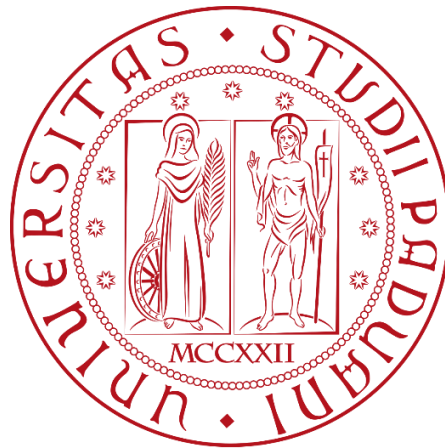


University of Padova

Department of Industrial Engineering DII

Master's Thesis in Aerospace Engineering



**Propulsive and mechanical design of a CubeSat mock-up
for low-gravity testing**

Author:

Giuliano Degli Agli

Supervisor:

Francesco Branz

Academic Year 2022/2023

*A mio padre Ingegnere,
fonte di ispirazione*

Abstract

In the last few years, smaller companies and research laboratories are focusing on satellite solutions with reduced dimension, production time and costs to conduct scientific experiments and technological demonstrations in orbit. The advent of small satellites in space missions has changed the standards of the research with a consequent increasing interest in miniaturized systems. In this scenario the most relevant studies currently under way involve micro-propulsion systems, high-throughput telecommunication terminals, guidance algorithms and navigation sensors, capture and servicing technologies. In particular the development of different technologies for proximity operations and docking manoeuvres are under investigation for several applications like on-orbit servicing (OOS), retrieval, i.e. capture and return to ground, of spacecraft and the assembly of large in-space structures (On -Orbit Assembly, OOA).

Experimental Rendezvous in Microgravity Environment Study (ERMES) is a technological demonstrator designed to test an autonomous docking manoeuvre between two free-flying CubeSats mock-ups. The experiment has been selected by the *European Space Agency* (ESA) education office, for the *Fly Your Thesis! 2022* (FYT), and performed during the 79th *ESA Parabolic Flight Campaign*.

This thesis presents the design and development of the propulsive and mechanical subsystems of the main ERMES CubeSat mock-up, the *Chaser*, which is also equipped with *Guidance Navigation and Control* (GNC) system and miniaturized docking interfaces. The *Chaser* is the active part, based on 2U CubeSat and propelled by a cold gas propulsive system based on expendable CO_2 cartridges.

Relevant focus is given to the design of the two subsystems, their implementation, testing and safety management. Moreover a first analysis of the data collected during the campaign is given, and compared with the theoretical results.

Contents

List of Figures	9
1 Introduction	12
1.1 ERMES Experiment	12
1.1.1 Scientific Background	12
1.1.2 Rendezvous and Docking Manoeuvre	13
1.2 Fly Your Thesis! Programme 2022	15
1.2.1 Parabolic flight	15
1.3 Experiment Procedure	16
2 Concept	19
2.1 Chaser	20
2.1.1 Chaser: Actuators	20
2.1.2 Chaser: Structure	23
2.1.3 Chaser: Other Subsystems	24
2.1.3.1 Docking interfaces	24
2.1.3.2 On board Computer System	26
2.1.3.3 Electronics	30
3 Propulsive Design and Development	34
3.1 Actuators design	35
3.1.1 Sizing	37
3.2 Dynamic Simulation	43
3.3 Safety Management	48
3.4 Test Integration	52
3.4.1 PWM control of electrovalves	52
3.4.2 Authority of the thrusters	57
3.4.3 Complete docking manoeuvre	60
4 Experiment Results	67
4.1 Medium Level Analysis	72
4.2 Camera Recording Analysis	79
4.2.1 Detection and Tracking	79
4.2.2 Trajectory Computation	82
Conclusions	90
Aknowledgments	96
Bibliography	98

List of Figures

1.1	Rendezvous and Docking manoeuvre	14
1.2	Gravity level during a parabola	15
1.3	ERMES experiment procedure	16
1.4	Parabolic flight profile manoeuvre	17
2.1	Rendering of the ERMES experiment set-up	19
2.2	Rendering of the Chaser mock-up	19
2.3	Schematic of pneumatic system	20
2.4	CO ₂ Cartridge	21
2.5	Pressure Regulator	21
2.6	Rupture Disk	22
2.7	Electrovalve	22
2.8	Nozzle	22
2.9	Rendering of the Chaser Structure	23
2.10	Modular structure of CubeSat mock-up and FEM analysis . . .	23
2.11	Probe-Drogue mechanical system	24
2.12	Rendering of the mechanical Probe-Drogue configuration	24
2.13	Exploded view of the Probe assembly	25
2.14	Rendering of the magnetic Probe-Drogue configuration	25
2.15	Rendering of the Androgynous configuration	25
2.16	On board Computer System scheme	26
2.17	Ranging performance of the VL6180x as reported by the datasheet	27
2.18	Example of fiducial markers used in computer vision applications	28
2.19	Software Architecture scheme	29
2.20	Chaser electrical circuit scheme	30
2.21	PCB electrical scheme	32
3.1	Thrust and Mass Flow respect to the nozzle diameter and pres- sure inlet	37
3.2	Mach variation with respect to the pressure regulator set	38
3.3	Analysis of the line with Fanno flow and isentropic motions with cross-sectional variation	39
3.4	Single convergent nozzle	41
3.5	Thruster Configuration	42
3.6	Attitude of the Chaser mock-up during the worst-case configu- ration with a generic nozzle active	45
3.7	Velocity and Energy of the Chaser mock-up during the worst- case configuration with a generic nozzle active	46
3.8	Paths of the manoeuvre with only one nozzle active	47
3.9	Nozzle setup on the mock-up structure	48
3.10	Extract of the Experimental Safety Data Package - Hazard Risk Explosion of Pressurized Systems	48
3.11	Pneumatic system of the test <i>E1</i>	50
3.12	<i>E1</i> - Experiment set-up	50
3.13	<i>E1</i> - Nozzle tested	51
3.14	Pneumatic system of the test <i>E2</i>	51

3.15	<i>E2</i> - Experiment set-up	52
3.16	Test PWM set-up	53
3.17	Response in thrust of one nozzle if controlled with a 30Hz PWM. The dotted lines represent the values of the PWM to map to the 16-levels values	55
3.18	Relation between the thrust of the nozzle and the PWM with the remapped values: step number 15 is $D = 1$	55
3.19	Response in thrust of one nozzle if controlled with a 122Hz PWM. The dotted lines represent the values of the PWM to map to the 16-levels values. The points between 0 and 200 are omitted because the mean thrust for them is 0	56
3.20	Relation between the thrust of the nozzle and the PWM with the remapped values: step number 15 is $D = 1$	56
3.21	Test Chaser and Target on the frictionless table	57
3.22	Authority and weight for trajectory corrections	58
3.23	Authority and weight for attitude corrections	59
3.24	Snapshots of the complete docking manoeuvre on the frictionless table	60
3.25	Testing on frictionless table - Relative trajectory of the Chaser with the respect to the Target with reference trajectory from external camera tracking (OptiTrack)	64
3.26	Testing on frictionless table - Relative velocity of the Chaser with the respect to the Target with reference trajectory from external camera tracking (OptiTrack)	65
4.1	ERMES post-processing flowchart	69
4.2	Acceleration profile of the Flight #2 (credits: Novespace)	70
4.3	Acceleration profile of the Flight #2 - Parabola #27 (credits: Novespace)	71
4.4	Rendering of the docked configuration	72
4.5	Target and Chaser in docked configuration	73
4.6	Output graphs from the analysis: relative trajectory of the Chaser with the respect to the Target	77
4.7	Output graphs from the analysis: relative velocity of the Chaser with the respect to the Target	78
4.8	Flowchart of the algorithm	80
4.9	Sequence of frame of the manoeuvre of Flight #2 - Parabola #27	80
4.10	Extrema point research of surface of the mock-up	81
4.11	Screen Reference System - SRS	81
4.12	Real Reference System - RRS	82
4.13	Reference Point and Selected Vertices	83
4.14	Output graphs from the analysis: relative trajectory of the Chaser with the respect to the Target	87
4.15	Output graphs from the analysis: relative velocity of the Chaser with the respect to the Target	88
4.16	Weight for global succes given to each objective	92
4.17	ERMES Experiment operating in microgravity - ©Novespace	94

1 Introduction

1.1 ERMES Experiment

Rendezvous and docking (RVD) [1] is a key operational technology, which is required for many missions involving more than one spacecraft. In particular the process consists of orbital manoeuvres and controlled trajectories, which bring the active vehicle, the chaser, into the vicinity of, and eventually into contact with, the passive vehicle, the target. The rendezvous and docking is divided into three phases: (1) the first phase used to insert the chaser in an orbit around the target; (2) then an approaching phase to reduce the distance; (3) lastly, a phase of proximity navigation that includes all the adjustments before the actual docking.

During small satellite-based missions, the proper progress of the operation relies on sensors and software that accomplish their tasks. For this reason more efficient and reliable proximity navigation and control systems for autonomous small satellites are of great interest, due to their effectiveness for several applications in space missions. A further point of interest stands in the development of miniaturized docking interfaces suitable for the satellites involved in the missions, considering their reduced sizes.

Experimental Rendezvous in Microgravity Environment Study (ERMES) is a technological demonstrator that aimed at testing the last phase of proximity navigation and the autonomous docking between two free-flying CubeSats mock-ups, the *Chaser* and the *Target*.

ERMES aims to integrate different subsystems for autonomous docking, to increase the Technology Readiness Level and to study possible applications for in-orbit servicing. The test is performed in low-gravity conditions achieved on board of the airplane Airbus A310 Zero-G operated by Novespace, during the 79th ESA Parabolic Flight Campaign.

The ERMES experiment setup consists of two free-flying CubeSat mock-ups equipped with Guidance Navigation and Control (GNC) systems. In particular, the Chaser is a 2U CubeSat mock-up ($20 \times 10 \times 10$ cm) and is equipped with a cold gas (CO_2) propulsive system characterized by a set of 8 simple convergent nozzles divided in two groups of four and fixed on opposite faces; while the Target is a 1U CubeSat mock-up ($10 \times 10 \times 10$ cm) equipped with 3 reaction wheels (RW). Therefore, the Chaser has control over its attitude and position (6 DoF), while the Target only over its attitude (3 DoF). The manoeuvre is accomplished by releasing the CubeSat mock-ups from their initial electromagnetic constraints into a free flying condition, then the dedicated localization and proximity navigation software controls attitude and position of the Chaser and slowly approaches and docks the Target. This one in the meanwhile works cooperatively by contrasting attitude disturbances.

1.1.1 Scientific Background

Autonomous space systems have always been an interesting topic in the scientific community because they allow particularly useful applications related to

large structure assembly, active space debris removal and management. The Automated Transfer Vehicle (ATV) [2] is one of the first examples of an autonomous space system that carried out rendezvous and docking with the International Space Station (ISS). Autonomous Satellite Docking System (ASDS) [3] from Michigan Aerospace Corporation (MAC) had the goal to demonstrate the capability of autonomous docking manoeuvres between satellites on orbit. More recently, the Crew-2 Mission [4] performed an autonomous docking manoeuvre with the ISS. Regarding small satellites studies some examples are: Synchronized Position Hold, Engage, Reorient, Experimental Satellites (SPHERES) [5] aboard the International Space Station (ISS) that 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. CubeSat Proximity Operations Demonstration (CPOD) [6] mission led by Tyvak Nano-Satellite Systems focused on a docking manoeuvre of two 3U CubeSats. Finally, regarding autonomous small satellite systems and, in particular, software architectures for autonomous operational capabilities, Astrobee [7] can be cited: it has been built on the legacy of SPHERES and it is composed of three autonomous cubic shape robots that help the astronauts in all their duty in the ISS.

Moreover, the University of Padova has a grounded heritage on autonomous docking manoeuvres studies for example: Flexible Electromagnetic Leash Docking system (FELDs) [8] studied an electromagnetic soft docking technology, participating in the Drop Your Thesis! Programme achieving a post-docking mechanical connection with a flexible wire; Autonomous Rendezvous Control And Docking Experiment - Reflight 2 (ARCADE-R2) [9] correctly performed three release operations and two docking procedures between 2-DOF vehicles; Position and Attitude Control with Magnetic Navigation (PACMAN) [10] focused on how a Chaser can correct autonomously the relative attitude with respect to the Target through electromagnetic actuators.

In summary, regarding the miniaturized docking interfaces, concepts based on different configurations have been proposed, which can be classified into three main categories: probe-drogue, androgynous, magnetic. A compact system based on probe-drogue configuration has been developed and already proven functional in the laboratory, on a low friction table, able to guarantee a docking manoeuvre with tolerance to misalignment [11]. Other researches focused on semi-androgynous systems that merge the advantages of the androgynous mechanism to the simplicity of the probe-drogue configuration. In this scenario must be mentioned SAM (Semi-androgynous mechanism) [12] and MUSA (MUltifunctional Semi-androgynous interface) [13]. Moreover, androgynous mechanical interface is currently being developed and there are other research programs, within the University of Padova, that aim to explore these topics.

1.1.2 Rendezvous and Docking Manoeuvre

In order to better understand how rendezvous and docking manoeuvres are accomplished, it is here presented a summary of the phases this kind of oper-

ation. In particular, referring to Figure 1.1, the manoeuvre consists of a first part of phasing, in which the misalignment between the chaser and the target is reduced. This phase ends with the acquisition of an "initial aim point" or with the achievement of a set of margins for position and velocity at a certain range. Then starts the far range rendezvous, whose objective is the reduction of distance between the two satellites, and in other words the achievement of position, velocity and angular rate conditions, necessary for the initialization of the next phase. Next, the proximity navigation phase starts, in which it is possible to distinguish two subphases: a preparatory phase called "closing", whose main objective is to prepare the chaser to start the final approach, and the final approach phase, which leads to the mating conditions [1].

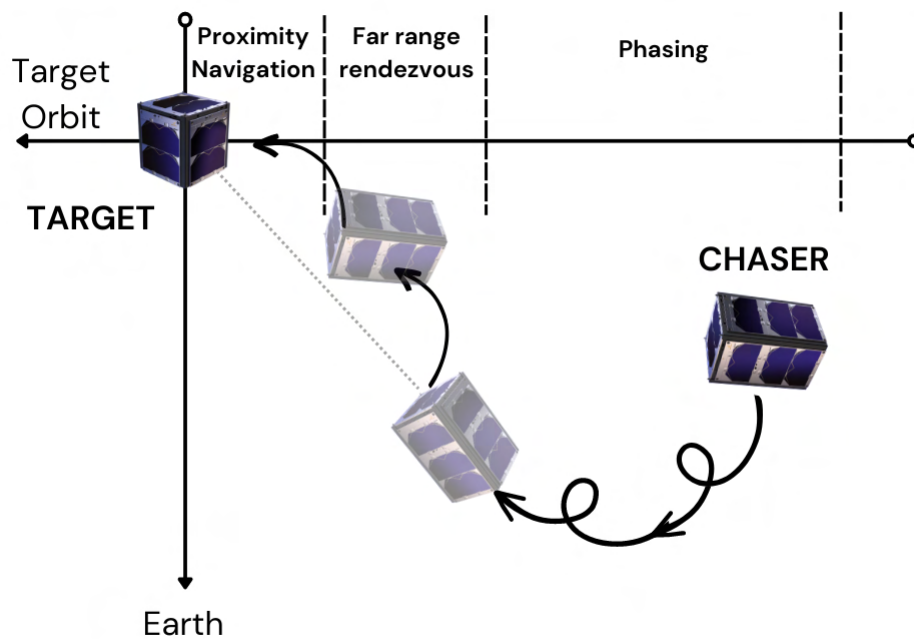


Figure 1.1: Rendezvous and Docking manoeuvre

From the navigation point of view, the proximity navigation phase is the most expensive and demanding, due to the strict safety constraints that limit the possible approach strategies and the manoeuvring velocity. For these reasons mistakes during this phase could easily lead to mission failure.

