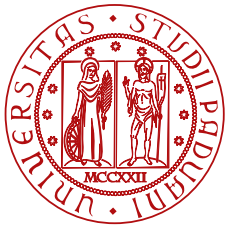

Università degli Studi di Padova

Dipartimento di Ingegneria Industriale

Master Thesis in Aerospace Engineering

ERMES: reduced-gravity testing of an autonomous docking manoeuvre between small satellites

Master Candidate

Alessandro Bortotto

Student ID 2018732

Supervisor

Prof. Francesco Branz

Università degli Studi di Padova

Co-supervisor

Eng. Lorenzo Olivieri

Academic Year
2022/2023

"Non exiguum temporis habemus, sed multum perdidimus."
Lucius Annaeus Seneca

"I don't have time to explain what I don't have time to understand."
Cayde-6

Abstract

Last decades, space applications and technological demonstrations involving small satellites have significantly increased. Many companies, which invest in developing miniaturized systems suitable for various space missions, are currently thriving in the so-called New Space Economy framework. As a consequence, the number of orbiting CubeSats has increased markedly. In this context, the University of Padua has shown a strong interest towards proximity operation systems for small autonomous satellites due to their effectiveness in several applications. The heritage of the University includes research in topics such as on-orbit servicing, active space debris removal, assembly of large in-space structures and flight formations control.

Experimental Rendezvous in Microgravity Environment Study (ERMES) is a student project that has as its primary objective the design and test of an autonomous docking manoeuvre between two free-flying CubeSats mock-ups in a reduced gravity environment. The two mock-ups involved in the experiment are equipped both with a Guidance Navigation and Control (GNC) system and miniaturized docking interfaces. During the manoeuvre, they work in a Target-Chaser configuration, where the Chaser is active while the Target is cooperative. The reduced gravity conditions are achieved by participating in the 79^{th} ESA Parabolic Flight Campaign, following the selection for the "Fly Your Thesis! Programme" 2022.

This thesis presents an overall picture of the ERMES project. It focuses firstly on framing the experiment in the current State Of Art (SOA) regarding autonomous docking manoeuvres; secondly on depicting the design of the experiment and the experimental procedure; and finally on discussing the outcome of the test campaign through results analysis.

Sommario

Negli ultimi decenni, le applicazioni spaziali e le dimostrazioni tecnologiche che coinvolgono piccoli satelliti sono aumentate in modo significativo. Molte aziende, che investono nello sviluppo di sistemi miniaturizzati adatti a varie missioni spaziali, stanno attualmente prosperando nel quadro della cosiddetta *New Space Economy*. Di conseguenza, il numero di CubeSat in orbita è aumentato notevolmente. In questo contesto, l'Università degli Studi di Padova ha mostrato un forte interesse verso i sistemi per operazioni di prossimità per piccoli satelliti autonomi, vista la loro efficacia in diverse applicazioni. L'eredità dell'Università in merito comprende ricerche su temi quali l'assistenza in orbita, la rimozione attiva dei detriti spaziali, l'assemblaggio di grandi strutture nello spazio e il controllo delle formazioni di volo.

Experimental Rendezvous in Microgravity Environment Study (ERMES) è un progetto studentesco che ha come obiettivo primario progettare e testare una manovra di docking autonomo tra due prototipi di CubeSats rilasciati in un ambiente a gravità ridotta. I due prototipi coinvolti nell'esperimento sono dotati di sistemi di Guida, Navigazione e Controllo (GNC) e di interfacce di docking miniaturizzate. La manovra avviene in una configurazione Target-Chaser, nella quale il Chaser è attivo mentre il Target è cooperativo. Le condizioni di gravità ridotta sono state ottenute partecipando alla 79^a Campagna di Voli Parabolici dell'Agenzia Spaziale Europea (ESA), a seguito della selezione per il "*Fly Your Thesis! Programme*" 2022.

Questa tesi presenta un quadro generale del progetto ERMES. Si concentra in primo luogo sull'inquadramento dell'esperimento nell'attuale Stato dell'Arte (SOA) relativo alle manovre di docking autonomo; in secondo luogo sulla descrizione del design dell'esperimento e della procedura sperimentale; infine sulla discussione dell'esito della campagna di test attraverso l'analisi dei risultati.

Contents

List of Figures

List of Tables

Nomenclature

Indices and Acronym

ARCADE-R2 Autonomous Rendezvous Control And Docking Experiment - Reflight 2

ATV Autonomous Transfer Vehicle

CM Centre of Mass

CPOD CubeSat Proximity Operations Demonstration

DoF Degree of Freedom

ERMES Experimental Rendezvous in Microgravity Environment Study

FELDS Flexible Electromagnetic Leash Docking system

FYT Fly Your Thesis

GNC Guidance Navigation and Control

ISS International Space Station

MIT Massachusetts Institute of Technology

MOSFET Metal Oxide Semiconductor Field Effect Transistor

MSDS Material Safety Data Sheet

OBCS On Board Computer System

PACMAN Position and Attitude Control with Magnetic Navigation

PID Proportional Integrative and Derivative

PLA Polylactic acid

PWM Pulse Width Modulation

RRS Real Reference System

RW Reaction Wheel

SOA State Of Art

SPHERES Synchronized Position Hold, Engage, Reorient, Experimental Satellites

SRS Screen Reference System

ToF Time of Flight

TRL Technology Readiness Level

Symbols

γ Heat capacity ratio

ω Angular velocity

ϕ Roll

ψ Yaw

θ Pitch

A_{ex} Exit area

f_{RS} Scaling factor of a Reference System

h_w Angular momentum

I Moment of inertia

M Torque

$M_{ROT_{RS1 \rightarrow RS2}}$ Matrix of rotation from Reference System 1 to Reference System 2

p^* Sonic pressure

p_0 Total pressure

p_{atm} Atmospheric pressure

p_{ex} Exit pressure

R Gas characteristic constant

T Thrust

v_{ex} Exit velocity

Chapter 1

Introduction

Recently, space applications and technological demonstrations involving small satellites have significantly increased. The reason behind this recent momentum is found in the access to space that has become more economically affordable thanks to new technologies and advanced manufacturing processes. Consequently, small satellites consolidated themselves as a low-cost solution for many applications due to their reduced dimensions and mass. Many companies, which invest in developing miniaturized systems suitable for various space missions, are currently thriving in the so-called New Space Economy framework. Therefore the number of CubeSats in orbit has increased markedly. In this scenario, studies that focus on miniaturized satellites, investigating more reliable and efficient Guidance Navigation and Control (GNC) systems, adaptable docking interfaces and autonomy-related softwares, are of high interest.
In particular, the University of Padua has shown a strong interest towards proximity operation systems for small autonomous satellites due to their effectiveness in several applications. The heritage of the University includes research in topics such as on-orbit servicing, active space debris removal, assembly of large in-space structures and flight formations control.
Experimental Rendezvous in Microgravity Environment Study (ERMES) is a project of the University of Padua carried out by Master Degree students from different academic backgrounds. The main goal of ERMES is the development and testing of an autonomous docking manoeuvre between two free-floating CubeSat mock-ups in a reduced-gravity environment.

A space rendezvous is a series of consecutive orbital manoeuvres which allow two spacecraft to reach the same orbit and get closer up until a very short distance. The docking manoeuvre is completed when a connection between the two is es-

Figure 1.1: ERMES Official Logo

tablished. In particular, space rendezvous are divided into three major phases: (1) the first phase of fly-around, which is needed to insert one spacecraft in an orbit around the other; (2) then an approaching phase, which aims at closing the distances; (3) lastly, a phase of proximity navigation that includes all the adjustments before the actual docking.
ERMES focuses on the last phase (see figure 1.2) because proximity navigation manoeuvres are notoriously difficult and risky since mistakes during this phase could easily lead to mission failure. Normally the vast majority of docking manoeuvres are carried out by astronauts, whose role is to check the proper progress of the operation. However, space missions involving small satellites can not always afford human monitoring, therefore they must rely on sensors and software solutions to accomplish this particular task. Thereby, more efficient and reliable proximity navigation and control systems for autonomous small satellites are of great interest, due to their effectiveness in several applications.

The two mock-ups involved are named ”Target” and ”Chaser” after their role in the manoeuvre. The Chaser actively perform the manoeuvre by approaching the Target and dock, while the Target acts cooperatively by contrasting unwanted attitude disturbances. Both mock-ups are equipped with Guidance Navigation and Control (GNC) systems suitable for their role. In particular, the Chaser is equipped with a cold gas (CO_2) propulsive system characterized by a set of 8 simple convergent nozzles, which allows the Chaser to control all its 6 Degrees of Freedom (DoF); while the Target is equipped with three Reaction Wheels (RW),

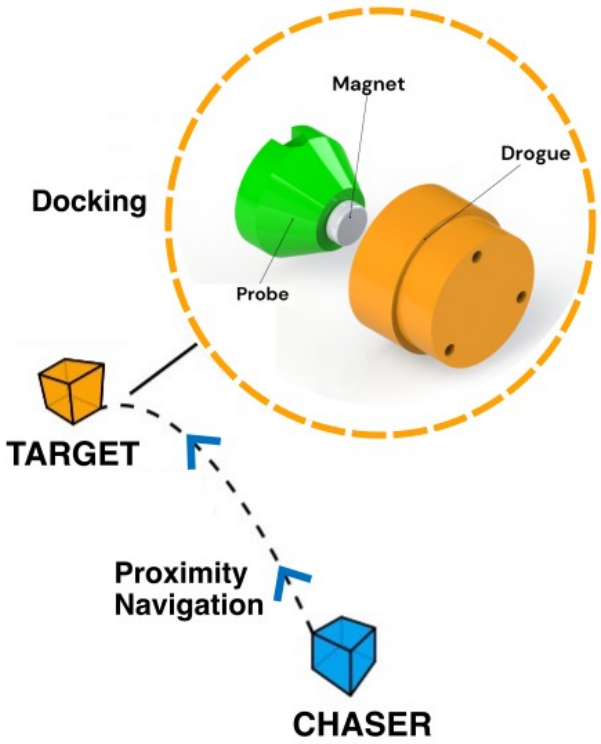

Figure 1.2: Schematic description of ERMES manoeuvre

which allow controlling only its attitude (3 DoF). The manoeuvre is accomplished by releasing the CubeSat mock-ups from their initial electromagnetic constraints into a free-floating condition, then the dedicated localization and proximity navigation software permits the Chaser to find and reach the Target.
The reduced-gravity environment is achieved by boarding the experiment on a parabolic flight. This particular experimental platform guarantees a low level of gravity aboard, that can reach values lower than $0.01g$.

1.1 State of Art

Autonomous space systems have always been an interesting topic for the scientific community. The reason behind this wide interest is found in their usefulness for several applications, such as large structure assembly, active space debris removal or formation control. The *Automated Transfer Vehicle* (ATV) [1] is one of the first examples of an autonomous space system that carried out on multiple occasions space rendezvous with the International Space Station (ISS). Other examples of an autonomous vehicle are the *Dragon-2* of SpaceX [15], which is a class of partially reusable spacecraft that on several occasions brought astronauts to the Station.
For what concerns studies regarding small satellites, some notable examples are:

(1) *Synchronized Position Hold, Engage, Reorient, Experimental Satellites* (SPHERES) [9] aboard the ISS, which consists of a series of miniaturized satellites developed by the Massachusetts Institute of Technology (MIT) used to test flight formations, rendezvous and autonomy algorithms in view of future implementations; (2) *Astrobee* [10], which is a free-flying robotic system developed by National Aeronautics and Space Administration (NASA) from the legacy of SPHERES, which helps astronauts reduce the time they spend on routine duties on the ISS; (3) *CubeSat Proximity Operations Demonstration* (CPOD) [5] mission led by Tyvak Nano-Satellite Systems focused on a docking manoeuvre of two 3U CubeSats.

The University of Padua has a grounded heritage in the field of autonomous docking manoeuvres, thanks to a series of successive student projects, that participated in different Programs proposed by the ESA Education Office. These projects are: (1) *Flexible Electromagnetic Leash Docking system* (FELDs) [14], which participated in the ESA "Drop Your Thesis! Programme" 2014 to test an electromagnetic soft docking technology, and achieved a post-docking mechanical connection with a flexible wire; (2) *Autonomous Rendezvous Control And Docking Experiment - Reflight 2* (ARCADE-R2) [2], which participated in the ESA "BEXUS" 17 and correctly performed three release operations and two docking procedures between 2-DoF vehicles; (3) *Position and Attitude Control with Magnetic Navigation* (PACMAN) [7], which participated in the ESA "Fly Your Thesis! Programme" 2017 and tested electromagnetic navigation for autonomous attitude control during a soft docking manoeuvre.
Therefore ERMES is a successor of the previous student experiments of the University and shares similar objectives. The experiment aims at further increasing the Technology Readiness Level (TRL) regarding autonomous docking systems starting from this well-established heritage.

1.2 Scientific Objectives

The main objective of the ERMES is to perform an autonomous docking manoeuvre between two small CubeSat mock-ups. This major objective can be achieved thanks to a series of low-level goals:

1. Firstly, design, develop and assembly both Target and Chaser accordingly to their different contributions in the manoeuvre;

2. Secondly, design, develop and test the software to control both Target and Chaser. In particular, it comprehends a dedicated proximity navigation soft-

ware for the Chaser, which should be able to autonomously locate the Target, compute the trajectory to reach it and send the command to the actuators, and a software for attitude control for the Target.

3. Finally, design, develop and test the miniaturized docking interfaces, suitable for this particular application.

In general, the ERMES experiment poses itself as a technological demonstrator of the integration of different subsystems to perform autonomously such difficult manoeuvres with small satellites. Simultaneously, it aims to prove their feasibility and versatility for several space applications. This general objective can be achieved by validating and testing the entire system in a relevant reduced-gravity environment. To fulfil this objective, the ERMES project has been selected in the ESA "Fly Your Thesis! Programme" 2022 (1.3).

Finally, the objectives of ERMES, considered as a student project, are various and equally fundamental. Firstly, it aims at complementing the standard academic course by improving the working skill and scientific knowledge on space topics of all team members. Moreover, another objective regards developing the interpersonal and management skills in view of occupations in future work environments, thanks to the help and interactions with the professors, researchers and various experts in the space field.

1.3 ESA Fly Your Thesis! Programme 2022

The ESA "Fly Your Thesis! Programme" [8] is a once-in-a-lifetime opportunity held by the ESA Education Office for students to propose, design, build, test and fly their experiment on board a parabolic flight. The Parabolic Flight Campaign takes place in Bordeaux (France), at the facilities of Novespace [12], which is the company that owns and operates the Airbus A310. The Campaign features 3 flights with 31 parabolas per flight.
This Programme aims at giving students the experience of working in a professional project environment, while simultaneusly allowing them to improve their *Curriculum Vitae* and their professional network.

1.3.1 Parabolic Flight

Parabolic flights are a particular type of testing platform, which provides a brief near-weightless environment for scientific and technological investigations. They

Figure 1.3: ESA "Fly Your Thesis! Programme" Logo

are characterized by a particular trajectory, that guarantees $22s$ of low-gravity phase (generally lower than 0.01g) at the top of the parabola. The trajectory is divided into four main phases (1.4):

1. **Standard-gravity phase**: initially the plane is flying straight without accelerating, therefore the gravity level perceived is 1g. During this phase all the preparations for the experiment take place. The duration of this phase can vary depending on a series of factors.

2. **First Hyper-gravity phase**: secondly the plane starts accelerating up $45°$. The perceived g-level is around 1.8. In this phase, the experimenters are anchored to the floor thanks to seatbelts. This phase lasts around $20s$.

3. **Low-gravity phase**: when the plane reaches the top of the parabola, the pilot lowers the engine power and the plane starts free-falling. Inside the plane, the perceived g-level drops under 0.01g for around $22s$. During this phase, experiments involving low-gravity texting take place.

4. **Second Hyper-gravity phase**: finally, the plane starts accelerating down $45°$ and return to the standard gravity conditions. This phase is analogous to the previous hyper-gravity one.

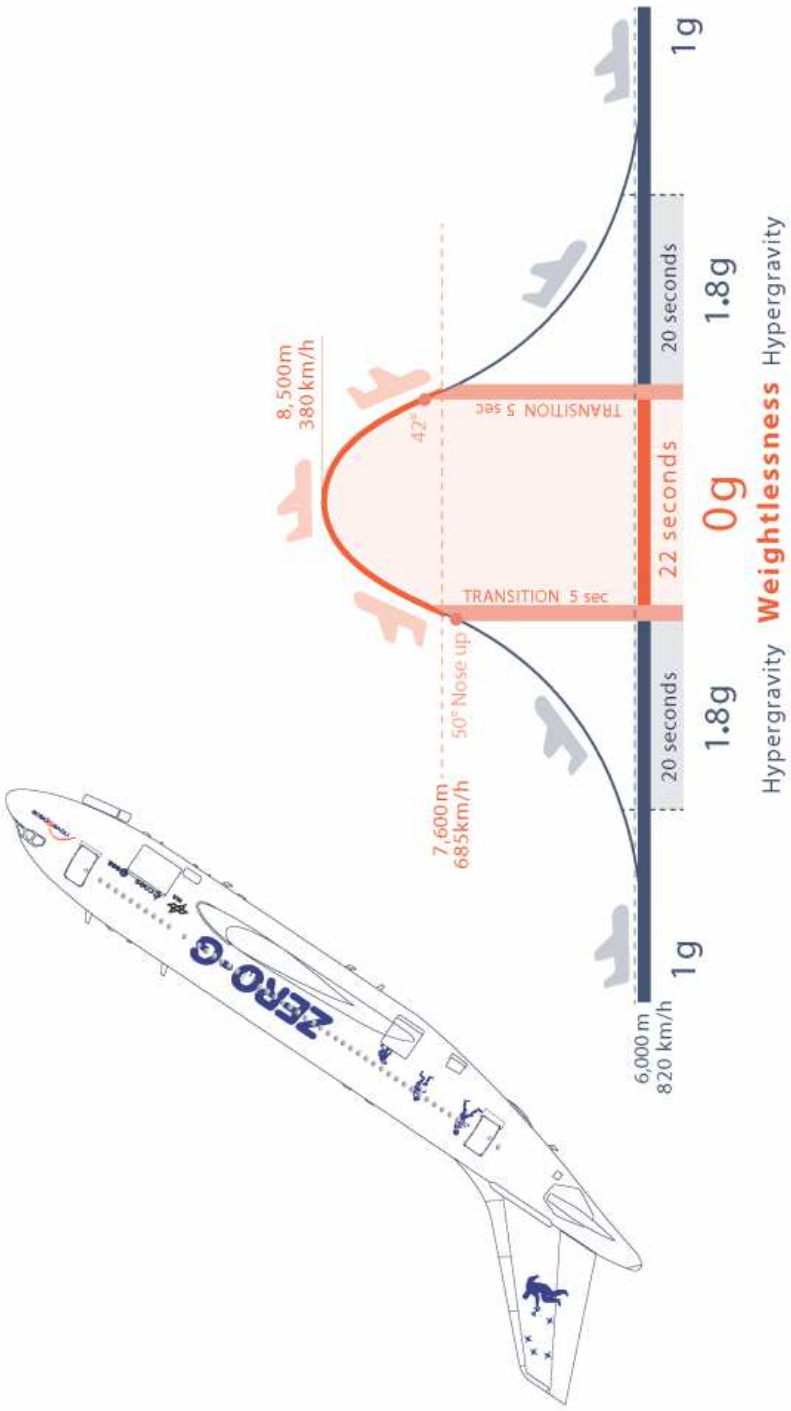

Figure 1.4: Parabolic Flight Scheme

Chapter 2

Experimental Setup

This chapter is dedicated to the description of all the main subsystems of ERMES, highlighting their role in the manoeuvre, how they interact and the key features and characteristics of their design. ERMES is composed of three main subsystems: the Chaser (figure 2.1 and 2.2), which is the active one during the capture operations; the Target (figure 2.3), that acts cooperatively trying to facilitate the Chaser; and finally the Release Structure (figure 2.4), that is used to position and release the mock-ups prior to the starting of the manoeuvre.

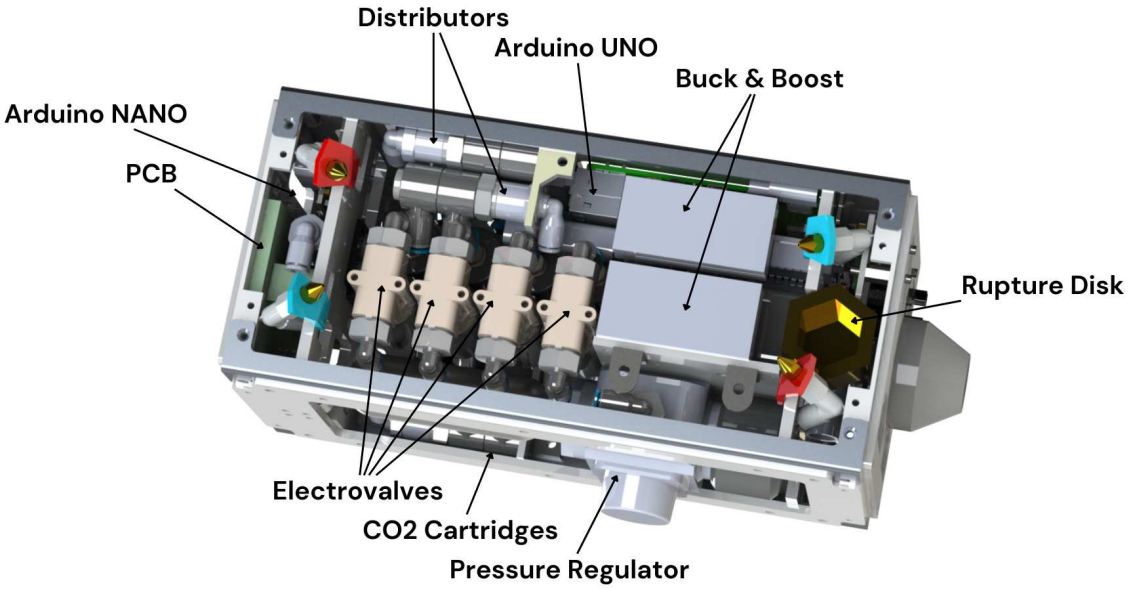

Figure 2.1: Mock-ups - Chaser [side A]

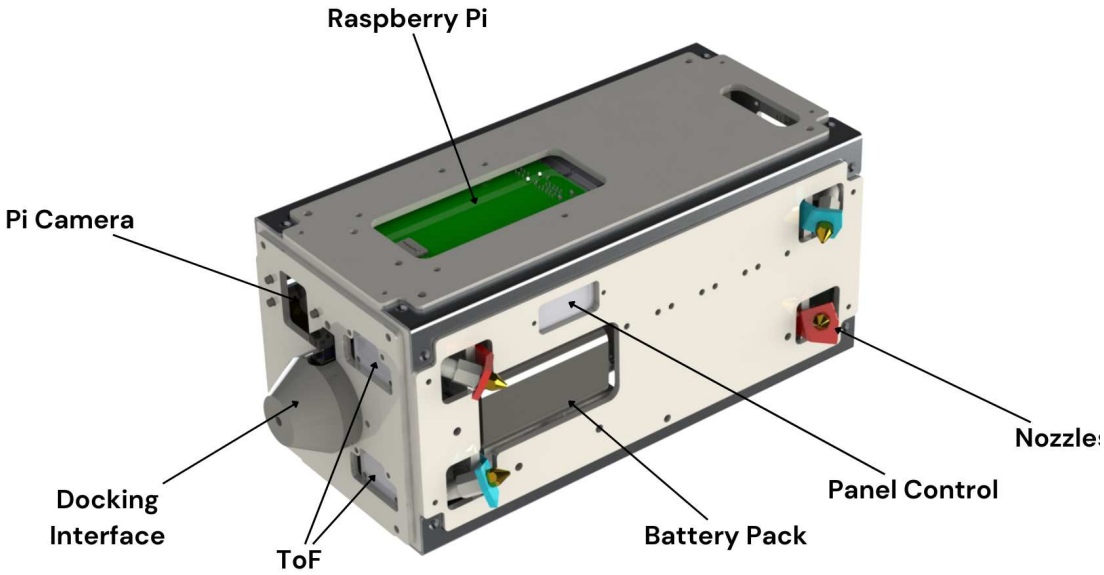

Figure 2.2: Mock-ups - Chaser [side B]

