only able to provide the nominal forces \( F_x, F_y, F_z \) and their opposites on the \( X, Y \) and \( Z \) axis. Thus, the only way to get a given impulse \( \vec{\mu} \) at an instant \( t \) is to modulate the thrusters opening durations so that:

\[
t_x F_x \vec{u}_x + t_y F_y \vec{u}_y + t_z F_z \vec{u}_z = \vec{\mu}(t)
\] (2.7)
2.2 SPHERES constraints

Where $t_i$ and $\overrightarrow{u_i}$ represent respectively the opening time and the unitary vector for direction $i$.

2.2.2 Actuation period

For the spheres are run by a clocked digital control computer, control commands arise at fixed time intervals, which duration $t - t_0$ defined previously is actually named control period.

In SPHERES though, the actuation period cannot last for the whole control period because of the perturbations the CO$_2$ expelled by the thrusters cause in the US pulses times of flight. Global telemetry updates and thruster firings must be separated to avoid contamination of the US readings.

Therefore, and as shown on Figure 2.1, the actuation period is defined as a fraction of the control period according to the relation $T_{actuation} = T_{control} \times \eta$, where $\eta < 1$ is named duty cycle.

\[ T_{actuation} = T_{control} \times \eta \]

Figure 2.1: Actuation period, control period and firing time for a single thruster. The control period is the reproduced time interval during which control tasks are performed.

2.2.3 Proof of SPHERES-constrained actuation

So, the SPHERES thrusters are far from being perfect. We want to prove though that the actuation they provide enables the control of the spacecrafts.

The problem being symmetrical for $X$, $Y$ and $Z$ axis, we are going to consider only the projection of system 2.6 on the $X$ axis for instance, and show how it is possible to fix the thrusters opening times so that the created impulse satisfies all the requested conditions.
Calling $T$ the control period, $S_0$ the known initial state at $t = 0$, and $S_1$ the desired state at $t = T$, the system 2.6 becomes:

$$\begin{cases} 
\mu_x(0) = 0 \\
\mu_x(T) = \mu_f \\
\int_0^T \mu_x(t) \, dt = \xi_x \end{cases} \quad (2.8)$$

Then, defining the $X$ axis thrusters opening time $t_{on}$, as well as the $X$ axis thrusters on time $t_x$ by:

$$\begin{cases} 
t_x = \frac{|\mu_f|}{F_x} \\
t_{on} = T - \frac{|\mu_f|}{2F_x} - \frac{\xi_x}{\mu_f} \end{cases} \quad (2.9)$$

Where the absolute values are only used here to ensure that the commanded impulse is on the positive $X$ axis. Otherwise, we just would have to consider the firing schedule for $-X$ thrusters.

It can be proven\(^\text{2}\) that the created impulse satisfies to the system 2.8 if the firing sequence is the following:

1. Do not fire in time-interval $[0, t_{on}]$.
2. Fire in $[t_{on}, t_{on} + t_x]$.
3. Do not fire in $[t_x, T]$.

Since we obviously need $t_{on} > 0$ as well as $t_{on} + t_x < \eta T$, we are introducing constraints on the command $\mu'$ and $\xi'$. We will see later that these constraints are equivalent to bounding the amplitude of the state shift that is asked of the vehicle, and can be satisfied by taking obvious precautions with the control command.

So, and as it has been previously mentioned, equations of the same kind can be obtained for $Y$ and $Z$ forces, as well as for the actuation of the attitude quaternion and the angular rate. What we have proven then, is that it is possible to control the spheres actuation using only the thrusters. What we are going to see next, is how the actuation is physically realized in SPHERES.

\(^{2}\)The proof is given on appendix A.3.
2.3 From the controller to the mixer

2.3.1 Description of the controller mechanism

Unfortunately, the SPHERES controller does not work exactly according to the mechanisms that have been described above.

For a particular vehicle, considering on one hand the actual state at instant \( t \), and on another hand the desired state, the controller can determine the state error vector at \( t \). Then, to save computation time, it works in a linear way: gains are affected to each coefficient of the state error vector, that convert them into forces and torques that the vehicle has to provide over the whole actuation period to correct its state. For instance, an error of \( \vec{e}_p \) in position and \( \vec{e}_v \) in velocity will be converted into a commanded force \( \vec{F}_c = G_p \vec{e}_p + G_v \vec{e}_v \) which, if provided over the whole actuation period, will create the required correction impulse.

What has been proved previously though, is that the state of the vehicle both depends on \( \vec{\mu}(T) \), the value of the impulse after a control period, and on \( \int_0^T \vec{\mu}(\tau) \, d\tau \), the time integral of the impulse over a control period.

However, by commanding constant forces and torques that have to be generated over the whole actuation period, the controller cannot set both \( \vec{\mu}(T) \) and \( \int_0^T \vec{\mu}(\tau) \, d\tau \). Indeed, the actuation period being fixed to \( T_a \), what is actually commanded by the controller is the impulse at \( T \):

\[
\vec{\mu}_c(T) = T_a \vec{F}_c
\]

Which fixes the time integral of this commanded impulse to:

\[
\int_0^T \vec{\mu}_c(t) \, dt = T_a \left( T - \frac{T_a}{2} \right) \vec{F}_c
\]  

(2.11)

Since it is not the purpose of this report to detail exhaustively the controller used in SPHERES, we are going to trust that the gains are set properly enough so that the commanded force \( \vec{F}_c \) optimizes the correction that is performed on both position and velocity, as well as the commanded torque \( \vec{M}_c \) optimizes the correction on attitude and angular rate. Thus, it is trusted that commanding \( \vec{F}_c \) and \( \vec{M}_c \) over an actuation period is the most effective way to control the vehicles.
2.3.2 Introduction to the mixer

The inputs to the thruster management algorithm (also known as the mixer), are the forces and torques commanded by the controller. And yet we have shown that the SPHERES actuators cannot generate variable thrust, meaning that $F_c$ and $M_c$ will most likely not match with the vehicle's physical capabilities.

A way to overcome this problem would be to properly set thrusters firing times so that we still get the equivalent thrusters provided impulse $\vec{\mu}_p$:

$$
\begin{align}
\vec{\mu}_p(T) &= \vec{\mu}_c(T) \\
\int_0^T \vec{\mu}_p(t) dt &= \int_0^T \vec{\mu}_c(t) dt
\end{align}
$$

This is possible by wisely determining thrusters on-time according to the force they can provide, and by centering their impulse on the middle of the actuation period, as shown on Figure 2.2. Indeed, projecting equations 2.12 and 2.13 on $X$ axis for instance, we can determine the opening time $t_x$ for thrusters in direction $X$ that enables to satisfy the equation 2.12 condition:

$$
t_x = T_a \frac{F_{c_x}}{F_x}
$$

Then, by firing in the time interval $[t_{on} = \frac{T_a-t_x}{2}, t_{off} = \frac{T_a+t_x}{2}]$ we get:

$$
\int_0^T \mu_{px}(t) dt = \int_0^{t_{on}} \mu_{px}(t) dt + \int_{t_{on}}^{t_{off}} \mu_{px}(t) dt + \int_{t_{off}}^T \mu_{px}(t) dt \\
= F_x t_x \left( T - \frac{T_a}{2} \right)
$$

Considering then equations 2.14 and 2.10 it comes:

$$
\int_0^T \mu_{px}(t) dt = \int_0^T \mu_{c_x}(t) dt
$$

Which proves that the controller command can be performed without inducing any error in the actuation: a perfect mixer should be feasible.
2.3 From the controller to the mixer

Figure 2.2: Difference between commanded impulse and performed impulse. It is clear here that the areas under the two curves are the same, as well as the value of the impulses at $T$, which proves that the performed impulse thus satisfies equations 2.12 and 2.13.
Chapter 3

The existing mixer and its limitations

It is thus the role of the mixer to properly set the thrusters firing times, so that the physical actuation perfectly matches the control command. Now that this has been proven feasible, and that all useful concepts have been made clear, we are going to discuss the problems raised by the current mixer design.

In this chapter, we are first going to describe the mechanism of the existing mixer: by putting forward the assumptions it makes on the SPHERES physical properties, and then explicitly detailing the algorithm it uses. Next, we are going to confront these assumptions with the actual parameters, showing the necessity of designing a new mixer.

3.1 Mechanism description

3.1.1 SPHERES physical properties assumption

To determine the thrusters firing schedule from the controller, the mixer obviously needs a thorough description of the thrusters mechanism. However, in developing the existing mixer, several assumptions were made, which we are going to describe here.

Assumption 1 (Geometry) All thrusters are perfectly located on the vehicles and their thrust direction is perfectly aligned with the body axis.
The existing mixer and its limitations

In this particular case, in here the description of thruster geometry and thrust direction is given on Figure 3.1, this means that every thruster has the same moment arm \( r \), and no off-axis force or torque is induced by firing the thrusters corresponding to a particular body-axis. Writing \( \vec{d}_i \) as the normalized thrust direction for thruster \( i \), \( \vec{d}_i \) is thus joined to a primary body direction.

![Figure 3.1: Thruster geometry. This geometry allows for pure body-axis force or torque using only two thrusters](image)

**Assumption 2 (Force)** All thrusters provide the exact same nominal force \( F \), no matter how many thrusters are on, and their reaction to on-off commands is instantaneous.

All thrusters are thus supposed to have the same behavior. Therefore, we can bring together opposite thrusters in pairs, the thruster pair \( p_i \) being defined as the ordered pair of opposite thrusters \( (i, i + 6) \). Then, the force \( \vec{f}_{p_i} \) provided by the thruster pair \( p_i \) is:

\[
\vec{f}_{p_i} = F(\delta_i - \delta_{i+6})\vec{d}_i \tag{3.1}
\]

Where \( \forall i \in [0, 11] \):

\[
\delta_i = \begin{cases} 
1 & \text{if thruster } i \text{ is commanded on} \\
0 & \text{otherwise}
\end{cases}
\]
3.1 Mechanism description

3.1.2 Mixer description

Considering this description of the thrusters, we can now introduce the mixer mechanism itself.