1.2 Fly Your Thesis! Programme 2022

The ERMES experiment, supported by the University of Padova, has been proposed to ESA and selected for the ESA Education Fly Your Thesis! 2022 (FYT) campaign, that allows the to conduct the experiments in microgravity environment during a series of parabolic flights on the Airbus A310 Zero-G operated by NoveSpace. The Parabolic Flight Campaign (PFC) lasts two weeks and takes place in Bordeaux. It consists of three flights with 31 parabolas each: as one parabola contains approximately 22 seconds of microgravity, the participants experience over 10 minutes of microgravity per flight and thus 30 minutes in total. Furthermore, before and immediately after each parabola there are two periods of increased gravity ($\sim 1.5 - 1.8g$) during which, if necessary, the experiments can also be tested [14].

This particular flights are characterized by a parabolic trajectory relative to the center of Earth. When the aircraft is on its top the gravitational force is fully balanced causing a sensation of weightlessness. That is beacuse, following this path, the aircraft and its payload are in free fall at certain moments.

1.2.1 Parabolic flight

The parabolic flight manouvre consists of the following steps: from a stabilized level-flight attitude, the pilot gradually pulls up the nose of the aircraft, which starts climbing at a steadily increasing angle. In this phase, which lasts for about 20 seconds, inside the airplane everything is subjected to a force between 1.5 and 1.8 times its normal weight. Once the aircraft is climbing at about 50 degrees, the pilots reduce the thrust to the minimum, adjusting the engine thrusts to compensate the air-drag. At this point, the aircraft follows a free-fall ballistic trajectory, which lasts for about 20 seconds and during which everything inside the airplane is experiencing weightlessness.

At the end of this phase, the aircraft must pull out of the parabolic arc, subjecting everything inside of it to hyper gravity conditions, as the airplane accelerates upwards. Finally, after about 20 seconds, the aircraft flies a steady horizontal path at $1g$ and scientists have about 1 minute and 40 seconds minimum before the next parabola to set up their experiments again [15].

Figure 1.2 shows the acceleration profile of the aircraft during a parabolic flight.

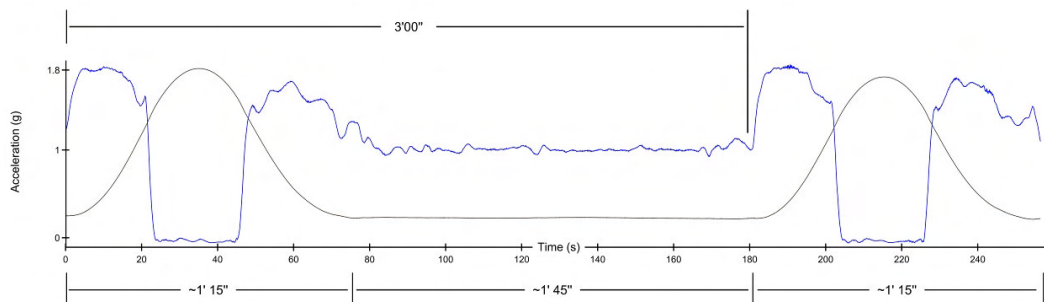


Figure 1.2: Gravity level during a parabola

1.3 Experiment Procedure

The testing of ERMES is performed during the parabolic flight described in the last paragraph, from the injection phase of the weightlessness to the starting of the pull-out. In particular for each parabola the experiment involves four main phases, marking the management and data collection procedure. The phases are described below:

- 1 **Release Phase**: the mock-ups are released from their initial electromagnetic constraints into a free-flying condition.
- 2 **Path Planning Phase**: the Chaser localizes the Target and compute the trajectory to reach it.
- 3 **Proximity Navigation Phase**: the Chaser approaches the Target by controlling its velocity and attitude, while the Target maintains the initial alignment.
- 4 **Docking Phase**: the Chaser is brought into contact with the target, and the soft docking based on magnetic constraint acts passively.

This procedure concludes with the relocation of the two mock-ups to their starting position on the release structure, and so the cycle repetition. Figure 1.3 shows the schematic representation of the experiment procedure.

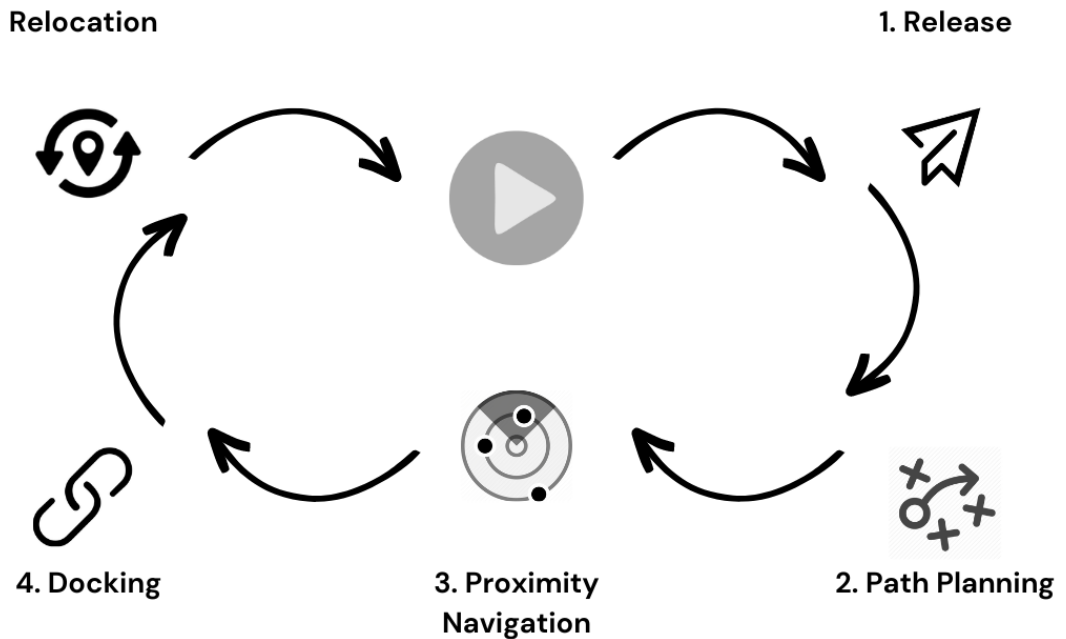


Figure 1.3: ERMES experiment procedure

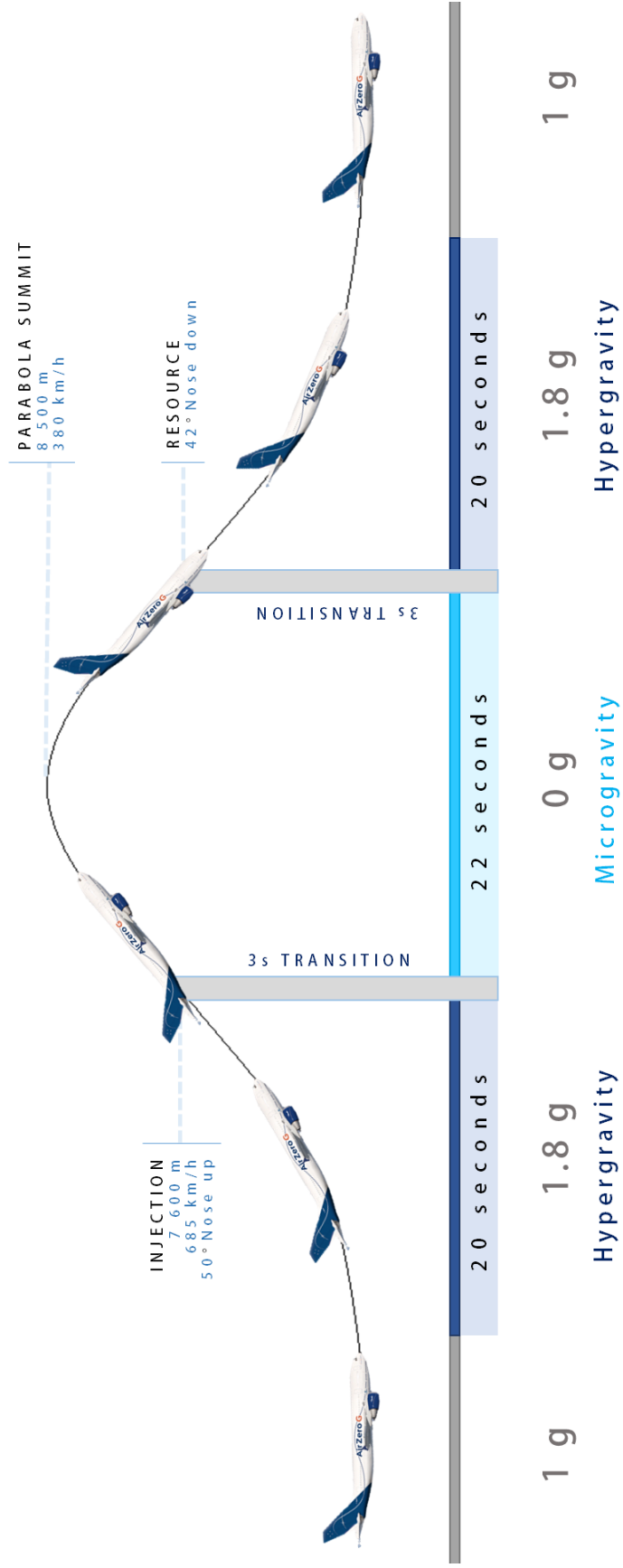


Figure 1.4: Parabolic flight profile manoeuvre

2 Concept

ERMES is composed of three main subsystems: two CubeSats mock-ups, one acting as active Chaser and the other as cooperative Target; a release structure that is used to accelerate and release the mock-ups prior to the starting of the manoeuvre (Figure 2.1). This thesis presents the propulsive and mechanical design of the Chaser mock-up. In the following subsections, the entire Chaser is described, analysing its subsystems and highlighting their role in the manoeuvre and design characteristics.

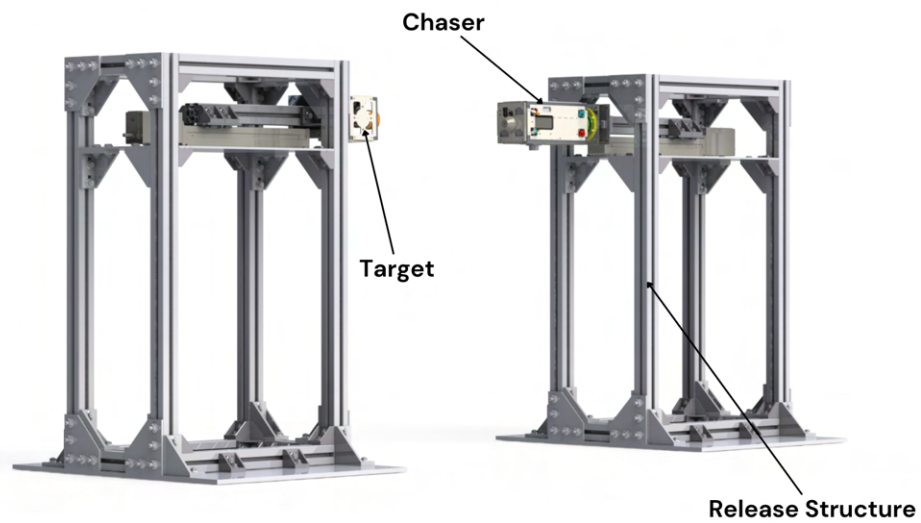


Figure 2.1: Rendering of the ERMES experiment set-up

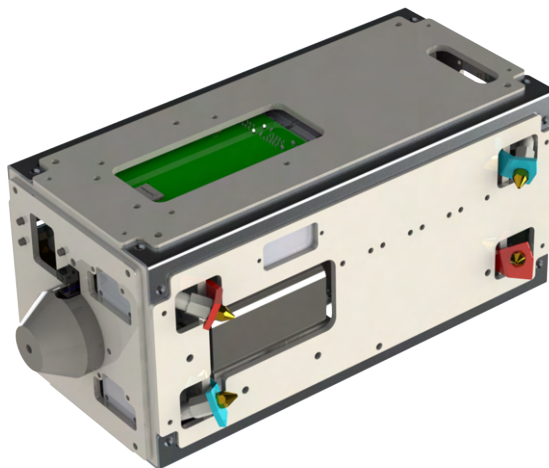


Figure 2.2: Rendering of the Chaser mock-up

2.1 Chaser

2.1.1 Chaser: Actuators

The Chaser mock-up is equipped with a set of eight actively controlled thrusters connected to a pneumatic subsystem (shown schematically in Figure 2.3) which consists of: (1) a CO_2 threaded cartridge (C1); (2) a pressure regulator directly attached to the cartridge thanks to the threaded input port (PR1); (3) one rupture disk to ensure a safety margin on the working pressure (RD1); (4) tubing and connector or adapters (T1-21 and D1-5); (5) miniaturized solenoid electrovalves (SV1-8); (6) simple convergent nozzles (N1-8).