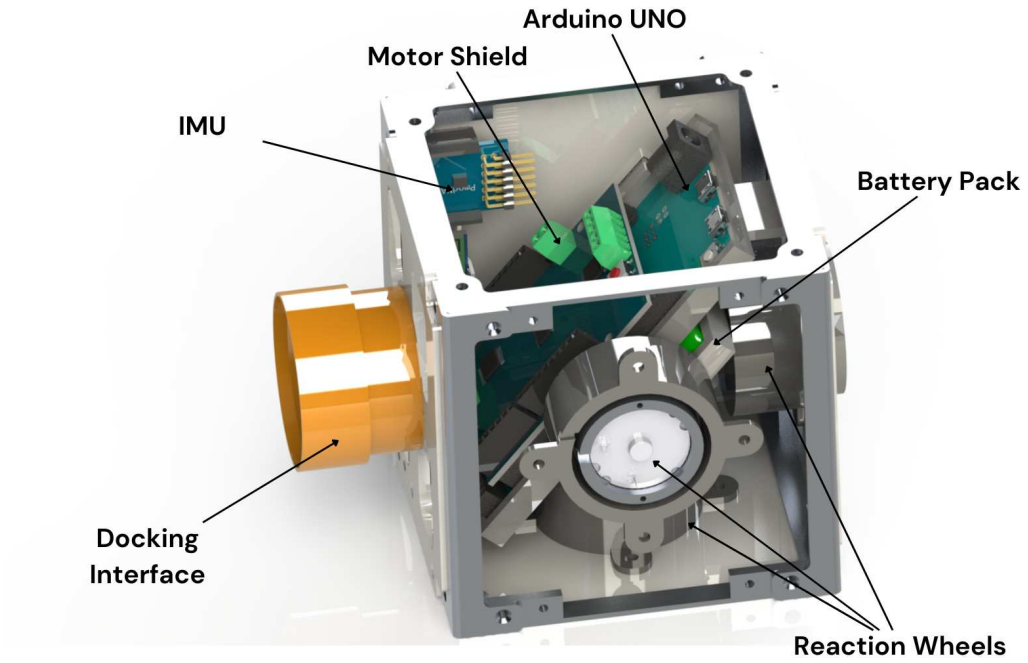

Figure 2.3: Mock-ups - Target

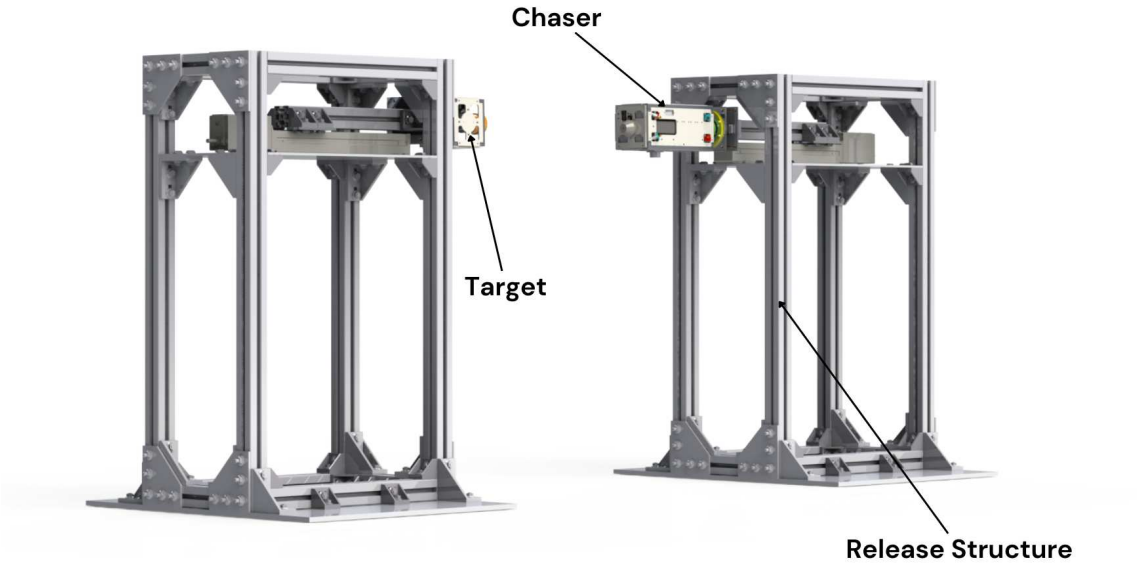

Figure 2.4: Release Structures

2.1 Mock-ups

The mock-ups do not share a lot of similarities due to their different role in the manoeuvre. Their resemblances are mainly in terms of external structure. Whereas the main differences lie in the GNC systems, in particular in the type of actuators and the support electronics needed for their powering and control. Therefore, different subsections are dedicated to describing the GNC systems separately.

2.1.1 Mock-ups Structure

The structures of the mock-ups include both the external frames and the internal walls. The purpose of the structure of the mock-ups is to hold in place and protect all the components of the systems. The design of the external structure follows the CubeSat standards. In particular, the structure of the Chaser is 2U ($200 \times 100 \times 100mm$), while the Target is 1U ($100 \times 100 \times 100mm$). Since the structure is modular, the design and development, as well as structural tests and solutions regarding the safety management of the possible hazards, are shared between both mock-ups.
The columns and the bases that compose the external structure are made of aluminium and have been manufactured with laser cutting and milling processes; whereas the lateral faces have been milled from polycarbonate sheets; finally, some small supports for the various components have been 3D-printed in PLA (Polylactic acid) plastic, which is a vegetable-based plastic material.

Finally, before the flights both structures must have been covered in a soft damping material (foam) to avoid damage from falling during the hyper-gravity phases (in accordance with Novespace Guidelines regarding free-flying objects). This foam is applied directly on the structure and covered in black tape to avoid debris formation from the foam itself.

2.1.2 Chaser GNC

This subsection is dedicated to the GNC subsystem of the Chaser, in particular the type of actuators of which the Chaser is provided and the On Board Computer system used to control them are presented.

⋆ Actuators

The Chaser is equipped with a set of 8 actively controlled thrusters connected to a pneumatic subsystem (shown schematically in figure 2.5). The propulsive system is powered by a small reservoir of liquid CO_2.
The propulsive system is composed of the following components:

- **CO2 cartridge** (C1): The selected cartridges contain $16g$ of CO_2. The pressure of the CO_2 inside the cartridge is around $57bar$ at standard condition of temperature ($25°C$). From the CO_2 Phase Diagram, these condition of pressure and temperature denotes a liquid state.
  In terms of hazardousness, the official Material Safety Data Sheet (MSDS) of the cartridges reports that they are IATA 2.2 approved, therefore they can be boarded on the plane. Moreover, their "high pressure per volume factor" is relatively low ($1.76bar \cdot L$).
  For what concerns connections, they have a threaded output and consequently are directly connected to the Pressure Regulator that follows.

- **Pressure Regulator** (PR1): The selected pressure regulator stands a maximum pressure of $82bar$ of input and can regulate pressure between 0 and $5.5bar$.
  It sets a working pressure of $2.5bar$. According to the MATLAB code used to study the fluid dynamics of the pneumatic system, this particular working pressure value guarantees over $100mN$ of thrust per nozzle. This analysis was carried out by considering the atmospheric pressure (p_{atm}) of both the laboratory tests [$1bar$] and the parabolic flight environment [$0.8bar$]).

- **Rupture Disk** (RD1): A rupture disk is implemented to avoid high-pressure bursts in the pneumatic system in case of a failure of the regulator. The pressure at which it bursts is $6.5barg \pm 10\%$, where $"barg"$ mean $"bar"$ relative to the atmospheric pressure.

- **Tubings and connectors/adapters** (T1-21 and D1-5): The tubings, connectors/adapters are needed to link all the components together. The tubings implemented have an external diameter of $4mm$ and an internal diameter of $2mm$.

- **Solenoid Electrovalves** (SV1-8): They represent the key component of the pneumatic system because they are the actual actuators controlled by the microcontroller (Arduino) as explained in section 2.1.2.
  Throughout various testing, the reaction time of the electrovalves has been measured to be $4ms \pm 10\%$. They have a throat diameter of $1.6mm$, opportunely chosen to be bigger than the exit diameter of the nozzles in order to avoid flow throttling inside the electrovalves, because it could have damaged the component.

- **Nozzle** (N1-8): The 8 nozzles are selected to be simple convergent with an exit diameter of $1mm$. This choice is in contrast with a classic convergent-divergent configuration due to the necessity of avoiding supersonic flows. In fact, since the experiment takes place at standard atmospheric pressure and not in a vacuum chamber, low exit pressure could lead to unwanted shock waves, that cause the transition from a supersonic flow to a subsonic flow at the exit. These shock waves travel inside the circuit, increase the pressure in the pneumatic system and possibly damage the components or simply alter the performances of the propulsive system.
  However, another advantage of this design is that, for total pressures (p_0) high enough, the flow exits at a sonic state (Mach number = 1 and $p_{ex} = p_{sonic} = p^*$). Increasing the total pressure of the system, the exit pressure increases linearly (equation 2.1.1).

$$p^* = p_0 \cdot (\frac{2}{\gamma - 1})^{\frac{\gamma}{\gamma+1}} \tag{2.1.1}$$

  Where the working pressure set on the regulator is assumed to be the total pressure of the gas flow because the flow is still when the electrovalves are closed. γ is the Heat capacity ratio of the CO_2.
  The sonic exit condition guarantees that the thrust (T) output (equation

2.1.2) is linear with respect to the pressure set too (equation 2.1.5).

$$T = \dot{m} \cdot v_{ex} + (p_{ex} - p_{atm}) \cdot A_{ex} \tag{2.1.2}$$

where $\dot{m}$ at a sonic state is (2.1.3):

$$\dot{m} = \frac{p^* \cdot A_{ex}}{\sqrt{\gamma \cdot R \cdot T^*}} \cdot \sqrt{\frac{2}{\gamma - 1} \cdot [(\frac{p^*}{p_0})^{\frac{1-\gamma}{\gamma}} - 1]} \tag{2.1.3}$$

and v_{ex} at a sonic state is (2.1.4):

$$v_{ex} = \sqrt{\gamma \cdot R \cdot T^*} \tag{2.1.4}$$

$$T = (p_0 \cdot \Gamma_{(\gamma)} - p_{atm}) \cdot A_{ex} \tag{2.1.5}$$

where $\Gamma_{(\gamma)}$ is a costant depending only on γ.
According to the MATLAB code used to study the fluid dynamics of the pneumatic system, the minimum value of total pressure for a sonic exit is $1.8bar$ (at $p_{atm} = 1bar$). Therefore a working pressure higher than that value (in our case $2.5bar$) guarantees certainly a sonic exit velocity and a linearly increasing thrust.

In summary, this particular design allows having a generally low-pressure system that generates an amount of thrust (at least $100mN$), that has been assumed a suitable value to perform quickly the docking manoeuvre accordingly to the preliminary simulations.
Moreover, it must be highlighted that since the working pressure is low and the CO_2 is a compressible gas, the phenomena related to the presence of back pressures are negligible. In particular, preliminary analysis and testing showed that the working pressure, set by the regulator, decreases by less than 5% at the nozzle. The evaluation has been performed by comparing data of thrust with two systems, which differ in the number of obstructions and flux deviations.

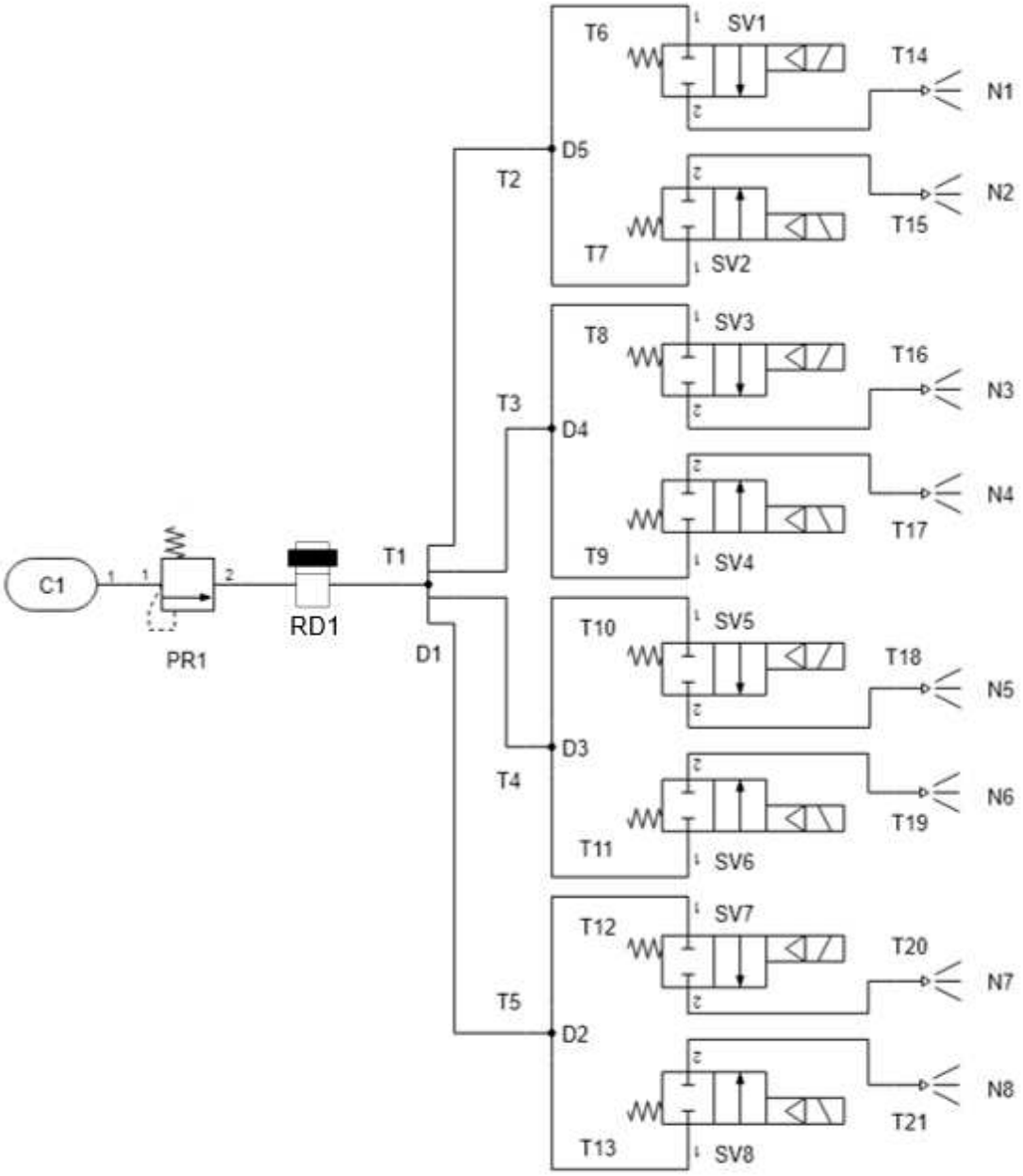

Figure 2.5: Chaser - Propulsive System Scheme

The configuration of the thrusters (figure 2.6) has a particular design, that has been created for this particular experiment and that allows controlling both attitude and trajectory (all 6 DoF) of the Chaser.
The 8 thrusters are divided into two groups of four each that are positioned in two opposite faces of the Chaser. Each thruster is tilted $\pm 45°$ with the respect to the normal vector to the face at which they are anchored, $\pm 30°$ and $\pm 60°$ with respect to the parallel axes of the face. Considering every couple of adjacent thrusters, in which a quartet can be divided, they point towards the same direction.
Finally, the reason why this particular subdivision in quartets and couples guarantees to control every DoF is that the just mentioned pointing directions are different for every possible couple. To move or rotate along a single axis four thrusters must be actuated together as shown in table 2.1.

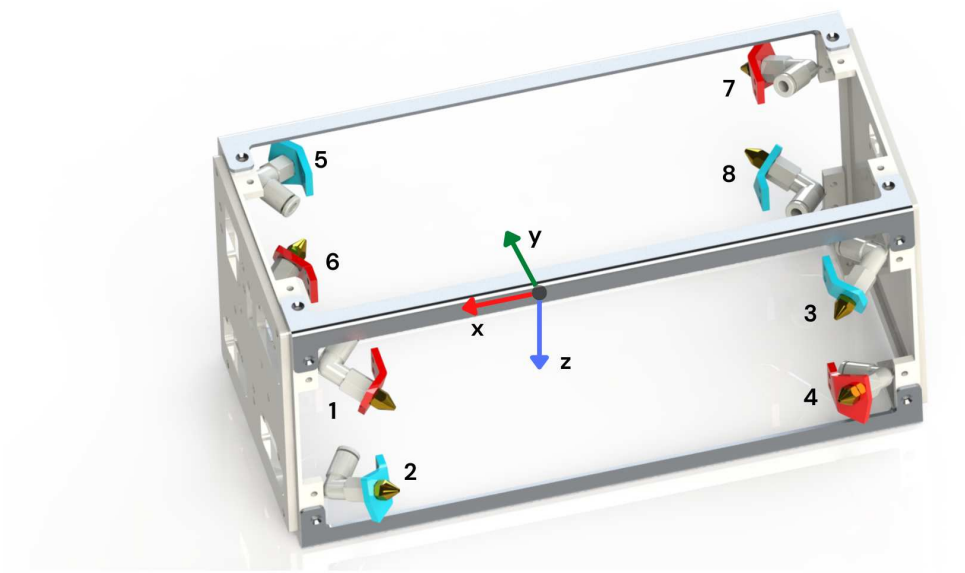

Figure 2.6: Chaser - Thrusters configuration

	Translation		**Rotation**	
	pos	**neg**	**pos**	**neg**
x	1256	3478	1368	2457
y	1234	5678	1458	2367
z	2468	1357	1278	3456

Table 2.1: Thrusters to be actuated to control the DoFs

The thrusters are controlled with a $30Hz$ Pulse Width Modulation (PWM) with 16 possible steps of discretization, that determines the duty cycle (DC) of the valve.

The choice of using a $30Hz$ PWM is derived from a trade-off based on the results of the testing of the electrovalves. The 16 steps are not related to a standard $4-bit$ PWM, but a fictitious $8-bit$ PWM, but with only 16 allowed values instead of 256. The number of steps is found by elevating 2 at the power of the number of bits.

The propulsive system of the Chaser has been profoundly tested. First of all, tests have been carried out to confirm the preliminary simulations and validate the designs. Then safety-related tests have been carried out according to the Novespace Guidelines. Finally, the focus shifted towards testing the control of the electrovalves.
In particular, the schedule of the tests has been the following:

- **Tests on Authority Validation**
  The first tests, which were focused on preliminary validations, aimed at measuring the maximum thrust the system could produce.
  The pneumatic circuit considered was initially simplified, it consisted of: a nozzle, an electrovalve, a pressure regulator and a CO_2 cartridge. Thrust data were acquired with a load cell connected to a supportive aluminium plate where the nozzle was attached. The electrovalves were controlled by an Arduino board. The setup of this laboratory experiment is shown in figure 2.7. The pressure regulator was set at various working pressure in order to confirm the linearity of the thrust trend.
  The results of these tests relatively the linearity of the trend were reported in [4].

- **Tests on Actuators control**
  Secondly, after choosing the value of working pressure more suitable for the experiment, the focus shifted towards validating the control of the electrovalves. The objective of these tests was to find a method to linearize the behaviour of the electrovalves in order to simplify their control. In particular, this method is based on reducing the number of steps in a PWM based on $8-bit$ (256 steps) commands to a fictitious $4-bit$ (16 steps) PWM.
  The experimental setup was the same as the previous ones. The procedure was based on sending repetitively step-like commands to the electrovalves and registering the thrust with the load cell.
  Initially, the data acquisition regarded the real trend that links thrust and number of steps of the command of the $8bit$ PWM. As expected, the real trend is a sigmoid (figure 2.8) with the initial values around zero, because

for low values of steps the electrovalve is not able to react in time and completely open. Afterwards, the number of steps available has been reduced from 256 to 16 to linearize the thrust trend with the respect to the step values.
This solution aimed at simplifying the control of the actuators. The simplification lies in the shift from a real sigmoid trend to an ideal linear trend of the thrust. This new fictitious $8 - bit$ PWM is characterized by only 16 precise values out of the 256 possible ones (see table 2.2).

[!h] **Real 8-bit**	**Fictitious 4-bit**		**Real 8-bit**	**Fictious 4-bit**
0	0		8	157
1	83		9	172
2	94		10	187
3	103		11	206
4	111		12	223
5	121		13	244
6	132		14	250
7	144		15	255

Table 2.2: Relation between fictitious 4-bit PWM steps and real 8-bit PWM

Finally the resulting PWM has been tested by sending random step-like commands with the new fictitious $4 - bit$ commands. The resulting trend well approximates the desired linearity (figure 2.9).
An article about the results of such tests regarding the control of the electrovalves and the equalization of the thrust trend has been presented in [3].

- **Safety tests**
  Lastly, the tests regarding the safety of the system have been carried out. They focused on verifying the resistance of the entire pneumatic system in case of failure of the burst disk.
  Therefore, the pneumatic system downstream from the burst disk (distributors, tubings, 8 electrovalves and nozzles) has been connected to an external pressure regulator and reservoir of an air compressor. The working pressure of the entire system was raised to $10barg$ which is the maximum working pressure of the system considering a safety factor of 1.5. The endurance test lasts $15s$, which was higher than the expected time to empty a cartridge of CO_2. The test was then repeated three times to assure the results.

The pneumatic system passed the test without any major damage to the components and without displaying any leakages on the sealed connections. All these tests have been carried out according to the Novespace Guidelines regarding the safety of pressurized systems.