Using equation 3.1 and Figure 3.1, it is possible to obtain a relationship mapping the body-frame forces and torques \( f_x, f_y, f_z, m_x, m_y, m_z \) which the vehicle is exposed to, and the forces provided by the thruster pairs:

\[
\begin{bmatrix}
  f_x \\
  f_y \\
  f_z \\
  m_x \\
  m_y \\
  m_z 
\end{bmatrix}
= \mathcal{M}^{-1}
\begin{bmatrix}
  f_{p_0} \\
  f_{p_1} \\
  f_{p_2} \\
  f_{p_3} \\
  f_{p_4} \\
  f_{p_5} 
\end{bmatrix}
\]  

(3.2)

Where \( \mathcal{M}^{-1} \) is defined according to Figure 3.1 as:

\[
\mathcal{M}^{-1} =
\begin{bmatrix}
  1 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 1 & 0 & 0 \\
  0 & 0 & 0 & 0 & 1 & 1 \\
  0 & 0 & 0 & 1 & 0 & 0 \\
  1 & -1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & -1 & 0 & 0 
\end{bmatrix}
\]  

(3.3)

Defining now \( u_{p_i} \), the opening time for pair \( p_i \), considering the impulse \( \vec{\mu}_{p_i} \) provided by the thruster pair \( p_i \) and the thruster nominal parameters:

\[
\vec{\mu}_{p_i} = u_{p_i} F d_i
\]  

(3.4)

We can transform the body-frame forces and torques commanded by the controller for the actuation period\(^1\) into thruster pairs opening time by:

\[
\begin{bmatrix}
  u_{p_0} \\
  u_{p_1} \\
  u_{p_2} \\
  u_{p_3} \\
  u_{p_4} \\
  u_{p_5} 
\end{bmatrix}
= \frac{T_a}{F} \times \mathcal{M}
\begin{bmatrix}
  f_x \\
  f_y \\
  f_z \\
  m_x \\
  m_y \\
  m_z 
\end{bmatrix}
\]  

(3.5)

\(^1\)See section 2.3 on page 11
And then determine single thrusters opening time according to the equation 3.4:

\[
\forall i \in [0, 5] \left\{ \begin{array}{ll}
  u_i & = \begin{cases} 
  u_{p_i} & \text{if } u_{p_i} > 0 \\
  0 & \text{otherwise} 
  \end{cases} \\
  u_{i+6} & = \begin{cases} 
  0 & \text{if } u_{p_i} > 0 \\
  -u_{p_i} & \text{otherwise} 
  \end{cases}
\end{array} \right.
\] (3.6)

Where \( u_i \) is the opening time for thruster \( i \).

Thus, the mixing-matrix \( M \) gives us a direct relationship between the controller command and the thrusters opening times that will allow the sphere to satisfy this command. Though, we can already see two issues:

**Maximal firing duration:** obviously no opening time can be greater than the actuation period \( T_a \), so the command sometimes might have to be bounded. What actually happens for practical purpose is a scaling of the firing durations, should any of them be greater than \( T_a \): \( u_{\text{max}} \) being the greatest commanded on-time, all commanded opening times are scaled by a factor \( \frac{T_a}{u_{\text{max}}} \) if \( u_{\text{max}} > T_a \). This way, no firing duration exceeds \( T_a \) and the resulting impulse is a known fraction of the controller-commanded one.

**State evolution during an actuation period:** the controller commands global-frame forces and torques, that are converted into body-frame commands knowing the state of the sphere at an instant \( t \): the mixer does not take into account the sphere state evolution during the actuation period. If this evolution is fast enough though, the body-frame actuation commanded at \( t \) could be totally improper approaching the end of the actuation period. For instance, let’s assume that the global axis \( X \) and the sphere axis \( x \) are the same at instant \( t \), that the sphere is rotating around its \( z \) axis with an angular rate \( \Omega \), and that a global-frame force \( F_X \) is commanded at \( t \). What will happen is the following sequence:

1. The global-frame force \( F_X \) is converted into a body-frame force \( F_x \) (\( X \) and \( x \) are joined at \( t \)).
2. According to equation 3.5, the mixer commands the opening of thrusters 0 and 1.
3. During the thrust, the global-frame thrust direction varies according to \( \Omega \), meaning that the resulting impulse over the actuation period won’t match the command.
3.1 Mechanism description

This behavior has to be discussed, so that we can know the order of magnitude of the errors we are looking at.

### 3.1.3 Quantification of inherent errors

Considering the development above, let’s determine the error induced by the mixer mechanism itself, and the fact that the state evolution of the vehicle during an actuation period is not taken into account.

Considering the simple 2D-case described above, where a sphere rotates around its $z$ axis at an angular rate $\Omega$ constant for now, that body-frame equals global-frame at $t = 0$, and that the controller commands an $F_X$ force at $t = 0$, meaning a $\mu_X$ impulse, let’s determine the actual impulse $\vec{\mu}_a$ provided by the sphere over the actuation period.

Introducing the $\theta$ angle describing the rotation of the sphere, we can easily determine the global-frame force $\vec{F}_a$ actually produced by the sphere when it fires its $x$ or $y$ axis thrusters, producing the forces $F_x$ and $F_y$:

$$\vec{F}_a = (F_x \cos \theta - F_y \sin \theta) \vec{X} + (F_x \sin \theta + F_y \cos \theta) \vec{Y} \quad (3.7)$$

We also have the relationship:

$$\theta(t) = \theta_{t=0} + \Omega t = \Omega t \quad (3.8)$$

Considering then that the firing-times are centered in the middle of the actuation period, as mentioned in section 2.3, and calling $t_x$ and $t_y$ the opening time respectively for $x$ and $y$ thrusters, it can be determined that:

$$\vec{\mu}_a = \mu_{aX} \vec{X} + \mu_{aY} \vec{Y} \quad (3.9)$$

Where:

$$\begin{cases} 
\mu_{aX} = \frac{2}{\Omega} \left[ F_x \sin(\Omega \frac{t_x}{2}) \cos(\Omega \frac{T_a}{2}) - F_y \sin(\Omega \frac{t_y}{2}) \sin(\Omega \frac{T_a}{2}) \right] \\
\mu_{aY} = \frac{2}{\Omega} \left[ F_x \sin(\Omega \frac{t_x}{2}) \sin(\Omega \frac{T_a}{2}) - F_y \sin(\Omega \frac{t_y}{2}) \cos(\Omega \frac{T_a}{2}) \right] 
\end{cases} \quad (3.10)$$

In our case, considering that an $F_X$ force is commanded at $t = 0$, where the body-frame equals the global-frame, we have $t_y = 0$. Considering then a fast rotation at $\frac{\Pi}{4}$ radians per second, a $t_x$ opening time of 100 ms, and an actuation period of 250 ms, we get:

$$\vec{\mu}_a = 0.0994 \times F_X \vec{X} + 0.0098 \times F_X \vec{Y} \quad (3.11)$$
The existing mixer and its limitations

However, what we should obtain is:

$$\vec{\mu}_0 = t_x F_x \vec{X} = 0.1 \times F_x \vec{X}$$ (3.12)

Which represent less than 1% error on the impulse command for the X axis. The same kind of calculation also shows error of this order of magnitude as far as the impulse time-integral.

Similar developments would also prove that in the worst case, entailing a moving sphere and commanded forces and torques, this very same error would not exceed 3% of the command. This error is due to the facts that:

- the actuation frequency is not infinite;
- the state evolution of a vehicle during an actuation period is not taken into account in the mixer.

Thus, it cannot be shrunk without designing a completely different—and far more complex—mixer, which cannot be done considering the computation time constraint. Therefore, we have to know that the mixer mechanism is *per se* not perfect, and that it is not possible to improve it to an 100% accuracy. But considering this architecture, we still need to know whether our physical assumptions are reliable or not, and to quantify the error they might induce in the actuation mechanism.

### 3.2 Actual physical properties

Experiments run with the mixer we described show unexpected results for the vehicles, such as incapability to fly straight or control imprecisions.

Based on the previous development, a natural idea would be to look for the cause of these errors in the actual SPHERES physical properties, and more precisely in the propulsion sub-system properties.

The research conducted by Chen [3] demonstrated multiple imperfections in this subsystem, and his work is briefly summed up here.

#### 3.2.1 Thrusters strength

The thrust provided by the thrusters differs from one to another, due to multiple factors: nozzle exit area imperfections, thermal effects … These
3.2 Actual physical properties

differences can be up to approximately 10% of the nominal thruster force of 0.12 N.

By firing several thrusters, this could lead to the creation of undesirable off-axis torques that are not predicted by the mixer described above, and that could explain the difficulties encountered with straight line translation motion.

3.2.2 Thrusters misalignments

The thrusters in the spheres are actually not perfectly aligned with their principal body axis, which means that every single thruster, when firing, creates a force on:

- its primary direction—the direction where it is supposed to thrust—for 99% of the created force;
- the two secondary other directions, for 1% of the created force.

3.2.3 Effects of multiple thrusters firing

Only one gas tank is used to provide all twelve thrusters with the CO$_2$ propellant. Then, the fact that the gas pressure, and so the force of the thrusters, remains steady as several thrusters are open, can be in doubt.

This phenomenon has been proven experimentally, and entails produced impulses far less powerful than commanded ones.

3.2.4 Thrusters time of response

We have assumed so far that the thruster response was instantaneous. Actually, delays in the system leads to a non-zero time of response when thrusters are commanded on or off.

This effect has to be corrected, especially since it could prevent the vehicle from producing very short impulses.

3.2.5 Plan for designing a more accurate mixer

In order to enhance our mixer, we are going to correct separately all these off-nominal effects so that we can see the contribution of these imperfection to the spheres thrusting capability.
The existing mixer and its limitations

As described on Figure 3.2, we are first going to look into the thrusters induced errors: thrusters misalignments, and differences in thrusters strengths. Then we will look into the errors due to what surrounds the thrusters: the gas supply and the opening/closing system.

![Diagram](image)

**Figure 3.2**: Plan for the design of a new mixer
Chapter 4

Correction of thrusters induced errors

In this chapter, we are focusing on the imperfections in the propulsion system that are directly linked to the thrusters, their mechanism, and the way they are mounted on the spheres. What we are going to show is that it is actually possible to design a mixer that completely takes into account the misalignments of the thrusters and their strengths differences.

We are first going to introduce an experiment that enables to measure some important thruster properties. Then, we will see how to take advantage of these measurements to let the mixer know exactly the resulting impulse it would get by firing a specific thruster.