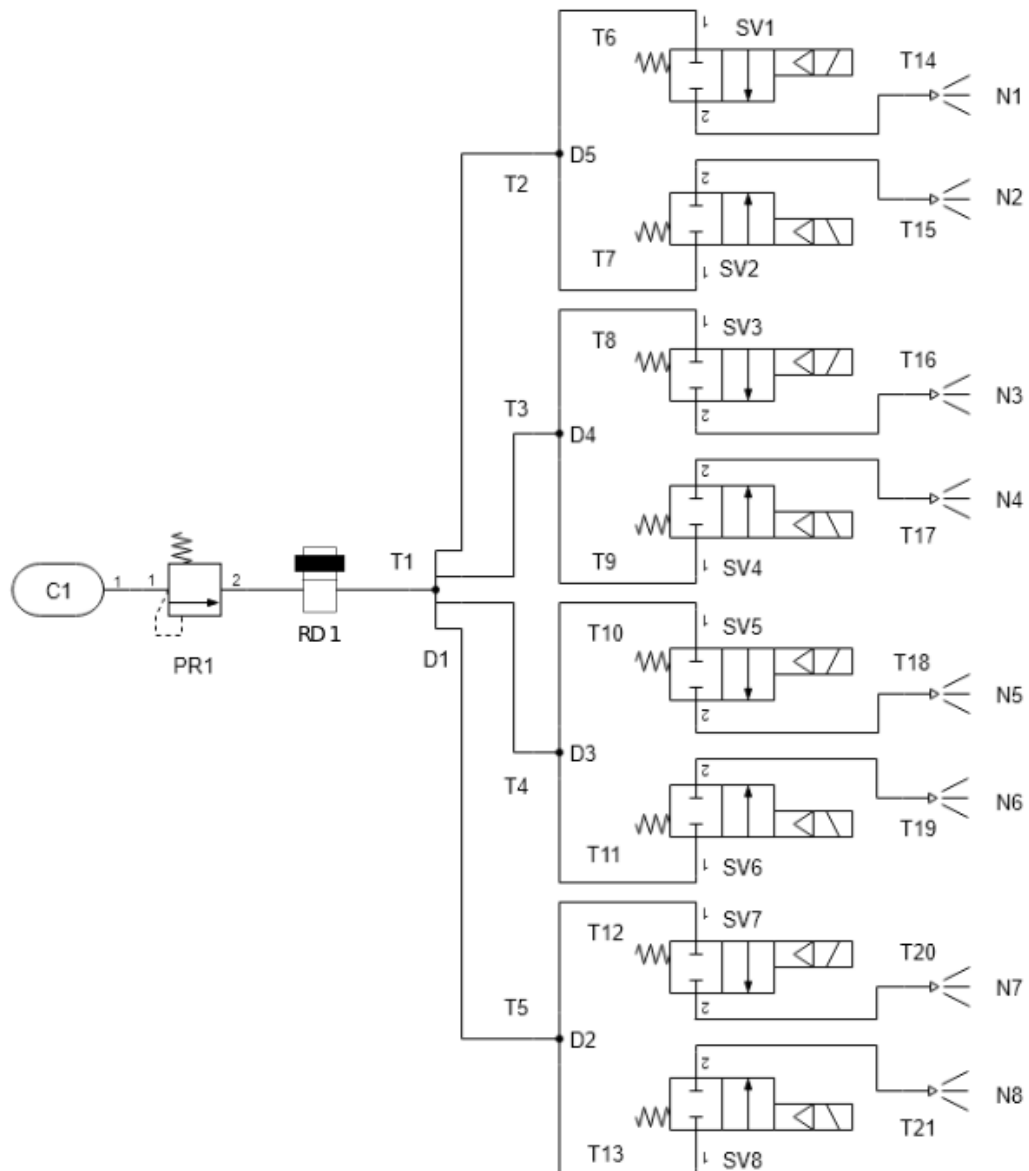


Figure 2.3: Schematic of pneumatic system

- 1 The selected CO_2 cartridges contain 16g of CO_2 and are commercially available, from *Barbieri snc.*. They are threaded (3/8 UNF-24) and consequently easily connectable to the pressure regulator with the same input thread. According to the datasheet and the information provided by the manufacturer, the pressure of the CO_2 is around 57 bar at standard condition of temperature, and assuming liquid state. Their technical specifications are conformed to constraints regarding "high pressure per volume factor":

$$\frac{M_{[CO_2]}}{\rho_{[liquid-CO_2]}} \cdot P_{[liquid-CO_2]} = P_{[liquid-CO_2]} \cdot V_{[liquid-CO_2]} = 1.76bar \cdot L \quad (2.1)$$

with $M_{[CO_2]}$ the mass of CO_2 contained in the cartridge, $\rho_{[liquid-CO_2]}$ the density of the liquid CO_2 , $P_{[liquid-CO_2]}$ the pressure of the liquid CO_2 contained in the cartridge and finally $V_{[liquid-CO_2]}$ the volume of liquid CO_2 contained in the cartridge.

During the usage, the cartridge freezes up to $-8^\circ C$, for this reason it is covered by a polyurethan insulation sleeve that withstand up to $-60^\circ C$.

- 2 The pressure regulator is *Ikegger model Mini*. The regulator stands a maximum pressure of 82 bar of input and can regulate pressure between 0 and 5.5 bar.

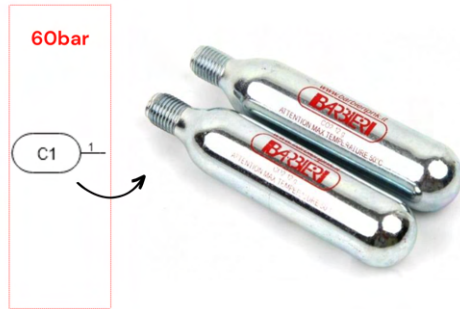


Figure 2.4: CO_2 Cartridge

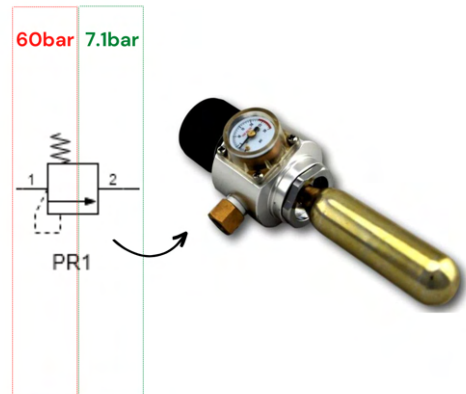


Figure 2.5: Pressure Regulator

- 3 A rupture disk is included to avoid high pressure bursts in the pneumatic system in case of a failure of the regulator. The selected configuration is the *Zook TA series*. It is a custom designed component, manufactured specifically following the ERMES requirements. It has a rupture pressure of ~ 6.5 barg with a precision lower than 10%. Therefore, the maximum design pressure (MDP) is set at 7.1 barg.

4 The miniaturized electrovalves are commercially available, from *SMC*, in particular the model *PVQ31-1.6* has been selected. They withstand a maximum pressure of 10 bars guaranteed by the relief valves. They represent the key component of the pneumatic system because they are the actual actuators controlled by the On board Computer System.

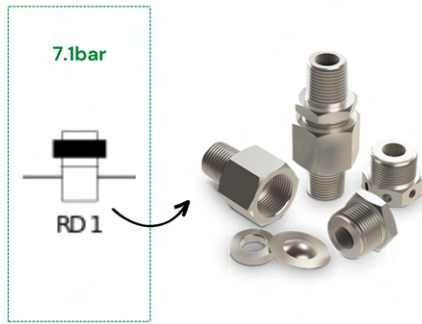


Figure 2.6: Rupture Disk

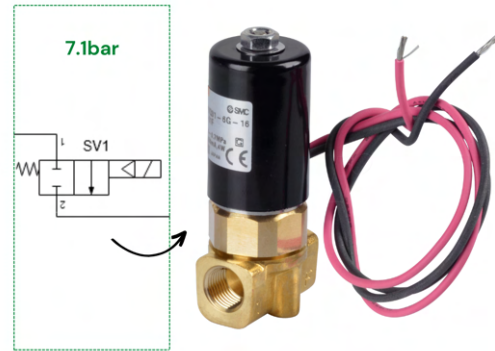


Figure 2.7: Electrovalve

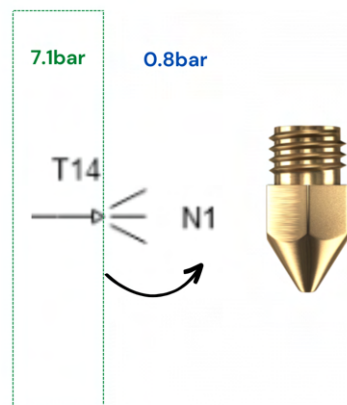


Figure 2.8: Nozzle

5 The eight nozzles are selected to be simple convergent instead of a classic convergent-divergent configuration. This comes from the necessity of avoiding supersonic flows since the experiment takes place at standard atmospheric pressure and not in a vacuum chamber. In particular 3D-printer nozzles have been selected, with an orifice diameter of 1 mm. This choice is a perfect compromise between simplicity and requirement of the system.

- 6 The selected tubes are all PVC 4×1 mm due to their relatively high flexibility and from the conducted sizing.

2.1.2 Chaser: Structure

The structure provides the mechanical connection and support to electronics, boards, actuators and internal structure in place. The design of the structure follows the CubeSat standards, which are a type of miniaturized satellites made up of $10 \times 10 \times 11.35$ cm units, designed to provide $20 \times 20 \times 20$ cm of useful volume (2U CubeSat). As already mentioned the Chaser is a 2U CubeSat structure as shown in Figure 2.9. In order to prevent unwanted damages and risks during the free-flying phase, the structure has been externally covered by damping material, in particular elastomeric foam based on synthetic rubber. Moreover it is important to underline that preliminary design and safety considerations for the design and manufacture of the structure were based on the heritage from the PACMAN experiment. The study has been validated by FEM analysis on deformations and equivalent stress. The simulations were made considering the PSD profile and the launching loads of a Dnepr launcher, in the prospect of a future space mission. In Figure 2.10 the standard module derived from PACMAN experiment [16].

The load-bearing structural parts are represented by the 4 main columns, manufactured in aluminium, through bending process. While all the plates are manufactured in polycarbonate, through milling process. All the junctions of the structure are bolted.

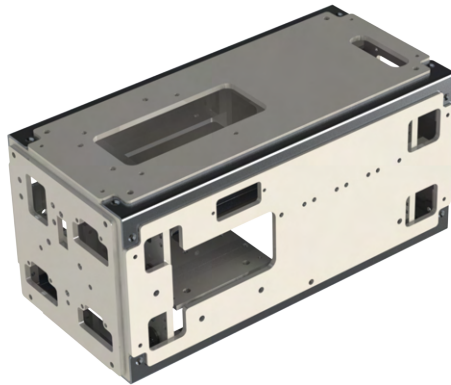


Figure 2.9: Rendering of the Chaser Structure

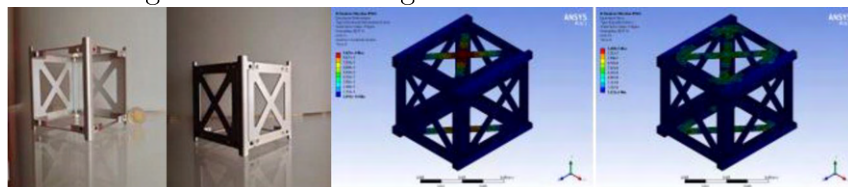


Figure 2.10: Modular structure of CubeSat mock-up and FEM analysis

2.1.3 Chaser: Other Subsystems

2.1.3.1 Docking interfaces

Two different configurations are taken into account for the flight tests: a probe-drogue configuration and an androgynous configuration. Only the first one is used during the campaign. The connection between the two mock-ups is guaranteed thanks to a magnetic link, in fact both configurations are characterized by a small permanent magnet on one side and a magnetic counterpart on the other side.

In the first configuration, the Chaser is equipped with a probe and the Target with a drogue. The shape of the probe-drogue configuration facilitates the manoeuvre by reducing the possibility of unwanted bouncing which can separate the mock-ups. The original design of the probe-drogue miniaturized docking interface is shown in Figure 2.11. The difference between the ERMES probe-drogue configuration and the original one described in the article [11], is the above mentioned magnetic constraint, in fact the tip of the probe have been replaced with a small magnet, in order to move from a mechanical docking to a soft docking configuration (see Figure 2.12 and 2.14). This solution derived also from the fact to avoid damage due to the servomotor dedicated to the lock (see Figure 2.13). In fact referring to the ERMES Experiment Safety Data Package, the hazard risk connected to contamination of the environment due to debris has to be managed. In particular debris generation following a fall of the prototype on the floor during the 2G phase or the 0G to 2G transition. Verification and testing of the equipment expects drop test, in which the foam covered mock-up has been dropped from a height of 1.5m on a mattress. This puts restriction to protruding and moving parts, that have to be impact-resistant.

Furthermore to test the strength of the link given by the magnetic docking interfaces, the drop test with mock-up docked has been successfully executed.

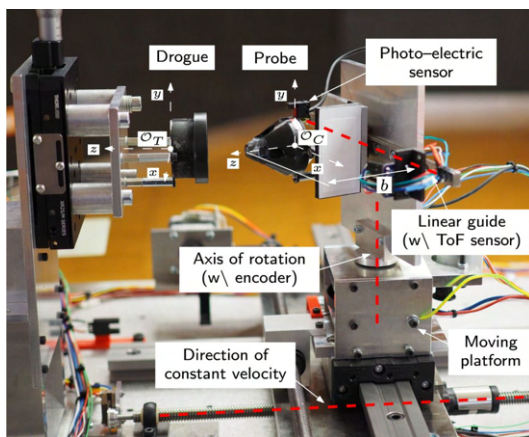


Figure 2.11: Probe-Drogue mechanical system

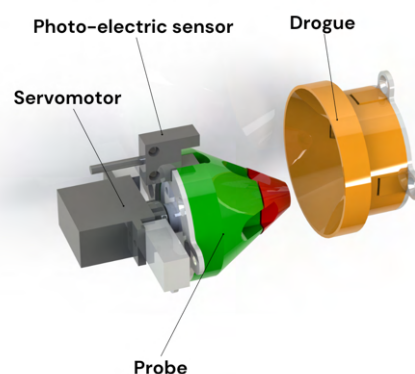


Figure 2.12: Rendering of the mechanical Probe-Drogue configuration

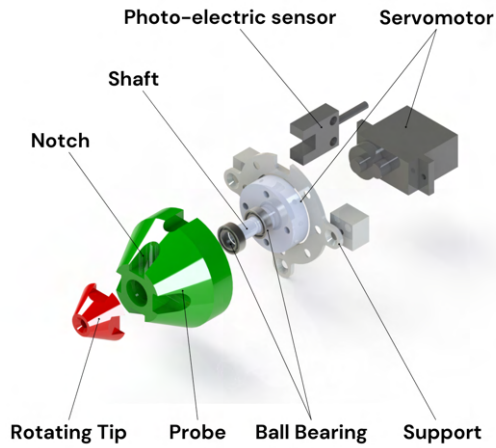


Figure 2.13: Exploded view of the Probe assembly

In the androgynous configuration, shown in Figure 2.15, the docking interface is the same for both Chaser and Target. The connection between the mock-ups is achieved thanks to the magnet and its counterpart placed at the centre of them. This configuration has been designed to have a shape that helps the docking by auto centring the interfaces.

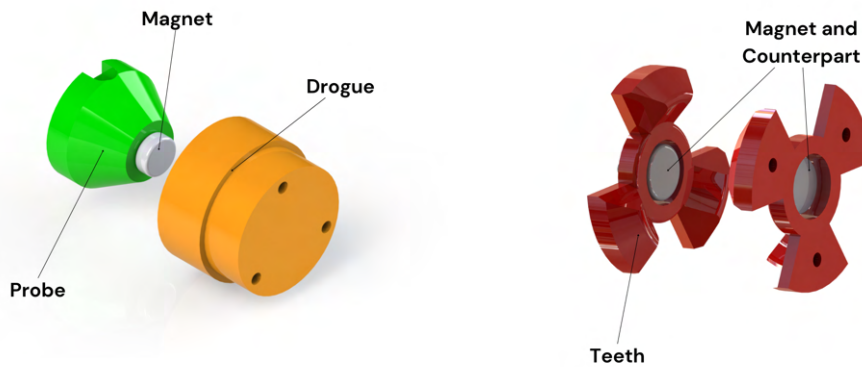


Figure 2.14: Rendering of the magnetic Probe-Drogue configuration

Figure 2.15: Rendering of the Androgynous configuration

2.1.3.2 On board Computer System

Both mock-ups are equipped with on board computer system composed by a group of sensors, the computer unit, and the actuators. The Chaser has the following sensors (On board Computer System scheme in Figure 2.16): two IMU units, three Time of Flight sensors (ToF) and a Pi Camera.