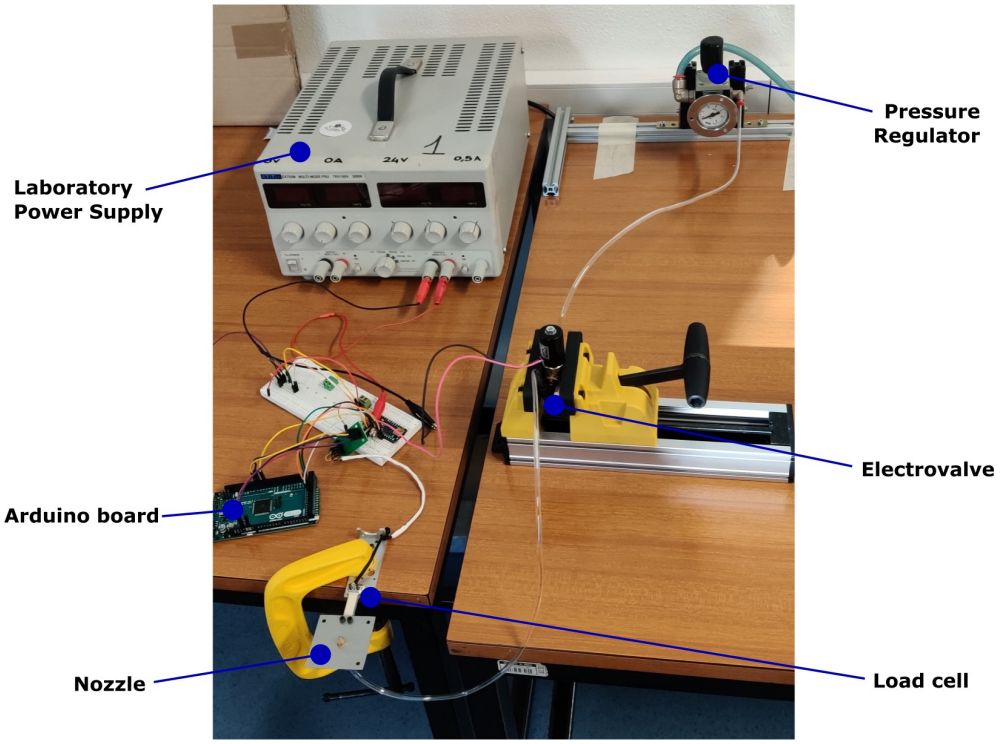

Figure 2.7: Experimental setup for tests on propulsive system

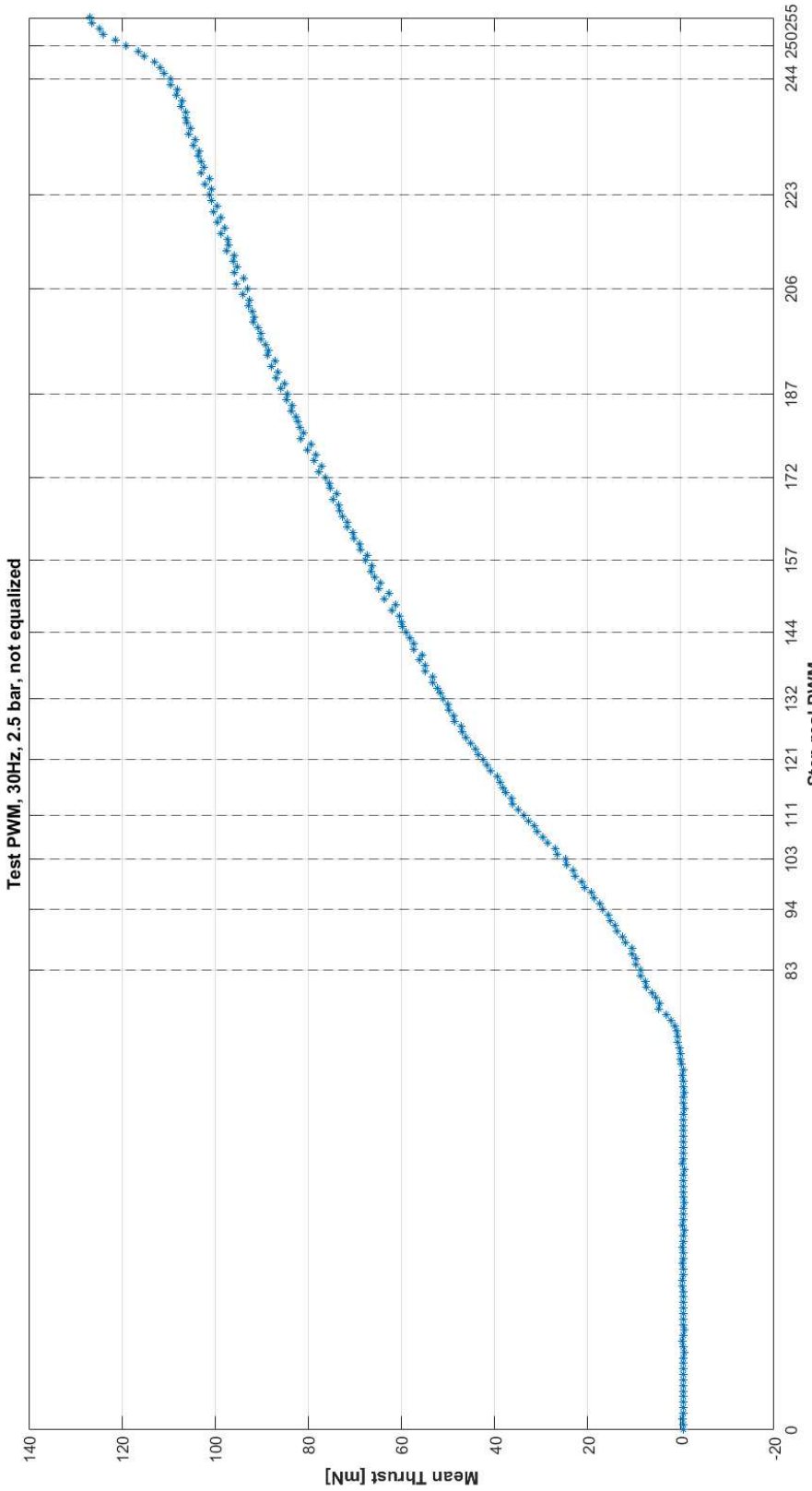

Figure 2.8: PWM tests – Real Trend

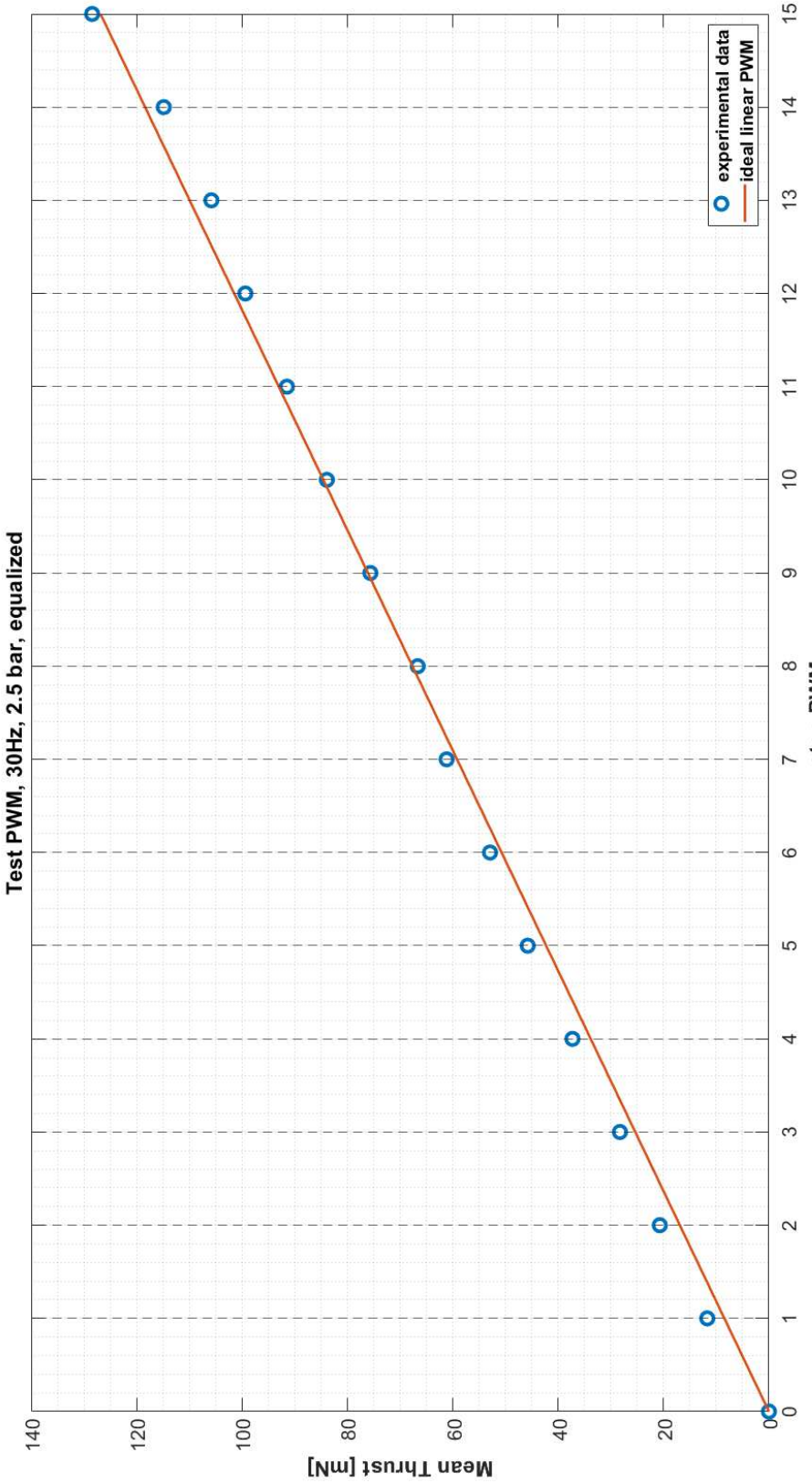

Figure 2.9: PWM tests – Equalized Trend

⋆ On Board Computer System and Electronics

This subchapter is dedicated to the On Board Computer System (OBCS) of the Chaser. To better understand the role of each components and how they interact, firstly it is presented a list of all the electronic components, then the functioning of the software is discussed explaining how each part of the OBCS cooperates with the others during the manoeuvre.
The list of components is the one that follows:

- **Battery pack**: It is needed to power all the system components.
  Since the overall requirement of power consumption is relatively high, a pack of 10 batteries with a total voltage output of $12V$ and $4A$ has been used. The battery pack chosen is composed by NiMh cells (nickel-metal hydride), because of reasons linked to the safety and hazards management. In fact, this particular type of battery is not inflammable differently from the standards with Lithium-ions.
  Moreover, in addition to a switch to act directly on the power supply of the system, a panel to monitor the level of the voltage has been placed in order to change the battery in case the voltage decreases too much.

- **Converters**: They are needed to reduce or increase the voltage of the battery to correctly supply all the components.
  In fact, while the boards that run the code and power the sensors require $5V$, the electrovalves chosen require $24V$. The maximum requirement of current is $4A$ when all the electrovalves are opened, while nominally the consuption varies from $2A$ and $3A$.

- **Electrovalves**: The eight electrovalves are needed to stop or let the flow pass through.
  The electrovalves are powered directly from the boost converter. They need $24V$ and $160mA$ while actuated, whereas nominally they are closed. However the power flow is controlled by a Metal Oxide Semiconductor Field Effect Transistor (MOSFET), which is a field-effect transistor that has three pins: two of them are power and ground, while the last one is the gate one that stops or allows the passage of power between the two pins.
  The MOSFET used are N-type, which means that the load is placed before the power pin and therefore the voltage drop after the ground pin is zero. Consequently, due to the nominal functioning of the N-types, the gate voltage must be just above zero, so any positive value under the electrical tolerance of the transistor is acceptable.

Each electrovalve is protected with a diode to ensure that no back current occurs.

- **Pi Camera**: It is used to register the data regarding the position the Target as explained later.
  The data are sent directly to the Raspberry Pi without any corrections or filtering. It is directly connected to the Raspberry Pi thanks to the CSI port. The focus distance is between $50mm$ and $250mm$ with maximum precision at around $150mm$.
  It is placed on the frontal face of the Chaser next to the docking interface.

- **Time of Flight (ToF) sensors**: Three Time of Flight sensors are used to evaluate the distance of the Target. They have been implemented to increment the precision of the visual system while the Chaser is closer to the Target, specifically under $150mm$. They are connected to a multiplexer that is then connected to the Arduino.
  They are placed on the frontal face of the Chaser next to the docking interface.

- **Raspberry Pi board**: It is the main board of control and it is needed to run the code to perform the manoeuvre. It is delivered with an Ubuntu 20.04 Server installation and the ROS Noetic framework.
  It receives the data from the ToF and the Pi Camera, then compute the position of the Target and a trajectory to follow to reach it. Afterwards, it sends the commands needed to perform the manoeuvre in order to actuate the electrovalves. Moreover, the Raspberry Pi is also connected via an *ssh* connection to the laptop in order to receive the message about the beginning of the manoeuvre.

- **Arduino boards**: Two Arduino board are implemented: one that deals only with actuating the electrovalves; while the other reads the data from a bunch of proximity sensors and send them to the Raspberry Pi.
  The one that controls the electrovalves receives the messages from the Raspberry Pi and opens or closes accordingly the gates of the MOSFETs. The messages to send include both the duty cycle of the command and the axis to actuate.
  While the second one deals with the ToF sensors by reading the data, correcting the bias of the sensors and sending the information to the Raspberry Pi. In summary, the Raspberry Pi deals with Soft Real-Time tasks, while the Arduino boards are needed for Hard Real-Time tasks.

All components are protected with dedicated fuses. The considerations and solutions on safety and hazard management as well as the overall design have been carried out according to the Novespace Guidelines.

The proximity navigation software allows the Chaser to perform the manoeuvre autonomously. To better understand how it works, the control software can be divided into three main levels that deal with different tasks. These Levels have been called respectively Low, Medium and High Level.

- **Low-Level control**
  The Low-Level deals with the control of the electrovalve. This part of the code runs in the Arduino board dedicated to these eight actuators. It is composed of a routine that checks if there are any commands sent by the upper level, then if so it actuates the MOSFETs accordingly. The code is intentionally extremely simple because it must be very quick since the reaction time of the electrovalves is around $4ms$. Therefore any error in the duration of the actuators due to the code must have been lower than this value, consequently a very fast and simple code was a necessity.

- **Medium-Level control**
  The Medium-Level is the more complex out of all three parts. It deals with three very important tasks: (1) Target localization and pose estimation, (2) path computation, and finally (3) commands calculation and delivery.
  (1) The localization task uses the camera data to localize a set of four Apriltags ([11], [13], [16]) placed on the Target frontal face (figure 2.10). From the Detection algorithm, the relative position and attitude of the Target are calculated. Then, the velocities are calculated by deriving the pose provided by the computer vision algorithms. The information obtained from the camera are then improved thanks to the readings from the ToFs thanks to a sensor fusion. It is important to underline that the precision of the localization depends also on the reflectance of the Target other than its distance, which is why the Target has a white frontal face.
  (2) Then the path computation takes place thanks to the data acquired by the localization and the pose estimation. The trajectory is obtained by evaluating a path that can align two precise systems of reference: one is the system of reference of the Chaser, presented in the previous subsection, which is centred in the centre of mass, has the x-axis perpendicular to the frontal face, z-axis along the gravity direction and y-axis to close the tern; while the other is the system of reference body of the Target centred in the centre of

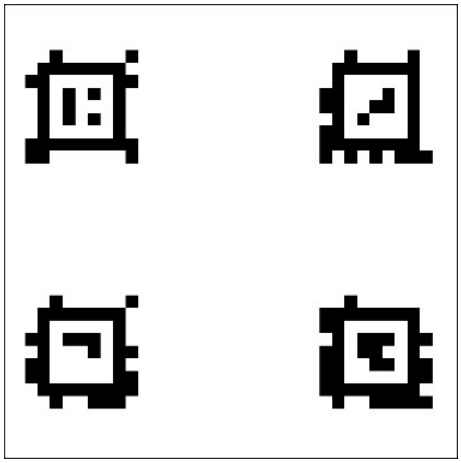

Figure 2.10: Apriltags placed on the face of the Target

mass of the Target and with the axes that are defined with the same logic as in the Chaser.
(3) The trajectory is then divided into a series of small corrections of just one axis at a time. Although the propulsive system is way capable to perform more than one axis correction at a time, the software has been voluntarily simplified to assure the simplicity of the control. After being calculated the commands are sent to the Low-Level to be actuated.

The choice of which particular axis to correct first follows simple rules. The main objective is to follow the path computed; however, the misalignment of every axis is always checked with respect to the tolerances. The definition of these tolerances or constraints ensures that not every small misalignment is correct. The priority of a possible command to send over another is given based on a series of parameters found empirically. Some of these parameters act similarly to the proportional coefficients in the PID-type of control, others instead are only constraints of the software useful to force a certain type of behaviour.
The parameters are:

1. **Authority for trajectory and attitude corrections**
   This category comprehends six parameters, each referring to a specific DoF (translations or rotations along x, y and z). These parameters indicate how strong a certain correction along one particular axis should be. Technically, it is just an empirical proportional coefficient to apply to the theoretical number of cycles used to correct a certain trajectory misalignment. This coefficient depends mostly on the value of thrust along

each axis, which is indeed different due to the particular thrusters configuration, but they have also been tuned to get the desired results.
For example, if the translations are considered, it can be stated that the y-axis and z-axis have a lower value with the respect to the x-axis due to the physical difference in thrust and the fact that the mock-up has to correct smaller misalignment along that two axis while having to cover more space along the x-axis. Instead regarding the rotations, the x-axis (roll) and z-axis (yaw) have a lower value with the respect to the y-axis (pitch) due to the physical difference in thrust and consequently in torque.

2. **Weight of trajectory and attitude corrections**
   This category comprehends six parameters, each referring to a specific DoF (translations or rotations along x, y and z). These parameters refer to how much important is considered correcting one DoF with the respect to the other possible corrections given the same percentual misalignment with the respect to the perfect condition.
   For example, the translation along the x-axis has the biggest weight because otherwise, the Chaser would not approach the Target, rather it would continue correcting just misalignment of attitude or trajectory along the y and z-axis. Moreover, due to the symmetry of the system, to the y and z-axis, the same weight is given, since there are not any preferred directions between these two DoFs. Regarding the rotations, to y (pitch) and z-axis (yaw), the same weight is given due to the symmetry of the approach, where there is no preferred axis of rotation. Instead, to the x-axis (roll) the least importance is given because of the symmetry of the docking interface, which allows withstanding the biggest misalignment out of the three attitude angles.

3. **Approaching threshold**
   This parameter refers to the minimum distance at which the Chaser has to engage a control with higher frequency commands to dock more precisely. Nevertheless, if a high misalignment is detected, the Chaser can accelerate the control even before this threshold. Including this eventuality was necessary because of the difference in the number of DoFs between testing on the frictionless table and testing on the parabolic flight. This distance has been set to be around $70mm$ (face-to-face).

4. **Distance at which tolerances decrease linearly**
   This parameter refers to the distance at which the tolerances of the ap-

proach phase are reduced linearly up to the docking ones, which are clearly stricter. This linear decreasing trend describes a 3D cone of approach, which is a standard and, most importantly, very simple type of approach. It is used to be sure that the Chaser can be more misaligned at the beginning due to not having extremely strict tolerances since the beginning of the manoeuvre. This distance is set to be $150mm$ (face-to-face).

In summary, the objective of the Medium-Level is to apply translations and rotations to the system of reference of the Chaser with the respect to the one of the Target. The "docked conditions" are reached when the origins are $190mm$ apart, and the y and z-axis are aligned along the same directions between Target and Chaser ($\pm 7mm$ and $\pm 5°$). The distance that should separate the two origins takes into account the dimension of the docking interface. In the case of the probe-drogue one, which is $40mm$ long, the distance is $190mm$(see figure 2.11 for the "docked condition"), while in the case of the androgynous is $170mm$.

Figure 2.11: Chaser and Target in docked configuration

- **High-Level control**
  The High-Level is the easiest of the three in terms of role since it deals only with:

  1. receiving of the messages about the beginning of the manoeuvre from the laptops;

2. sending back information about the position of the Target and the trajectory to follow. These information compose the log of the Chaser that is then saved for post-processing purposes.

2.1.3 Target GNC

The complexity of the GNC of the Target is consistently reduced with the respect to the one of the Chaser. The difference clearly lies in the disparity between the role in the manoeuvre of the two mock-ups. In fact while the Chaser has to perform complex 3D manoeuvring, the Target has just to control its attitude and behave cooperatively. It is important to underline that any type of communication between Target and Chaser has not been established, therefore the Target does not perform corrections based on the commands received by the Chaser, whether it just acts by contrasting disturbances and maintaining a fixed attitude with respect to the initial alignment.

⋆ Actuators

Since the Target has to control just the DoF related to the rotations, a set of 3 reaction wheels (RW) has been implemented. Each reaction wheel is composed of a flat brushless DC motor and a flywheel of about $0.02kg$ made to have enough torque for attitude control. The flywheel is a hollow cylinder made of aluminium, that has been manufactured by lathing. Each RW has the rotation axis along one of the directions perpendicular to the faces, that are mutually perpendicular (figure 2.12).
To control the RWs a simple code has been programmed for an Arduino board: this code implements a Proportional-Integrative-Derivative controller (PID) that aims at reducing the misalignment (error) regarding the attitude with the respect to the initial alignment. The initial alignment is measured when the Target receives the message about the beginning of the manoeuvre from the laptop.
The safety and hazard management for these actuators focuses mainly on the possibility of the flywheel detaching and becoming a dangerous free-flying object. Therefore in order to avoid this eventuality, a cover for the RW made of PLA has been 3d-printed. Resistance tests of the covers have been carried out according to Novespace Guidelines regarding free-flying objects.

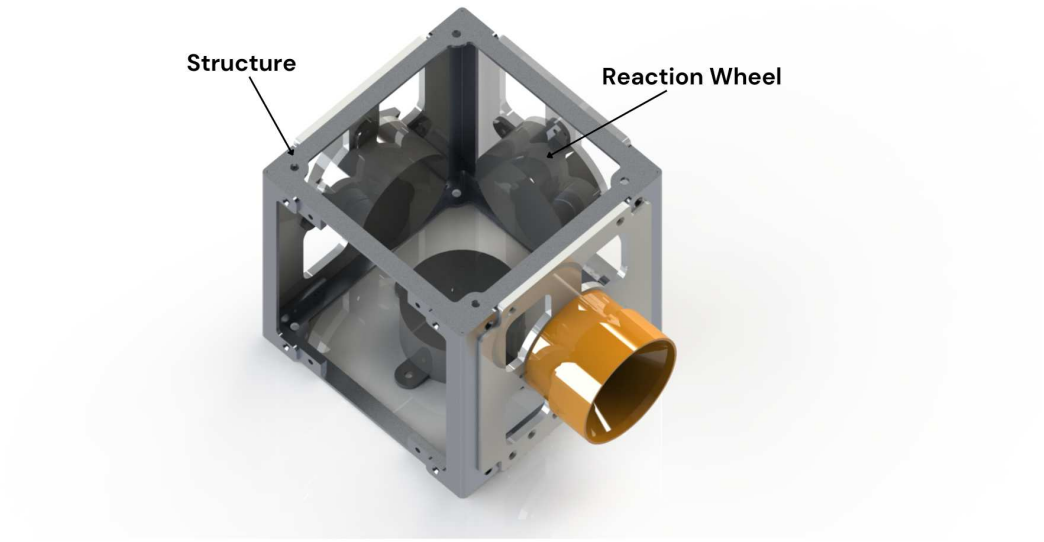

Figure 2.12: Target - Reaction wheels configuration

⋆ On Board Computer System and Electronics

Consequently to the simplicity of the actuators, the On Board Computer System (OBCS) of the Target has a pretty straightforward design. As in the case of the Chaser, firstly the list of all electronic components is presented, then the code programmed to control the actuators is discussed.
The list of components is the one that follows:

- **Battery pack**: It is needed to power all the system components. Since the components do not have high power consumption and voltages requirement, a simple pack of 5 batteries with a total voltage output of $6V$ has been used. The battery pack has been chosen made by NiMh cells for the same reasons that drove the choice of the Chaser battery pack. Moreover, also the implementation of a switch and a control panel has common reasons.

- **Arduino board**: It is needed to process the data and run the code to control the three RWs.

- **Arduino Shield**: It is needed to translate the commands to the RWs sent by the Arduino board in electrical pulses to send directly to the motors. It is directly connected to the power of the DC motors and plugged into the Arduino board.

- **Brushless DC Motors**: The three Brushless DC Motors are needed to rotate the flywheel and induce the rotation of the Target as a direct effect due to

the conservation of the moment of inertia. They are directly connected to the Arduino Shield, that powers and control them.

- **Inertial measurement unit (IMU)**: It is used to register the data regarding the velocity of rotation and accelerations of the mock-up. The data acquired are sent directly to the Arduino board that processes it to obtain information about the attitude of the Target. It is directly connected to the Arduino.

- **Bluetooth Module**: It is used to communicate with the laptops in order to receive the command that the experiment has started. The information acquired is sent directly to the Arduino board. It is directly connected to the Arduino. The type of connection is different than the one implemented on the Chaser (ssh communication) because the Arduino can not support the same type of connection.

All components are protected with dedicated fuses. The considerations and solutions on safety and hazards management as well as the overall design have been carried out according to the Novespace Guidelines as in the case of the Chaser.

The code programmed to control the RW of the Target is based on a high-level feedback loop on the attitude of the mock-up with also an internal feedback loop on the current. The output data from the external loop is the velocity of the RW that is sent directly to the board that controls the motors. For the external feedback loop, it uses the data acquired from the IMU sensor. Before entering the loop, the data from the IMU get filtered with a low pass filter to eliminate high frequency fluctuations.
The controller implemented is a PID (K_{prop}-K_{int}-K_{der}) based on the error in the attitude (ϵ), that gives the torque (M) as shown in equation 2.1.6.

$$M = K_{prop} \cdot \epsilon + K_{int} \cdot \frac{\epsilon}{s} + K_{der} \cdot s \cdot \epsilon \tag{2.1.6}$$

With proper design of the integrative controller (K_{int2}) of the internal current loop, the transfer function of the DC motor results in a low pass filter that transform the torque from the PID error into the derivative of the angular momentum of the RW ($\dot{h}_w$) as shown in equation 2.1.7.

$$\frac{\dot{h}_w}{M} = \frac{1}{1 + \frac{s \cdot R_m}{K_{int2}}} \tag{2.1.7}$$

From this value, the velocity of the RW is found by integrating and dividing by the momentum of inertia of the RW (I_w). Saturation blocks are also included to ensure that the commands do not exceed the maximum motor authority. Since the assumption of angular velocities is valid, the gyroscopic couplings between the three rotations in the Euler equations (2.1.8) have been removed, in order to simplify the equation deeply (see the final system of equation in 2.1.9. Moreover, to support such a strong hypothesis, it must be highlighted that the moments of inertia of the mock-ups are all very similar, therefore the cross products in the equation such $(I_i - I_j)$ are very small.

$$\begin{cases} I_x\dot{\omega}_x + (I_z - I_y)\omega_y\omega_z &= M_x \\ I_y\dot{\omega}_y + (I_x - I_z)\omega_z\omega_x &= M_y \\ I_z\dot{\omega}_z + (I_y - I_x)\omega_x\omega_y &= M_z \end{cases} \tag{2.1.8}$$

$$\begin{cases} I_x\dot{\omega}_x = M_x \\ I_y\dot{\omega}_y = M_y \\ I_z\dot{\omega}_z = M_z \end{cases} \tag{2.1.9}$$

To test the control system of the Target was dedicated less time with the respect to the Chaser due to the clear disparity in complexity. Nevertheless, it was important to test it prior to the end of the testing of the ML of the Chaser so that the integration tests could take place. The testing procedure of the Target has been a series of validation of the controller chosen focusing also on better tuning the proportional factors in the PID.