4.1 Measurements of thruster properties

It has been shown by Dustin Berkovitz [2] that by using a 6 DOF load cell and a proper mounting, it would be possible to measure the force and torques produced by a sphere to a relatively good accuracy. This is what is going to be discussed in this section.

4.1.1 Experiment description

The load cell that has been used is a 6 DOF JR-3 load cell. To mount the sphere on the sensor, a special mounting plate has been made as shown on Figure 4.1.
Correction of thrusters induced errors

Figure 4.1: Load cell experiment setup: the mounting plate, the load cell that has been used, and the final setup.

In this experiment, each thruster was turned on and off for one second. The measured forces and torques over time in each axis were collected with SigLab and processed with Matlab scripts to filter out teststand vibration, change coordinate frames, and decouple forces and moments. SigLab’s handbook claims to have an accuracy of 0.0025% of the selected full-scale range for the \( y \) measurement. In our case, the resolution of the measurement system thus is \( 2.10^{-5} \) N for the measured forces, with a bandwidth up to 20 khz.

To get rid of ignition and shut-off undesirable effects, the force components were averaged over 500 ms in the middle of the one second pulse to create a nominal vector of created force in all 3 axes for each thruster. From this, we could determine the resultant thrust magnitude as well as its direction.

These nominal values were averaged over many test sessions, and the results are shown below.

4.1.2 Results

What appears after the experiments is that the thrusters are often far from producing their nominal force of 0.12 N. An example of data measured can be seen on Appendix B. For Sphere S/N \#2 for instance, and in agreement with
4.1 Measurements of thruster properties

Berkovitz’s results, the measured thrusters forces are shown on Table 4.1.

Table 4.1: Measured forces for Sphere S/N #2. The $x$, $y$ and $z$ axis are stated in the vehicle frame.

<table>
<thead>
<tr>
<th>Thruster #</th>
<th>$F_x$ (N)</th>
<th>$F_y$ (N)</th>
<th>$F_z$ (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory</td>
<td>Actual</td>
<td>Theory</td>
</tr>
<tr>
<td>0</td>
<td>0.12</td>
<td>0.0995</td>
<td>0.</td>
</tr>
<tr>
<td>1</td>
<td>0.12</td>
<td>0.0965</td>
<td>0.</td>
</tr>
<tr>
<td>2</td>
<td>0.</td>
<td>0.0012</td>
<td>0.12</td>
</tr>
<tr>
<td>3</td>
<td>0.</td>
<td>-0.002</td>
<td>0.12</td>
</tr>
<tr>
<td>4</td>
<td>0.</td>
<td>-0.0001</td>
<td>0.</td>
</tr>
<tr>
<td>5</td>
<td>0.</td>
<td>-0.0008</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-0.12</td>
<td>-0.1058</td>
<td>0.</td>
</tr>
<tr>
<td>7</td>
<td>-0.12</td>
<td>-0.1025</td>
<td>0.</td>
</tr>
<tr>
<td>8</td>
<td>0.</td>
<td>0.0041</td>
<td>-0.12</td>
</tr>
<tr>
<td>9</td>
<td>0.</td>
<td>-0.003</td>
<td>-0.12</td>
</tr>
<tr>
<td>10</td>
<td>0.</td>
<td>0.0009</td>
<td>0.</td>
</tr>
<tr>
<td>11</td>
<td>0.</td>
<td>-0.0005</td>
<td>0.</td>
</tr>
</tbody>
</table>

For all of the primary axis\(^1\) measurements, the standard deviation is less than or equal to $2.10^{-3}$ N, meaning that our average seems acceptable considering the order of magnitude of the measured quantities.

For the secondary axis measurements however, the standard deviation sometimes reaches $10^{-4}$ N depending on the considered thruster, or axis. Considering the order of magnitude of the measured forces on secondary axis, some of our measurements might thus not be accurate.

Considering the produced torques, the results obtained by calculating the vector product of thruster position in the vehicle frame, given by the sphere CAD, and thruster force measured on the load cell, match the torques measurements. Thus, this calculation is the \textit{modus operandi} we are going to use from now for the computation of produced torques.

What is also shown is that the undesirable forces on secondary axis actually exist, and that the largest ones can be characterized with an acceptable precision. Then, it should be possible to take into account these drifts in the mixer.

\(^1\)For each thruster, the axis on which the thruster is expected to thrust.
4.2 Mixer and thrusters imperfections

The 6×6 mixing matrix \( M \) played a crucial role in the “old” mixer, since it directly mapped the commanded forces and torques to the corresponding thrusters on-times.

A huge assumption has been made though, that enabled to use this matrix: opposite thrusters were expected to produce the exact same opposite force. This way, the 12 thrusters could be rearranged in 6 thrusters pairs.

By seeking for 6 pairs on-times instead of 12 thrusters on-times, and thanks an obvious mathematical artifice, the 6 equations of motions—3 commanded forces and 3 commanded torques—thus happened to be sufficient to solve our problem.

But according to the results of Table 4.1, we obviously cannot arrange our thrusters in pairs anymore, if we want an acceptable precision for the mixer. We thus have now to find 12 firing times out of our 6 equations of motion.

4.2.1 A 12×6 “mixing matrix”

Efforts have been made to stay as close as possible to this simple relation between thrusters firing times and commanded forces and torques. Therefore, a kind of new 12×6 “mixing matrix” has been developed.

For every thruster \( i \), let us call \( \vec{F}_i \) the force produced by the thruster, and \( f_{ix} \) the normalized projection of this force on the vehicle \( x \)-axis:

\[
\vec{F}_i = F_i(f_{ix}\vec{x} + f_{iy}\vec{y} + f_{iz}\vec{z}) \tag{4.1}
\]

With of course \( \sqrt{f_{ix}^2 + f_{iy}^2 + f_{iz}^2} = 1 \).

Also, calling \( \overrightarrow{OP}_i \) the position vector of thruster \( i \) from the sphere center of mass, we can define \( \overrightarrow{M}_i \), the torque produced by the thruster \( i \). And according to the section 4.1.2 this torque can be calculated by:

\[
\overrightarrow{M}_i = \overrightarrow{OP}_i \times \vec{F}_i = F_i \left[ \overrightarrow{OP}_i \times (f_{ix}\vec{x} + f_{iy}\vec{y} + f_{iz}\vec{z}) \right] \tag{4.2}
\]

It is then easy to define the coefficients \( m_{ix} \) such that:

\[
\overrightarrow{M}_i = F_i(m_{ix}\vec{x} + m_{iy}\vec{y} + m_{iz}\vec{z}) \tag{4.3}
\]
4.2 Mixer and thrusters imperfections

Then, calling \( u_i \) the opening time for thruster \( i \) (with \( u_i > 0 \)) it is obvious that the total force impulse \( F \cdot T_a \) and torque impulse \( M \cdot T_a \) produced by the firing of thrusters 0 to 5 over an actuation period can be described as:

\[

T_a \begin{bmatrix}
F_x \\
F_y \\
F_z \\
M_x \\
M_y \\
M_z
\end{bmatrix} = \mathcal{H}_1 \begin{bmatrix}
F_{0u_0} \\
F_{1u_1} \\
F_{2u_2} \\
F_{3u_3} \\
F_{4u_4} \\
F_{5u_5}
\end{bmatrix} = \begin{bmatrix}
f_{0x} & f_{1x} & \cdots & f_{5x} \\
f_{0y} & f_{1y} & \cdots & f_{5y} \\
\vdots & \vdots & \ddots & \vdots \\
m_{0x} & m_{1x} & \cdots & m_{5x}
\end{bmatrix} \begin{bmatrix}
F_{0u_0} \\
F_{1u_1} \\
F_{2u_2} \\
F_{3u_3} \\
F_{4u_4} \\
F_{5u_5}
\end{bmatrix}
\]

Thus, the equation 4.4 puts forward a relationship between the command and the thrusters on-times. Considering \( \overline{F}_c \) and \( \overline{M}_c \), the commanded forces and torques, we indeed have:

\[

\begin{bmatrix}
F_{0u_0} \\
F_{1u_1} \\
F_{2u_2} \\
F_{3u_3} \\
F_{4u_4} \\
F_{5u_5}
\end{bmatrix} = T_a \mathcal{H}_1^{-1} \begin{bmatrix}
F_{c_1} \\
F_{c_2} \\
F_{c_3} \\
M_{c_1} \\
M_{c_2} \\
M_{c_3}
\end{bmatrix}
\]

But we should not forget here that we are seeking for thruster opening times, that by definition cannot be negative. All the \( F_i \) being positive here, nothing can guarantee us that the equation 4.5 will output positive firing times . . . We still have 6 more thrusters at our disposal though, and remembering Table 4.1, these last 6 thrusters happen to be those that are almost directly opposed to the ones we have considered so far. Besides, it also exists a matrix \( \mathcal{H}_2 \), built the same way as \( \mathcal{H}_1 \)—and thus entailing the same risk of getting negative firing times—such that:

\[

\begin{bmatrix}
F_{6u_6} \\
F_{7u_7} \\
F_{8u_8} \\
F_{9u_9} \\
F_{10u_{10}} \\
F_{11u_{11}}
\end{bmatrix} = T_a \mathcal{H}_2^{-1} \begin{bmatrix}
F_{c_4} \\
F_{c_5} \\
F_{c_6} \\
M_{c_4} \\
M_{c_5} \\
M_{c_6}
\end{bmatrix}
\]

What has been done then, is the building of the 12\( \times \)6 “mixing matrix” \( \mathcal{M}_1 \) as following:

\[

\mathcal{M}_1 = \begin{pmatrix}
\mathcal{H}_1^{-1} \\
\mathcal{H}_2^{-1}
\end{pmatrix}
\]

27
Then, and according to the equations 4.5 and 4.6 we can obtain a relation between the commanded forces and torques and what would be the virtual on-times of the thrusters, if they could be commanded on for a negative time:

\[
\begin{bmatrix}
F_0u_0 \\
F_1u_1 \\
\vdots \\
F_{11}u_{11}
\end{bmatrix} = T_a M_1
\begin{bmatrix}
F_{cx} \\
F_{cy} \\
F_{cz} \\
M_{cx} \\
M_{cy} \\
M_{cz}
\end{bmatrix}
\]  \tag{4.8}

Now, the negative firing times problem is almost overcome. Since our system is redundant, the odds are big that if a negative firing time is output by the equation 4.8 for a thruster \(i < 6\), a positive firing time is also output for the thruster \(i + 6\): a negative force asked to thruster \(i\) can be almost compensated by asking a positive force to thruster \(i + 6\).