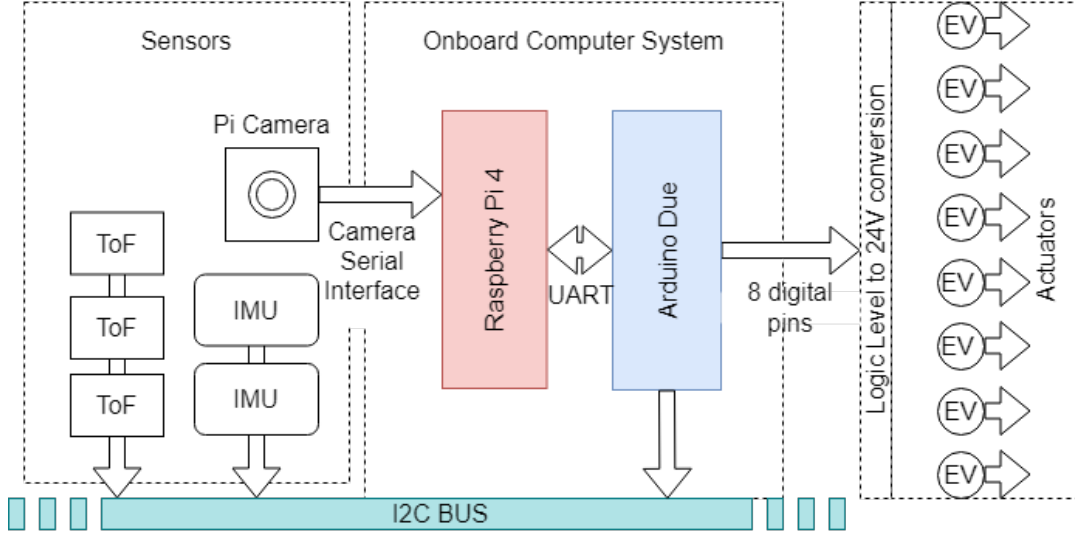


Figure 2.16: On board Computer System scheme

The IMU are two Pmod NAV sensors, they have the LSM9DS1 chip module that provides a 3-axes accelerometer, 3-axes gyroscope and 3-axes magnetometer. Then the three ToF sensors are the VL6180x model. They provide the distance of the target and also its orientation. According to the datasheet these sensors are unreliable under 1cm (Figure 2.17), for this reason their measurements are considered only during the path planning phase of the manoeuvre. The camera is used to implement a computer vision system. The principal algorithm is the AprilTag Detection [17] [18] [19], a fiducial tag recognition system.

The computer module is composed by two parts: two Arduino to compute and control all Hard Real Time systems, a Raspberry Pi 4 computer unit for the Soft Real Time. The actuators are eight valves open by one MOSFET each.

The Proximity navigation software is based on three main levels.

- Low Level Software (LLS), with the actuator and control systems, runs on the Arduino boards;
- Medium Level Software (MLS), with the Guidance, Navigation and Control, runs on the Raspberry Pi 4;
- High Level Software (HLS), that takes care of the communication between the operators and the Chaser, runs on the laptop.

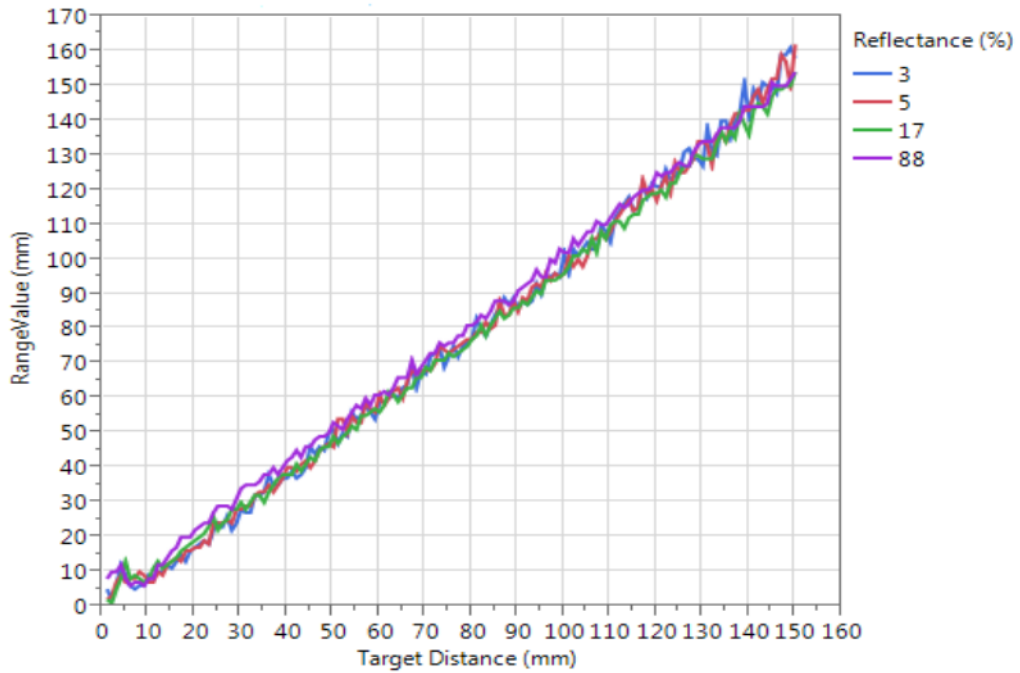


Figure 2.17: Ranging performance of the VL6180x as reported by the datasheet

The Low Level Software deals with the control of the opening of the electrovalves, the reading of the data from sensors connected to the I2C line, the deserialising of the command received from the MLS and the sending data to the MLS. In particular, it is composed of 6 independent closed-loops Single Input Single Output (SISO) controllers dedicated to a single Degree of Freedom each. The feedback is guaranteed by the IMUs, from which the low-level extracts information about the linear and angular accelerations. As mentioned, the input is a step-like command of acceleration, while the output is the duty cycle of the valve needed to actuate the command.

The Medium Level Software deals with the recognition of the Target, path calculation and command transmission (via UART communication). In the control theory, its architecture is identified through the Guidance, Navigation and Control scheme.

The Guidance refers to the determination of the trajectory from the Chaser current location to the Target, as well as desired changes in velocity and rotation. For the software, the trajectory consists of a series of states, each containing position, rotation and linear and angular velocity information.

The Navigation is the determination, at a given time, of the state of the vehicle. Its state is determined by the position, velocity and attitude. In particular to determine this status, the Chaser needs data from sensors, including a vision system capable of locating the Target. There are a lot of solutions for a computer vision based system, but ERMES software focused in a fiducial tag based recognition. Fiducial tags are objects positioned in the field of view of the camera for use as a point of reference or a measure, examples are shown in Figure 2.18. They are used to recognise the position of the Target, so it will

have at least one tag with a defined pattern and a known size. The computer vision software will recognize the tag and compute the position and orientation of the Target with respect to the camera.

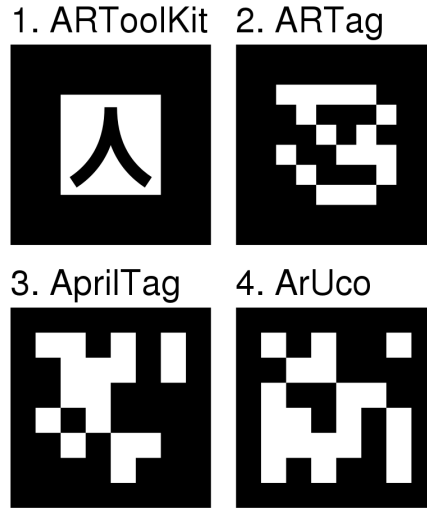


Figure 2.18: Example of fiducial markers used in computer vision applications

The MLS processes the sensors data provided by the LLS. Considering the nature of the data recorded by the sensors, in addition to the position relative to the target provided by the vision system, there are also the acceleration along the axes and velocity around them, provided by IMU and the relative distance between target and chaser, provided by the 3 ToF.

Information from the control system is also used in determining the status of the Chaser. At each control cycle, a predicted state is generated in accordance with the command sent. The control system sends commands in the form of an applied force. Therefore the Control can generate a prediction of what the next state of the Chaser will be. To do this it must integrate acceleration with respect to time, and use the previous state as an initial condition. The first state used by the control is initialised with the initial conditions of the experiment, which are known. The Navigation, considering all the data from the sensors explained above, considering the predictive model of the control system and also the status generated in the previous step, will generate the estimate of the current status of the Chaser. Taking into account that each piece of information has an uncertainty associated with it, the localization can also estimate this uncertainty of the current generated state. To do this, a filter is needed, and in particular it is used the Extended Kalman filter (EKF). It is the nonlinear version of the Kalman filter [20] which linearizes about an estimate of the current mean and covariance.

The Control takes care of generating the messages that can be executed by the LLS. They are a tuple with two main elements: the force required and the time. Using the trajectory provided by the guidance, the Control must generate messages allowing it to move from one point on the trajectory to the next. Since it has been assumed that guidance always generates trajectories in which acceleration along only one axis is required between each point and the

next, the control could generate two commands at most between two points. Therefore, the Control plans the above accelerations, with time constraints: between two different points of the trajectory it can use a percentage of the time for the two commands of acceleration and deceleration, and keep the rest for the commands of trajectory adjustment. The percentage of the time available for the acceleration and deceleration commands is a parameter set by the project, and can be tweaked to have the right compromise: the lower this parameter, the longer the time needed to travel the distance between one point on the trajectory and the next. During the adjustment time, the control will generate the necessary impulses to maintain a certain attitude and direction. The attitude and the translation movement are corrected with the information received from the location relative to “odom”. Using the information from the Navigation relative to Target, it could estimate the real target position in the odom coordinate system, and use this information to correct the manoeuvre.

Finally, the High Level Software is used to interact with the experiment. The two main tasks of the HLS are to start (and stop) the experiment and to make the Chaser telemetry available to the user.

The software architecture is summarized in Figure 2.19.

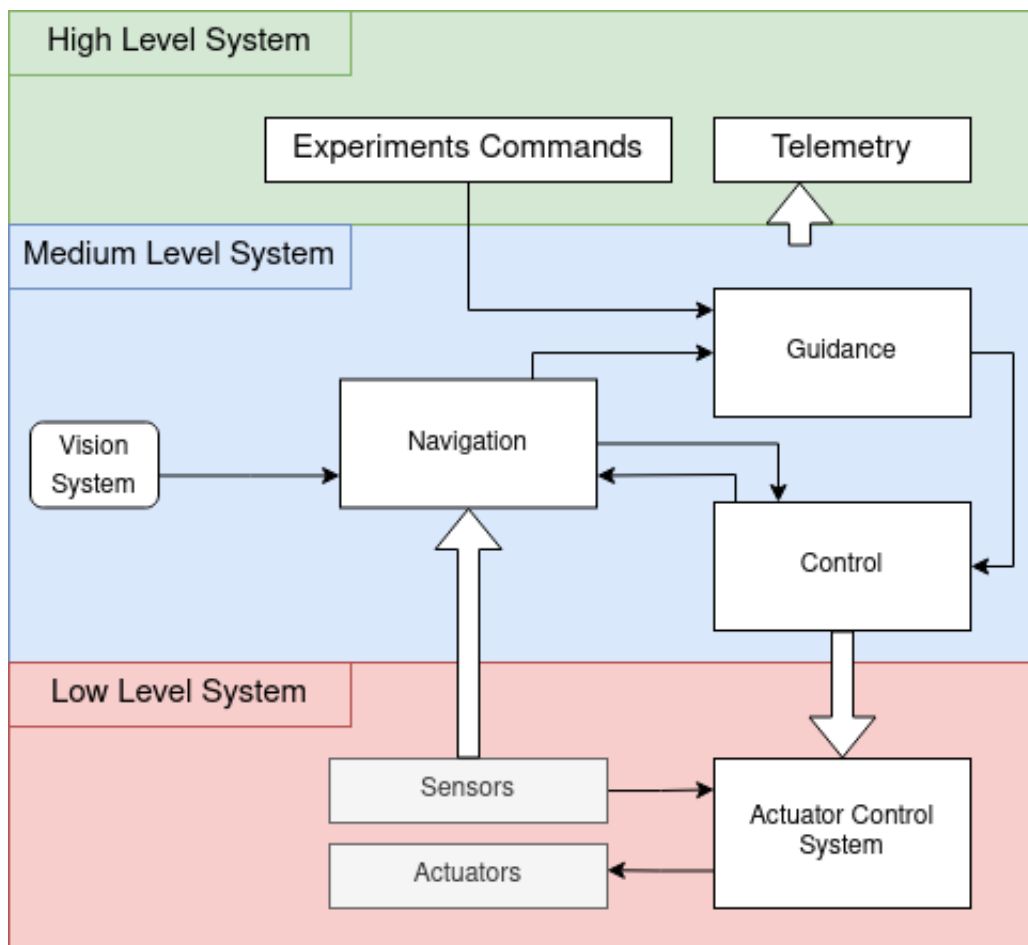


Figure 2.19: Software Architecture scheme

2.1.3.3 Electronics

The electrical system of the Chaser (showed in Figure 2.20) includes: (1) a battery pack; (2) a control panel; (3) one boost converter; (4) one buck converter; (5) a break out board; (6) a Raspberry Pi4; (7) an Arduino UNO; (8) an Arduino NANO; (9) eight electrovalves; (10) a PCB for the electrovalves control; (11) a multiplexer to deal with a lot of sensors together; (12) sensors (ToF, IMU and a PiCam); (13) a Bluetooth module for the laptop high level communication; (14) the fork photosensor for the docking mechanism.