2.1.4 Docking Interfaces

Two different configurations have been planned to be tested during the parabolic flights. In particular, the two interfaces are a probe-drogue configuration and an androgynous configuration. The two configurations differ in geometry but have few features in common in terms of the type of connection post docking and in terms of solution for the design in view of facilitating the manoeuvre.
The connection between the two mock-ups is granted thanks to a small permanent magnet placed on the Chaser This magnet attaches to a ferromagnetic counterpart on the Target. Initially, both docking interfaces had been designed with a mechanical connection. The connection was obtained through a rotating component which would have locked the two interfaces. However, it has been replaced by a magnetic constraint because of risks and hazards analysis. In fact, the shaft of the servomotor used to rotate the mechanical locking components could have been severely damaged if hit directly, or just by falling during the hyper-gravity phase. Therefore, in accordance with the Novespace Guidelines, it has been replaced with a non-mechanical one.
The probe-drogue configuration is characterized by a probe on the Chaser and a drogue on the Target. The shape of the probe-drogue configuration facilitates the manoeuvre by permitting a sliding motion of the two interfaces in order to better align the two mock-ups. The probe-drogue miniaturized docking interface is shown in 2.13. The difference between the probe-drogue configuration used in ERMES and the one described in ([6]) is the above-mentioned magnetic constraint, in fact the tip of the probe has been replaced with a small magnet, in order to move from an active mechanical docking to a passive soft docking solution.

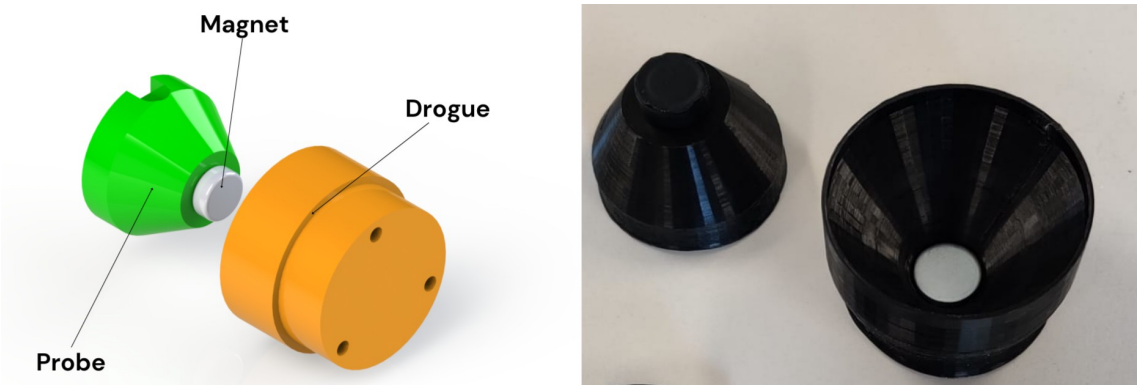

Figure 2.13: Probe-Drogue docking interface

Whereas in the androgynous configuration (shown in figure 2.14.) the docking interface is symmetric, therefore it is the same for both Chaser and Target. The

connection between the mock-ups is achieved thanks to the magnet and its counterpart, which are both placed at the centre of the interface. This configuration too has been designed to have a shape that helps the docking by auto-centring the interfaces.

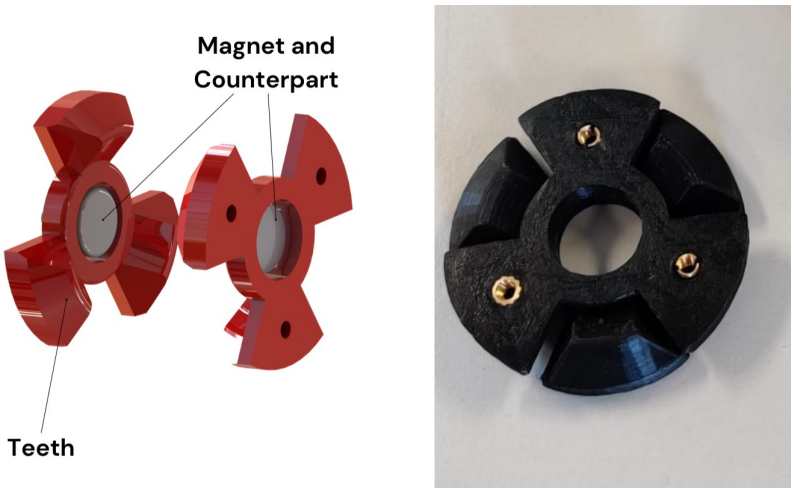

Figure 2.14: Androgynous docking interface

2.2 Release Structure

The Release Structures (figure 2.15) have the main purpose of releasing the mock-ups during the low-gravity phase and sustaining them during the hyper-gravity phase. The two Release Structures, one dedicated to the Chaser and one to the Target, are referred to respectively as ”Rack#1” and ”Rack#2”.
At the beginning of the low-gravity phase, both mock-ups are fixed in their initial position by small electromagnets, that keep them attached to the release interface. When the message regarding the starting of the manoeuvre is received, the electromagnets are turned off in order to release the mock-ups in a free-floating condition. The Release is obtained thanks to Slider Type Electric Actuator with Built-in Controller (from now on referred to as ”Slider”), that uses a sled attached to a motorized endless screw to accelerate or just position the two mock-ups. After the release, the Slider retracts in order to increase the space available for the manoeuvre. In the following subsections, firstly the designs of the mechanical frame and of the Release Interface are presented, and then the Release Mechanism is discussed.

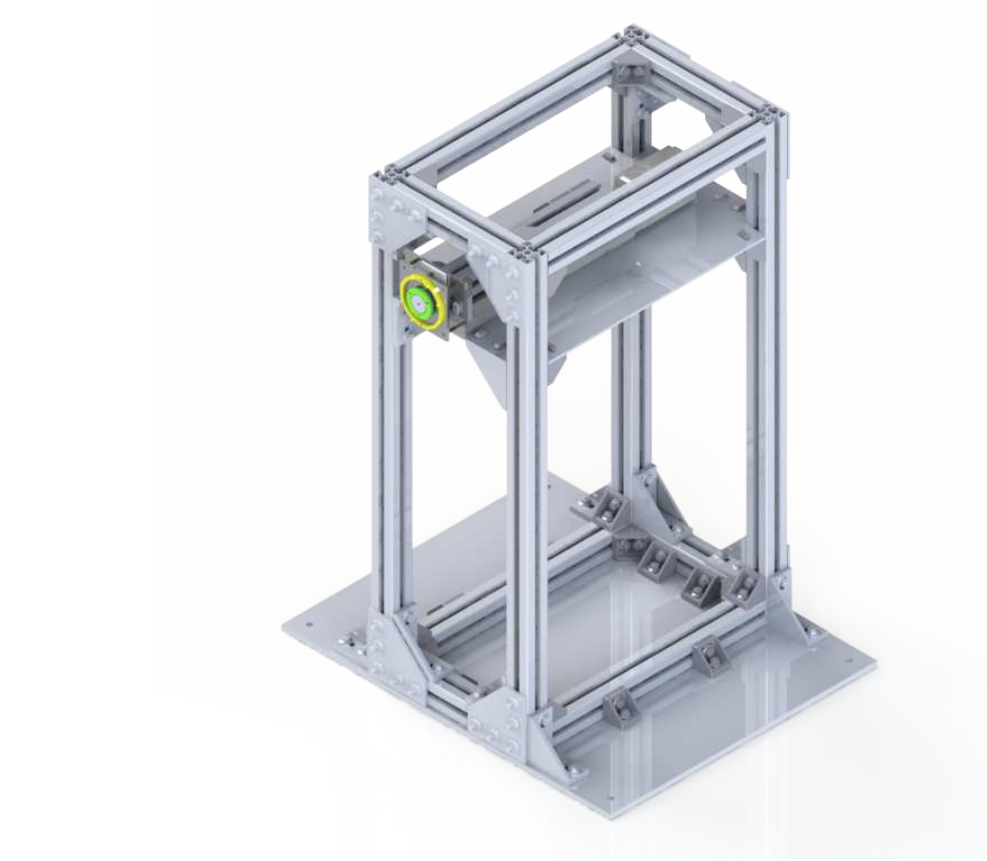

Figure 2.15: Release Structure

2.2.1 Structure

The metallic frame of the Release Structure is composed of Bosch profiles connected with angular Bosch connectors. Moreover, it features two plates made of aluminium: one as a light shelf (to which the Slider is connected); the second one as a base-plate to fix the rack to the plane surface. The base-plate is anchored to the railways of the plane with four $M12$ screws. The first has a thickness of $5mm$, while the last one of $10mm$. Its design, in terms of the type of profile used, masses balance, protections implemented and overall dimensions strictly follows the Novespace Guidelines. The mechanical design of the two Racks is the same because they have the same purpose, which is to support the Slider and lock it in place.
The aim of the metallic frame is to sustain the Slider at the highest position possible. The light shelves have been placed to a height of $720mm$ from the base-plate of the racks; the decision to place it above half of the structure was made so that the tests take place as high as possible from the plane surface always considering that the overall structure is still compliant to the general rules imposed by Novespace Guidelines. In the final design, the resulting centre of mass is at $351mm$, which is lower than half of the total height.

2.2.2 Release Interface

The Release interface (shown in figure 2.16) is composed of a holding system and a centring structure. The holding system consists of the electromagnets above mentioned, which are mounted directly to a metallic plate connected to the Slider with Bosch profiles and brackets.
In order to choose the correct electromagnet, it has been considered the worst condition scenario regarding the amount of weight to sustain as a safety factor. The worst case scenario happens during the hyper-gravity when the mass of the mock-up is around twice the initial one (from $2.5kg$ to $5kg$ for the Chaser and from $0.8kg$ to $1.6kg$ for the Target). This consideration includes the effect of the axial, shear and bending forces on the electromagnet. The selected electromagnets provide $150N$ ($15kg$) of holding force for the Target and $280N$ ($28kg$) for the Chaser. The total safety factor behind this choice is around 2 for both mock-ups. These electromagnets are anchored to a simple structure connected to the Slider with a Bosch profile and relative brackets. The magnetic connection between the electromagnets and the face of the mock-ups is obtained thanks to a ferromagnetic metallic plate attached to the face of the mock-ups that acts as a direct counterpart

for the electromagnets.
The last component of the Release Interface is a centring component, that has been 3d-printed in PLA. This centring component is placed on the face of the mock-ups facing the Release Interface and extends farther than the metallic plate. It penetrates the upper plate of the Release Interface which has a hole that acts as its counterpart. This simple structure is characterized by three teeth (2.17) that have been designed to help to centre the mock-ups and avoid unwanted rotations.

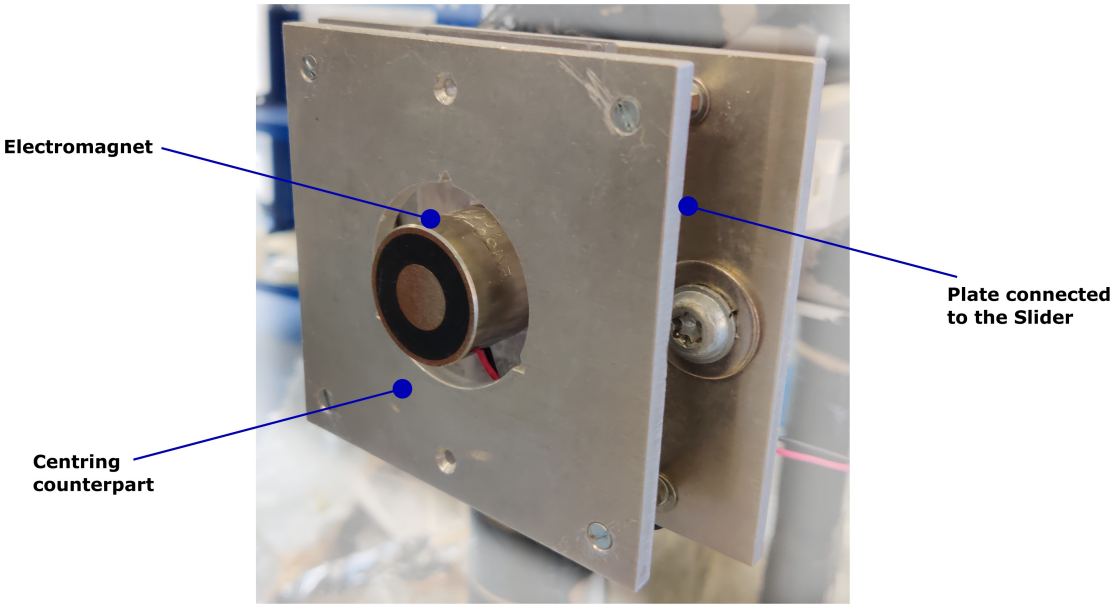

Figure 2.16: Release Structure - Release Interface

Figure 2.17: Release Structure - Centring component of the Release Interface

2.2.3 Release Mechanism

This section is dedicated to how the Slider has been powered and controlled. The list of components needed for the Release includes:

- A power supply to power the Slider. It provides $24V$ and exactly $2A$ to each Slider.

- A power supply to power the electromagnets. It provides $24V$ and around $0.5A$ in total for both electromagnets.

- The Slider itself. In particular, a set of cables are needed to power and control it (reported in figure 2.18). To actuate the control commands the voltage applied to that particular cable must be raised to $24V$.

- The electromagnets for the magnetic constraints. They require $24V$ and around $0.5A$ in total.

- An Arduino board to control the Slider and the electromagnets. It simply recalls pre-chosen trajectory uploaded into the Sliders directly from their proprietary software running in the laptop using a standard serial connection via USB. To recall the position the Arduino simply opens the gates of dedicated MOSFETs. Moreover, the Arduino board is needed to perform Real-time tasks such as synchronising the movement of the sled with the magnetic release.
  The Arduino board is powered directly from the laptops with a standard USB connection. This USB connection is used also to plot the real-time position of the sled.

- A series of MOSFETs P-type, that are characterized by the necessity to put the load after the ground pin, is connected to the Arduino with an electrical board. The loads in this case are the cables of the positions or commands (such as the safety break) of the Slider. The difference with the N-type is the position of the load, which consequently impacts the voltage needed to open the gate that is very high because the voltage drop after the ground pin is $18V$.

- An electrical cabinet in which all the cables of the slider are connected to the power supply or to the Arduino board.

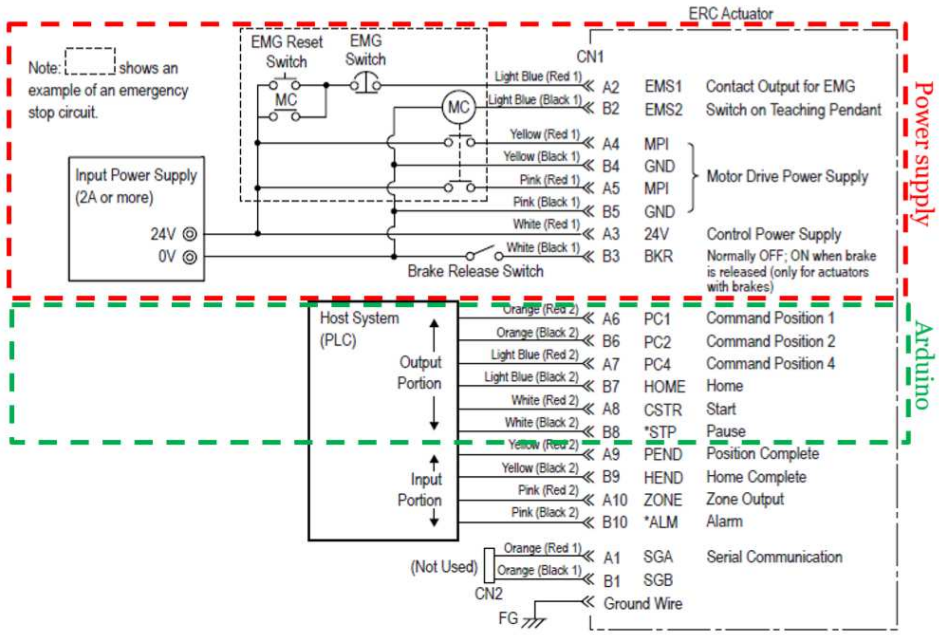

Figure 2.18: Scheme of the electrical connections of the Slider

Finally, the type of releases programmed into the Arduino code must be described. The releases are based just on simple commands that recall the positions as explained above.

Two types of solutions have been implemented: the first one regards ”static releases”, in which both mock-ups are released statically; while the second one includes ”pushing releases”, in which the Chaser is accelerating before the magnetic disconnection, while the Target is released statically.

The Slider can move the sled to any position along its length. In particular, three positions have been programmed into the Slider. The first one called ”HOME” or ”Position 0”, is the position in which the sled is completely retracted, therefore is the last position of the release procedure. In fact, after the magnetic release, the Slider retracts automatically to the HOME position. The other two are called ”Position 1” and ”Position 2”. ”Position 1” is nearly halfway between ”Position 2” and HOME. It is needed in case of releases in which the Chaser is placed further to Target. ”Position 2” instead is achieved when the Slider is fully extended. It is needed in case of releases in which the Chaser is placed closer to the Target. In summary, six possible releases have been programmed to be used during the flights (see figure 2.19).

The six releases are the ones that follow:

- **Static Release - Mod#1**: Both Chaser and Target are released at Position 1, then the sleds come back at Position 0.
- **Static Release - Mod#2**: The Chaser is released at Position 2, while Target is at Position 1, then the sleds come back at Position O.
- **Static Release - Mod#3**: Both Chaser and Target are released at Position 2, then the sleds come back at Position O.
- **Pushing Release - Mod#4**: The Chaser is accelerated and released at Position 1, while the Target is released statically at Position 1, then the sleds come back at Position O.
- **Pushing Release - Mod#5**: The Chaser is accelerated and released at Position 2 (same velocity as Mod#4, but closer to the Target), while the Target is released statically at Position 1, then the sleds come back at Position O.
- **Pushing Release - Mod#6**: The Chaser is accelerated and released at Position 2 (same velocity as Mod#4, but closer to the Target), while the Target is released statically at Position 2, then the sleds come back at Position O.

The variant releases have been programmed in order to adapt as fast as possible to the possible difficulties encountered during the flight. With the same logic hybrids between all these six "main" releases have been prepared.
It is important to underline that, even though each release may differ in position recalled by the software at a certain time, they all last the same amount of time. The duration of the release considers both positioning and eventual acceleration. Instead, the possible discrepancy between the release of the Chaser and the Target has been measured at less than $1ms$.
The reason for having all the releases last a certain value is that mock-ups need to be released at a precise moment. In particular, the exact moment chosen for the release is set to be around the middle of a parabola because it is when the low gravity is at the lowest. Precisely, the release happens around $8-10s$ from the start of the low-gravity phase. The discrepancy of two seconds is a versatile parameter to be changed even during the flights in order to adapt better to the particular day of flights. The choices made about the design of the software for the release are based on information provided by Novespace.

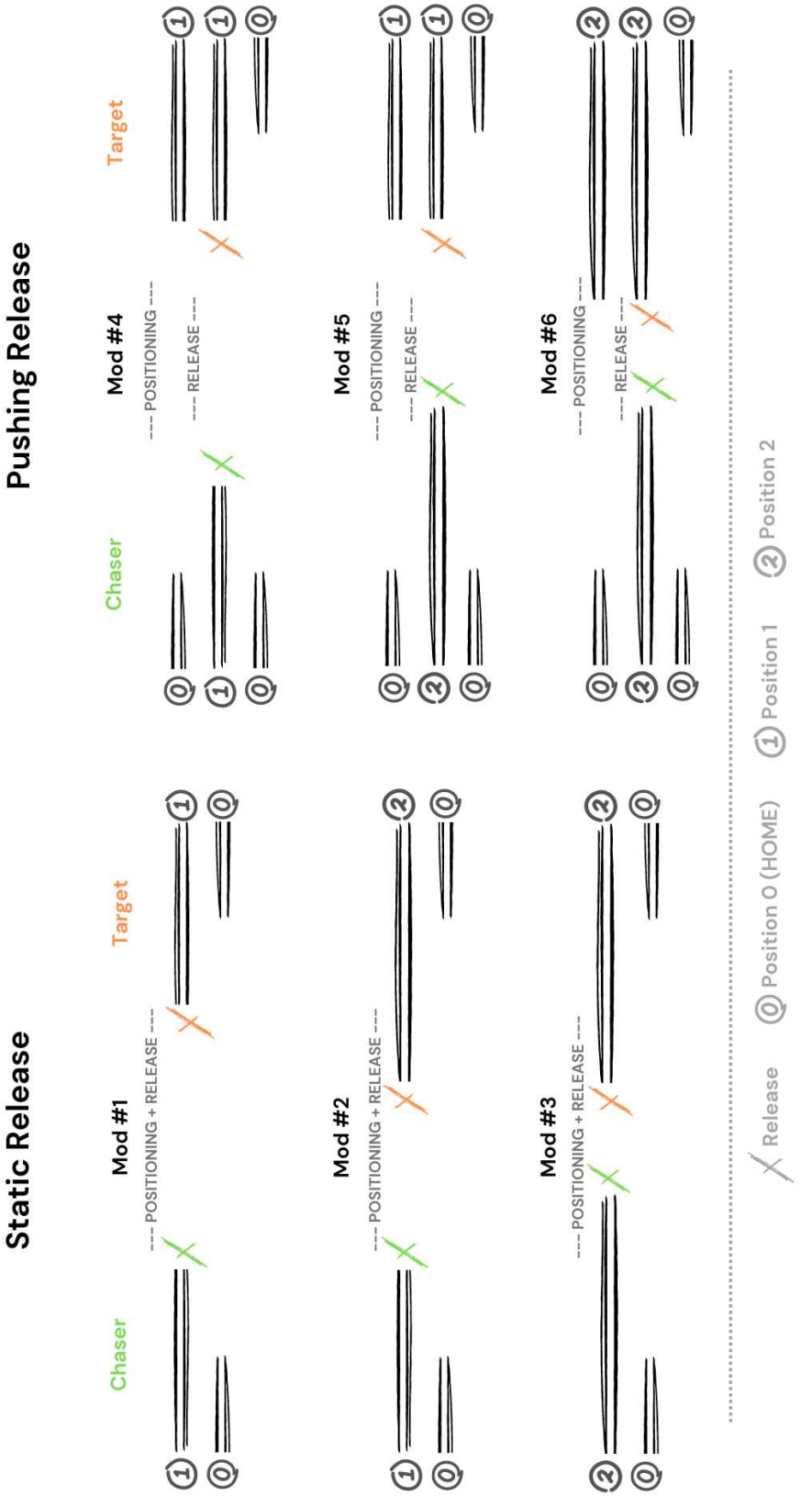

Figure 2.19: Schematic of the release modes

2.3 Cabin Layout

Aboard the Airbus A310, a large number of experiments take place, therefore the space dedicated to each experiment is extremely precise. The choice regarding both the Cabin Layout and position in the plane of ERMES has been an interface requirement of the experiment. These important decisions are made by Novespace. Taking into consideration its placement, the aim is to increase the free-floating time for both mock-ups. In order to achieve this goal, the space dedicated to the entire experiment has been optimized to have the 3-dimensional free-floating area as wide as possible. This has been achieved thanks to a few logistic solutions.
A total area of $2 \times 2 \times 2m^3$ was avaible as a free-floating area. This area has been surrounded by a five-sided net. In this space, the two Racks have been allocated along the side of the net facing the cockpit of the plane. Moreover, the entire space dedicated to ERMES is placed towards the cockpit, after the centre of mass of the plane, so that the motion of the plane would induce a backwards motion of the mock-ups. This effect would have guaranteed that the mock-ups would not hit the side of the net near them during the release but, instead, they would have moved towards the tail of the plane. The access to the floor between the two Release Structures has been prevented by putting a smaller elastic net from one Rack to another. The remaining part of the floor has been covered with a soft and thick mattress to avoid damage due to the falling of the mock-ups during the hyper-gravity phase. The access to the free-floating area is allowed only during the standard gravity phase and just for one experimenter only.
All the electronics to power the experiment and laptops to monitor it are anchored to a single small base-plate (called "BasePlate#1"), itself anchored to the floor thanks to the plane railways, outside of the above-mentioned free floating area in order to occupy less space. Bags dedicated to expandable components, such as for example CO_2 cartridges or batteries have been anchored outside the net to the BasePlate#1 with Velcro strips.
Alongside the BasePlate, two experimenters of ERMES are seated: they avoid free-flying uncontrollably by fastening their seatbelts, which are attached to the railways of the plane.
All the safety and hazard management solutions above discussed, such as the net surrounding the free-floating area, the mattress on the floor and the seatbelts to protect the experimenters, refer to the Novespace Guidelines regarding Cabin Layout designs.