Almost, but not exactly. This is the reason why a simple loop algorithm has been used for determining the actual thrusters firing times \(u_i\) from the commanded force and torque \(\vec{F}_c\) and \(\vec{M}_c\):

1. The virtual on-times \(u'_i\) are determined from equation 4.8;
2. The real on-times \(u_i\) are determined by canceling out the thrusters that have a negative virtual on-time \(u'_i\);
3. Force \(\vec{F}_a\) and torque \(\vec{T}_a\), that would be produced with these firing durations \(u_i\), are computed from equation 4.4, and a similar equation for thrusters 6 to 11;
4. The errors \(\vec{F}_e = \vec{F}_c - \vec{F}_a\) and \(\vec{T}_e = \vec{T}_c - \vec{T}_a\) are computed;
5. The on-times \(u_{e_i}\), necessary to compensate the error, are determined as in (1)-(2);
6. The final commanded on times are \(u_i + u_{e_i}\).

By looping once, the results provided by this algorithm are excellent. For instance, by commanding a force of 0.15 N for 1 s on \(x\)-axis, the output firing
4.2 Mixer and thrusters imperfections

times are (in milliseconds):

\[
\begin{bmatrix}
821.6 \\
862.1 \\
2.8 \\
9.4 \\
0 \\
0 \\
0 \\
0 \\
0.2 \\
0.6 \\
11.8 \\
15.6
\end{bmatrix}
\]  
\quad (4.9)

Which gives us an actual produced impulse of (in N.s):

\[
T_a \begin{bmatrix}
F_x \\
F_y \\
F_z \\
M_x \\
M_y \\
M_z
\end{bmatrix} = \begin{bmatrix}
0.15 \\
0 \\
0.002 \\
0 \\
0 \\
0
\end{bmatrix}
\]  
\quad (4.10)

Which means that the output impulse theoretically exactly matches the command. For a comparison, the impulse that would have been produced by using the same algorithm without looping, i.e. by stopping after step (2), would be:

\[
T_a \begin{bmatrix}
F_x \\
F_y \\
F_z \\
M_x \\
M_y \\
M_z
\end{bmatrix} = \begin{bmatrix}
0.15 \\
0 \\
0 \\
0.002 \\
0 \\
0
\end{bmatrix}
\]  
\quad (4.11)

While the impulse that would have been produced by using the old mixer
Correction of thrusters induced errors

would be:

\[
T_a \begin{bmatrix}
F_x \\
F_y \\
F_z \\
M_x \\
M_y \\
M_z \\
\end{bmatrix} = \begin{bmatrix}
0.134 \\
-0.0009 \\
-0.0023 \\
0 \\
0.0004 \\
0 \\
\end{bmatrix}
\]  \hspace{1cm} (4.12)

4.2.2 Preliminary results

Without actually testing it in the testbed, we can theoretically obtain some interesting results with this new mixer A, that then takes into account the thrusters misalignments and their different strength values.

First, for a simple force command that has the direction of one of the vehicle primary axis, it appears that the mixer A asks for firing more than the old one. Indeed, for the example discussed above, and back on equation 3.5, the old mixer would only have commanded thrusters 0 and 1 on, while the mixer A, as shown on equation 4.9, commanded 8 thrusters on\textsuperscript{2}.

This feature can be easily explained: since the primary thrusters—here 0 and 1—are inducing undesirable secondary forces and torques when opened, other thrusters have to fire to compensate this imperfection. This is the reason why thrusters 2,3,8,9,10,11 were actuated.

What also appears is that the error made by the old mixer on the secondary axis—here \(F_y\), \(F_z\) and \(M_y\)—are not significant: about 1\% of the \(F_x\) primary impulse. However, by over-estimating the force of the thrusters, the old mixer makes an error of almost 20\% on the primary axis impulse—the impulse on the axis it was commanded. This is obviously the main source of error here, and this requires further investigation.

4.2.3 Tests

This new mixer A, taking into account thrusters misalignments and differences in thrusters strengths, has been tested in conditions similar to flight: a sphere is mounted on the load cell and is asked to provide forces and torques of different magnitudes and directions for an actuation period.

This means that forces and torques input the mixer, exactly as if they were output by the controller. The load-cell and SigLab are used to measure

\textsuperscript{2}We do not care yet about the delays that prevent from getting short opening durations.
4.2 Mixer and thrusters imperfections

the forces and torques time functions created by the sphere. The SigLab data storage capacity is limited to 8192 samples, which represent approximately 160 ms of data at a 51200 hz sample frequency (a 20 kHz bandwidth) and 8 s of data at a 1024 hz sample frequency (a 400 hz bandwidth).

The sample frequency that has been used for all the tests that are discussed in this chapter and the following is 1024 hz, which correspond to a 400 hz bandwidth and a capacity of running 8 s tests. Thus, we cannot see the effects of impulses shorter than 2.5 ms. To make sure that the tests would show the small corrections that the mixer A is expected to provide, the commanded impulses are then set according to the preceding example and the equation 4.9: main corrections should thus be visible.

The produced impulse is then easy to compute by integrating the measured forces and torques over the control period. Thus, comparisons of the old mixer and the mixer A have been made, about:

- precision on primary axis;
- precision on secondary axis;
- fuel consumption.

What we have just shown though, is that the old mixer had a tendency to over-estimate the thrusters forces, which led to a smaller impulse in the primary axis. What has also been done then, is that the impulses commanded to the old mixer have been over-estimated, to compensate for this error. This way, what are compared are 2 mixers that produce the almost exact same primary axis impulse: comparing their errors on secondary axis and their fuel consumption thus becomes meaningful. Once again, the results have been averaged out over many tests sessions.

What appeared during these tests, as shown on Figure 4.2, is that the mixer A, then taking into account thrusters misalignments and different thrusters forces, is more accurate. The Figure 4.2 shows it with respect to $F_x$, $F_y$, $F_z$ and $M_z$ commanded impulse: the undesirable impulses created by mixer A sometimes are 50% lower than those of the old mixer. Also, the errors that show up here with this new mixer—although the theory did not mention them—are due to the fact that first, the thrusters are controlled at 1 kHz, meaning that they can only fire for integer durations (in milliseconds), and second, the delays we talked about earlier prevent from getting short “compensation” firing durations, thus limiting the efficiency of the mixer. Besides, as mentioned earlier, our measurement precision is not infinite.
Correction of thrusters induced errors

Figure 4.2: Secondary axis errors for old mixer and mixer A (absolute values). For both mixers, the primary produced impulse are the same. The bars represent an average on 10 tests. For each of them, the standard deviation is less than 10% of the considered unit. The tank use—index corresponding to the amount of propellant used—indicates an increase of 9% to 10% of fuel consumption for the mixer A compared to the old mixer.
4.3 Considering only different thruster forces

However, by taking a closer look, these errors we are looking at correspond to, at most, 2% of the impulse commanded on the primary axis for the old mixer. This is better than our overall mixer accuracy, discussed in section 3.1.3. And even if the mixer A shrinks these drifts to, at most, 1%, it entails an increase of the fuel consumption of 9%.

Besides, the measurement of the computation time gives, in average, 0.003 milliseconds\(^3\) for the old mixer, and 0.011 milliseconds for the mixer A. Thus, even if the new mixer is not overloading the sphere CPU, it uses more computational power.

For this two reasons, it has been decided not to follow through the development of this mixer A, that seems to bring small improvements at a rather huge cost. However, what we have learned might be useful for improving the old mixer and having it taking into account the actual thrusters forces—what has been tacitly made for the tests. This way, we could reduce the main error made by the old mixer.

### 4.3 Considering only different thruster forces

By only taking into account the differences in thrusters forces, and preserving the 6×6 mixing matrix described in section 3.1.2, we are going to suppress the "compensation" firings introduced by considering the thrusters misalignments, and we should thus limit the fuel consumption.

#### 4.3.1 Implementation

Since we do not care about thrusters misalignments anymore, then we can arrange the thrusters in pairs again. By calling \( p_i \) the impulse commanded for the thrusters pair \( p_i \), we thus have:

\[
\begin{bmatrix}
  p_{p0} \\
  p_{p1} \\
  p_{p2} \\
  p_{p3} \\
  p_{p4} \\
  p_{p5}
\end{bmatrix} = T_a \times \mathcal{M} 
\begin{bmatrix}
  f_{x_c} \\
  f_{y_c} \\
  f_{z_c} \\
  \frac{m_{xx}}{m_r} \\
  \frac{m_{yy}}{m_r} \\
  \frac{m_{zz}}{m_r}
\end{bmatrix}
\]

\( (4.13) \)

---

\( ^3\)This computation has been made on Visual C++, which outputs a result measured in \( \frac{1}{\text{ClocksPerSecond}} \), that is milliseconds.
Correction of thrusters induced errors

From this equation, it is simple to determine the thrusters firing times following:

\[
\forall i \in [0, 5] \left\{ \begin{array}{ll}
u_i &= \frac{p_{p_i}}{F_i} & \text{if } p_{p_i} > 0 \\
0 & \text{otherwise}
\end{array} \right.
\]

\[
u_{i+6} = \left\{ \begin{array}{ll}
0 & \text{if } p_{p_i} > 0 \\
-\frac{p_{p_i}}{F_{i+6}} & \text{otherwise}
\end{array} \right.
\]

From now on, we are thus going to define the “new” mixer, or mixer 1, as the original mixing matrix \( M \) associated with the equation 4.13 and 4.14, that take into account the exact force of the thrusters.

4.3.2 Tests

The exact same tests were conducted as those described in section 4.2.3: impulses were commanded to old mixer and mixer 1, still over-estimating the ones commanded to the old mixer, and the produced impulses on primary and secondary axis were compared, as well as the fuel consumption.

What stood out from these tests and can be seen on Figure 4.3, as far as the secondary axis precision, is that for the same primary axis produced impulse, those two mixers are almost indistinguishable: the order of magnitude of the standard deviation does not allow to discriminate between one and another.