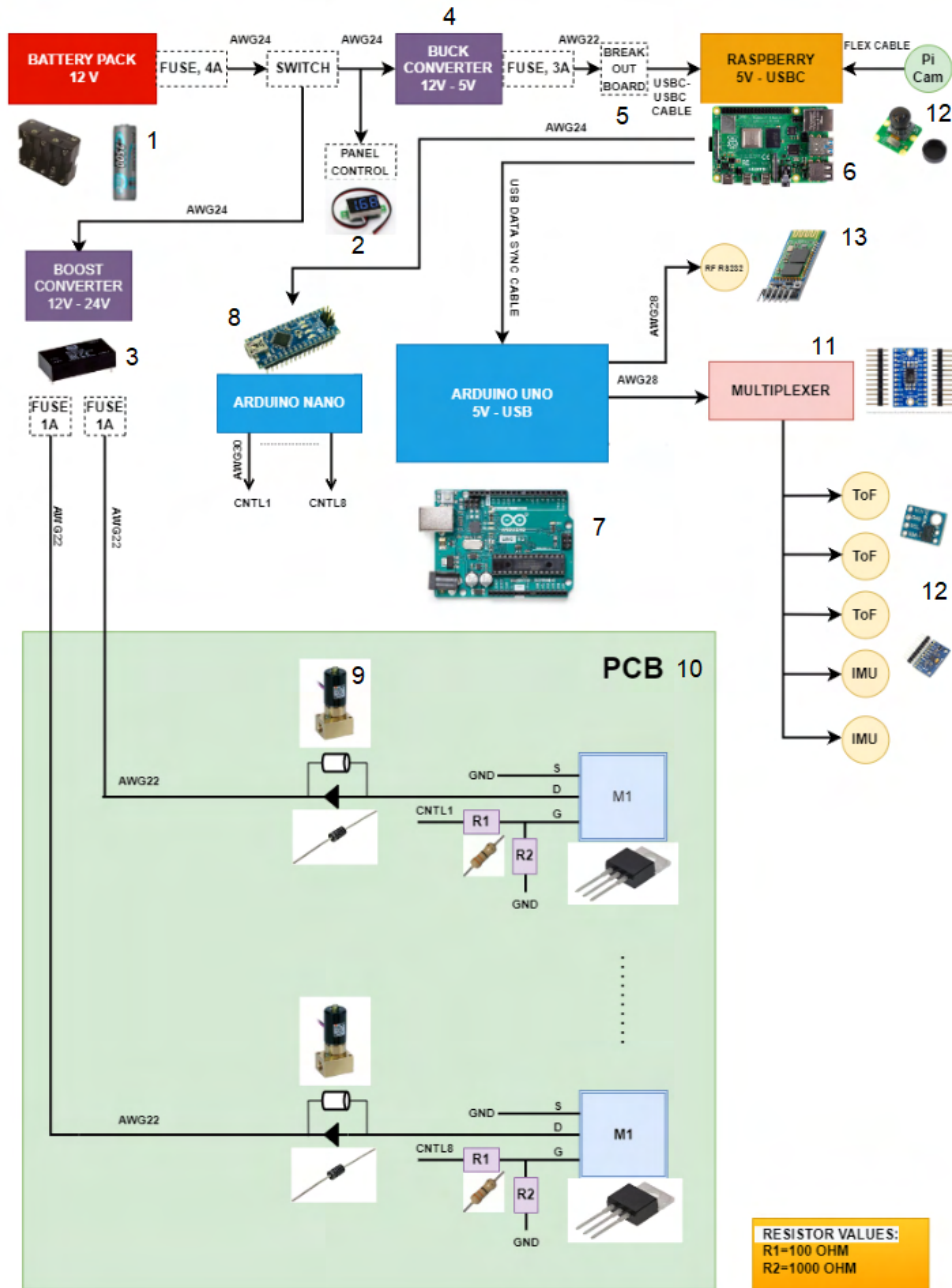


Figure 2.20: Chaser electrical circuit scheme

- 1 The battery pack is composed of 10 rechargeable AA NiMh (Nickel Metal Hydride) cells "Ansmann - 1.2V, 2.5Ah, NiMH". It provides a total of 24 Wh. It provides a 12 V with a maximum discharge current of 5 A. The voltage is then reduced to 5 V for the boards and increased to 24 V for the electrovalves.
- 2 The control panel shows the level of the battery. This feature is included for safety reason in order to monitoring possible voltage peaks and eventually intervene via switch included in the circuit to manually stop the power supply.
- 3 The boost converter is the "Mean Well MDS20A-24". It increases the voltage from 12V to the 24V needed for the correct electrovalves control.
- 4 The buck converter is the "Mean Well MDS20A-5". It decreases the voltage from 12V to the 5V needed for the powering of the Raspberry Pi4.
- 5 A break-out board is implemented so that it works as an adapter between the cable from the buck converter and the USBC cable for powering the Raspberry Pi4. The break-out board is the "SparkFun BOB-15100 USB-C Breakout Board".
- 6 The main board is the Raspberry Pi4. It is connected to the breakout board via USBC cable. To the Raspberry it is directly connected the PiCam and both Arduino UNO and NANO. The Raspberry PI4 has as its the main role to elaborate the data of the sensors sent by Arduino UNO, to compute the trajectory and send the relative commands to the Arduino NANO to properly actuate the electrovalves
- 7 The first connection with Raspberry Pi4 is the Arduino UNO. It has connections with the Bluetooth module, the electromagnets, the servo motor of the docking interface and finally the Multiplexer. Arduino UNO is linked to the multiplexer and allows the connection I2C with three ToF and two IMU, taking information from sensor and sending these to the Raspberry PI4 that proceeds with the computation.
- 8 The second connection with Raspberry Pi4 is the Arduino NANO. It is used to control eight MOSFET linked to the electrovalves. The selected MOSFET are IRL40B215.
- 9 As cited in the *Section 2.1.1*, the Electrovalves chosen for the experiment are "SMC series PV Q31-1.6". Each electrovalve is powered by the boost converter and is controlled electrically by one of the eight MOSFET: when Arduino NANO set its output pin as high, one MOSFET allows the current to flow, turning on the electrovalve. Each electrovalve is connected to the Drain of each MOSFET and to the boost converter via cable through a Printed Circuit Board (PCB).

10 A PCB has been implemented to ease the connection of the electrovalves with MOSFET and the boost converter, acting as the source of power for the electrovalves. In this PCB there are eight MOSFET that have for each a protection circuit composed of two resistors and a diode connected in anti-parallel with the electrovalve. In Figure 2.21 the scheme of the PCB.

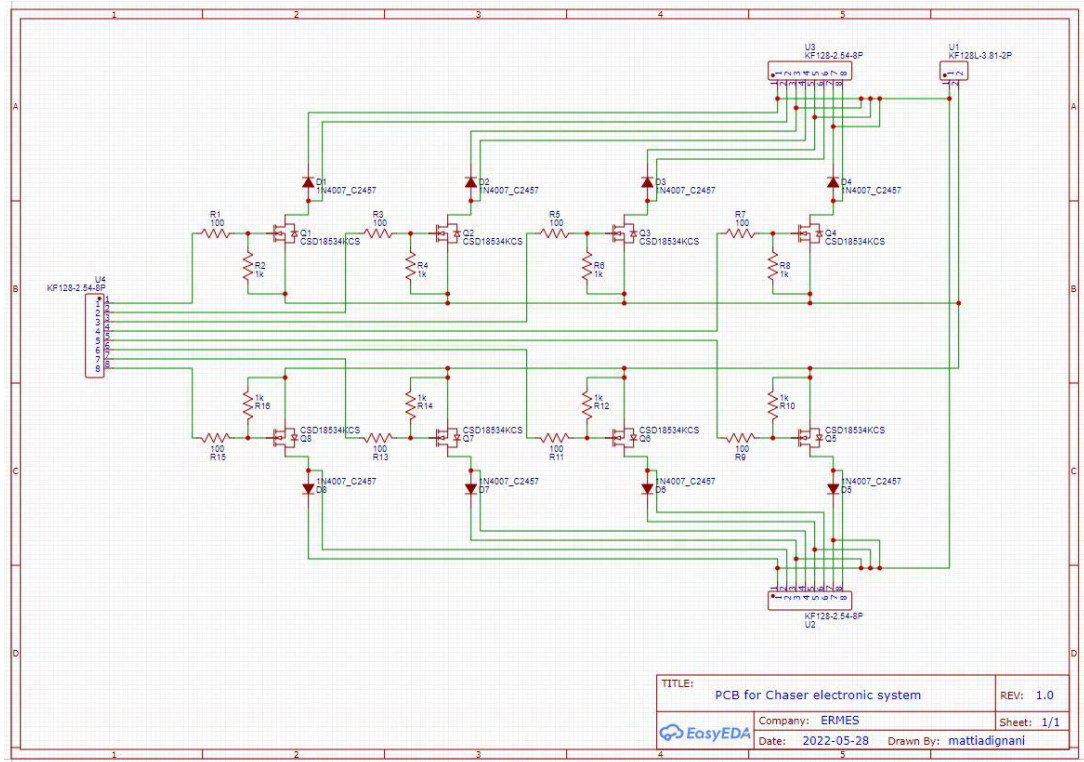


Figure 2.21: PCB electrical scheme

11 The Multiplexer chosen is the “Adafruit TCA9548A” and it is used to manage and the communication I2C between the Arduino UNO and three ToF and two IMU.

12 The Sensors that are managed by the multiplexer are three ToF and two IMU. ToFs and IMUs are connected to the multiplexer. The other sensor is the Pi Camera, connected with a flex cable to the Raspberry Pi4. It is used during the first phase of the experiment to check the distance between the Target and the Chaser and it is the main sensor for the localization system via AprilTags.

13 The Bluetooth module is used for high communication with the laptop needed for the telemetry.

3 Propulsive Design and Development

The propulsion system technology chosen by ERMES is cold gas propulsion. It has been developed because of its simplicity, compact interface to the mock-up, low power requirement, and its ability to incorporate a variety of inert propellants.

Cold gas propulsion technology relies on gas expansion, in particular through the acceleration of the gas via a nozzle using only the available stored enthalpy without any combustion, heat addition, or other mechanism for adding energy. The propellant feed system consists of a pressure differential provided by the pressurized tank or a self-pressurization feed system between the propellant tank and the nozzle. The only required power is to control valves that open and close the line, managing the activation of the thrusters. The thrust produced is directly proportional to the pressure of the propellant inside the propellant storage tank. The tank pressure decreases over the course of the mission due to the use of the propellant stored in the tank, consequently, the maximum thrust produced by the system decreases. Furthermore, cold gas propulsion systems are characterized by low specific impulse, relative to chemical or electric propulsion systems, depending on the propellant atomic mass. In addition to its attributes, cold gas propulsion technology has a well-established space operation heritage.

A basic propulsion system consists of a propellant storage tank, a valve and a nozzle. Each of these is a distinct component and is typically joined to the others by tubes. A joint occurs where each tube interfaces with a component and it is the most likely point for leaks. The tank houses the propellant required for attitude control of satellite for its operation. The propellant is used either in liquid or gaseous state (compressed). Thrusters provide sufficient amount of force to maintain equilibrium in pitch, yaw and roll of the mock-up. As mentioned the thrusters are managed by valves, in particular solenoid valves based on electrical control. The solenoid valve opens and closes the propellant flow to the nozzle. With the help of this, the thrust production frequency can be adjusted. Moreover, the period can also be controlled. The working principle of a solenoid valve is as follows: when no voltage is applied to the valve, the line to the nozzle is cut off by the gas pressure and the force of a spring. On the other hand, when the voltage is applied, a magnetic field is created, which forces the spring to open the line. After passing the solenoid valve, propellant enters the nozzle. Here, the propellant accelerates, and the exhaust gas velocity depends on factors such as inlet pressure, exhaust pressure, nozzle profile such as area ratio, diverging angle, etc.

Regarding the propellants, a higher atom weight of the expelled gas is desirable for the third Newton law. Together with the contamination free property, the moderately low boiling and melting temperatures are preferable features for a propellant gas from system point of view. However, mass efficient storage of the gas is a major concern. The following table illustrates the common propellants used in cold gas propulsion system.

Propellant	Molecular Weight (kg/kmol)	Density (g/cm ³)	Specific Thrust (s)	
			Theoretical	Measured
Hydrogen	2.0	0.02	296	272
Helium	4.0	0.04	179	165
Nitrogen	28.0	0.28	80	73
Ammonia	17.0	Liquid	105	96
Carbon dioxide	44.0	Liquid	67	61

Table 3.1: Propellant Technical Data

The main roles of the cold gas system is the attitude control of the mock-up, orbit maintenance and manoeuvring. The benefits connected to the developing of the propulsion system for the experiment are numerous, in particular:

- control of the 6 degrees of freedom of the Chaser mock-up, covering all the possible translations and rotations required by the manoeuvre, with a dedicated thruster configuration;
- control of the velocity of the mock-up for the manoeuvre with dedicated solenoid valve control;
- customization of the system for the experiment setup, with a wide choice of components commercially available.

3.1 Actuators design

In order to design a cold gas propulsion system, it is imperative to start from the Ideal Rocket equation of Tsiolkovsky and its corollaries, that are used to convert the maximum change of velocity requirements into propellant requirements.

$$\Delta V = I_{sp}g_0 \cdot \ln(m_i/m_f) \quad (3.1)$$

where $I_{sp}g_0$ is the effective exhaust velocity, with I_{sp} the specific impulse and g_0 the standard gravity, then m_i is the initial total mass and m_f the final total mass.

Thrust is generated by momentum exchange between the exhaust and the mock-up and by the pressure imbalance at the nozzle exit. It can be expressed by the relation:

$$T = \dot{m}v_{eq} = \dot{m}I_{sp}g_0 \quad (3.2)$$

where v_{eq} is the effective exhaust velocity and \dot{m} is the mass flow, obtained by the gas dynamic and function of the gas characteristics:

$$\dot{m} = \frac{p^* \cdot A_e}{\sqrt{\gamma \cdot R \cdot T^*}} \cdot \sqrt{\frac{2}{\gamma - 1} \cdot \left[\left(\frac{p^*}{p_0} \right)^{\frac{1-\gamma}{\gamma}} - 1 \right]} \quad (3.3)$$

From the second Newton law it is possible to express the Equation 3.2 as the combination of the two cited thrust generating effects:

$$\begin{cases} T_v = \dot{m}v_e \\ T_p = (p_e - p_a)A_e \end{cases} \quad (3.4)$$

As consequence, the general thrust equation is:

$$T = \dot{m}v_e + (p_e - p_a)A_e \quad (3.5)$$

where p_e the pressure of the exhaust and p_a is the ambient pressure.

Introducing c^* , the characteristic velocity, that is a figure of thermochemical merit for a particular propellant and may be considered to be indicative of the combustion efficiency. The expression for ideal c^* is given in Equation 3.6, and is seen to be solely a function of the products of combustion (k , MM , T_o).

$$c^* = \sqrt{\frac{R'/MMT_o}{k\left(\frac{2}{k+1}\right)^{\frac{k+1}{k-1}}}} \quad (3.6)$$

remembering that the specific gas constant is given by the ratio of the gas constant to the molar mass of the specific gas.

Considering the thrust coefficient c_f , that is a function of nozzle inlet pressure, nozzle outlet section pressure and expansion ratio, and it expresses how well kinetic energy is converted in the nozzle. The equation for c_f is the following:

$$c_f = \sqrt{\left(\frac{2}{\gamma-1}\right)\left(\frac{2}{\gamma+1}\right)\left[1 - \left(\frac{p_e}{p_0}\right)^{\frac{\gamma-1}{\gamma}}\right]} + \frac{p_e - p_a}{p_0} \frac{A_e}{A_{throat}} \quad (3.7)$$

where p_0 is defined as the pressure at the inlet of the nozzle. For the type of propellant gas it is known the γ gas constant, and from which the graph of the thrust coefficient as a function of the expansion ratio is determined.

The expansion ratio of the nozzle is given as:

$$\frac{A_e}{A_{throat}} = \frac{1}{M_e} \left[\left(\frac{2}{\gamma+1}\right) \left(1 + \frac{\gamma-1}{2} \cdot M_e^2\right) \right]^{\frac{\gamma+1}{2\gamma-1}} \quad (3.8)$$

Considering an isentropic transformation at the nozzle, the pressure ratio transformation is given as:

$$\frac{p_e}{p_0} = \left(1 + \frac{\gamma-1}{2} \cdot M_e^2\right)^{\frac{\gamma}{\gamma-1}} \quad (3.9)$$

Knowing c^* and c_f it is possible to find the relation with I_{sp} .

$$\begin{cases} c^* = p_0 A_{throat} / \dot{m} \\ c_f = \frac{T}{p_0 A_{throat}} \\ I_{sp} = c^* c_f / g \end{cases} \quad (3.10)$$

Also the thrust is function of c^* and c_f . The objective is to determine the inlet pressure required at the nozzle for a given expansion ratio. Alternatively, known the inlet pressure of the nozzle, the expansion ratio of the nozzle and thus its geometry can be optimized.

3.1.1 Sizing

Defining as boundary conditions the A310 Zero-G ambient, with a pressure of 0.8 bar, and as propellant the CO_2 , whose technical data are:

Propellant	Molecular Weight ($kg/kmol$)	Temperature (K)	Density (g/cm^3)	γ Constant ($-$)
Carbon dioxide	44.0	288.15	0.00187	1.3

Table 3.2: CO_2 Technical Data

the following design evaluations are conducted.