Finally, three external cameras (referred to as "reference cameras") have been anchored to a handrail of the plane or the Release Structure. These have been used for post-processing analysis or for outreach purposes. In particular, reference cameras #1 and #2 are dedicated to recording the experiment, while #3 is for recording the two experimenters.
The final Cabin Layout is reported in figure 2.20.

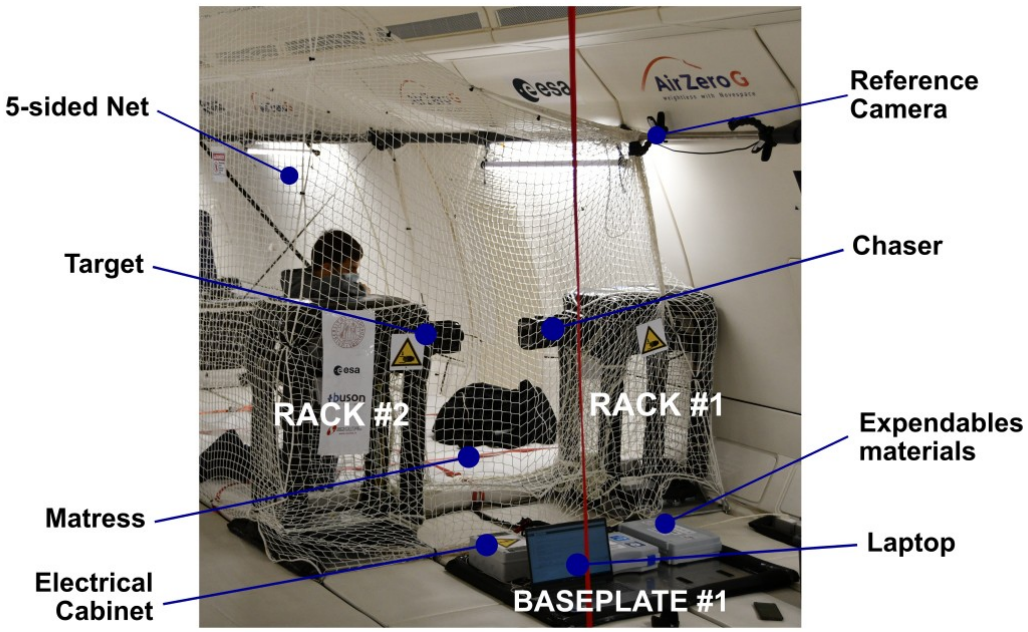

Figure 2.20: ERMES Cabin Layout on board the Airbus A310

2.4 Experimental Procedure

This section deals with two important topics: Manoeuvre Sequence and Flight Activities. The first one focuses on describing the sequence of events that compose a docking manoeuvre, while the second one deals with the tasks that the two experimenters of ERMES have to perform during the parabola. These two together compose the Experiment Procedure of ERMES (shown in figure 2.21).

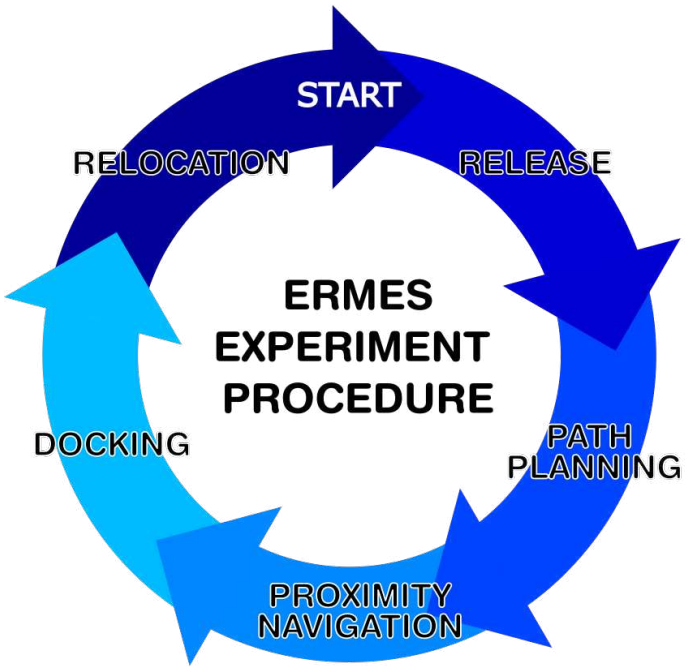

Figure 2.21: ERMES Experiment Procedure Scheme

2.4.1 Manoeuvre Sequence

The Manoeuvre is composed of four main phases:

- **Release phase**: the mock-ups are released from their initial electromagnetic constraints into a free-flying condition by actuating the Slider. The type of release is chosen by the Experimenters.
  This phase lasts exactly $8s$, no matter the type of release as explained in the section dedicated to the software of the release 2.2.3. It starts exactly when the signal of the low-gravity phase starting is delivered by the Novespace crew onboard. Therefore the actual release happens in the middle of the low-gravity phase, precisely when the g-levels are the lowest.
  The duration of this phase can be changed by the Experimenters even during the flights acting on the software of release by changing directly the duration of the waiting time prior to the release itself.

- **Path planning phase**: the Chaser localizes the Target and computes the trajectory to reach it. The duration of this phase depends on the condition after the release in terms of misalignment or relative position and velocity.
  Generally, this phase lasts less than $1s$. The end of this phase is characterized by the sending of the first command to the Low-Level by the Medium-Level.

- **Proximity navigation phase**: the Chaser approaches the Target by controlling its relative position and attitude actuating its thrusters according to the commands sent from the Medium-Level to the Low-Level. In the meanwhile, the Target contrasts disturbances in order to maintain the initial alignment by actuating its RW.
  This phase lasts from $5 - 7s$ depending on the type of release chosen. In fact, as explained in the section dedicated to the Release Software (2.2.3), pushing-type releases accelerate the Chaser and consequently reduce the time needed to reach the Target.

- **Docking phase**: the Chaser performs the navigation needed to correctly dock to the Target. The phase ends when the two mock-ups are connected by the magnetic constraint.
  It lasts between $1s$ and $3s$ depending on the type of release chosen.

The average total time needed for the free-floating manoeuvre is around $8s$. This duration has been chosen according to the advice from Novespace regarding the maximum time for the free-floating objects before hitting the ground, ceiling, net

or walls of the plane. As mentioned previously, the release phase is not included in the calculation because its duration has been programmed in order to release at an exact moment of the parabola according to the advice from Novespace.

2.4.2 Flight Activities

For what concerns the actual activities during the flight, the tasks should be divided according to the phases marked by the changes in g-levels (as explained in 1.3.1). Taking into consideration one parabola (from the standard gravity phase to the standard gravity phase of the next one), the procedure that the two Experimenters must follow is the following:

- **Standard-gravity phase**: During this phase the preparation prior to the manoeuvre initialization takes place.
  Firstly, Experimenter#1 enters inside the net and locates the mock-ups on the electromagnetic constraints. However, during the long break of $5min$ or $8min$, Experimenter#1 has also to check for the voltage of the battery of both mock-ups and in case change it before placing them on the magnets. Moreover, every break the cartridge of CO_2 has to be changed. In fact, although it is possible to change the cartridge during the standard-gravity phase between two parabolas since the task does not require much time to be completed, this task has been programmed only during the long breaks in order to decrease the number of tasks to perform during the short ones between parabolas.
  Meanwhile, Experimenter#2 monitors the overall state of the experiment from the laptop. He activates the electromagnets to lock the two mock-ups. Then, after the preparation for the experiment has been completed, each Experimenter sits near the laptop waiting for the start of the parabola. Before the starting of the hyper-gravity phase, each Experimenter fastens his seatbelts to assure to not free-float uncontrollably during the low-gravity phase.

- **Low-gravity phase**: The low-gravity phases start when the "injection signal" is given by the plane pilot. When the signal is given the Experimenters initiate the experiment by sending the High-Level commands. Afterwards, the manoeuvre is completely autonomous. The manoeuvre starts whenever the timer reaches a certain moment as explained in the previous section. In the meanwhile, each Experimenter sits near the laptop, both anchored firmly in place with seatbelts.

- **Hyper-gravity phase**: During the hyper-gravity phase, both Experimenters are not involved in any activity. The Experimenters sit near the laptop, both anchored firmly in place with seatbelts.

Chapter 3

Results

This chapter focuses on the validations from the on-ground testing carried out on a low-friction table and the results and outcomes of the 79^{th} ESA Parabolic Flight Campaign. In particular, firstly the data collected during the tests are presented and described (section 3.1), then a section focuses on presenting the post-processing procedure (section 3.2) and, finally, the interest shifts towards discussing the results of the testing perform in the laboratory (section 3.3) and the results from the 79^{th} ESA Parabolic Flight Campaign (section 3.4).

3.1 Experimental Data

The data collected during the tests (either on-ground or on the parabolic flights) regard the behaviour of the Chaser during the manoeuvres, in terms of interactions between Medium and Low-Level of the Software, the video recordings needed to add an external reference. Moreover, after the Campaign some other data regarding g-levels, trajectory of the plane etc. are provided by Novespace.
In particular the data stored include:

- **ToFs acquisition**: they include the information derived from the three Proximity sensors mounted on the front face of the Chaser.
- **Apriltag Pose**: they include the information derived from camera data acquisition used to find the relative position, attitude and velocity between Target and Chaser.
- **Medium-Level to Low-level commands**: they include a recollection of all the commands sent by the Medium-Level to the Low-Level of the Chaser to be actuated.
- **Data from the external cameras**: they include a recollection of all the video recordings from all 3 cameras during the 93 parabolas.
- **Data regarding the g-levels**: they include the data provided by Novespace regarding the g-levels and trends in every parabola of every flight.

The information provided by the Chaser, consisting of the position and attitude of the Chaser relative to the Target and commands actuated, are referred to an absolute time reference (Greenwich) in accordance to the one used for the camera recordings. The matrix containing the relative pose of the Chaser is expressed as: $[time; x; y; z; \phi; \theta; \psi]$. These data are defined in the reference system centred in the centre of mass of the Chaser, with the x-axis that starts from the centre of mass of the Chaser and perpendicular to the frontal face, the y-axis on the horizontal plane parallel to the ground and the z-axis to close the right-handed triad as shown in figure 3.1. The one of the Target is constructed equivalently.
When the manoeuvre ends, the commands sent to the Low-Level are saved inside the internal memory of the Chaser. They are saves as a matrix, with each column referring to one command; the information reported includes: the time of actuation, the axis to correct and the duty cycle of the electrovalve.

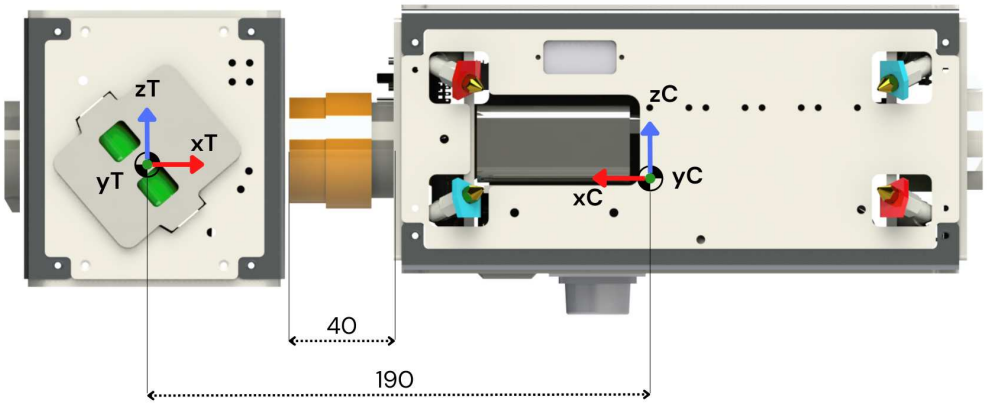

Figure 3.1: System of reference of the Chaser and Target

3.2 Data Analysis Procedure

All the data collected are then used together in the post-processing phase to obtain a comprehensive experiment data analysis. In particular, the analysis of the manoeuvre during the flights or in the laboratory is divided into two complementary sections: the "Medium-level Analysis" and the "Camera Recording Analysis".
The first takes into account all the handling of the data from the mock-ups (sensors, camera, software etc) and it focuses on whether the relative trajectory computed has been correctly followed by the Chaser. In particular, it discusses the choices of the commands sent to the Low-level.
The second one introduces an external reference to the trajectory computed by the Chaser thanks to the data from the external reference camera. While in the parabolic flights, 3 cameras have been placed to record the experiment, in the laboratory the external reference system is given by a Motion capture technology (in particular the OptiTrack technology has been used) that helped track the Chaser and the target precisely.

3.2.1 Medium-level Analysis

The main objective of the Medium-level Analysis is to investigate the functionality of the Medium-level of the proximity navigation software. In particular, the aim is to use the data collected by the OBCS of the Chaser in order to recreate

and analyse singularly the manoeuvre performed in each test. Then, thanks to the introduction of the Medium-level to Low-Level commands, the interest shifts towards studying the efficiency of the choices of the Medium-level. This task can be carried out by highlighting the effect of each command on the attitude and momentum of the Chaser, and by comparing its choices to a different kind of manoeuvring.
The output of this Analysis is composed of various graphs about the DoFs of the Chaser (six for the tests performed on the parabolic flight $[x; y; z; \phi; \theta; \psi]$ and three for the on-ground testing $[x; y; \psi]$) and as many for the velocities of these DoFs. These graphs are then discussed in order to find the strengths and weaknesses of the software.

3.2.2 Camera Recording Analysis

The second part of the post-processing, called "Camera Recordings Analysis", has the objective to introduce the reconstruction of the manoeuvre from the external reference represented by video recording from the external cameras. The tracking data from the video gives an important feature in order to help evaluate the precision and the functionality of all the systems, taking into consideration also the efficiency of the integration of all subsystems involved in the manoeuvre (OBCS, propulsive system and interfaces). In particular, the objective is accomplished firstly by tracking reference points of the geometry of the mock-ups in order to obtain their positions in every frame, which composes the video recording, and secondly by reconstructing the relative trajectory and attitude.
In the laboratory, the implementation of the OptiTrack technology simplifies this procedure because it automatically tracks the center of mass of the Chaser and Target and returns directly the data regarding the absolute trajectories. Whereas for the parabolic flights, a dedicated software has been prepared. This software follows the two procedures above mentioned, therefore it is divided into two tasks: Camera Tracking and Trajectory Computation.

⋆ Camera Tracking

The Camera Tracking is carried out using the Motion-Based Multiple Object Tracking method of MATLAB. In order to track both mock-ups, the algorithm requires detection, prediction, and data association of objects intended as general "points":

- **Detection** to detect objects of interest in a video frame.
- **Prediction** to predict the object locations in the next frame.

- **Data association** to use the predicted locations to associate detections across frames to form tracks.

Basically, background subtraction is performed to detect objects in motion. In particular, the foreground detector algorithm, based on Gaussian mixture models, compares the colours of the grayscale video frame to a background model to determine whether individual pixels are part of the background or the foreground. Then, the blob analysis (Binary Large Object analysis), detects groups of connected pixels, which are likely to correspond to moving objects. After using this method, it can be concluded that a large number of coherent moving pixels is the target object, while the small number of connected pixels are not of interest because these are noise and so they are cancelled. The detections are based solely on motion, therefore the movement of the experimenters or other sources (such as for example the Slider) must be accurately filtered.
A Kalman filter predicts the next location of an object, assuming that it moves according to a motion model, such as constant velocity or constant acceleration. It also takes into account process noise and measurement noise.
Finally, data association is the process of associating the detections to each other and recognising the trajectory of an object (or simply a point) that moved in frame 1 up to the last frame.

The application of the method to the ERMES video recordings captured during the campaign is here presented. The example focuses on tracking the Chaser from the video recording of a parabola (in particular the 27^{th} of the 2^{nd} day, which will be discussed deeply in 3.4.2).

- A frame discretization has been performed on the video recording of the manoeuvre during the parabola. The number of the total frames from the discretization was different for every parabola, but it mostly depend on the length of the manoeuvre and the background noises (higher background movement required a larger number of frames to get filtered). Indicatively, 2 to 4 frames per second are enough. An example of snapshots taken from the video is reported in figure 3.2, which shows three frames out of 20 of a manoeuvre lasting about $5s$.

- The algorithm executes the detection frame by frame and finds the outline of the mock-ups (figure 3.3). Then it saves the coordinates of each vertex of the Chaser on the screen plane (as shown in figure 3.4). The vertexes are found from frame 1 and then their position is iterated thanks to the other frames.

Figure 3.2: Sequence of frames from a video of a manoeuvre performed during a parabolic flight

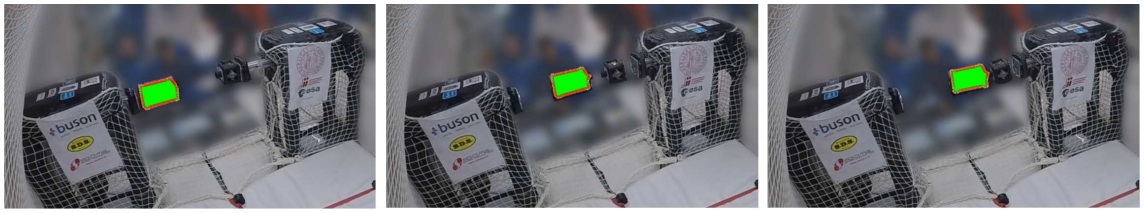

Figure 3.3: Outline research from object tracked

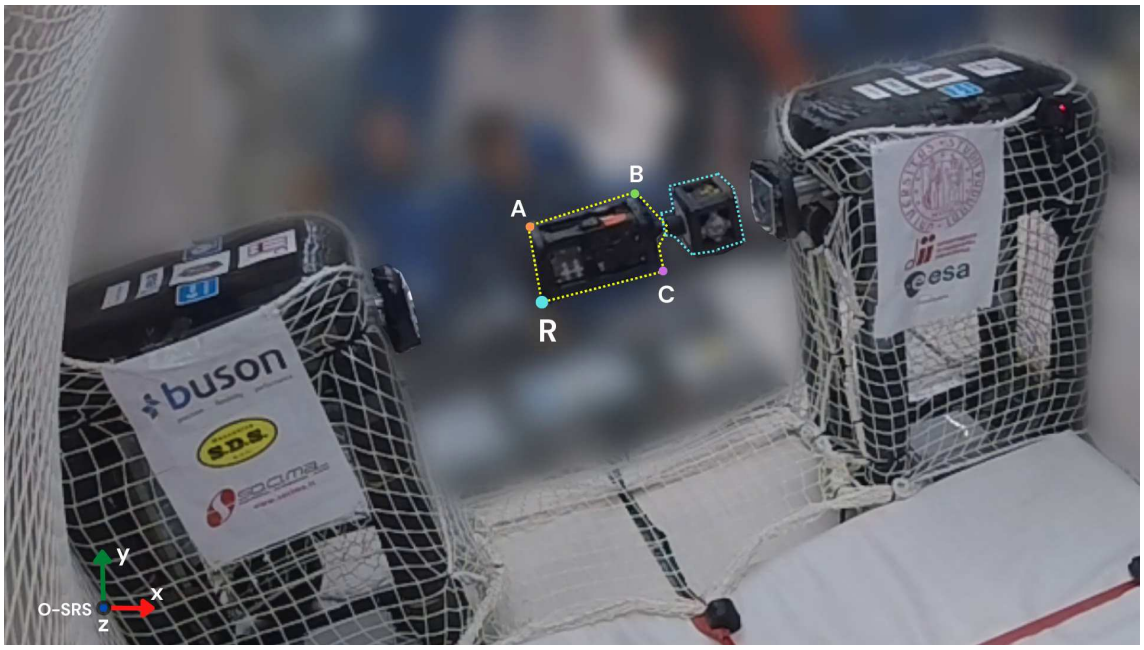

Figure 3.4: Vertexes detected in Screen Reference System

⋆ Trajectory Computation

The second part of the process focuses on reconstructing the trajectory of the mock-ups from the data regarding tracking. It is achieved through a process called "Trajectory Computation". In summary, this process is based on finding the transformation between two different systems of reference, one linked to the apparent trajectory in the video recording and one linked to the real trajectory. The two systems of reference are called respectively "Screen Reference System" (SRS) and "Real Reference System" (RRS) and they are defined as follows:

- Firstly, the SRS is defined as the system of reference where the tracking takes

place. The x and y-axis are on the plane defined by the screen, and the z-axis closes the right-handed triad. It must be noted that the z-axis|SRS is a fictional axis just to close the triad, because the information about the depth through the screen can not be extrapolated from video tracking. Therefore, this information is assumed an unknown to find in order to perform the transformation. Moreover in order to simplify the calculation, after the tracking, the origin of SRS is moved to the position that the vertex named R (named so to shorten "reference") has in the first frame (named "f1"): therefore $R|_{SRS,f1} == [0, 0, 0]$.

- Secondly, the RRS has the origin coinciding with the one of SRS, therefore $R|_{RRS,f1} == R|_{SRS,f1} == [0, 0, 0]$. The x-axis is along the conjunction of the Release interfaces, the y-axis is perpendicular to the ground and z-axis closes the triad (see figure 3.5). This choice of having the system of reference different to the one described in 2.1.2 has been made because from the video is not possible to track the centre of mass. Therefore it is easier to track the vertexes and only after reconstructing the trajectory of the centre of mass.

It must be highlighted that the trajectory can be computed because the geometry of the mock-ups, in terms of geometrical relations in RRS between all the vertexes detected, is known. For example, some relations that are used as boundary conditions are:

- Norm of the sides of the mock-ups ($\|\vec{AR}\|$, $\|\vec{BR}\|$, $\|\vec{CR}\|$).
- Relation between vertexes such as: $\vec{AR} + \vec{CR} = \vec{BR}$ and $\vec{AR} \cdot \vec{CR} = 0$.

The algorithm to reconstruct the trajectory is executed as follows:

1. Taking into account only the first frame, it solves a system of equations in order to find the rotation matrix from the RRS to the SRS $M_{ROT_{RRS \to SRS}}$. The system is solvable thanks to knowing the geometries in RRS from frame #1. The system of equations is obtained by considering the following equation (3.2.1) applied to more than one vertex:

$$\vec{P}_{SRS,f1} = \left\{ \begin{matrix} x \\ y \\ z \end{matrix} \right\}_{|SRS} = f_{SRS,f1} \cdot M_{ROT_{RRS \to SRS}} \cdot \vec{P}_{RRS,f1} \tag{3.2.1}$$

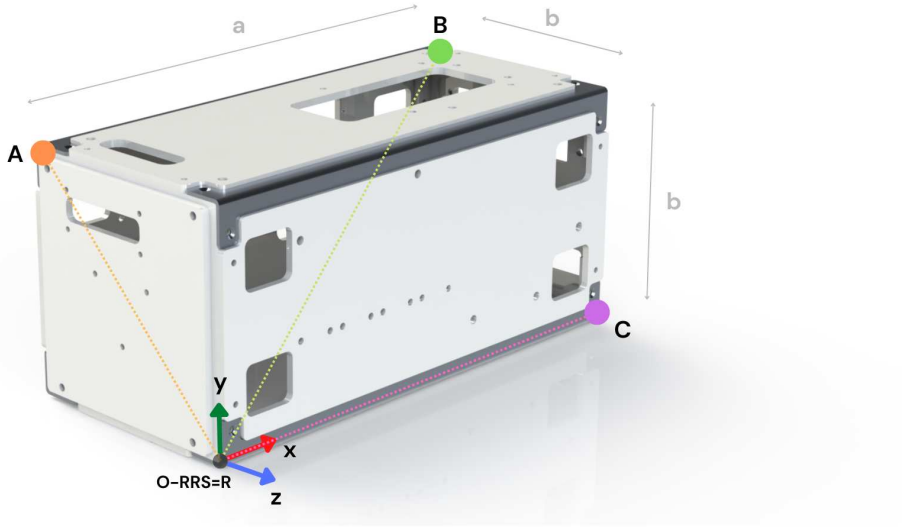

Figure 3.5: Vertices used for Trajectory Computation in Real Reference System

where $\vec{P}$ in both RRS and SRS refer to the coordinates of a vertex (A, B, C) with the respect to $R|_{RRS,f1}$ and $R|_{SRS,f1}$. Moreover, f_{SRS} is a scaling factor of the norm between $\vec{P}_{SRS}$ and $\vec{P}_{RRS}$ as shown in equation 3.2.2.

$$f_{SRS,f1} = \frac{\|\vec{P}_{SRS}\|}{\|\vec{P}_{RRS}\|} = \sqrt{\frac{(x_P^2 + y_P^2 + z_P^2)_{SRS,f1}}{(x_P^2 + y_P^2 + z_P^2)_{RRS,f1}}} \tag{3.2.2}$$

The matrix of rotation is a 3-2-1 $M_{ROT} = M_\psi \cdot M_\theta \cdot M_\phi$. The unknowns of this system are the $z\|SRS$ of each vertex and the angles of the rotation matrix (ϕ, θ, ψ), which are in common for each vertex. Therefore, in order to solve it two points are enough (5 unknowns and 6 equations).