This makes sense: by using the 6×6 mixing-matrix, we prevent the firing of secondary axis thrusters. Thus, the only difference concerns the firing times of the two primary axis thrusters. But by over-estimating the impulse commanded to the old mixer, we implicitly erase this difference, and “level” the firing times between the old mixer and the mixer 1. For this reason, no difference is seen in the fuel consumption.

Besides, almost no additional computation is needed: mixer 1 uses the exact same algorithm as the old mixer, and they only differ on the instant where a division is made.

It could be objected though, that mixer 1 should produce less undesirable torque than the old mixer. Indeed, looking back at Figure 3.1 on page 16, the old mixer should create a positive \( M_{x_p} \) secondary torque when commanded a positive \( F_z \) force\(^4\), if thruster 4 is stronger than thruster 5—as it is the case according to the Table 4.1. Also, the mixer 1 should correct this error, since it considers this force difference.

\(^4\)Which would entail firing of thrusters 4 and 5.
4.3 Considering only different thruster forces

Figure 4.3: Secondary axis errors for old mixer and mixer 1 (absolute values). The propellant use is now the same for both mixer.
Actually, and considering that thrusters 4 and 5 moment arm around $x$-axis is approximately $r = 10 \text{ cm}$, the undesirable torque impulse created during a $t = 800 \text{ ms}$ firing duration (according to what has been developed earlier) is:

$$M_{x} \cdot t = (F_{4} - F_{5}) \cdot r \cdot t$$

$$= 1.36 \cdot 10^{-4} \text{ N.m.s}$$

This is one order of magnitude less than the measurements shown in Figure 4.2, and also unfortunately less than the standard deviation on our tests average. We were not really able to see this error then—neither its correction.

So, does the fact that the old mixer and the mixer 1 are comparable mean that they are as good? Of course not, because we mentioned that they were comparable for a same primary axis produced impulse. This means that, as long as the produced primary impulses are the same, the mixer behave almost the same. The point is that to get the same primary impulses, the command of the old mixer has been over-estimated compared to the command of the mixer 1. We should then also take a look at the differences between the impulse commanded and the impulse got on the primary axis.

What appears then, shown in Figure 4.4, is that for the old mixer, the impulse on the primary axis consistently is around 25% lower than the command, while it is around 10% lower for the mixer 1. The mixer 1 thus shows an improvement here, but since it takes into account the actual thrusters forces, it was expected to exactly match the command.

The undesirable forces shown on Table 4.1, that “weaken” the thrusters on their primary axis, are actually too small to explain this 10% drop. We should thus look for its cause somewhere else: it is the purpose of the next chapter.

4.4 Outcome

But before focusing on the thrusters environment, let us sum up in a table what has been done here. This is what is shown on Table 4.2.

To summarize, correcting the thrusters misalignments entails an improvement of the secondary axis precision, as well as an increase of the fuel-consumption. Since the error made by the old mixer and the $6 \times 6$ mixing matrix are acceptable (around 1% error on secondary axis), it might not be worth it to spend more propellant to correct it, as mixer A does.
4.4 Outcome

Figure 4.4: Primary axis precision for old mixer and mixer 1. Impulses were commanded on different directions $F_x$, $F_y$, $F_z$, and the created primary impulses were measured.

Table 4.2: Comparison between the discussed mixers. The number of symbols ✓ in a cell indicates whether the considered mixer is good in this category.

<table>
<thead>
<tr>
<th></th>
<th>Old mixer</th>
<th>Mixer A</th>
<th>Mixer 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel consumption</td>
<td>✓✓</td>
<td>✓</td>
<td>✓✓</td>
</tr>
<tr>
<td>Computation time</td>
<td>✓✓✓</td>
<td>✓</td>
<td>✓✓✓</td>
</tr>
<tr>
<td>Primary axis precision</td>
<td>✓</td>
<td>✓✓</td>
<td>✓✓</td>
</tr>
<tr>
<td>Secondary axis precision</td>
<td>✓</td>
<td>✓$\frac{2}{3}$</td>
<td>✓</td>
</tr>
</tbody>
</table>
Correction of thrusters induced errors

On another hand, considering the actual thrusters forces implies an improvement of primary axis accuracy, and no additional cost: same fuel consumption, computation time or secondary axis precision. For this reason, we are now only going to follow through the development of the mixer 1, and look for a way to improve it over.
Chapter 5

Physical characteristics of thrusters

In this chapter, we are only considering the errors induced by the physical characteristics of the thrusters: the gas supply system, and the on/off command electronic system. What we are going to show is that it is possible to improve mixer 1 so that the output impulse it produces on the primary axis matches the commanded, without adding too much computation time or fuel consumption.

As such, we are focusing on the two known causes of error: the reduction in force produced by each thruster when several thrusters are firing, and the difficulty in getting short firing time due to non-zero times of response.

5.1 Force drop due to multiple thrusters firing

Since each SPHERES satellite only disposes of a single CO$_2$ gas tank, opening multiple thrusters at a time is likely to affect the value of the force produced by each one.

Indeed, firing multiple thrusters simultaneously will likely deplete the gas capacitor and make the gas pressure drop in the supplying pipes. And a propulsive force depending on the pressure of the expelled gas, the force produced by each thruster may be expected to drop if several thrusters are open. What needs to be done then, is to quantify this thrust loss, and to design an algorithm that takes it into account in the mixer.
5.1.1 Evaluation of the attenuation factor on the load cell

Test setup

The same test setup was used as the one described in section 4.1, page 23. But this time, instead of having the thrusters firing one at a time, multiple thrusters were opened at the same time for one second.

What has been measured then is the force of each thruster on its primary axis, considering the additional opening of 0 to 6 other thrusters. Of course, attention had to be paid to the fact that the additional opened thrusters could not have the same primary direction as the one that was measured. For instance, and considering the Figure 3.1, additional thrusters 1,7,0,6,3,9 were opened while measuring the $F_z$ force produced by thruster 4.

In this particular case, this means that for the measurement, the $F_z$ undesirable force created by the thrusters 1,7,0,6,3,9 has been neglected compared to the $F_z$ force produced by thruster 4 on what is its primary axis, which could lead to a loss of precision.

However, Table 4.1 indicates that these undesirable forces are two orders of magnitude below the primary forces. We should be allowed to neglect them then, although it also means that we should not expect an accuracy better than 1% in our measurements.

Results

Once more, the results were averaged out over 10 tests sessions. What showed up is that no matter which thruster was considered, or which “secondary” thrusters were opened, an obvious loss of thrust appeared on the primary axis during the multiple openings.

Considering for example the thruster 4\(^1\) of the sphere 4, the results are shown in Table 5.1.

No tests were conducted with more than 7 thrusters open at the same time, for the following reasons. First, it seemed unlikely—at least in 2D experiment—that a vehicle would have to open as many thrusters simultaneously. Besides, considering that while measuring the force drop for a particular thruster, the 3 other thrusters that have the same primary axis are not allowed to fire, we are left with 8 additional thrusters that can be

---

\(^1\)Which primary direction is the $z$-axis.
5.1 Force drop due to multiple thrusters firing

Table 5.1: Force drop due to several thrusters open shown for thruster 4 of sphere number 4. The z-axis here is the primary direction of the considered thruster. The last line $\frac{F_n}{F_0}$ indicates the ratio between the force produced with $n$ other open thrusters $F_n$ and the “alone” produced force $F_0$.

<table>
<thead>
<tr>
<th>Number of additional thrusters opened</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_z$ (N)</td>
<td>0.1224</td>
<td>0.1179</td>
<td>0.1104</td>
<td>0.102</td>
<td>0.0912</td>
<td>0.0827</td>
<td>0.0652</td>
</tr>
<tr>
<td>Drop (%)</td>
<td>0</td>
<td>3.7</td>
<td>9.8</td>
<td>16.7</td>
<td>25.5</td>
<td>32.4</td>
<td>46.8</td>
</tr>
<tr>
<td>$\frac{F_n}{F_0}$</td>
<td>1</td>
<td>0.96</td>
<td>0.90</td>
<td>0.83</td>
<td>0.75</td>
<td>0.66</td>
<td>0.53</td>
</tr>
</tbody>
</table>

opened. We stopped at 6 because following through the experiment did not seem to be worth it at the time, considering the relatively good estimation that we already had.

Indeed, for a particular thruster $i$ that produce alone the force $F_{i_0}$ on its primary axis, calling $n$ the number of other thrusters that are opened at the same time and $F_{i_n}$ the force produced by thruster $i$ on its primary axis while $n$ other thrusters are open, we can write:

$$F_{i_n} = \alpha_n \times F_{i_0}$$ (5.1)

Where the attenuation factor $\alpha_n$ is defined as follows, according to Table 5.1 and other measurements for other thrusters and other spheres:

$$\forall n \in [1, 11], \ \alpha_n = \begin{cases} 0.96 & \text{if } n = 1 \\ 0.90 & \text{if } n = 2 \\ 0.83 & \text{if } n = 3 \\ 0.75 & \text{if } n = 4 \\ 0.66 & \text{if } n = 5 \\ 0.53 & \text{if } n \geq 6 \end{cases}$$ (5.2)

We then have at our disposal a quantification of the force drop factor. What needs to be done now, is the implementation of a correction for this effect in the mixer.

5.1.2 Implementation on SPHERES

We mentioned earlier that on the SPHERES vehicles, the thrusters impulses were centered in the middle of the actuation period. Here, we are then going
to describe what this implies, and how to take into account the force drop in
the mixer.

Considering that two thrusters are commanded on for a different duration,
let us rename them according to their on-time: thruster A being the one that
fires for the shortest duration. Let us also name $t_1$ and $t_2$ the on-times. The
Figure 5.1 shows us the arrangement of the firings over the actuation period.