Firstly, the pulses in acceleration and deceleration of the mock-up are the same so the choice of thrust is not governed by external effects. For this reason, having a higher or lower value simply affects the speed of execution of the manoeuvre but not its feasibility. One therefore chooses a thrust such that it handles the sizing. In particular thrust between 0.01 and 0.25 N is assumed. These reference values are calculated by assuming uniformly accelerated motion for $t/2$ and then decelerated for the remaining time to cover the distance between the two mock-ups, assumed around 200 – 300 mm and staying within the window of the maximum 20 seconds of manoeuvring time.

The graphs show, respectively, the thrust curve as the nozzle outlet diameter changes and as the pressure upstream of the nozzle varies, and the flow curve again as the varying nozzle outlet diameter and as the pressure upstream of the nozzle varies. The approach to follow with the use of the two graphs is to first evaluate of the feasible nozzle outlet dimensions and then following a choice of the pressure upstream of the nozzle. The realizable thrust in that configuration is inferred as well as the flow rate in the tubes.

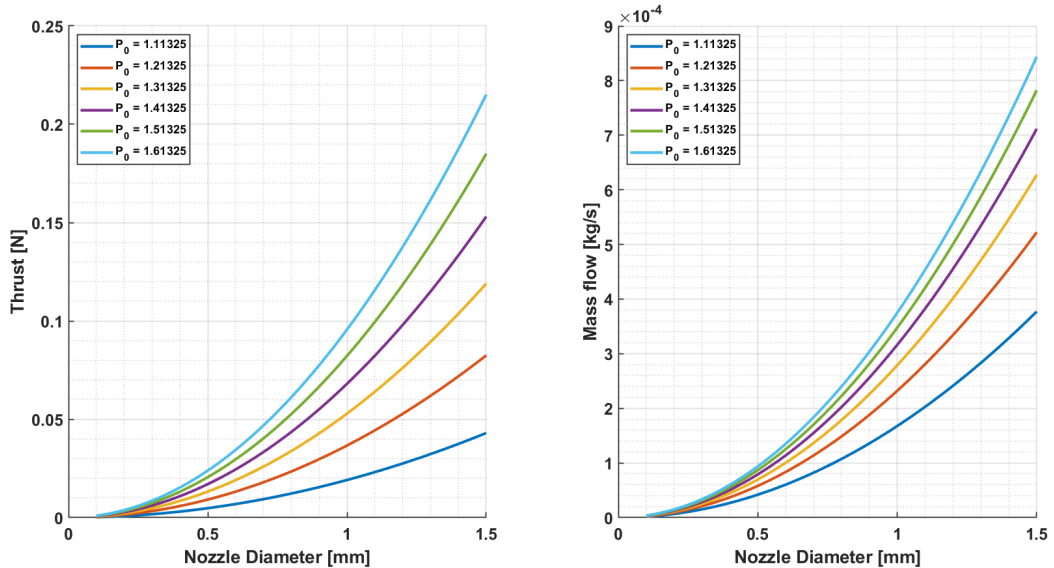


Figure 3.1: Thrust and Mass Flow respect to the nozzle diameter and pressure inlet

Assuming the isentropic transformation at the nozzle, the value of Mach at the throat of the nozzle as the pressure of the regulator changes is derived. These values are assumed to be the corresponding at the outlet of the line.

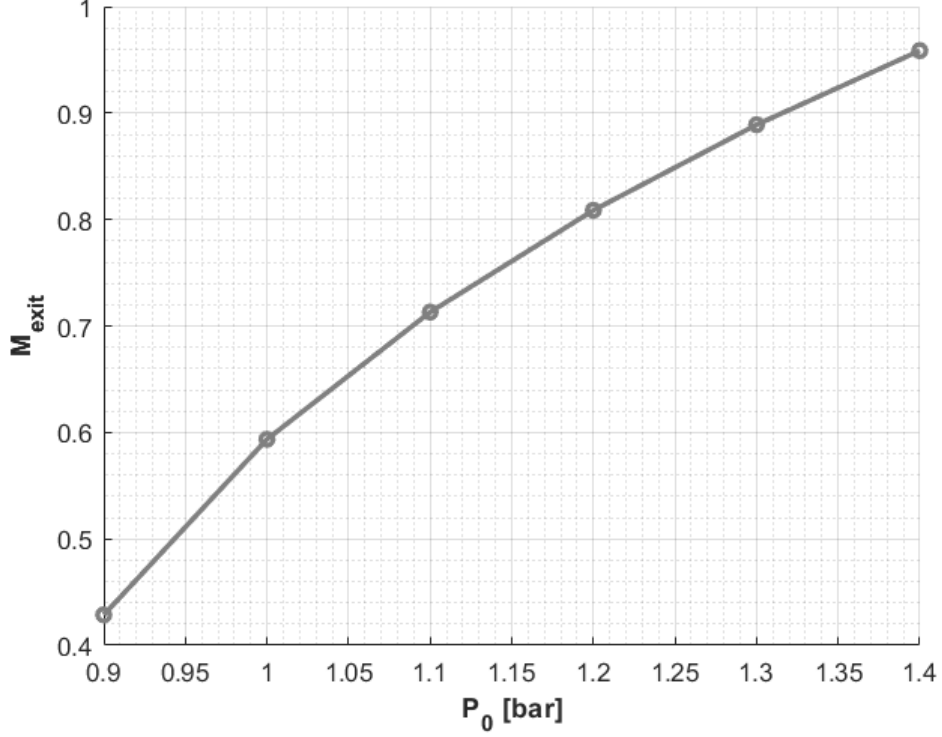


Figure 3.2: Mach variation with respect to the pressure regulator set

The line are studied assuming Fanno flow, also considering the friction effect along the tubes and leaks in corispondence of the connections.

The forcing of Fanno is defined as:

$$4\frac{fL}{d} = \left(\frac{1 - M^2}{\gamma M^2}\right) + \left(\frac{\gamma + 1}{2\gamma}\right) \ln\left[\frac{M^2}{\left(\frac{2}{\gamma+1}\right)\left(1 + \frac{\gamma-1}{2}M^2\right)}\right] \quad (3.11)$$

the friction parameter f is assumed equal to 0.005, the line tubes are considered of 4×1 mm and also the length of the tubes in relation to the 2U mock-up structure, are set in two sections, the first one from the cartridge to the faucet equal to 0.25 mm, and the second one from the faucet to the nozzle equal to 0.40 mm, The pressure ratio for the Fanno flow is defined as:

$$\frac{p}{p^*} = \frac{1}{M} \frac{1}{\sqrt{\left(\frac{2}{\gamma+1}\right)\left(1 + \frac{\gamma-1}{2}M^2\right)}} \quad (3.12)$$

The entire propulsive system is simulated and the data are collected in Table 3.3.

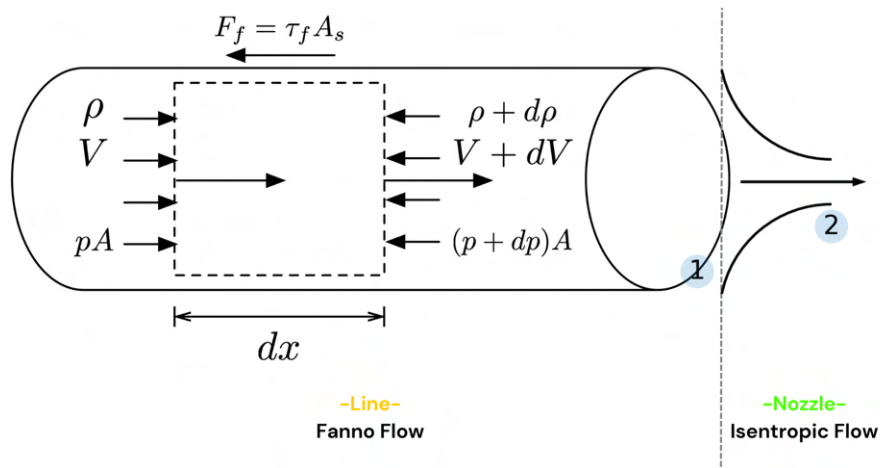


Figure 3.3: Analysis of the line with Fanno flow and isentropic motions with cross-sectional variation

Pressure Inlet p_0 (bar)	Mach Exhaust M_e	Mass Flow \dot{m} (g/s)	Thrust S (N)
1.10	0.38	0.15	0.015
1.19	0.52	0.21	0.028
1.28	0.63	0.25	0.040
1.38	0.71	0.29	0.052
1.47	0.78	0.32	0.063
1.56	0.85	0.35	0.073
1.65	0.90	0.37	0.083
1.74	0.95	0.40	0.093
1.83	1.00	0.42	0.102

Table 3.3: Simulation of the complete propulsive system with a nozzle diameter of 1 mm

With a nozzle diameter of 1 mm it is possible to have a thrust value around 0.1 N with a pressure regulator setup around 2–2.5 bar. This level of pressure allows to have a low-pressure system that generates a great amount of thrust capable of performing quickly the manoeuvre. It is important to underline that, since the system works at very low pressure and with a compressible gas, the back pressure is very low, in fact from the analysis the working pressure set by the regulator decreases of less than 5% at the nozzle. The evaluation has been performed by comparing data of thrust with two systems, which differ for the number of obstructions and flux deviations.

The diameter of the nozzle cited is the corresponding one to the throat of the nozzle. An important consideration regarding the geometry of the nozzle and the dynamic of the exhaust flow has to be conducted. In particular, the nozzles are selected to be simple convergent instead of a classic convergent-divergent configuration due to the necessity of avoiding supersonic flows since the experiment takes place at standard atmospheric pressure and not in a vacuum chamber. In fact, supersonic flows lead to shock waves, that could cause unwanted increase in pressure in the pneumatic system and, consequently, damages to the system or simply alter its performance.

Considering the converging nozzle, the distribution of the pressure ratio p/p_0 along its length is shown in Figure 3.4. The pressure p is the static pressure. The inlet conditions of the gas are at the stagnation state (p_0, T_0) which are constants. The pressure at the exit plane of the nozzle is denoted by p_E and the back pressure is p_B which can be varied.

- i The pressure p_0 is throughout, i.e. $p_0 = p_E = p_B$. There will be no flow through the nozzle.
- ii Decreasing gradually p_B , the flow rate will increase. The pressure will decrease in the direction of the flow. The pressure p_E at the exit plane of the nozzle shall remain equal to p_B as long as the maximum discharge condition is not reached. The flow rate is directly proportional to mass flow rate, so as the flow rate increases mass flow rate will also increase.
- iii In this case the pressure distribution is in the maximum discharge situation. The flow rate has attained its maximum value, i.e. when Mach, $M = 1$ is achieved and the nozzle is said to be choked. The pressure p_E is equal to p^* (pressure for Mach 1 flow). Since the nozzle does not have a diverging section, further reduction in p_B will not accelerate the flow to supersonic condition. As a result, p_E will continue to remain at p^* even though p_B is lowered further.
- iv Since p_B is less than p^* , the flow leaving the nozzle has to expand to match the lower back pressure. This expansion is three dimensional and the pressure distribution cannot be predicted by one dimensional theory.

For the mock-up, the thrusters are designed for adapted nozzle, which means maximum dynamic thrust, where the pressure at the exhaust equals the ambient pressure, $p_e = p_a$. The choice of a convergent solution allows to

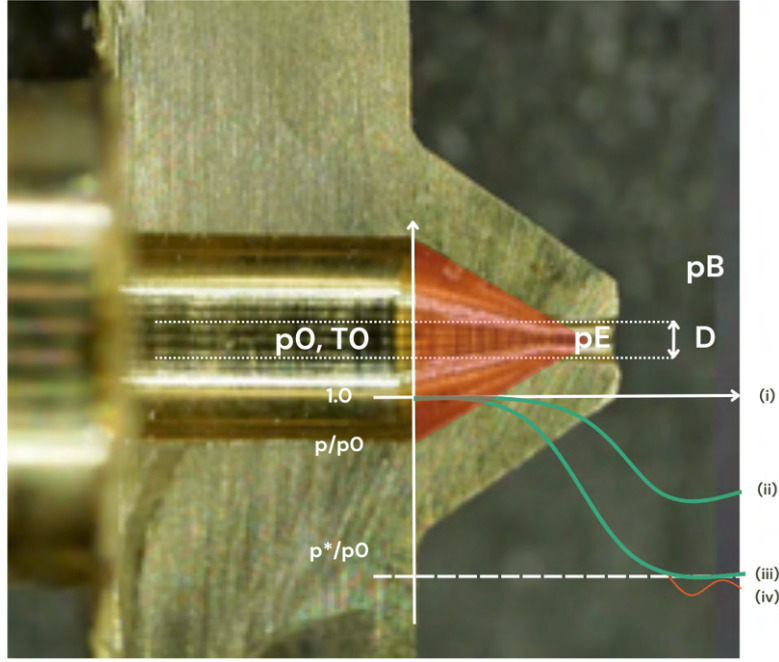


Figure 3.4: Single convergent nozzle

model the thrust output linearly with the respect to the pressure set on the regulator. In fact, for pressure higher than 1.8 bar, the flow exits at a sonic state, Mach number equal to 1 (condition *iii*). Therefore, the exit pressure is a function of only the total pressure set on the regulator:

$$p_{sonic} = p_0 \cdot \left(\frac{2}{\gamma + 1}\right)^{\frac{\gamma}{\gamma - 1}} \quad (3.13)$$

Consequently, the thrust T output is linear with respect to the pressure set too. In fact starting from Equation (6), using the expression of Equation (4) and knowing that the sonic exhaust velocity is:

$$v_e = \sqrt{\gamma RT_0} \Gamma(\gamma) \quad (3.14)$$

$$\begin{aligned} T &= \frac{p^* A_e}{\sqrt{RT^*}} \sqrt{\gamma} \sqrt{RT^*} \sqrt{\gamma} + p_0 \left(\Gamma(\gamma) - \frac{p_a}{p_0} \right) A_e \\ &= [p_0 \Gamma(\gamma)^2 + p_0 \left(\Gamma(\gamma) - \frac{p_a}{p_0} \right)] A_e \\ &= p_0 A_e \left[\Gamma(\gamma) - \frac{p_a}{p_0} \right] \\ \Rightarrow T &= (p_0 \cdot \Gamma(\gamma) - p_{atm}) \cdot A_e \end{aligned} \quad (3.15)$$

The thrusters are divided into two groups of four thrusters each that are positioned in two opposite faces; these groups are further divided into couples of thrusters pointing towards the same point (Figure 3.5). This tilted configuration allows control over both its attitude and position (6 DoF). The thrusters are tilted of $\pm 45^\circ$ with the respect to z axis, $\pm 30^\circ$ with the respect to y axis and $\pm 60^\circ$ with the respect to x axis. To move or rotate along a single axis four thrusters must be actuated together as shown in Table 3.4.