2. The next step is to solve a set of equations found by inverting the previous one about the $\vec{P}_{RRS}$ (as shown in equation 3.2.3).

$$\vec{P}_{RRS} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{|RRS} = f_{RRS} \cdot M_{ROT_{SRS \to RRS}} \cdot \vec{P}_{SRS} = \frac{1}{f_{SRS}} \cdot M_{ROT_{RRS \to SRS}} \cdot \vec{P}_{SRS} \tag{3.2.3}$$

It can can be computed in each frame by knowing $\vec{P}_{SRS}$ for each vertex. The complexity of the system of equations to solve is greater due to the larger number of unknowns. In particular, the unknowns of each frame after the

first one, are $x, y, z|RRS$ and $z|SRS$ per vertex.
For each vertex, 3 equations can be derived (one for each axis), but with a total of 4 unknowns, therefore not solvable. To solve the problem, three more equation must be added. These additional equations regard the known geometries and relation between vertexes that should be maintained in each frame. The relations are reported in the system 3.2.4.

$$\begin{cases} \|\vec{A}_{RRS} - \vec{R}_{RRS}\| = \|\vec{AR}\| \\ \|\vec{C}_{RRS} - \vec{R}_{RRS}\| = \|\vec{CR}\| \\ (\vec{A}_{RRS} - \vec{R}_{RRS}) \perp (\vec{C}_{RRS} - \vec{R}_{RRS}) \end{cases} \tag{3.2.4}$$

The system of equations resulting from this procedure is:

$$\begin{cases} \vec{R}_{RRS} = \frac{\|\vec{R}_{RRS}\|}{\|\vec{R}_{SRS}\|} M_{ROT_{SRS\to RRS}} \vec{R}_{SRS}, & +3 \text{ eq}, +4 \text{ unkn} \\ \vec{A}_{RRS} = \frac{\|\vec{A}_{RRS}\|}{\|\vec{A}_{SRS}\|} M_{ROT_{SRS\to RRS}} \vec{A}_{SRS}, & +3 \text{ eq}, +4 \text{ unkn} \\ \vec{C}_{RRS} = \frac{\|\vec{C}_{RRS}\|}{\|\vec{C}_{SRS}\|} M_{ROT_{SRS\to RRS}} \vec{C}_{SRS}, & +3 \text{ eq}, +4 \text{ unkn} \\ \|\vec{A}_{RRS} - \vec{R}_{RRS}\| = \|\vec{AR}\|, & +1 \text{ eq}, +0 \text{ unkn} \\ \|\vec{C}_{RRS} - \vec{R}_{RRS}\| = \|\vec{CR}\|, & +1 \text{ eq}, +0 \text{ unkn} \\ (\vec{A}_{RRS} - \vec{R}_{RRS}) \cdot (\vec{C}_{RRS} - \vec{R}_{RRS}) = 0, & +1 \text{ eq}, +0 \text{ unkn} \end{cases} \tag{3.2.5}$$

3. This is then iterated for every frame. The starting point of the iterations to find the solution for every frame is derived from the final step of the previous frame.

4. The final step is a simple process that finds the data about the trajectory of the centre of mass and attitude of the mock-up starting the ones regarding

the vertices in RRS. The equations used are the following:

$$\begin{cases} \vec{CM}_{RRS} = \frac{\vec{B}_{RRS}+\vec{R}_{RRS}}{2}, & \text{Trajectory} \\ \vec{Mv}_{rot,RRS} = M_{ROT_{attitude}} \cdot Mv_{RRS}, & \text{Attitude} \end{cases} \tag{3.2.6}$$

Mv_{RRS} is the matrix of the vertexes not rotated (therefore with attitude null): each column contains the position of one vertex with the respect to the centre of mass when $[\phi, \theta, \psi] = [0, 0, 0]$. $Mv_{rot,RRS}$ indicates the matrix of the orientation in any moment of time: : each column contains the position of one vertex with the respect to the centre of mass with any possibile orientation $[\phi, \theta, \psi]$. Finally, $M_{ROT_{attitude}}$ indicates the matrix of rotation 3-2-1 that describes the attitude of the mock-up, that is a function of $[\phi, \theta, \psi]$.

After applying such process to the Chaser, it is applied also to the Target to compute its trajectory and attitude. Then to compare the results to the Medium Level Analysis the relative trajectory and attitude of the Chaser with the respect to the Target are calculated.

3.3 On-Ground Testing

The On-Ground Testing is based of a series of tests to validate singularly some aspects of Target and Chaser up to simulate the entire manoeuvre. All the tests were carried out on a low-friction table in the laboratory. In order to make the Chaser and the Target levitate over the table, sleds with flat round air bearings have been designed to hold them (see figure 3.6).
Initially, the two mock-ups were tested separately. The tests of the Target focused on attitude control. As explained previously, the aim was improving the controller by tuning the coefficient of the PID.
Whereas, the tests regarding the control of the Chaser were more varied, from validating simply the Low-level up to the integration of all systems used in the manoeuvre.
All the tests have been carried out implementing a Motion capture technology, in particular the OptiTrack technology, in order to track the mock-ups.

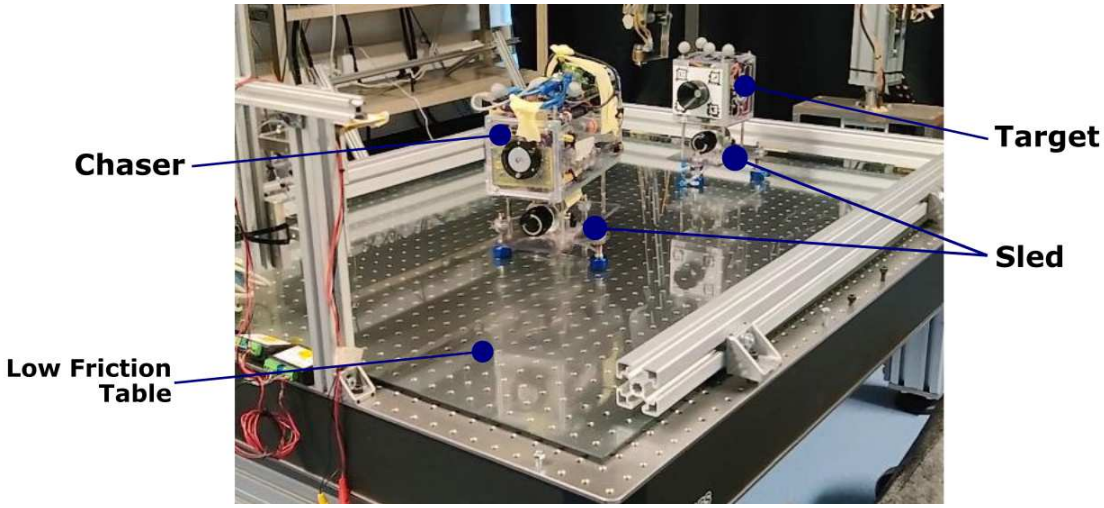

Figure 3.6: Chaser and Target on the low friction table

During the simulations of the manoeuvre, the Chaser was placed in front of the Target and then it tried autonomously to reach it. Basically, these simulations aim at improving the docking manoeuvres performed by the Chaser in view of the testing on the parabolic flight. In figure 3.7 snapshots from the video recording of a test performed on the low friction table are shown.
The graphs reported in figure 3.8 and figure 3.9 show the relative trajectory followed by the Chaser and the relative velocity with the respect to the Target. In particular, the graphs show three DoFs: the trajectory along the x-axis, represented as the distance face-to-face so that it is clear at which position the Chaser

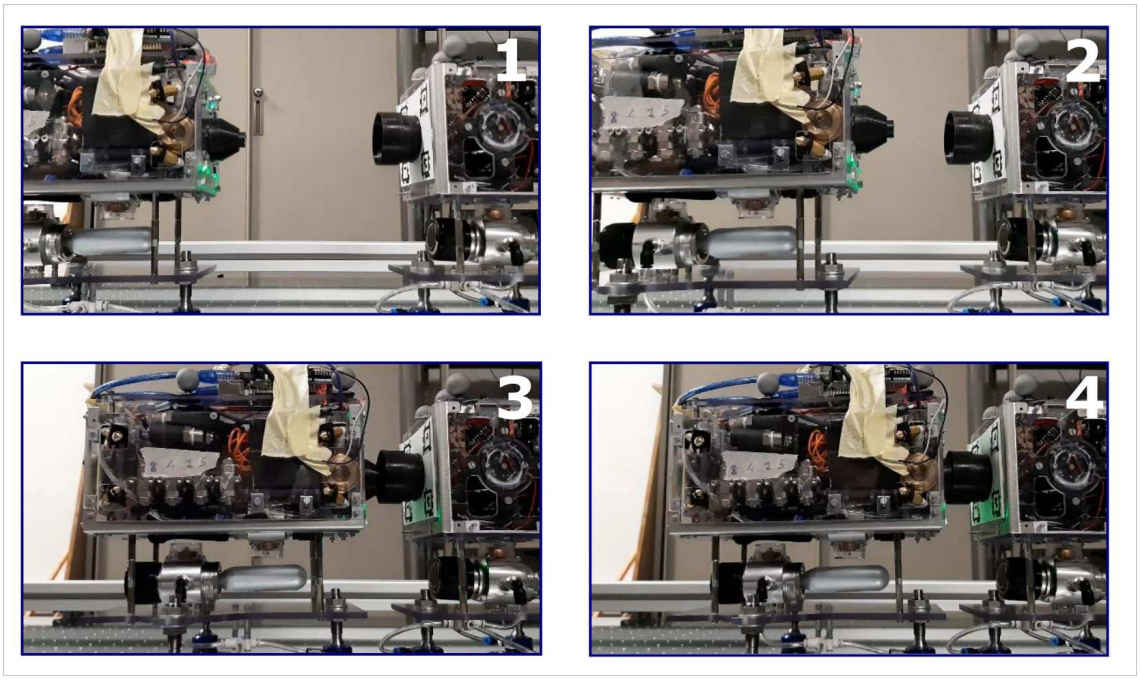

Figure 3.7: Testing on low friction table – Snapshots from the video recordings

was placed at the beginning of the manoeuvre; the trajectory along the y-axis; and the rotation along the z-axis (yaw).

Before discussing the results, the layout of the graphs must be described:

1. The **trajectory computed by the Chaser** is represented with a pale blue line with orange circles. Where the circles represent the data from the localization printed on the screen and saved, while the pale blue line is the interpolation between them. Therefore this data takes into consideration all the calculations made by the Medium-level.

2. The **trajectory reconstructed thanks to the reference camera** is represented with a green line. The trajectory reconstruction is performed thanks to the OptiTrack for the tests in the laboratory and with a dedicated software (see subsection 3.2.2) for the tests on the parabolic flights as explained previously. The importance of including these data is to validate the localization and path computation of the Chaser by introducing an external reference.

3. **The commands sent to the Low-level** are represented with vertical coloured dotted lines. In particular, blue dotted lines indicate corrections along the positive direction of the axis (accelerations), while red dotted lines indicate corrections along the negative direction of the axis (decelerations). More-

over, the number of cycles, that is labelled alongside the dotted line, indicates the duration of the impulse.

4. The black dotted lines, instead, represent all the constraints of the software. In particular, two types are present:

   (a) The horizontal black dotted line present in the graph of the x-axis highlights the "**approaching threshold**". This threshold is a software constraint on the face-to-face distance of the mock-ups. It is equal to $70mm$. It identifies the distances at which the tolerances on the other DoF become linearly stricter up to the "docked conditions".

   (b) In the graphs of the y and z-axis, θ and ψ, the black dotted lines are the **tolerances conditions** of maximum misalignment. When the Chaser reaches the approaching threshold these tolerances start decreasing. The roll (ϕ) has not a strict misalignment tolerance because the interfaces have axial symmetry.

Regarding the results of tests several considerations can be pointed out covering various aspects of the manoeuvre.

- Regarding the **duration of the manoeuvre**, the tests aimed generally to have a manoeuvre of around $8s$ or fewer starting from around $300mm$ of initial face-to-face distance. The reason is that, although the low-gravity phase is theoretically longer, it is important to have margins. In the graphs is reported one such kind of test.

- The **release** has been performed manually with a preestablished initial misalignment. The initial drift velocity in all three DoF is just a consequence of the manual release, in fact a small residual velocity was a nearly inevitable effect after the release.

- The **initial misalignment** is a wanted condition. The initial misalignment (around $25mm$ for the y-axis and around $13°$ for the yaw) was always chosen to be near the tolerances of the software ($20mm$ for the y-axis and $12°$ for the yaw) because the focus was on seeing if the Chaser was able to correct it and to stabilize the approach. This method has been used also to find empirically the maximum allowable misalignment.
  However, some tests were performed also with a nearly perfect alignment (considering negligible small errors due to small disturbances that could cause drafts-like effects). In these tests, the sequence of commands sent

was mainly composed of forward thrusts (positive x-axis) with a series of small adjustments in lateral misalignment and attitude. When such a fortunate alignment happens, the frequency and number of the commands sent are less than in the presented case because the Chaser finds himself already aligned.

- From all the tests, it is clear that generally the software acts with **low cycles commands**. This behaviour is due to the tuning of the authority parameters of the Medium-Level. Those tests found that a control based on commands sent with high frequency and with low thrust was the more efficient solution. The reason is the possibility of correcting the measured misalignments more frequently.
  Regarding the tests in the parabolic flights, the only expected difference regarding the types of commands sent was an increase in the frequency of the commands, linked to the increase in the number of DoF to control.

- The **approaching threshold** is reached at around $7s$ after the beginning of the manoeuvre. However, after around $5s$ the frequency of the commands increases even if the threshold has not been passed. The reason is because there was the necessity of stabilizing faster the y-axis. Then, after passing through it, the Chaser clearly favours attitude control over the position. The increase in the frequency of the commands during the docking phase improves the control by adding layers of precision.

- The results suggest that the authority of the command was enough to 2d manoeuvring on the low friction table with consistency by performing a series of small adjustments. In fact, the Chaser was able to correct the initial misalignment on both attitude (yaw) and position (y-axis) and simultaneously get closer to the Target.

- From the results of these tests, it was also possible to validate the integration of the localization system because the precision of the computed manoeuvre and the errors were acceptable to assume the success of the tests. Moreover, it can be seen that the error on the localization decreases with a decreasing face-to-face distance, where the minimum is reached around $150mm$, which is the focal length of the camera. When in close proximity, the errors increase by a small amount but are still under the requirement of the localization system. Nevertheless, it is interesting to point out that at the beginning of the test, when the Chaser is farther away, the errors are still relatively low.

- The tests regarding the localization software showed a dependence between the velocity of the approach and the precision of the localization, however, this trend was not explored deeply. The reason for this choice is that in the range of distances ($< 300mm$) and velocities ($< 30mm/s$) relevant to the experiment, the localization system was characterized by acceptable errors with respect to the requirements ($< 5mm$ for the positions and $< 5°$ for the rotations). Therefore, considering these tests, it can be confirmed that when the Chaser was moving slowly the pose computing was more precise.

Other tests were performed similarly to the one presented. The results are consistent with the ones already discussed.

One final remark regarding the testing in the laboratory focuses on the simplicity of the entire system. The simplicity of the entire system was a key requirement in our design, although it was assumed that this kind of approach would have affected the efficiency. Actually, the series of tests performed in the laboratory suggests that the simplicity of the software does not impact the results deeply. Nevertheless, on several occasions, the system did not perform well while testing manoeuvres with a drastically decreased duration. In fact, if the Chaser was tested with the same initial distance but by imposing via software fast manoeuvre approaches (high initial acceleration) or by inducing a strong initial velocity (around $30mm/s \pm 15\%$), the precision of the docking clearly decreased.

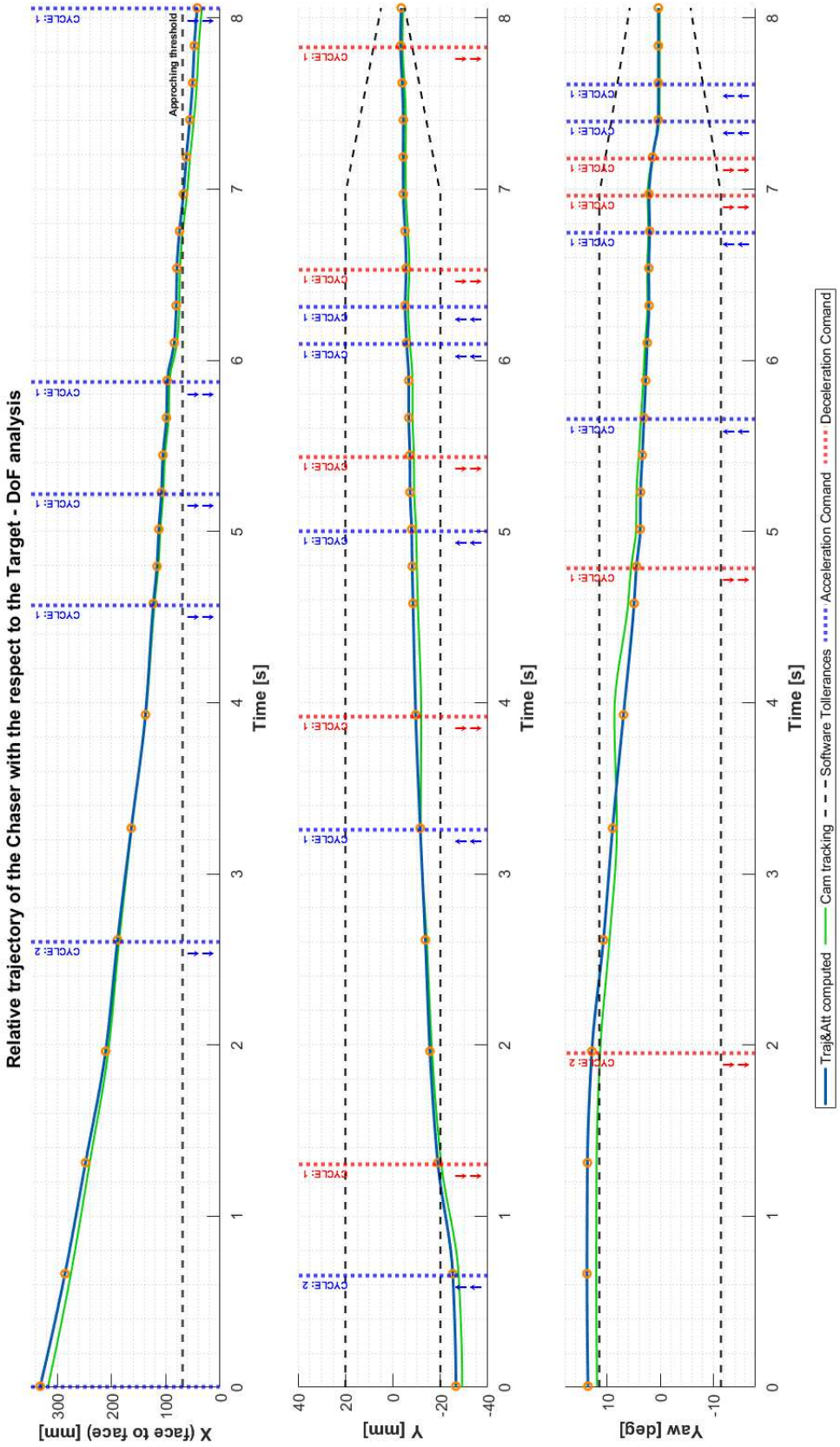

Figure 3.8: Testing on low friction table - Relative trajectory of the Chaser with the respect to the Target with reference trajectory from external camera tracking (OptiTrack)

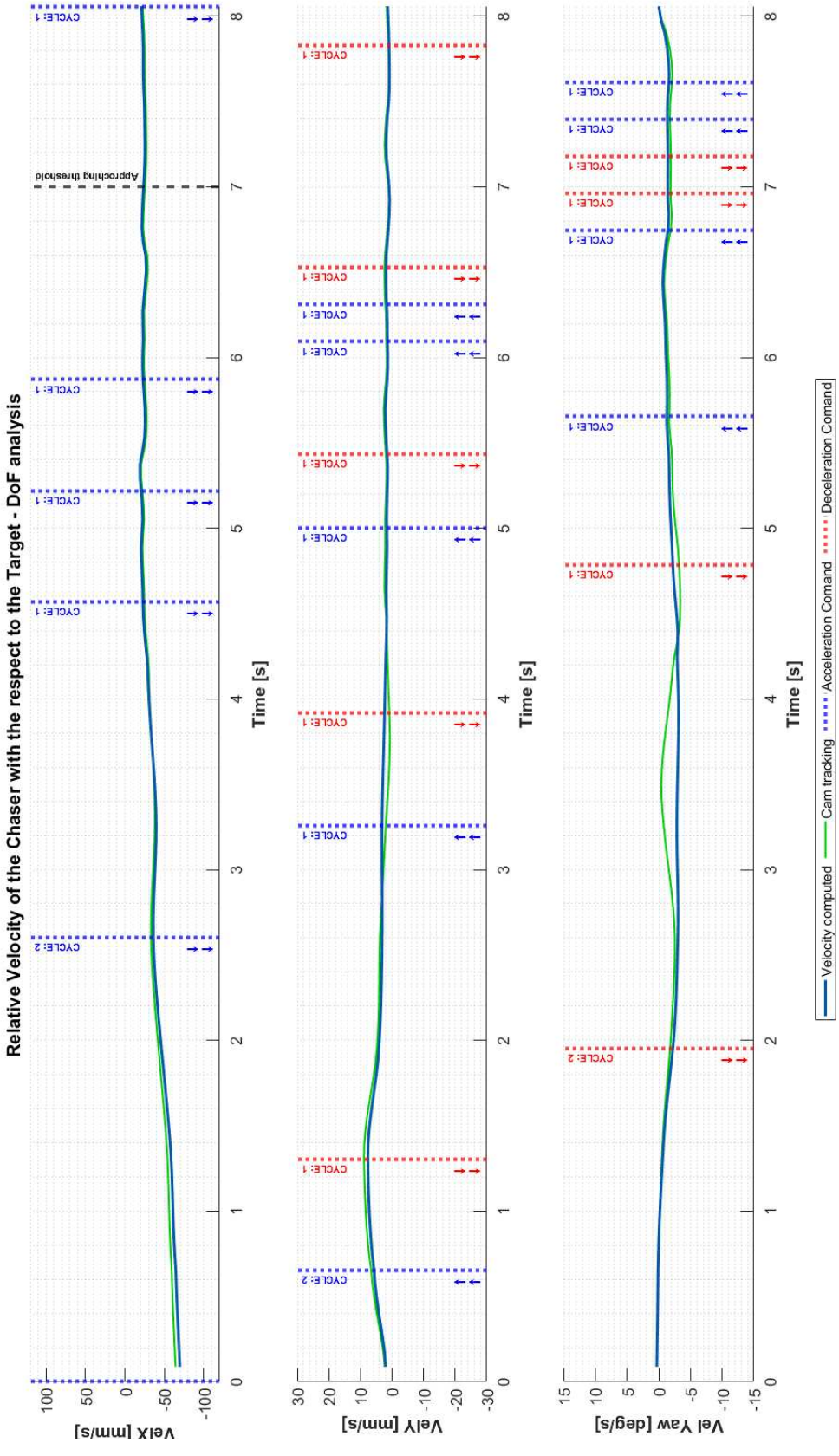

Figure 3.9: Testing on low friction table - Relative velocity of the Chaser with the respect to the Target with reference velocity from external camera tracking (OptiTrack)

3.4 Campaign

The 79^{th} ESA Parabolic Flight Campaign took place between the 17^{th} and the 28^{th} of October 2022 at the Novespace facility in Bordeaux. I would like to dedicate this small introduction to remembering my personal experience of the Campaign. It has been an amazing experience, thanks to which I had the pleasure to get in touch with a lot of amazing experts and people. Generally speaking, this experience has been a once-in-a-lifetime opportunity that will always be remembered gladly.
The two weeks passed through a series of hard work tasks, various meetings, team discussions, technical testing, and preparations for the flights themselves, which really tired and put pressure on the team. However, the feeling of getting aboard the Airbus A310 and experiencing $0g$ and seeing ERMES working in low-gravity, all the hard work throughout the week and the year prior paid off.
Figure 3.10 shows the ESA FYT 2022 participants and our supervisors.