![Figure 5.1: Firing schedule for two thrusters. The bold lines represent
the opening duration of the thrusters, centered in the middle of the
actuation period $T_a$](image)

Considering the quantities that we previously introduced, and since one
other thruster is open for the whole time of the thruster 1 firing, it is then
obvious that the efficient force impulse produced by the thruster 1 on its
primary axis is:

$$\mu_i = t_1 F_{10} \alpha_1$$  \hspace{1cm} (5.3)

Which is equivalent to the force impulse that would be produced by an
imaginary thruster having the same direction as thruster 1, the same “alone”
produced force $F_{10}$, and that would fire during the efficient time $t_{1e}$:

$$t_{1e} = t_1 \alpha_1$$  \hspace{1cm} (5.4)

The same way, it is simple to get the efficient on-time for the thruster 2:

$$t_{2e} = t_1 \alpha_1 + (t_2 - t_1)$$  \hspace{1cm} (5.5)

Then, a way to actually obtain efficient opening times of $t_1$ and $t_2$ would
be to command the over-estimated times $t_{1e}$ and $t_{2e}$ such as:

\[
\begin{align*}
   t_{1e} &= \frac{t_1}{\alpha_1} \\
   t_{2e} &= t_{1e} + (t_2 - t_1)
\end{align*}
\] (5.6)

Indeed, and thanks the equation 5.4 and 5.5 that map the effective on-time to the physical on-time, we would thus get:

\[
\begin{align*}
   t_{1e} &= t_{1e} \alpha_1 = t_1 \\
   t_{2e} &= t_{1e} \alpha_1 + (t_2 - t_{1e}) = t_1 + (t_{1e} + (t_2 - t_1) - t_{1e}) = t_2
\end{align*}
\] (5.7) (5.8)

So, what we have just proven is that if mixer 1, described in the preceding chapter, asks for the opening of two thrusters, with a firing duration of $t_1$ and $t_2$, $t_1 < t_2$, what should be commanded are $t_{1e}$ and $t_{2e}$ described on system 5.6. This way, the actual output impulse should match the command from the controller.

Now, and to follow through this development, let us consider what would happen if three thrusters were commanded open, as shown on Figure 5.2. It is clear that for all the thrusters, the relationship mapping the efficient on-times to the physical on-times would now be:

\[
\begin{align*}
   t_{1e} &= t_{1e} \alpha_2 \\
   t_{2e} &= t_{1e} + (t_2 - t_1) \alpha_1 \\
   t_{3e} &= t_{2e} + (t_3 - t_2)
\end{align*}
\] (5.9)
Physical characteristics of thrusters

Which means that to actually obtain \( t_1 \), \( t_2 \) and \( t_3 \) as efficient opening durations, the over-estimated commanded on-times should now be:

\[
\begin{align*}
    t_{1c} &= \frac{t_1}{\alpha_2} \\
    t_{2c} &= t_{1c} + \frac{(t_2 - t_1)}{\alpha_1} \\
    t_{3c} &= t_{2c} + (t_3 - t_2)
\end{align*}
\]

(5.10)

This allows us to find a recursive relationship between the commanded firing times, considering that a total of \( n \) thrusters are commanded on during the control period, and that they are sorted by growing mixer 1-commanded firing durations \( t_i \):

\[
\begin{align*}
    t_{1c} &= \frac{t_1}{\alpha_{n-1}} \\
    t_{2c} &= t_{1c} + \frac{t_2 - t_1}{\alpha_{n-2}} \\
    \ldots \\
    t_{ic} &= t_{i-1c} + \frac{t_i - t_{i-1}}{\alpha_{n-i}} \\
    \ldots \\
    t_{nc} &= t_{n-1c} + (t_n - t_{n-1})
\end{align*}
\]

(5.11)

Thus, considering that the mixer 1 outputs the opening durations \((u_i)_{0 \leq i \leq 11}\), what needs to be done is a sorting of these durations, and then a computation of the over-estimated times that should be commanded to get an impulse equivalent to these \( u_i \) firing times.

The algorithm that is used on SPHERES for this purpose is the following, considering the input \((u_i)_{0 \leq i \leq 11}\) given by the mixer 1:

1. Sorting of the \( u_i \) firing times thanks a primary “merge sort” algorithm.

The merge-sort method is optimal: its complexity is in \( O(n) \)—\( n \) being the size of the array that has to be sorted. It thus enables to perform the sorting without using too much resources.

2. Computation of the over-estimated on-times using the system 5.11.

Let us then call mixer 2 this mixer that takes into account the different thrusters forces and the force drop due to multiple firings, as shown on Figure 5.3. Now that its mechanism has been explained, let us have a look at its comportment on SPHERES.
5.1 Force drop due to multiple thrusters firing

Figure 5.3: Explanation of the described mixers

5.1.3 Tests of mixer 2

Again using the load cell, the sphere was mounted the exact same way, and the mixer was asked to provide different kind of impulses over an actuation period, as if this command had been output by the controller.

This time though, instead of commanding one primary impulse at a time—and then having 2 thrusters firing simultaneously at the most—we commanded different primary impulse for a single actuation period: force on $F_x$, on $F_y$, and torque on $M_z$ for instance. This way, the effect of the force drop correction is expected to be outstanding. We compared here two different things:

1. The old mixer and mixer 2 with the exact same command, so that we can quantify the probable gain of precision on the primary axis impulse.

2. The old mixer and mixer 2 with the same produced primary impulse, so that we can quantify the differences in fuel consumption and in second-axis errors. As explained before, the command that inputs the old mixer has to be over-estimated to allow this comparison.

Since mixer 2 still uses the $6 \times 6$ mixing-matrix though, no noteworthy difference is expected to be seen neither on the secondary axis errors nor on the fuel consumption. However, what constituted the main error of the old mixer and mixer 1—the error on primary axis—will be reduced.
Fuel consumption and errors in secondary axis

Over the several tests that were executed, no difference stood out between the old mixer and mixer 1, as far as secondary axis accuracy and fuel consumption.

Again, this is explained by the $6 \times 6$ mixing matrix that both old mixer and mixer 1 use: no secondary thrusters are opened to correct errors due to primary thrusters, entailing no improvement of secondary axis accuracy, as well as no increase of fuel consumption for the same primary axis impulse produced.

Errors in primary axis

However, the results in the primary axis impulses are very interesting. Considering Figure 5.4 (data for Sphere S/N #4), it is clear that mixer 2 brings in a substantial gain in precision on the primary axis impulse, no matter how many thrusters are on.

Mixer 2 thus seems to produce impulses that almost match the commanded input. As to the remaining errors that are left on the primary axis, they can be due to:

- measurements errors: our measurements are not perfect and we cannot be perfectly sure that what is plotted here is what exactly happened, considering our standard deviation;

- undesirable effects due to thrusters delays, that still need to be corrected.

If we look at the computation time though, we can see that the execution of this algorithm now lasts 0.007 ms, versus 0.003 ms for the old mixer, which represents a significant increase. It still is less than a millisecond though. The spheres being controlled at 1 KHz, running this mixer with a sphere in a flight condition test is possible. As such, the algorithm that that has been used for taking into account the force drop (mixer 2) might still be improved. It is presented in appendix C in this purpose.

Also, we can now consider the following table, that compares the mixers we have been working on so far.

What remains to be done then, is the correction of the delays of the thrusters, and their non instantaneous response to a command.
5.1 Force drop due to multiple thrusters firing

![Force drop due to multiple thrusters firing](image)

**Figure 5.4**: Comparison between old mixer and mixer 2. The table gives the commanded impulse according to the test number. The bars represent an average over 10 tests, and the standard deviation is approximately 2%.

**Table 5.2**: Comparison between the discussed mixers. The number of symbols ✓ in a cell indicates whether the considered mixer is good in this category.
5.2 Thruster delays, actuation and shut-off durations

We now have mixer 2 that seems to have a really good accuracy on primary axis for long commanded impulses—approximately 450 ms firing duration for the ones described before. However, we have not yet cared about the last known source of errors: delays at actuation and shut-off induced by the solenoids deadbands, and non-instantaneous rise and shut-off durations.

The work that has been done on this aspect so far assumes an order of magnitude of 10 ms delay for each thruster. Considering the tests we have been conducting—the commanded on times are longer than 400 ms—this delay would introduce less than 2% error on the primary impulse measurement: it is understandable that nothing has been noticed so far. However, this undesirable effect exists and has to be quantified.

5.2.1 Measuring the delays

Procedure

Obviously, our 400 hz bandwidth will be able to measure the 1 KHz effect we are interested in. So, for these tests we will be using the maximum SigLab sample frequency: 51.2 Khz, which gives us a 20 Khz bandwidth.

At this frequency, the longest test duration we can get is 160 ms. For the tests that had been conducted so far, SigLab was manually triggered after the sphere mounted on the load cell entered its “test” state—after the laptop commanded the vehicle to start the test. Considering now a 160 ms window for commanding the sphere to run the test and triggering SigLab, the test protocol is more complicated.

Therefore, two spheres have been used for these tests:

A trigger sphere, that is only used to trigger the data recording. A simple connection has been made between a SigLab “trigger” channel and the sphere FPGA: at the instant a specific thruster is commanded on, this actuation signal is sent to SigLab and used to trigger the measurement.

A load cell sphere, that is mounted on the load cell and used to actually measure the thrusters properties. The SPHERES testbed allows a synchronization better than 1 ms between different vehicles, so a thruster
5.2 Thruster delays, actuation and shut-off durations

on the load cell sphere can easily be commanded on 3 ms after the ignition signal has been sent by the trigger sphere, once the measurement has started. This timing sequence is illustrated on Figure 5.5.

![Figure 5.5](image)

**Figure 5.5**: Timing sequence for measurement of ignition delay. The trigger sphere and the load cell sphere are synchronized prior to what is shown here. The actual SigLab data recording is represented by the bold line.

**Results**

The raw data that we get from SigLab are excessively noisy. First, there obviously are residual vibration in our test stand: by only mounting a sphere on the load cell, without firing any thrusters, SigLab records an harmonic oscillation which frequency is between 20 and 30 hz.

Besides, firing a thruster constitutes without a doubt an impulsive shock for the sphere and the load cell, and shall thus excite various modes of the test stand.

However, a Fourier analysis of the raw data shows us pretty clearly the frequency of these modes, the most important ones being around 30, 150, and 240 hz. These modes are due to vibrations inherent to the testbed: they show up even if the sphere stays still on the load cell. Through Matlab, band stop filters have then been designed to remove these frequency contents.

Figure 5.6 shows us an example of what has been obtained for thruster number 0 of sphere number 2. The tests have been conducted at a 35 psi tank gas pressure, and the results are pretty consistent for the 3 different thrusters that have been tested. The results are presented on Table 5.3.