	Translation		Rotation	
	+	-	+	-
x	2367	1458	1278	3456
y	1256	3478	2358	1467
z	1234	5678	2468	1357

Table 3.4: Thrusters to be actuated, referring Figure 30, to control the different Degrees of Freedom

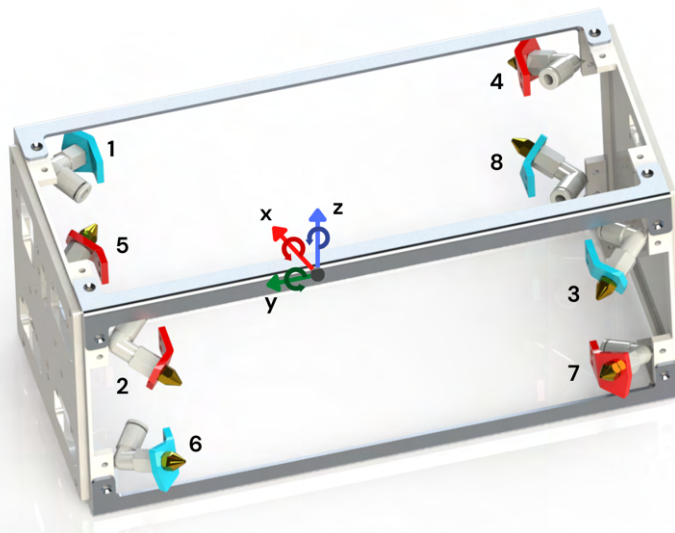


Figure 3.5: Thruster Configuration

3.2 Dynamic Simulation

The study of the system dynamics is conducted in order to evaluate the velocity and trajectory followed by the Chaser mock-up and also to give a graphical representation of what the design of the nozzle configuration entail during the manoeuvre. In particular considering the angles of the configuration, $\alpha[x]$, $\beta[y]$, $\gamma[z]$, and knowing that $\alpha = 60^\circ = \pi/3$, $\beta = 30^\circ = \pi/6$, $\gamma = 45^\circ = \pi/4$ the vector of thrust is defined as:

$$S_{1-8} = \begin{Bmatrix} \pm \cos \alpha \sin \gamma \\ \pm \sin \alpha \sin \gamma \\ \pm \cos \gamma \end{Bmatrix} \cdot |S| \quad (3.16)$$

The components of thrust are $S_{x,y,z}$, and the sign of it is the direction $(\pm\alpha, \pm\gamma)$.

$$S_1 = \begin{Bmatrix} -S_x \\ -S_y \\ S_z \end{Bmatrix} \quad S_2 = \begin{Bmatrix} S_x \\ -S_y \\ S_z \end{Bmatrix} \quad S_3 = \begin{Bmatrix} S_x \\ S_y \\ S_z \end{Bmatrix} \quad S_4 = \begin{Bmatrix} -S_x \\ S_y \\ S_z \end{Bmatrix} \quad (3.17)$$

$$S_5 = \begin{Bmatrix} -S_x \\ -S_y \\ -S_z \end{Bmatrix} \quad S_6 = \begin{Bmatrix} S_x \\ -S_y \\ -S_z \end{Bmatrix} \quad S_7 = \begin{Bmatrix} S_x \\ S_y \\ -S_z \end{Bmatrix} \quad S_8 = \begin{Bmatrix} -S_x \\ S_y \\ -S_z \end{Bmatrix} \quad (3.18)$$

Proceeding with the moments, the arms are defined for the components of the thrust as:

$$r_1 = \begin{Bmatrix} L/2 \\ L \\ L/2 \end{Bmatrix} \quad r_2 = \begin{Bmatrix} -L/2 \\ L \\ L/2 \end{Bmatrix} \quad r_3 = \begin{Bmatrix} -L/2 \\ -L/2 \\ L/2 \end{Bmatrix} \quad r_4 = \begin{Bmatrix} L/2 \\ -L \\ L/2 \end{Bmatrix} \quad (3.19)$$

$$r_5 = \begin{Bmatrix} L/2 \\ L \\ -L/2 \end{Bmatrix} \quad r_6 = \begin{Bmatrix} -L/2 \\ L \\ -L/2 \end{Bmatrix} \quad r_7 = \begin{Bmatrix} -L/2 \\ -L \\ -L/2 \end{Bmatrix} \quad r_8 = \begin{Bmatrix} L/2 \\ -L \\ -L/2 \end{Bmatrix} \quad (3.20)$$

The time fraction $t = \tau/T$ is introduced, where τ is the timing of actuators activated, while T is the total time of the manoeuvre.

The equations to solve are:

- **Translations**

$$\begin{cases} T_x = +S_x(\tau_1 + \tau_4 + \tau_5 + \tau_8) - S_x(\tau_2 + \tau_3 + \tau_6 + \tau_7) \\ T_y = +S_y(\tau_1 + \tau_2 + \tau_5 + \tau_6) - S_y(\tau_3 + \tau_4 + \tau_7 + \tau_8) \\ T_z = +S_z(\tau_5 + \tau_6 + \tau_7 + \tau_8) - S_z(\tau_1 + \tau_2 + \tau_3 + \tau_4) \end{cases} \quad (3.21)$$

Note that the activation of four thrusters along x^+ have as consequence that $\tau_{y,z} = 0$, and the same for the other quartet.

- *Rotations*

$$\begin{cases} M_x = (S_z L + S_y L/2)(\tau_3 + \tau_4 + \tau_5 + \tau_6 - \tau_1 - \tau_2 - \tau_7 - \tau_8) \\ M_y = (S_x L/2 + S_z L/2)(\tau_1 + \tau_4 + \tau_6 + \tau_7 - \tau_2 - \tau_3 - \tau_5 - \tau_8) \\ M_z = (S_y L/2 - S_x L)(\tau_1 + \tau_3 + \tau_5 + \tau_7 - \tau_2 - \tau_4 - \tau_6 - \tau_8) \end{cases} \quad (3.22)$$

By implementing the system of equations it is possible to predict the path of the manoeuvre with the activation of a specific nozzle or group of nozzle. In particular, the attitude and the velocity in every time-step of the simulation are computed and also the trajectory followed by the center of mass of the mock-up with respect to a fixed reference system is derived. A simulation of a worst-case scenario in which only one electrovalve is open until the cartridge is empty has been carried out. This is an interesting case analysed for safety reason, in particular because with the failure of 7 nozzles out of 8 the trajectory needs particular attention in order to check if there is the risk of impact of the uncontrolled mock-up with the nets of the experiment area, causing eventually their breaking. Considering the mass of the mock-up, its inertia and also the timing for the simulation of 22 s, giving as input the command of only one thrust open, the graphs collected in the following Figure are given as output of the analysis.

It is important to underline that the time of the simulation is intended to be one parabola although the cartridges theoretically would last more than that. The reasons of this choice are, firstly because the thrust is not enough to beat the gravitational force therefore after the 0 g phase it is impossible for the Chaser to move both in hyper and normal gravity; secondly because when the Chaser hits the nets it loses a lot of energy and it is difficult to predict and simulate, therefore the interest lies more on evaluating the energy of the impact during the manoeuvre.

The simulation shows that the Chaser initially rotates a lot but then a jet-dumping effect occurs and stabilizes the trajectory, as shown in Figure 3.6. This new trajectory is based on an acceleration towards the direction of the angular momentum and a nutation around a tilted axis with respect to the angular momentum, depending on which thruster is open (see Figure 3.8). The stabilization phase needs more than half the time of simulation, then it is just quasi-linear acceleration. The energy of impact remains low at the beginning, then it increases nearly linearly (see Figure 3.7).

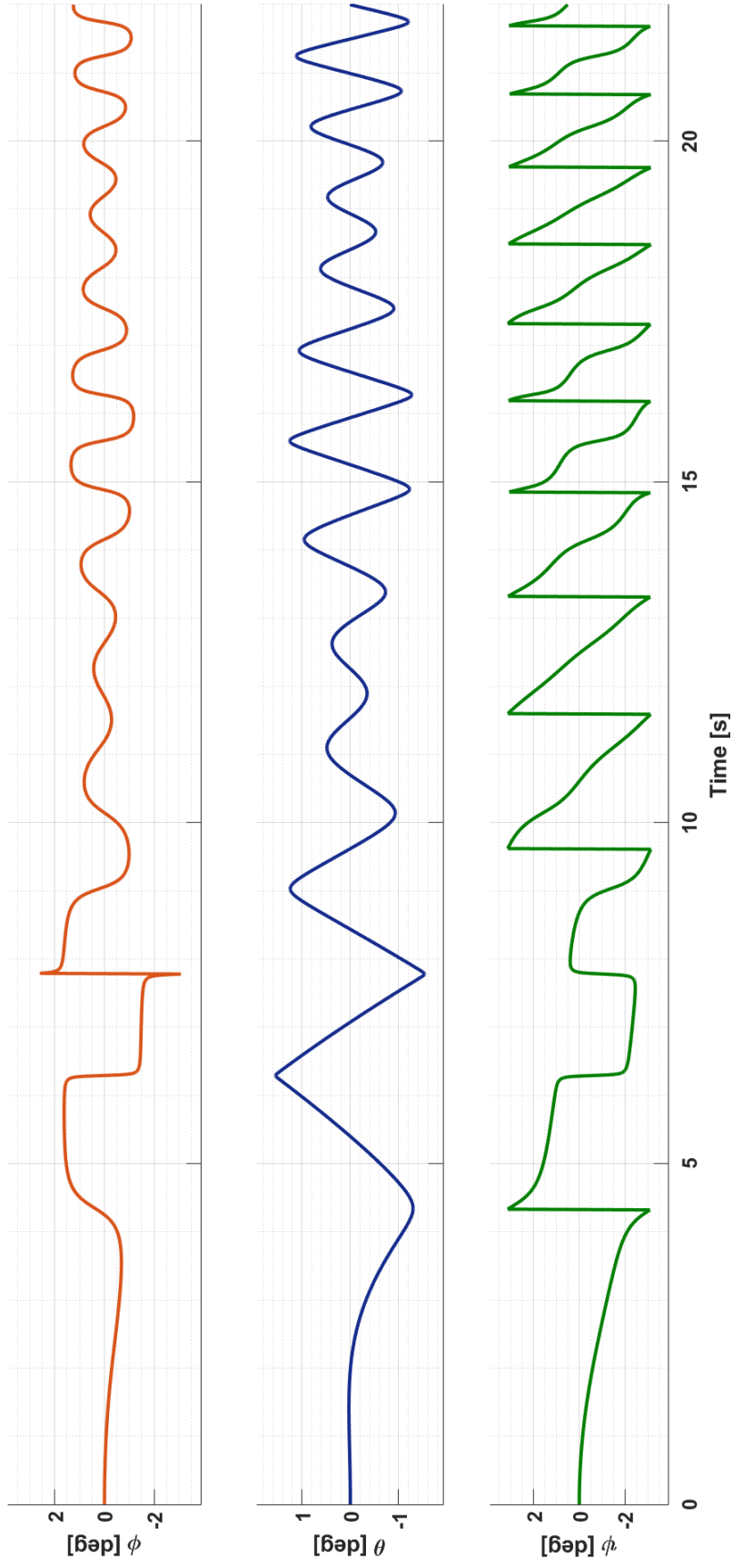


Figure 3.6: Attitude of the Chaser mock-up during the worst-case configuration with a generic nozzle active

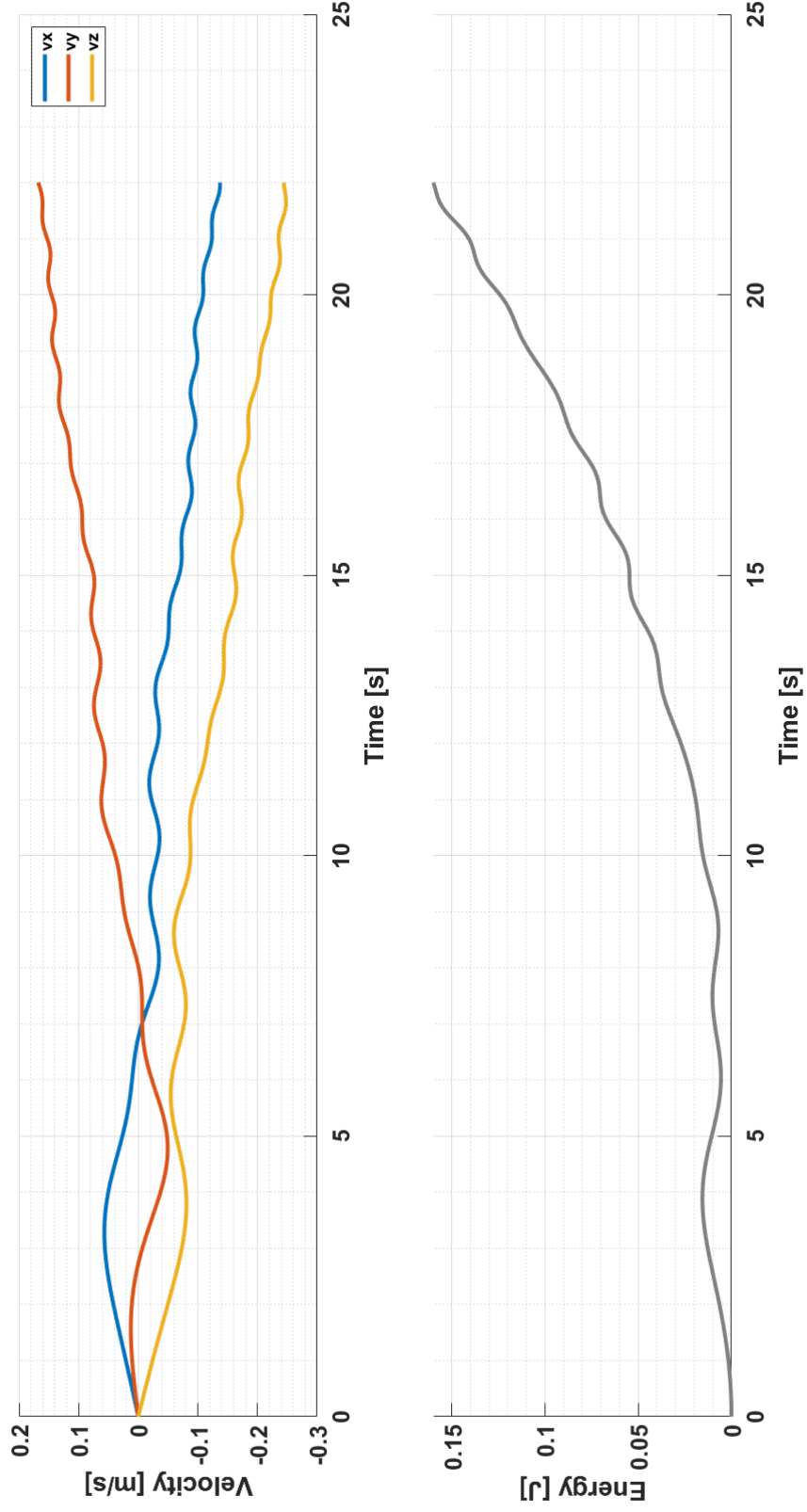


Figure 3.7: Velocity and Energy of the Chaser mock-up during the worst-case configuration with a generic nozzle active