Figure 3.10: Participants to the 79^{th} ESA Parabolic Flight Campaign

The first week was dedicated to the preparation of the experiment for the flights. The main on-ground activities aimed at preparing the experiment for the reviews and the consequent boarding on the Airbus A310 (in figures 3.11-3.12-3.13 the experiment boarded is shown).

Figure 3.11: Target boarded in the Airbus A310

Figure 3.12: Chaser boarded in the Airbus A310

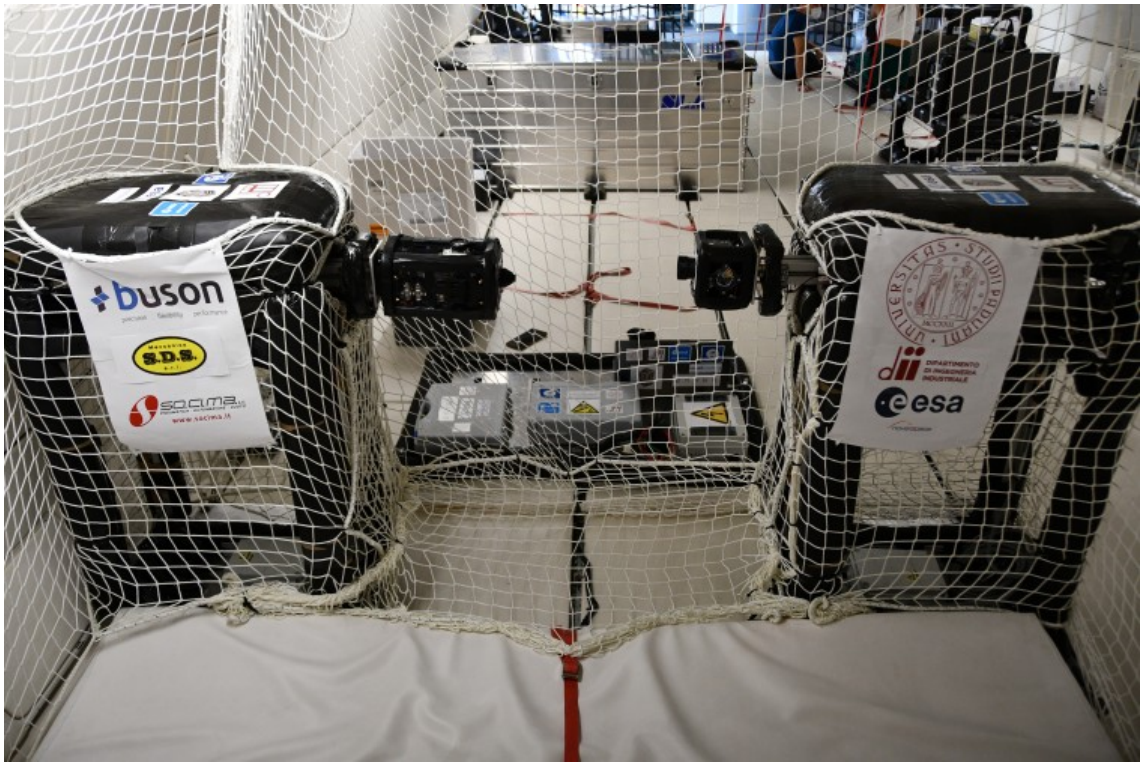

Figure 3.13: Experiment boarded in the Airbus A310

On the plane, instead, the activities included the assembly of the system (Racks, Baseplate, electrical connections etc), the installation of all the foam protections needed and the 5-sided net, and the final checking routine regarding the correct functionality of every subsystem of both the mock-ups and the Release Structure. The second week of the Campaign was instead dedicated to the flights: three flights were scheduled for three consecutive days starting from the 25^{th} of October.

3.4.1 Outcome of the Campaign

The 79^{th} ESA Parabolic Flight Campaign has been a very productive test campaign, but not all the parabolas provided the desired results. Nevertheless, considering all the possible parabolas the ERMES experiment succeeded to perform some manoeuvres that have been considered "successful".
Before proceeding with the discussion of the outcome of the Campaign the meaning of "successful" must be clarified: a parabola is considered "successful" if the Chaser was able to reach the Target and make the docking interfaces compenetrate remaining inside the tolerances during the proximity navigation. Moreover, since the software recognizes autonomously if the manoeuvre has been performed correctly, the "successful parabolas" are drawn by the group of the docking ma-

noeuvres characterized by a message that states "** *DONE* **", which is sent by the Chaser to the laptops. The message is sent only if the Chaser recognizes that the Target is around $40mm$ distance and with a low relative misalignment (for y and z-axis $\pm 7mm$ and $\pm 5°$) as explained in the subsection 2.1.2.
The parabolas that respect all the previously stated conditions and, therefore, that can be defined as "successful" are the 27^{th} of the 2^{nd} day (from now on referred to as $\#2/27$) and the 17^{th} of the 3^{rd} day (from now on referred to as $\#3/17$). These two parabolas shared similarities regarding the type of release chosen, precision of the manoeuvres and gravitational levels. However, the two parabolas differ just on the number of commands sent, since the $\#2/27$ has a larger quantity than the $\#3/17$. This difference is due to the fact that, during $\#3/17$, the disturbances by various factors (gravity level, release etc) have been very low.

In the majority of the manoeuvres in which the Chaser correctly manoeuvred up to reach the Target, the misalignment and the impact during the docking approach caused unwanted results. The misalignment caused a relevant disturbance to the attitude of Target. These disturbances should have not been registered because the shape of the docking interface and the gyroscope rigidity of the Target should have mitigated the effect. Moreover, the impacts during the docking capture happened at greater velocities than expected. These collisions induced an inevitable backwards motion of both mock-ups, which caused further collisions with some external surfaces, and, consequently, the disconnection of the two mock-ups. Nevertheless, in a "successful parabola", even if these disturbances and collisions are registered, the manoeuvre is still considered successful based on the conditions above-mentioned.
The reasons and causes behind this particular behaviour are discussed in the following subsection.

⋆ Encountered Problems

The ERMES experiment encountered some unexpected problems of different natures. These problems affected to some extent the functionality of the system as a whole. Nevertheless, at the same time, no subsystem malfunctioned and all of them worked as intended and according to the tests in the laboratory. Moreover, as it is discussed in the following bulleted list, even the precision of the localization and the manoeuvring remained comparable to the tests carried out in the laboratory.
To be precise, these problems occurred all together, therefore they are strictly con-

nected with one and another: it is their combined effect that is really impactful for what concerns the results of the tests. Despite their relevance, it was very difficult to predict and understand deeply these eventualities, or most importantly their coexistence.
In general, they can be divided into three main topics:

1. **Time for the manoeuvre**
   Considering all three days of flights, the duration of the low-gravity phase that could have been exploited is on average around $4 - 5s$: after this period the mock-ups would hit the Release Structures, ceiling or floor. Initially, it was supposed to be around $8s$ accordingly also to the information provided by Novespace. To a lesser extent, the mean differs between the days, with the first one being the best of all in terms of time dedicated to the free-floating condition.
   The reason behind the discrepancy between the expected value and the experienced one can not be found in one single aspect, but some factors shared the fault. The causes can be found firstly in the impact of the g-Jitter effect, then in the motion of the mock-ups with the respect to the plane and finally in the type of release chosen.
   Considering every cause singularly it can be stated that:

   - **g-Jitter effect** is a fluctuation in the perceived g-levels due to the movement of the crew or the vibrations of the plane. This effect affects both Target and Chaser in the same way, inducing the same acceleration, therefore it does not affect the relative motion. This is true for a coordinated and simultaneous release of the two mock-ups. The timing of the release of Target and Chaser differs by under a thousandth of a second and since the frequency is lower for the major harmonics that describe this type of fluctuation (i.e. those that have a greater impact), it is possible to consider the same g-Jitter effects for both of them.
     Therefore it is not the disturbance in the relative motion between the mock-ups that caused troubles, whether the main effect of inducing a relative motion with respect to the plane. In fact, when a "late" release made the mock-up free-float very near the Release Structures or even in between, the effect of the g-Jitter caused one of them to hit some surface. In those cases, the parabola is completely lost because both Target and Chaser lose control over their attitude.

- The second cause is the **relative motion between the mock-ups and the plane**. This problem has already been introduced in the description of the previous cause, but there are other causes of it other than the g-Jitter effect. In the design phase, it was assumed that after the Release the mock-ups would have moved towards the tail of the plane, where the mattress has been placed. This aspect of the Release is very important in order to maximize the space useful for the manoeuvre, therefore it was deeply discussed and analyzed with the Novespace personnel.
  However, often happened that the mock-ups would stay near the Release Structure instead of floating towards the tail of the plane as intended. Not going over the mattress meant not being able to take advantage of the dedicated free-floating space. In this extremely large space, the manoeuvre could have been taking place avoiding the collision with the Release Structures, ceiling or floor. In general, the first day has been the best in terms of avoiding impacts.
  The problem is deeply connected to the "Timing of release" problem that is explained after. This connection confirms again that the problems that occurred coexisted and that their combined effect is what caused the failure in the majority of the parabolas.

- The third cause is the **type of release** chosen in the flights. In the first 10 parabolas of the first flight, the experimenters tested different types of release to find the best solution. In the laboratory, a lot of attention was given to the static release. However, the team was prepared for the eventuality of using a pushing release. During the flights, it became practically imperative to use it because there was the necessity to reduce the time of the manoeuvre to avoid the above-mentioned collisions. In fact, when the mock-ups did not move towards the tail of the plane, with a static release the probability of one mock-ups hitting some surface was higher. Meanwhile, with a pushing release, only the Target could hit something right after the Release while the Chaser was instead moving away from the release interface immediately. Effectively, the chances of hitting something were simply nearly halved. At the same time, using a pushing release meant that the Chaser would have a high velocity of impact (still under the requirement of maximum allowable velocity $35mm/s$) and therefore it would have pushed the Target towards the release structure of the Target, causing an impact on it. This topic is related also to the problems of the timing of

impact and the gyroscope rigidity of the Target.

2. **Timing of the release**
   This problem refers to the fact timing the release of the mock-ups perfectly is a fundamental factor in order to increase the possibility to perform the manoeuvre correctly. Most of the effects of a bad release are reported in the previous problem analysis since they are deeply linked one to another. The design for the release took into consideration the data given by Novespace and simple simulations in which it seems that the perfect moment to release was around $8s$ from the start of the $0g$ phase. The experience acquired during the flights suggests that the perfect moment is an aspect way more difficult to evaluate. Most importantly, it depends deeply on the flight itself, and to a minor extent also on the parabola. On the last day of the flight, the team managed to basically avoid badly timed releases, however the exact moment had to be calculated during the flight thanks to the experience acquired on the previous flights.

3. **Gyroscope rigidity of the Target** The final problem to analyse is actually the one that impacted the results of the Campaign the most. It refers to the fact that the Target, when approached and hit at a relatively high velocity, loses control of its attitude. If the approach is highly misaligned and the Chaser hits the internal surface of the coned shape drogue, the Target suffers a high disturbance to its attitude and loses its control.
   This problem, at a low velocity of approach (as in the test performed in the laboratory with the static releases), is not present because the combination of the gyroscopic momentum of the Target and the shape of the docking interface can correct the misalignment while still get pushed backwards a little. This problem could have been solved in many ways, the simplest would have been by designing heavier flywheels for the RW.

⋆ Flights Overview

Before moving to the Results analysis, a description of the outcome and key event of each flight of the campaign is presented.

▷ **Flight** $\#1$
Flight $\#1$ was programmed for the 25^{th} of October.
The first set of parabolas was dedicated to getting used to the $0g$ and to the experiment routine as well as focusing on the tuning of the parameters of the release.

The actual testing of the docking manoeuvre lasts up to the 15^{th} parabola. The 16^{th} parabola was characterized by an unfortunate fall of the Chaser during the $2g$ phase, that damaged it. As a consequence of the fall, the Chaser hit frontally the metal baseplate of the Release Structure of the Target by passing through a very small hole in the net. This impact actually damaged the ToFs and disconnected the power wire of 4 electrovalves. The hole in the net was closed after the flight. Moreover, since the Chaser did not show any visible damage, the team was unaware of the effect of the fall and decided to not change the mock-up with the spare one during the flight. Consequently, in the remaining parabolas, the experiment has not functioned as intended because of it. In particular, from the 16^{th} parabola on, the Chaser kept sending a rapid sequence of two commands, one to move forward, while the other to correct a nose-down disturbance that was always detected after the first command was actuated. The reason behind this particular sequence was that when the "go forward" command was actuated, only two electrovalves out of the four needed for such a command were available. The effect of actuating only these two remaining thrusters is what caused the nose-down disturbance and, therefore, the necessity to correct it.
The completed manoeuvres were interesting although they did not manage to dock. More than one type of release was tested, showing that a pushing release was the best solution. Regarding the docking phase, the lack of more gyroscopic rigidity of the Target was the key problem.

▷ **Flight** $\#2$
Flight $\#2$ was programmed for the 26^{th} of October.
The second flight started with a major problem because an unexpected disconnection of the Chaser from the ssh-connection used to communicate to it from the laptop caused the loss of the majority of parabolas. Whereas, the remaining have proven to be actually positive parabolas. This type of disconnection was never experienced in any test in the laboratory and came completely unexpectedly. The disconnection lasts up until the 18^{th}, then the Chaser came back online. To avoid this problem, after the flight a backup connection of another kind was set up in case the Chaser would disconnect again. In the meanwhile, tests on the timing of release have been carried out, increasing the know-how on this particular problem.
The manoeuvres completed were interesting but with a better timing of release than Flight $\#1$ and, consequently, a larger time dedicated to the manoeuvres. The 27^{th} parabola is a particularly interesting one since it is characterized by a precise

approach and a large number of commands sent. Therefore it has been chosen to be the focus of the Results analysis in the following chapter.

▷ **Flight** $\#3$
Flight $\#2$ was programmed for the 27^{th} of October.
The last flight was actually the most positive one in terms of the manoeuvres tested. The manoeuvres completed were interesting and similar to Flight $\#2$ thanks also to the knowledge acquired since then. The 17^{th} parabola is the most interesting of the entire flight: the docking manoeuvre has been precise but fast. The reason for such precise manoeuvre is that the Chaser has not registered any major misalignment after the release. Therefore, the manoeuvre is characterized also by a fewer quantity of commands because fewer disturbances or misalignments had to be corrected. Unfortunately, the video recordings of the parabolas from 15 to 20 were lost due to the SD card of the external reference camera not being replaced. Consequently, eventual studies of these manoeuvres do not report the reference from the camera.
Finally, one last remark on the 3^{rd} flight concerns the docking interface tested: since the amount of parabola lost in the previous two flights, in accordance with the ESA supervisors, the tests continue focusing on the probe-drogue docking interface. Therefore, the comparison between the two docking interfaces was not performed because the androgynous interface has not been tested.

3.4.2 Testing on the Parabolic Flight

In this section, the focus shifts towards the study of a successful parabola. The post-processing procedure is divided into 2 sections as explained previously, one called "Medium Level Analysis" which is principally related to what the Chaser has computed regarding both recognition and trajectory computation, and the other called "Camera Recording Analysis" which is needed in order to have an external reference to the previous analysis. The parabola analysed is the 27^{th} parabola of the 2^{nd} day (referred to as $\#2/27$). Snapshots of the video recording of this parabola are shown in figure 3.14.

Figure 3.14: Parabola $\#2/27$ - Snapshots from the video recordings

The graphs reported in figure 3.15 and 3.16 show the relative trajectory followed by the Chaser and the relative velocity with the respect to the Target during this parabola. The graphs are coherent with the one presented in chapter 3.3 in terms of layout (dotted lines, colours, etc.). The data shown regards all the events up to

the impact against the Release Structure and the consequent complete separation of the mock-ups.
From the analysis of the graphs the following considerations can be made:

- The **duration of the manoeuvre** has been around $4s$. The reason behind the shortness of the manoeuvre is due to the combined effect of the g-Jitter and "imperfect" release as explained in the previous chapter. As explained earlier, this fast approach is due to the necessity of reducing the time needed for the manoeuvre to avoid collisions of any kind. To speed up the manoeuvre these tests have been carried out with a pushing release (in particular Mod#4) in order to induce a greater velocity to the Chaser. This choice is due to the experience acquired during Flight#1 in which the static releases have shown to be inefficient in assure that the mock-ups would not hit the floor, the ceiling or any other surface before even reaching the threshold.

- The **approaching threshold** is reached around $3s$ after the release. Just comparing this detail to the test in the laboratory gives a perspective on the difference in timing between the expected duration of the free-floating conditions and the actual one. Therefore the approach along the x-axis is faster than expected, but still inside the range determined by the requirements.
  After passing the approaching threshold the corrections focused mainly on the attitude. Moreover, the docking is clearly characterized by the faster frequency of commands sent to the Low-level. This is due to the fact that the Chaser is aiming at reducing as much as possible the angular velocity of the approach. During the tests in the laboratory, the behaviour was very similar, but more precise.

- Regarding the **initial attitude alignment**, only the pitch is not near zero (around $-5°$), while the roll and yaw have a lower misalignment ($< 5°$). This initial discrepancy in pitch could have been due to the light bending due to the placement of the Chaser on the magnetic constraints.

- The increase in the **pitch misalignment** up to around $12.5°$ could be due to two different factors coexisting: to a small nose-down disturbance during the release, or to the initial misalignment that has been enhanced by the thrusting along x-axis. However, it can be seen that when the mock-up reaches the tolerance limit ($12°$) it immediately corrects it.

- Differently, the **yaw trend** is near zero during the navigation, while increasing during the approaching phase. Therefore, it required fewer corrections

with the respect to the pitch.

- Furthermore, during the manoeuvre important influences on the **position along the y and z-axis** are registered. The initial misalignment is probably due to the same reason that caused the pitch one. Similarly to that case, it can be seen that when it reaches the limit ($20mm$) it corrects its trajectory.
- However, the **misalignment in different axis** did not happen at the same time, therefore the Chaser was able to correct them individually. The only two misalignments that happened at the same time are the pitch and z-axis ones: however fortunately the combined effect mitigated both of them assuring that the Target remained visible throughout the manoeuvre.
- Regarding the **velocities** during the manoeuvre, registering higher ones with the respect to the tests in the laboratory was expected. Consequently, also stronger thrusts were expected.
- As a consequence of all the previous analysis, the Chaser mock-up enters the approaching phase towards the Target with parallel face-to-face conditions (roll, pitch and yaw nearly null) but without a precise alignment along the y and z-axis (both lower than $10mm$). In this case, the conical design of the **docking interface** should have helped correct it. In fact, the misalignment can be recovered by sliding between the two contact surfaces and then establishing the connection thanks to the attraction between the two magnetic interfaces. Nevertheless, during the campaign, destabilising contact forces between the interfaces have been registered, causing in most cases the failure of the manoeuvre. In fact, as seen in the graphs, entering the interface the y and z axis remain basically constant, instead of slowly decreasing up to zero. In conclusion, the perfect alignment is not achieved, however, small corrections are consequences of the soft impact with the interface.
- As explained previously, another reason behind this unexpected behaviour during the docking final moments is also the insufficient gyroscopic rigidity of the Target. At such velocities (around $3cm/s$), it has not been able to stabilize itself while impacted. Therefore it can be stated that the Target performed in an acceptable manner throughout the entire experiment generally avoiding high misalignment but that suffered a lot the collisions with the Chaser.
- After the impact all the momentum is shared between the two mock-ups resulting in a backwards movement. This effect with the addition of g-Jitter-

related movements caused an impact on a surface of the plane or of the experiment setup, which ended completely the experiment.

- Generally speaking, the ability to recover misalignments confirms that the **authority of the command** was enough to 3d manoeuvring with consistency. The frequent commands during the approaching phase are translated in a good response of the system, with relatively precise corrections composed of short and consecutive impulse thrusters operations.

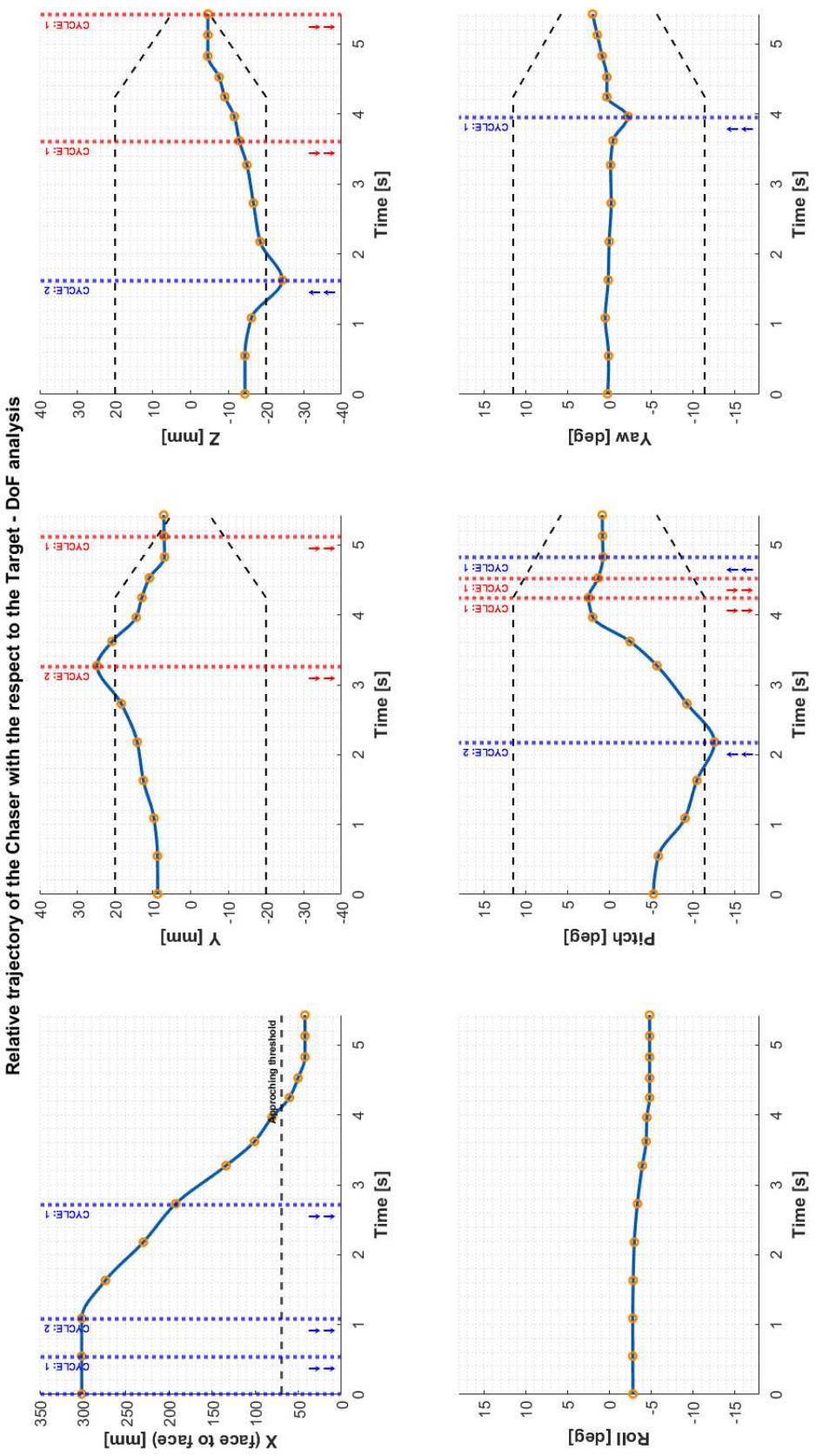

Figure 3.15: Parabola #2/27 - Relative trajectory of the Chaser with the respect to the Target