What appears then, is a 8 ms delay between the command and the actual actuation, probably due to the response of the solenoid. Then, the thruster reaches its steady-state force in approximately 2.4 ms. It is thus fully active approximately 10.4 ms after the command. But by this time, since the rise is not instantaneous, it already has produced an impulse of $1.18 \times 10^{-4}$ N.s.
Figure 5.6: Thruster rise time, raw data and filtered data. The bottom plot is a zoom on the ignition zone.

Table 5.3: Timing sequence from the instant a thruster is commanded on, based on the several tests that have been performed

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Event</th>
<th>Produced Impulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td>Actuation command</td>
<td>0</td>
</tr>
<tr>
<td>$t = 8$</td>
<td>Actual actuation</td>
<td>0</td>
</tr>
<tr>
<td>$t = 10.4$</td>
<td>Steady-state force</td>
<td>$1.18 \cdot 10^{-4}$ N.s</td>
</tr>
</tbody>
</table>
5.2 Thruster delays, actuation and shut-off durations

Considering the thruster force, this is equivalent to a 1.18 ms opening time at a steady state force.

We would thus have a pretty good model of our thruster actuation by considering that:

1. The thruster does not produce any force from $t = 0$ (ignition command) to $t = 9$ ms.

2. At $t = 9$ ms, the thruster starts instantaneously to produce its steady-state force.

The exact same kind of tests have been performed to obtain a correct approximation of the thruster shut-off dynamics, the results are presented in Figure 5.7 and Table 5.4.

**Table 5.4:** Timing sequence from the instant a thruster is commanded off, based on the several tests that have been performed

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Event</th>
<th>Produced Impulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td>Shut-off command</td>
<td>0</td>
</tr>
<tr>
<td>$t = 4.5$</td>
<td>Actual shut-off</td>
<td>$4.5 \times 10^{-4}$ N.s</td>
</tr>
<tr>
<td>$t = 6.5$</td>
<td>No more force</td>
<td>$5.7 \times 10^{-4}$ N.s</td>
</tr>
</tbody>
</table>

Thus, there also is an approximately 4.5 ms delay between the shut-off command and the instant where the thruster actually “starts” to shut down. Then, it takes 2 ms for the thruster to completely turn off. During this descent, it produces an impulse of $1.2 \times 10^{-4}$ N.s, which would be equivalent to 1.2 ms opening time at a steady state force.

Similar to the thruster ignition, we can then model the extinction as follows:

1. The thruster keeps producing the steady state force from $t = 0$ (extinction command) to $t = 5$ ms.

2. At $t = 5$ ms, the thruster instantaneously shuts off.
Physical characteristics of thrusters

Figure 5.7: Thruster shut-off time, raw data and filtered data. The bottom plot is a zoom on the extinction zone.
5.2 Thruster delays, actuation and shut-off durations

5.2.2 Implementation on SPHERES

This effect is simple to implement on the SPHERES mixer.

Indeed, considering what would be called the IGNITION_DELAY (9 ms here, according to the previous development) as well as the EXTINCTION_DELAY (5 ms), and naming \((u_i)_{0 \leq i \leq 11}\) the opening times output by the mixer 2, the algorithm we use is the following:

```c
for (i = 0; i < 12; i++)
{
    if (u(i) >= EXTINCTION_DELAY)
        u(i) = u(i) + IGNITION_DELAY - EXTINCTION_DELAY;
    else
        u(i) = 0;
}
```

Since we obviously can not get opening times shorter than the extinction delay.

5.2.3 As a conclusion on this subject

This model of the thrusters ignition and extinction obviously will enable to gain accuracy in the mixer: by taking into account the delays that were discussed and correcting them, the effective firing duration of the thrusters will now match the command output by the mixer.

However, by canceling out all commands shorter than 5 ms, our capacity to produce short impulse in a given direction still is limited. A way to overcome this problem, that has been evoked but neither tested nor implemented, would be to fire opposite thrusters for slightly different times—2 or 3 ms or so. This way, the resulting impulse would be equivalent to the one produced by a thruster that has been fired for a very short time. This could constitute a possible future work.
Chapter 6

Summary

A lot of concepts, errors, and way of corrections have been put forward so far, so it is the purpose of this chapter to summarize, and offer an overview of the results we got. Here we are first going to indicate the mixer that has been chosen, before focusing on its implementation on the SPHERES vehicles.

6.1 The choice of a mixer

As a reminder, the Table 6.1 that follows gathers all the features we have discussed so far that could improve the mixer precision, and their characteristics.

It shows us that, based only on accuracy considerations, the best mixer would be constituted by:

1. The 12×6 mixing-matrix, to improve secondary axis precision.
2. The consideration of thrusters actual forces, to improve primary axis precision.
3. The force drop correction algorithm.
4. The transitional regime correction algorithm.

And yet have we abandoned earlier the developments involving the 12×6 mixing matrix. Indeed, considering off-axis thrusters in the mixer entails a considerable increase of fuel consumption and computational time, and it only brings in a small secondary axis improvement. Besides, we have
Table 6.1: Features that have been discussed for mixer improvement, with their pros and cons.

<table>
<thead>
<tr>
<th>Features</th>
<th>Purpose</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>12×6 mixing matrix</td>
<td>Correct thrusters misalignments</td>
<td>Significant improvement for both primary and secondary axis</td>
<td>Big increase of the fuel consumption</td>
</tr>
<tr>
<td>Taking into account actual thrusters forces</td>
<td>Correct thrusters with different strengths</td>
<td>Improvement of the primary axis accuracy</td>
<td>No measured significant improvement on the second axis accuracy</td>
</tr>
<tr>
<td>Force drop correction algorithm</td>
<td>Correct force drop due to multiple thrusters firing simultaneously</td>
<td>Excellent experimental primary axis precision</td>
<td>Increase of the computation time</td>
</tr>
<tr>
<td>Transition correction algorithm</td>
<td>Correct effects due to thrusters transitional regime</td>
<td>Actual impulse matches the command</td>
<td>Still limited to impulses longer than 5 ms</td>
</tr>
</tbody>
</table>
6.2 Implementation on SPHERES

shown that the secondary axis errors made by the $6 \times 6$ mixing-matrix were acceptable, and for this reason it has been decided that spending more gas or computational time to correct them was not worth it. Therefore, the implementation of the $12 \times 6$ mixing matrix algorithm has not been followed through.

On another hand, considering the other features, the improvements they bring in seem to be worth the cost they impose. The final mixer that has been designed then contains the correction of different thrusters forces, the correction of force drop, and the correction of thrusters transitional regimes. It is simplified on the Figure 6.1.

![Figure 6.1: Mechanism of the finally designed mixer](image)

6.2 Implementation on SPHERES

6.2.1 Coding

A whole algorithm has been fully written and tested on the SPHERES testbed. The program that has been written uses the actual thrusters forces that have been programmed in the vehicles flash memory, so the code does not differ from one sphere to another.

All the scaling issues entailed by a maximum pulse duration, that have not really been discussed here, are also fully taken into account in the mixer. If a sphere is asked for more impulse than it can provide during an actuation
period\textsuperscript{1}, then all opening times are scaled by a factor $\frac{T_a}{u_{max}}$, $u_{max}$ being the longer commanded on-time. This way, no firing duration exceeds $T_a$ and the produced impulse has the same direction as the input. It has not the same intensity though, but this can be corrected at the next actuation period.

6.2.2 Testing

Load cell

The tests that have been conducted on the load cell with this mixer actually do not differ from the ones that have been described in the section 5.1.3 page 46, since only the transitional regime correction algorithm has been added since then.

This section thus can be consulted for whatever would concern the results on primary axis accuracy, secondary axis accuracy, fuel consumption or computation time.

We also know for a fact that the effects induced by the transitional regime correction algorithm are:

1. A small increase of fuel consumption, since an extra on time of 4 ms is commanded for each thruster.

2. An insignificant increase of time computation, regarding the total computation time for the mixer.

3. A gain of accuracy for primary axis impulse: the produced impulse now exactly matches the command.

6.3 Conclusion related to the mixer design

We thus have covered every known undesirable effects that could affect the impulse produced by the mixers: we have quantified these effects, and put forward a way to correct them. What has been shown then, is that a complex mixer, that would fully take into account all of these effects and correct them, can be developed.

However, this complex mixer would have a cost: fuel consumption and computation time costs. By considering every aspects of our testbed, what

\textsuperscript{1}If a thruster is asked for an impulse larger than $F_{thruster} \times T_a$. 

58
6.3 Conclusion related to the mixer design

has been done then is a choice, that optimizes the mixer performances versus its cost. It might not be perfect yet, but it is a significant improvement over the original mixer that had been used so far.
Appendix A

Mathematics for pulse modulation

A.1 Variation of spacecraft mass

In this section we are going to prove that for short actuation period, we can neglect the diminution of spacecraft mass by comparison with the other quantities involved in the spacecraft dynamics.