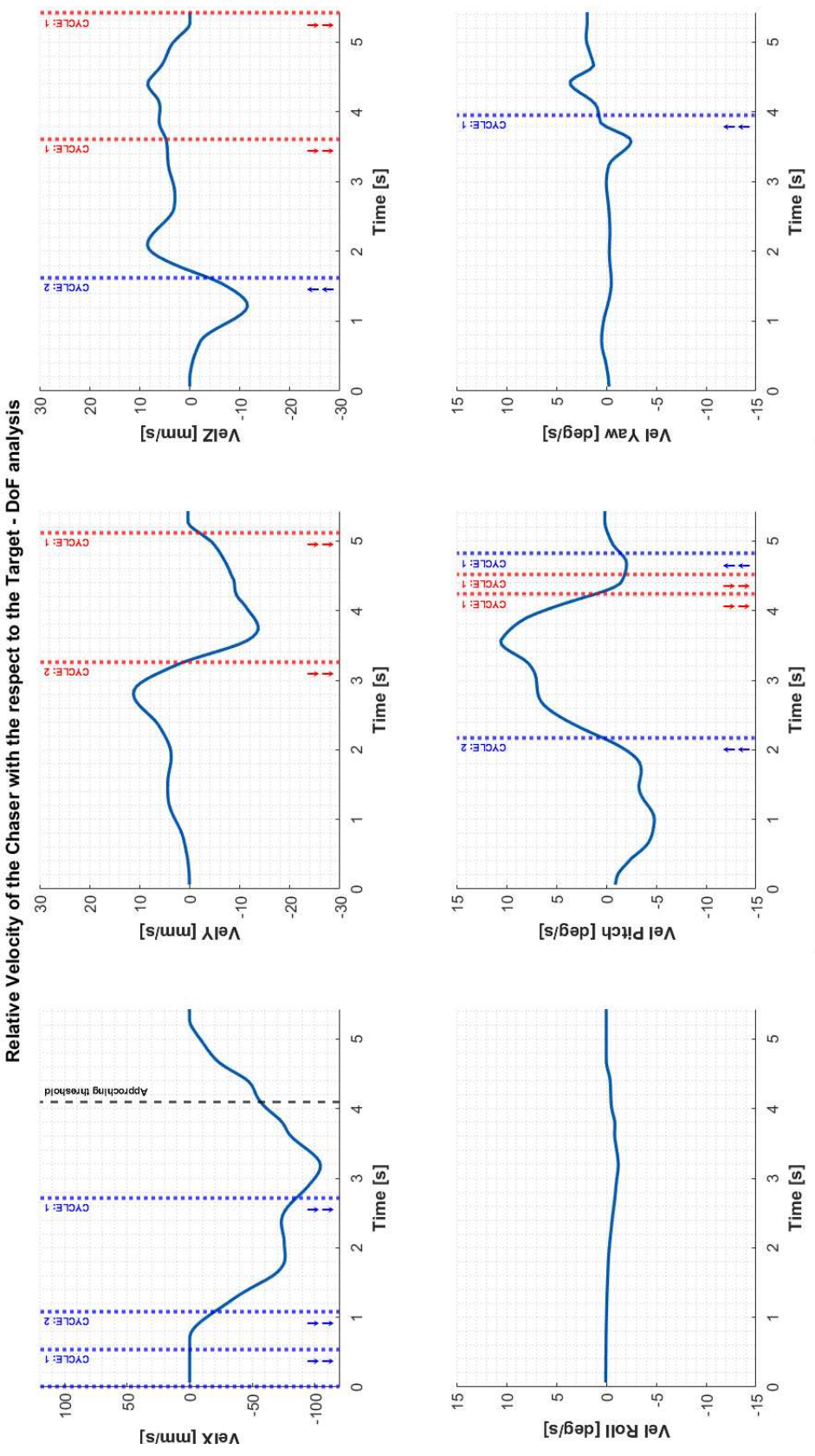

Figure 3.16: Parabola #2/27 - Relative velocity of the Chaser with the respect to the Target

Finally, the reference from the external cameras can be introduced.
It is obtained thanks to the tracking algorithm and trajectory computation from the video recordings as explained in section (3.2.2). The results are reported in figures 3.17 and 3.18. The trajectory computed with the algorithm of tracking and reconstruction is added in the previous graph with a green colour with the respect to the blue colour of the Medium Level Analysis. Important considerations regarding the performance of the proximity navigation software can be made by comparing the computed trajectory to the reconstructed one.

- A great influence on the Medium-level is represented by the Pi Camera. In fact, errors mainly occur when the PiCamera is out of focus. This happens with a face-to-face distance above $150mm$. The errors in localizing the Target lower under this threshold, in which the camera focusing improves. As a consequence, the control in both attitude and position improves.
  Another layer of perception is given by the data provided by the ToFs during the last part of the approach. The ToFs are useful in order to overcome possible errors in the calculation of distances, which could be caused by a partial eclipse of the Apriltags caused by the docking interfaces.

- Another interesting comment on the localization is that the error is clearly greater when the velocities are higher. This feature was present also in the tests on the low friction table.

- On the other hand, the attitude keeps a constant error under the threshold too. The roll, pitch and yaw are affected more by computational errors of the software and so it was foreseeable that they would have a higher error than distances. A constant integration error could propagate in presence of partial shading of the Apriltags. This shading could justify the small disagreement with the data reconstructed from the video recording.

- It should be noted that both attitude and trajectory are inside the expected errors and design margins. This leads to the conclusion that the localization system has still performed as expected and in an acceptable way even in the parabolic flight environment.

- Moreover, considering that the camera reconstruction seems to confirm the coherence between the Low-level commands and the effects on the trajectory and attitude.

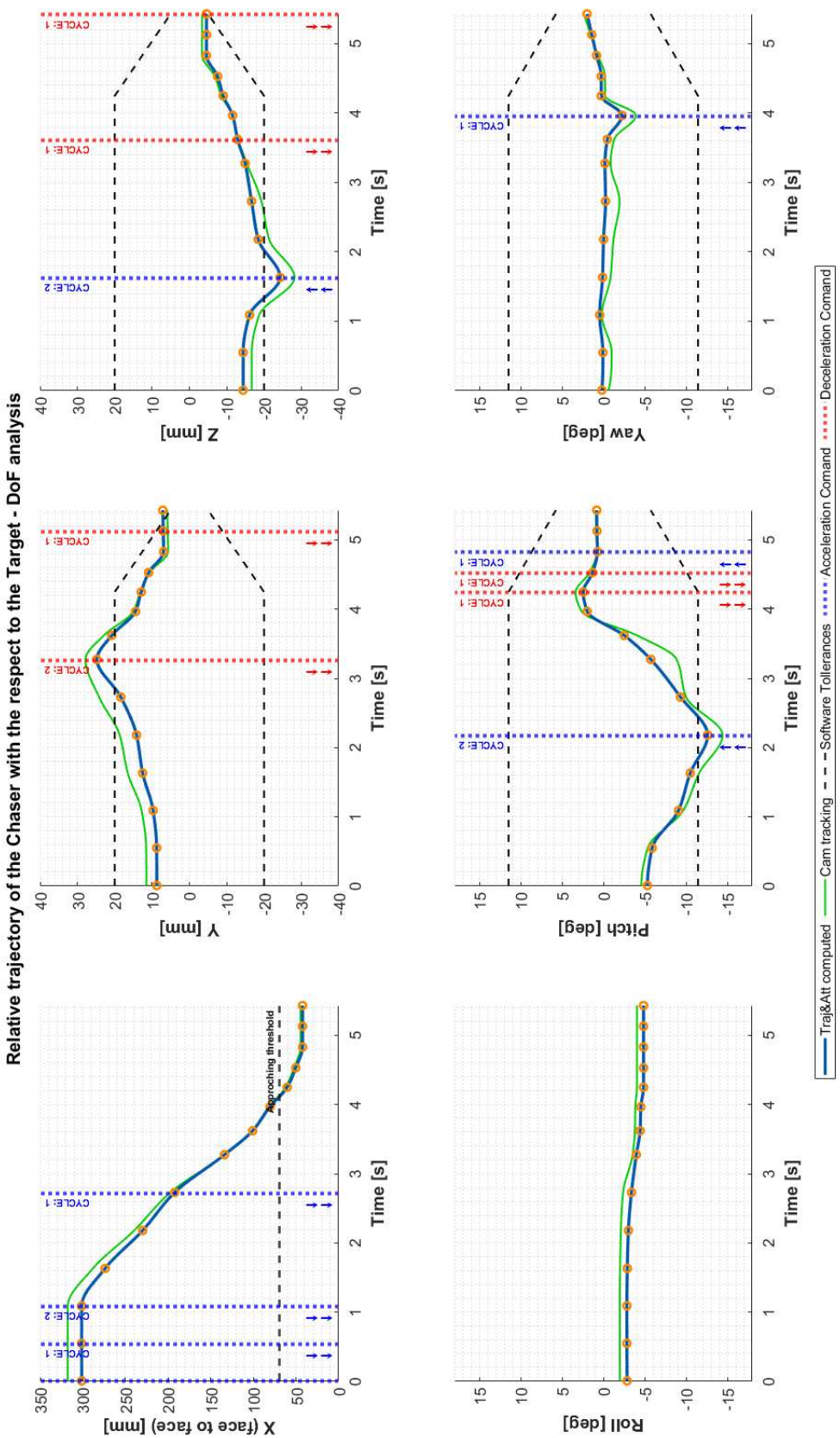

Figure 3.17: Parabola #2/27 - Relative trajectory of the Chaser with the respect to the Target with reference trajectory from external camera tracking

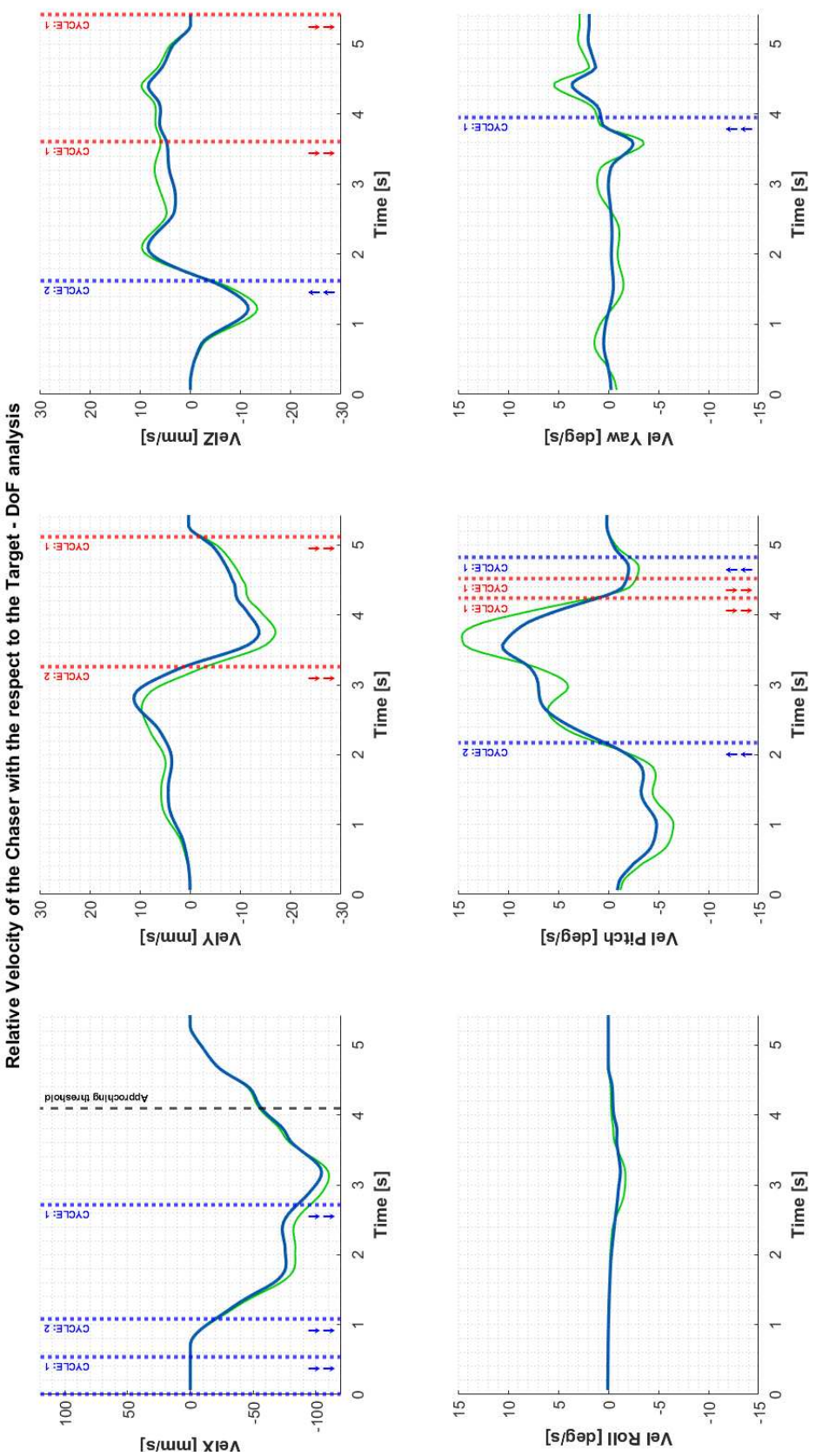

Figure 3.18: Parabola #2/27 - Relative velocity of the Chaser with the respect to the Target with reference velocity from external camera tracking

In conclusion, in the Parabola $\#2/27$ the docking manoeuvre was accomplished with an acceptable final alignment between the mock-ups.
The duration of the manoeuvre was not the one expected but by implementing pushing releases it has been possible to perform it up until docking. The pitch angle has been recovered from the initial misalignment and the misalignments encountered in the other axes have been recovered. Moreover, subsequently to the impact between the docking ports, a momentaneous state of good stability has been registered and the contact between the surfaces of the docking interfaces was soft enough to guarantee at least partially the sliding process. However, the collision pushed back the Target causing a collision with its Release Structure. Without an external object to hit or more time to perform the manoeuvre, probably the Chaser could have improved the connection.
In summary, the software performed as intended and with acceptable errors also considering its simplicity.

An equivalent results discussion could be carried out about other parabolas. The $\#2/27$ represents the clearest one in terms of manoeuvre since the remaining parabolas, which can be defined "successful", are characterized by a little more visible final misalignment. This difference can be imputed to a stronger impact during the docking phase, or even simply due to a greater misalignment to correct during the approach. An example in fact is the parabola $\#3/17$ in which the approach was even slower but poorer in commands since the misalignment remained inside the tolerances. However, during the approaching phase, passed the threshold, the yaw error was higher enough to make the Chaser entry the docking interface slightly tilted, causing an uncontrolled post-docking misaligning effect.

Chapter 4

Conclusions

In conclusion, it can be stated that during the 79^{th} ESA Parabolic Flight Campaign the experiment worked as intended, but at the same time it had to face some unexpected problems that partially impacted its functionality and effectiveness. Therefore, the results of the flight campaign are not entirely negative or positive. In such cases the experiment must be deconstructed up to its core objectives to identify exactly the strengths of this design and the eventual errors it had. Finally, facing a precise analysis of the experiment as a whole, it is interesting to evaluate improvements for future iterations.

Firstly, in order to evaluate the grade of success of ERMES, the objectives of ERMES have been deconstructed as follows. Then for each task, a grade of successfulness (called $VAL\%$) is given and compared to the weight (called $W\%$) with respect to the entirety of the experiment (see table 4.1).

- **Development [D]**
  It concerns all the tasks of Design, Manufacturing, Assembly and Testing of the experiment carried out prior to the campaign. The total weight of the contribution of this category on the successfulness of the experiment is 25%.

  1. Design, Manufacturing and Assembly of a cold gas propulsive system for the Chaser according to Novespace Guidelines;
  2. Design, Manufacturing and Assembly of a system of RWs for the Target according to Novespace Guidelines;
  3. Design and Testing a localization software implementing Apriltags;
  4. Design and Testing a simple Proximity Navigation Software;
  5. Design, Manufacturing and Assembly of a mechanical release system;

6. Testing the entire experiment on on-ground facilities;

Between all the above-mentioned objectives, half of the total weight of this category is given to objective $\#6$ due to its importance. The other objectives share the same weight (see table 4.1).

- **Safety [S]**
  It concerns all the tasks relative to the risk and hazard management. The importance and gravity of such topics are the reason why this category is separated from the previous one, although it complements it. The total weight of the contribution of this category on the successfulness of the experiment is 15%.

  7. Prepare a complete and precise risk and hazard analysis;
  8. Design and Implement solutions for safety management according to Novespace Guidelines;

  The total value is split in half between the two.

- **Campaign [C]**
  It concerns all the tasks that characterized the preparation for the campaign and the flights themselves. The total weight of the contribution of this category on the successfulness of the experiment is the highest, in particular 60%, because it has the most important objectives such as the one relative to the autonomous docking manoeuvre. Moreover, the high weight chosen for this category is further sustained by the fact that, if the ERMES experiment would have not fulfilled the following objectives, the final value of successfulness of the entire experiment would have been 40%, which would have been configured as a failure.

  9. Board the experiment on the Airbus310;
  10. Chaser - Follow the trajectory computed in a low-gravity environment;
  11. Target - Perform cooperative attitude control in a low-gravity environment;
  12. Manoeuvre - Dock accomplished;
  13. Test and compare the two docking interfaces;

  The one with the greatest significance is the objective regarding the Docking itself, whose weight is set to 25%. This objective regards the actual con-

nection between the mock-ups, therefore the "successful" manoeuvres described in the previous chapter are not considered an accomplished docking manoeuvres.

N	CAT	CRYTERIA	W%	VAL%
1	D	D.M.A. of a cold gas propulsive system	3	3
2	D	D.M.A. of of a system of RWs	3	3
3	D	D.T. a localization software	3	3
4	D	D.T. a simple Proximity Navigation Software	3	3
5	D	D.M.A. of a mechanical release system	3	3
6	D	T. the experiment on on-ground facilities	10	10
7	S	Prepare a risk and hazard analysis	10	10
8	S	D.I solutions for safety management	5	5
9	C	Board the experiment on the Airbus310	5	5
10	C	C - Follow the trajectory computed	10	10
11	C	T - Perform cooperative attitude control	10	10
12	C	Dock accomplished	25	5
13	C	Test and compare the two docking interfaces	10	2.5
TOT			**100**	**72.5**

Table 4.1: ERMES Objectives

As reported in table 4.1, the ERMES experiment achieved the 72.5% of all the objectives. The only two objectives that have not been fully achieved were the one regarding the accomplished docking manoeuvre ($\#12$), and the one regarding the testing of the two interfaces ($\#13$).
Considering the former, the task has been set partially achieved because the magnetic connection has not been established but the precision needed to dock has been confirmed to have been reached.
Moreover, for what concerns the docking interfaces, since only one docking interface has been partially tested (probe-drogue docking interface), for the task $\#13$ the achieved value is 2.5% out of the total weight of 10%. This value is still under half of the weight because, since the docking did not happen as planned, the probe-drogue docking interface can not be considered fully tested either.

Nevertheless, all the subsystems involved in both Target and Chaser worked well together and performed according to the test results in the laboratory. There-

fore, ERMES, as a technological demonstrator, has been able to confirm the validity of the integration of many different technologies and subsystems in view of an autonomous docking manoeuvre.

On the other side, considering the educational aspects, the ERMES project has been an amazing opportunity for all its member to enhance their academic career. In fact, during the journey of ERMES, the team members not only improved their knowledge of space-related topics but also their indispensable troubleshooting skills. In general, they experienced an all-around immersion in a work-like environment. In addition, the experience included presentations at important congresses and symposiums thanks to the different article productions. Finally, the project helped all the members to improve their interpersonal and management skills thanks to the help of the mentorship of the endorsing professors, researchers of the University and ESA supervisors with whom the team continuously interact.

In conclusion, the ERMES experiment has ensured the fulfilment of a lot of its objectives, meanwhile showing plenty of room for improvement. Unfortunately, the team did not achieve some of the desired results, but, given the complexity and ambitiousness of the experiment, the ERMES experiment can be considered a partial success.
Furthermore, since ERMES has been ideated as a successor of previous experiments of the *Università Degli Studi di Padova*, naturally I hope that future works by other student teams will rise inspired by the ERMES project.

References

[1] P Amadieu and JY Heloret. "The automated transfer vehicle". In: *Air & Space Europe* 1.1 (1999), pp. 76–80.

[2] Marco Barbetta et al. "Arcade experiment on board bexus 13 and 17: design, integration and flight of a technology test platform within a student balloon programme". In: Dec. 2015.

[3] Alessandro Bortotto et al. "ERMES: Design and preliminary simulations for an autonomous docking manoeuvre". In: *4th Symposium on Space Educational Activities*. Apr. 2022. DOI: 10.5821/conference-9788419184405.081.

[4] Alessandro Bortotto et al. "Ermes: Experimental Rendezvous in Microgravity Environment Study". In: *72nd International Astronautical Congress*. 2021.

[5] John Bowen et al. "CubeSat Proximity Operations Demonstration (CPOD) mission update". In: *2015 IEEE Aerospace Conference*. 2015.

[6] Francesco Branz et al. "Miniature docking mechanism for CubeSats". In: *Acta Astronautica* 176 (June 2020), pp. 510–519. URL: https://www.sciencedirect.com/science/article/pii/S0094576520304082.

[7] Matteo Duzzi et al. "Electromagnetic position and attitude control for pacman experiment." In: May 2017.

[8] ESA. *ESA Education Fly Your Thesis! (FYT) Programme*. URL: https://www.esa.int/Education/Fly_Your_Thesis/About_Fly_Your_Thesis (visited on 04/01/2023).

[9] A. Fejzic, S. Nolet, and D.W. Miller. "Results of SPHERES microgravity autonomous docking experiments in the presence of anomalies". In: 8 (Jan. 2008), pp. 4878–4890.

[10] Lorenzo Fluckiger et al. "Astrobee Robot Software: A Modern Software System for Space". In: 2018.

[11] Maximilian Krogius, Acshi Haggenmiller, and Edwin Olson. "Flexible Layouts for Fiducial Tags". In: Nov. 2019, pp. 1898–1903. DOI: 10.1109/IROS40897.2019.8967787.

[12] Novespace. *Novespace*. URL: https://www.airzerog.com (visited on 04/01/2023).

[13] Edwin Olson. "AprilTag: A robust and flexible visual fiducial system". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3400–3407. DOI: 10.1109/ICRA.2011.5979561.

[14] Davide Petrillo et al. "Flexible Electromagnetic Leash Docking system (FELDs) experiment from design to microgravity testing". In: Oct. 2015.

[15] SpaceX. *Dragon Spacecraft*. URL: https://www.spacex.com/vehicles/dragon/ (visited on 04/01/2023).

[16] John Wang and Edwin Olson. "AprilTag 2: Efficient and robust fiducial detection". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2016, pp. 4193–4198.

Ringraziamenti

Vorrei dedicare questo spazio della mia tesi per ringraziare tutti coloro che mi hanno supportato durante questo percorso di crescita personale.

In primis, vorrei ringraziare i miei genitori, a cui dedico questa tesi, per avermi cresciuto insegnandomi a seguire i miei sogni e a lottare per raggiungerli. In particolare mio padre per essere sempre stato il mio esempio di vita, il mio eterno giovane rivale sportivo, e per aver condiviso con me la sua passione per lo Spazio. Ugualmente, vorrei ringraziare mia madre per avermi fatto sempre sorridere e per essersi spesso caricata sulle spalle le mie difficoltà e i miei dolori affinchè questi pesassero meno sulle mie.
In secondo luogo, vorrei ringraziare la mia dolce metà Federica per essere rimasta sempre al mio fianco e per essere stata la valvola di sfogo senza la quale non sarei resistito. Colgo l'occasione di ringraziare anche tutti i miei parenti soprattutto i miei due cugini Simone e Giacomo con cui sono cresciuto.

Inoltre vorrei ringraziare tre importanti persone: il mio miglior amico Kevin per tutti questi anni passati insieme a condividere la nostra pazzia; il sommo But per l'infinità di ore spese a rispondere ad inutili domande *et Rufus*; e Stefano, compagno di sopravvivenza sia nell'appartamento signorile di Barbarigo, sia nella favela padovana tra loschi individui di dubbi costumi e tradizioni. In seguito ricordo il resto del gruppo di amici *Classmeils* ormai diventato famiglia: il colto rosa Lante, il decrepito caffeinomane Donny, il bradiposo Lollo, il veneto DOCG Dena, il mio amante Dario e infine Nicola aka Johnny-Timothée Depp-Chalamet. Ringrazio anche tutti i compagni dell'Università e di ERMES per tutte le esperienze passate insieme, dalle "aggressioni verbali", ai coefficienti "inventati" *k-Bortotto*, ai viaggi in neurochirurgia e all'ambiente *amichevole e accogliente* di ERMES.

Infine vorrei ringraziare i miei relatori Prof.Francesco Branz e Ing.Lorenzo Olivieri per la Pazienza e il supporto dimostrati durante il percorso di ERMES e della tesi.