Considering a spacecraft defined by its mass \( m(t) \), its initial state \( S_0 \) (position vector \( \vec{P}_0 \), velocity \( \vec{v}_0 \)), that gets applied a propulsive force \( \vec{F}(\tau) \) between \( t_0 = 0 \) and \( t \), the equation 2.3 shows that:

\[
\vec{v}(t) = \vec{v}_0 + \int_{t_0}^{t} \frac{\vec{F}(\tau)}{m(\tau)} d\tau \tag{A.1a}
\]

We know though that for a propulsive force, writing \( q(t) \) the mass flow of expelled gas and \( \vec{V}_e(t) \) its relative speed we have:

\[
\vec{F}(t) = q(t) \vec{V}_e(t) \tag{A.1b}
\]

\[
m(t) = m_0 - \int_{t_0}^{t} q(\tau) d\tau \tag{A.1c}
\]

Calling \( \vec{v}_a(t) \) the approximate velocity assuming that the spacecraft mass does not change: \( \vec{v}_a(t) = \vec{v}_0 + \frac{1}{m_0} \int_{t_0}^{t} \vec{F}(\tau) d\tau \), we are going to determine
the error made by neglecting the mass variation.

\[
\| \vec{v}(t) - \vec{v}_a(t) \| = \left\| \int_0^t \frac{F(\tau)}{m_0^2 - m_0 \int_0^\tau q(\sigma) d\sigma} d\tau \right\| \tag{A.2a}
\]

\[
\leq \int_0^t \left\| \frac{F(\tau)}{m_0^2 - m_0 \int_0^\tau q(\sigma) d\sigma} \right\| \left| \int_0^\tau q(\sigma) d\sigma \right| d\tau \tag{A.2b}
\]

But because of obvious physical limits, the gas mass flow is bounded: \( \forall t \in \mathbb{R}, q(t) < Q \). Also, since the thrusters cannot provide infinite force, \( \| F(t) \| \) can be bounded by \( F \) on \([0, t]\). Then:

\[
\| \vec{v}(t) - \vec{v}_a(t) \| \leq \frac{FQ}{m_0 \int_0^t \left| \int_0^\tau q(\sigma) d\sigma \right| d\tau} \tag{A.2c}
\]

But the spacecraft cannot expel more gas than it contains. Meaning that \( \forall \tau \in \mathbb{R} \), with calling \( m_d \) the dry mass:

\[
|m_0 - \int_0^\tau q(\sigma) d\sigma| \geq m_d \tag{A.3}
\]

By replacing in equation A.2c it comes:

\[
\| \vec{v}(t) - \vec{v}_a(t) \| \leq \frac{FQ}{m_0 m_d} \int_0^t \tau d\tau \tag{A.4a}
\]

\[
\leq \frac{FQ}{2 m_0 m_d} t^2 \tag{A.4b}
\]

Using Landau notation, we have just proven that:

\[
\vec{v}(t) = \vec{v}_a(t) + O(t^2) = \vec{v}_a(t) + o(t) \tag{A.5}
\]

Considering then that the position is the integral of the velocity we get:

\[
\vec{P}(t) = \vec{P}_0 + \int_0^t \vec{v}(\tau) d\tau = \vec{P}_0 + \int_0^t (\vec{v}_a(\tau) + o(t)) d\tau = \vec{P}_a(t) + o(t^2) \tag{A.6}
\]
A.2 Proof of minimal-constrained actuation

What we have demonstrated is that considering very short actuation width, we can consider that the real temporal velocity and position of a vehicle correspond with the approximate ones, where we assume that its mass does not change. In the particular case of the SPHERES vehicles, considering $\vec{P}_0 = 0$, $\vec{v}_0 = 0$, $t = 250\text{ms}$, $m_0 = 4.2\text{Kg}$, an actuation consisting of a firing among the $x$ direction for $200\text{ms}$, meaning that, according to SPHERES physical parameters:

$$ \vec{F}(\tau) = \begin{cases} 
0.24 \vec{u}_x & \text{if } \tau \leq 200\text{ms} \\
0 & \text{if } \tau > 200\text{ms}
\end{cases} $$

$q(\tau) = \begin{cases} 
7.10^{-4} & \text{if } \tau \leq 200\text{ms} \\
0 & \text{if } \tau > 200\text{ms}
\end{cases}$

$\vec{F}$ being in Newton and $q$ in kilograms per second. It comes then:

$$ \vec{v}'(t) = 114.287 \times 10^{-4} \vec{u}_x \text{ (m.s}^{-1}) $$

$$ \vec{v}'_a(t) = 114.285 \times 10^{-4} \vec{u}_x \text{ (m.s}^{-1}) $$

$$ \vec{P}(t) = 228.574 \times 10^{-5} \vec{u}_x \text{ (m)} $$

$$ \vec{P}_a(t) = 228.570 \times 10^{-5} \vec{u}_x \text{ (m)} $$

The difference between actual values and approximate values being less than .001\% for a typical SPHERES actuation, it is clear than we can consider the spacecraft mass constant during the actuation period.

A.2 Proof of minimal-constrained actuation

We are going to show here that the piecewise constant function $\vec{F}$ defined as follows creates an impulse function $\vec{\mu}$ solution of the system 2.6.

$$ \vec{F} : \begin{cases} 
[0, t] & \longrightarrow \mathbb{R}^3 \\
\tau & \longmapsto \vec{F}(\tau) = \begin{cases} 
\frac{1}{t} \left( \frac{4\pi}{3} - \frac{\vec{\mu}_f}{t} \right) & \text{if } \tau \leq \frac{t}{2} \\
\frac{1}{t} \left( 3\vec{\mu}_f - \frac{4\pi}{t} \right) & \text{if } \tau > \frac{t}{2}
\end{cases}
\end{cases} $$
A.2.1 Initial and final conditions

This system entails limits conditions on $\mu$ that can be easily verified. First, it is clear that $\mu(0) = \int_0^0 \mu(\tau) d\tau = 0$. Then, considering $\mu(t)$ we get:

$$\mu(t) = \int_0^t F(\tau) d\tau$$

$$= \frac{1}{t} \left[ \int_0^{t/2} \left( \frac{4 \xi}{t} - \mu_f \right) d\tau + \int_{t/2}^t \left( 3 \mu_f - \frac{4 \xi}{t} \right) d\tau \right]$$

$$= \frac{2 \xi}{t} - \frac{\mu_f}{2} + \frac{3 \mu_f}{2} - \frac{2 \xi}{t}$$

$$= \frac{\mu_f}{2}$$

(A.7a)

A.2.2 Integral constraint

The considered system also entails a condition on the value of the time integral of the impulse over a control period. The impulse function $\mu$ created by the force $F$ defined above is:

$$\mu(\tau) = \begin{cases} \tau \left( \frac{4 \xi}{t} - \mu_f \right) & \text{if } \tau \leq \frac{t}{2} \\ \frac{2 \xi}{t} - \frac{\mu_f}{2} + \left( \frac{\tau}{t} - \frac{1}{2} \right) \left( 3 \mu_f - \frac{4 \xi}{t} \right) & \text{if } \tau > \frac{t}{2} \end{cases}$$

Then it comes:

$$\int_0^t \mu(\tau) d\tau = \int_0^{t/2} \mu(\tau) d\tau + \int_{t/2}^t \mu(\tau) d\tau$$

After calculation, and considering the definition of $\mu$ given above this equality leads to:

$$\int_0^t \mu(\tau) d\tau = \xi$$

(A.8a)

Proving that $\mu$ is solution of the system 2.6.
A.3 Proof of SPHERES actuation

What we want to demonstrate here is the feasibility of accurate actuation considering the SPHERES physical constraints described on section 2.2.

For a control period, those constraints showed us that the only parameters that can be modulated are the thrusters firing times. Yet, as it is proven on equation 2.7, the opening times are constrained by the value of the commanded impulse at the end of the control period. Considering that a single thruster only fires once in a control period—which is what happens in SPHERES—we then can only set the opening instant $t_{on}$. Considering the $X$ projection of the time integral of the provided impulse over the control period, we then get:

$$\int_{0}^{T} \mu_{px}(t) \, dt = \frac{F_x t_x^2}{2} + F_x t_x (T - (t_{on} + t_x))$$

$$= F_x t_x \left( T - \frac{t_x}{2} - t_{on} \right)$$

(A.9a)

Which allows us to determine $t_{on}$ that will offer a solution to the system 2.8:

$$t_{on} = T - \frac{t_x}{2} - \frac{\xi_x}{F_x t_x}$$

(A.9b)

Again, the equation 2.7 implying $t_x = \frac{\mu_f}{F_x}$, we obtain the following:

$$\begin{cases} t_x = \frac{\mu_f}{F_x} \\ t_{on} = T - \frac{\mu_f}{2F_x} - \frac{\xi_x}{\mu_f} \end{cases}$$

(A.10)

This just proves that by properly setting the SPHERES firing time, it is possible to get an actuation that perfectly matches the commanded pulse as well as the commanded pulse integral, in spite of the constraints embedded to the testbed.
Appendix B

Load cell raw measurements

Here are shown some raw data output by siglab during the load cell experiments.

![Graphs showing force measurements](image)

**Figure B.1:** Force measurement raw data for Thruster #4 of Sphere #2.
Figure B.2: Comparison between old mixer and mixer 1 (primary axis accuracy), for a command of 0.15 N.s on $F_z$ axis. The first to fire is old mixer. We can see the longer opening time on $F_z$ for mixer 1.
Appendix C

Taking force drop into account

We have seen that the force drop due to the firing of several thrusters was the main cause of errors of the old mixer. Here is presented the algorithm that correct this effect. As mentioned earlier, it might still me improved, especially computation time wise.

C.1 The thruster command structure

This structure represent a thruster opening command by the thruster number and the commanded on-time. It is going to be useful later.

```c
typedef struct {
    float on_time;
    int index;
} thruster_command;
```

C.2 Taking force drop in account

We have seen that the duration times had to be sorted for computing the corrected on-times more easily. This is the purpose of the `mixer_mergesort` algorithm that is used, but not presented here. It takes as input an array of `thruster_command` representing the commands for the 12 thrusters, and sort it.

The algorithm that actually correct the thrust drop error is presented here. It takes as input the commanded on times `u` for the 12 thrusters, and knows the global variable `command`, that is an array of 12 `thruster_command`
representing the commanded on-times for all thrusters, and the attenuation array, that contains the force drop factor considering the number of other opened thrusters.

```c
void mixer_multipleFiringCorrection(float u[12],
                                     unsigned int maxPulseWidth)
{
    int i, processed_thrusters, nb_longer_thrusters;
    float corrected_u[12];

    // Initialize variables
    processed_thrusters = 0;
    memset(corrected_u, 0, sizeof(float)*12);

    // Sort pulse_length
    mixer_mergesort(command, on_count);

    /* Correct thrust drop, considering centered pulses */
    for (i=0; i<on_count; i++)
    {
        if (processed_thrusters==0)
        {
            nb_longer_thrusters = on_count - 1;
            corrected_u[command[i].index] =
                command[i].on_time / attenuation[nb_longer_thrusters];
        }
        else
        {
            // Determine number of longer (or equal) pulses
            if (command[i].on_time != command[i-1].on_time)
                nb_longer_thrusters = on_count -
                               (processed_thrusters + 1);

            corrected_u[command[i].index] =
                corrected_u[command[i-1].index] +
                (command[i].on_time - command[i-1].on_time) /
                attenuation[nb_longer_thrusters];
        }
        processed_thrusters++;
    }

    // Scale corrected on-times and save
    mixer_scale(corrected_u, maxPulseWidth);
    memcpy(u, corrected_u, sizeof(float)*12);
}
```
Bibliography