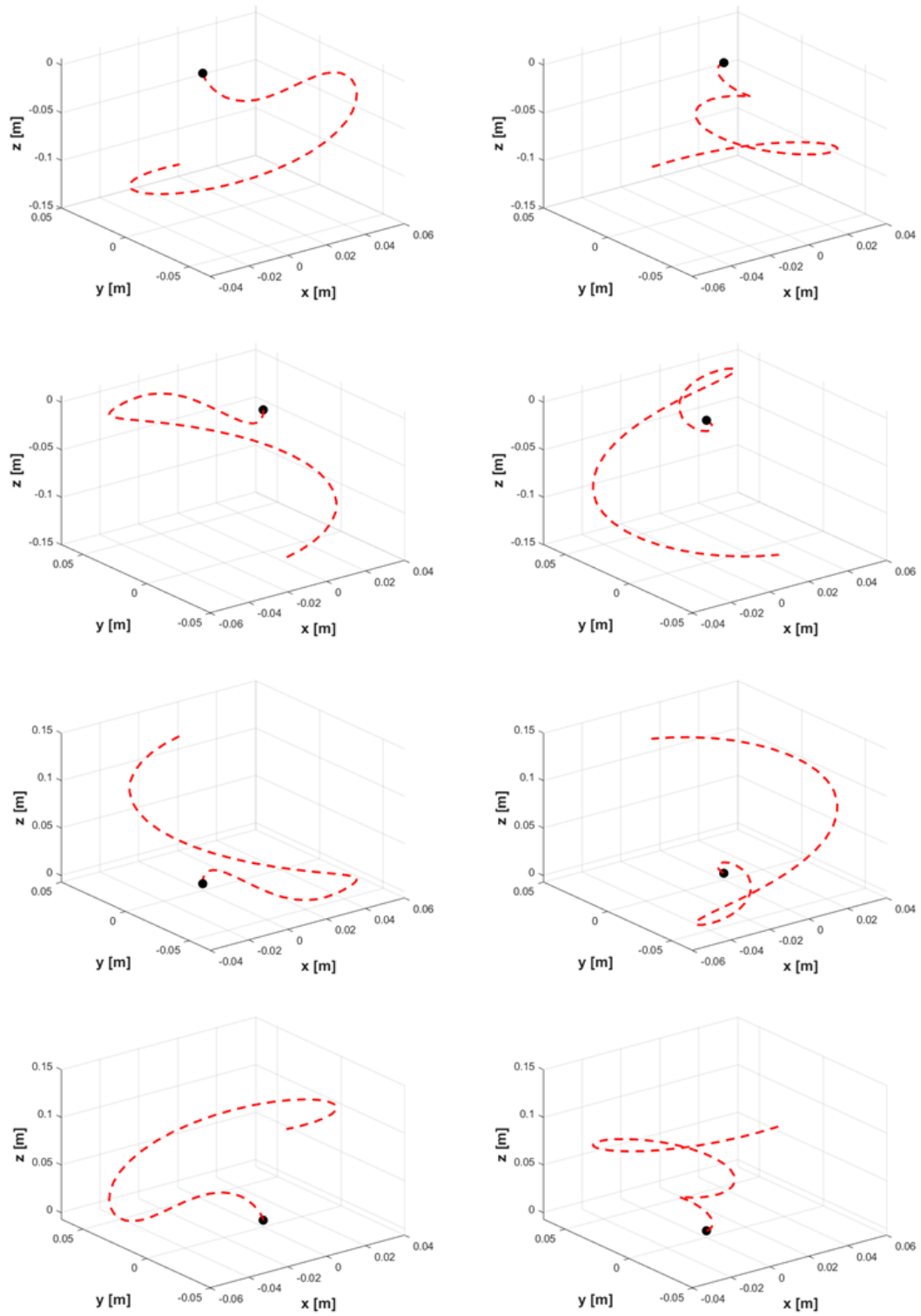


Figure 3.8: Paths of the manoeuvre with only one nozzle active

3.3 Safety Management

The complete propulsive system has been mounted and tested in order to check its reliability and safety.

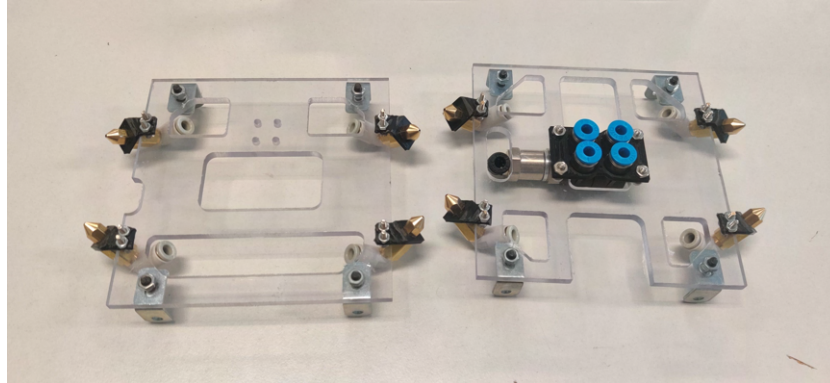


Figure 3.9: Nozzle setup on the mock-up structure

In particular the interest lied on testing the nozzle to assure that they could stand a 10 bar flow following the Novespace Guidelines regarding components in pressurized systems that are not properly documented. In fact, since they are nozzles for 3D printers, they are not guaranteed to stand high pressure flows. Secondly on testing that the whole system was checked to guarantee that all the connections were properly sealed. Therefore, a test involving emptying a cartridge with small bursts continuously from each nozzle at a time has been carried out. Finally, on testing that the whole system could stand a 7 bar pressurized flow following the Novespace Guidelines regarding pressurized systems. In fact, 7 bar is the maximum allowable pressure for the electrovalves, therefore the maximum pressure at which the electrovalves works correctly as a gate to avoid unwanted bursts.



Hazard cause	A component, between the pressure reducer and the nozzle explodes due to improper design	
Hazard Control	All components can sustain the Maximum Design Pressure	
Verification E1	Components, without any manufacturer specification, have passed a stress test: Nozzle - tested at 10 bar	closed
Verification E2	Check from datasheets that all commercial components sustain at least the Maximum Design Pressure: <ul style="list-style-type: none"> System - tested at 7 bar 	closed
 		

Figure 3.10: Extract of the Experimental Safety Data Package - Hazard Risk Explosion of Pressurized Systems

In order to check the integrity of the system at high pressure, single nozzle [*E1*] and complete pneumatic system [*E2*], two stress tests are executed. The objectives are:

- [*E1*] To test the nozzles at high pressure, 10 bar, to assure no damage to the nozzle or hazards for the experimenters could occur.
- [*E2*] To test the entire pneumatic system at high pressure, 7 bar, to ensure there are no leaks and that no damage to the electrovalves or hazards for the experimenters could occur.

The components used for the test are:

- Nozzle: 3D printer with orifice diameter of 1 mm.
- Tubing: PVC tube 4x1 mm for the last section of the line before the nozzle, and PVC 10x2 tube for the starting section of the line from the air compressor.
- Adapters/connectors with pressure sealing: 10 mm to *G1/2"*, *G1/2"* to *G1/8"* and *G1/8"* to 4 mm.
- Valve: manual valve T-type, 3 ways.
- Pressure regulator: Numatics pressure regulator 34203341, during *E1* it works just like a valve, while during *E2* it is needed to lower the pressure.
- Air compressor: Nardi Compressori ESPRIT SILENZIATO 2/60
- Clamp.
- Laboratory bench power supply.

The set input data for the two tests are:

TEST	Pressure (bar)	Expansion Ratio (-)	Mass Flow (g/s)
<i>E1</i>	10	16	1.3
<i>E2</i>	7	16	0.9

Table 3.5: Input data of the experiments

Starting with the single nozzle test, in order to have a large repeatability it is used the air compressor rather than cartridges with pressure regulator. The tested line can be represented by the scheme of Figure 3.11.

The line starts from the compressor and through the dedicated adapters is connected to the nozzle. Along the line a manual valve is set to manage the test. Then the compressor working pressure is set at 10 bar. From the previous propulsive analysis, the thrust at this pressure condition is estimated around 0.67 N. Then the valve is manually opened for 12 seconds. This timing has been chosen in order to be nearly the full consume of the 16g cartridge

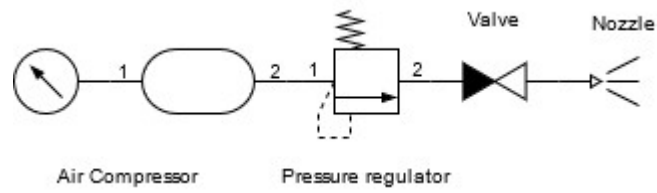


Figure 3.11: Pneumatic system of the test *E1*

at 10 bar. In fact from the set input data of Table 3.5, considering case *E1*, $16g/1.3g/s \sim 12s$ is the computed emptying cartridge timing. After the test, there are no visible damages to the nozzle as shown in Figure 3.13, and repeating the experiment does not damage the nozzle either.



Figure 3.12: *E1* - Experiment set-up

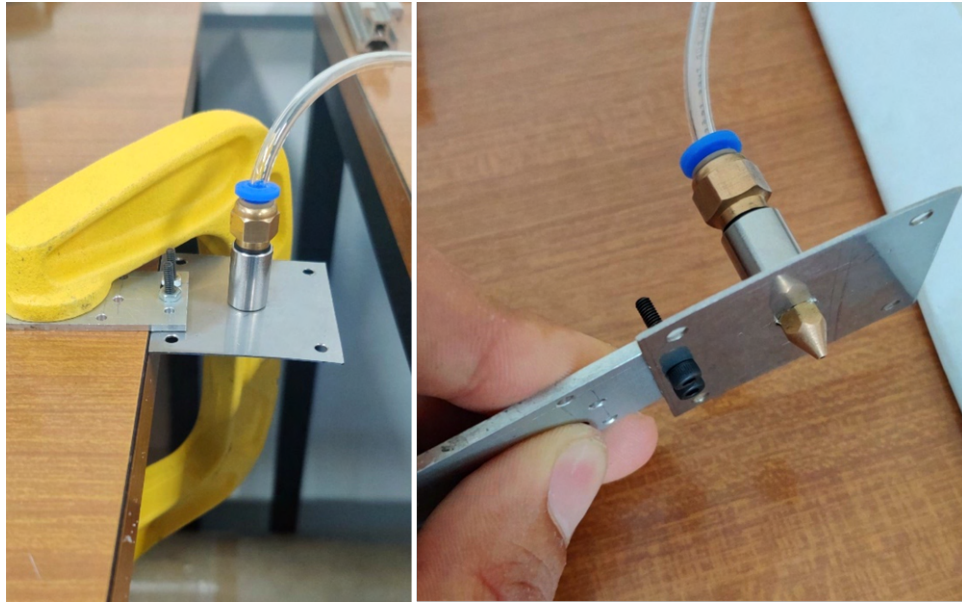


Figure 3.13: *E1* - Nozzle tested

For the second test regarding the complete pneumatic system it is used the complete designed system, mounted on the mock-up (Figure 3.15). The tested line can be represented by the following scheme:

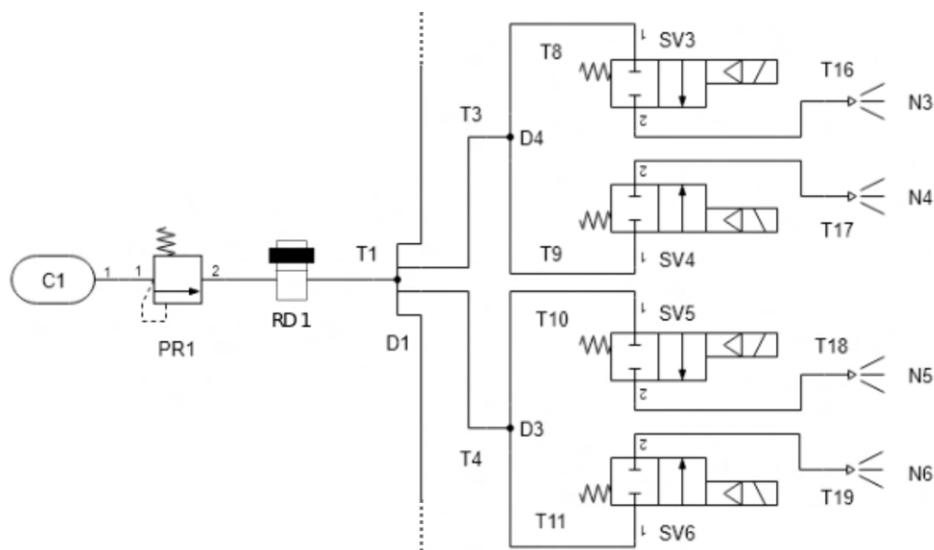


Figure 3.14: Pneumatic system of the test *E2*

The working pressure at 7 bar is set on the pressure regulator of the cartridge, then the valve is manually opened. The pneumatic system does not have leaks that could compromise the effectiveness of it. During the static part of the experiment the log of the regulator showed a fixed value (valves not opened). After the bursts, there are no visible damages to the nozzle/connections/electrovalves. Repeating the experiment does not damage the system either.

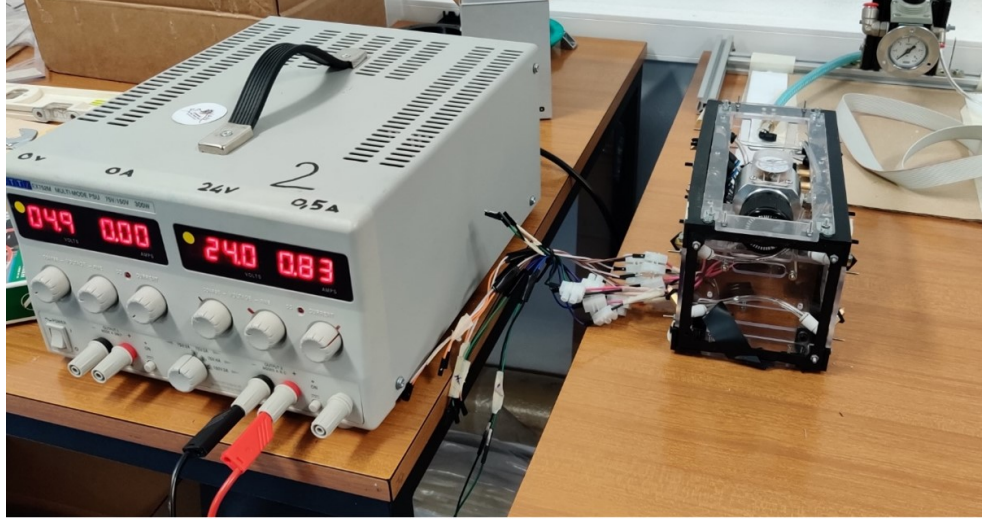


Figure 3.15: *E2* - Experiment set-up

3.4 Test Integration

3.4.1 PWM control of electrovalves

The electrovalves are controlled with a 30 Hz Pulse Width Modulation (PWM) with 16 steps, that determines the duty cycle (DC) of the valve. It is important to highlight that the 16 steps are not related to a standard 4 bit PWM, but a fictitious 8 bit PWM with only 16 possible values instead of 256. The number of steps can be obtained by elevating 2 at the number of bit. The objective of the dedicated tests is to find the relation between the thrust and the PWM control.

The Pulse-width modulation is a method to reduce the average power delivered by an electrical signal. Fixing a frequency f , a periodic signal with a period of T seconds, and the possible values for the signal as $[0, 1]$. In a period T the signal is defined as values 1, for $T \cdot D$ seconds, and 0, for $T \cdot (1 - D)$ seconds, where $0 \leq D \leq 1$. D is called duty cycle, defined as:

$$D = \frac{t_{ON}}{T} \quad (3.23)$$

with t_{ON} opening valves time that is between 0 and $T_{sample} = T$. Varying D results in a variation of the average power of the signal. The PWM signal is used to vary the output of the EV, maintaining the linear relation:

$$F_{thrust} = F_{max} \cdot D \quad (3.24)$$

Where F_{max} is the maximum thrust of a nozzle with the electrovalve completely open. In the implementation the duty cycle needs to be discretised, so the PWM generator, that the Arduino provides in some ports, was used to obtain this result.

The tests of the PWM control were based on a set of laboratory experiments. The pneumatic circuit consisted in a nozzle, an electrovalve, and an

air compressor with its pressure regulator, as shown in Figure 3.16. The pressure regulator guaranteed a constant pressure in the pneumatic system. Thrust data were acquired with a load cell connected to the support plate of the nozzle.

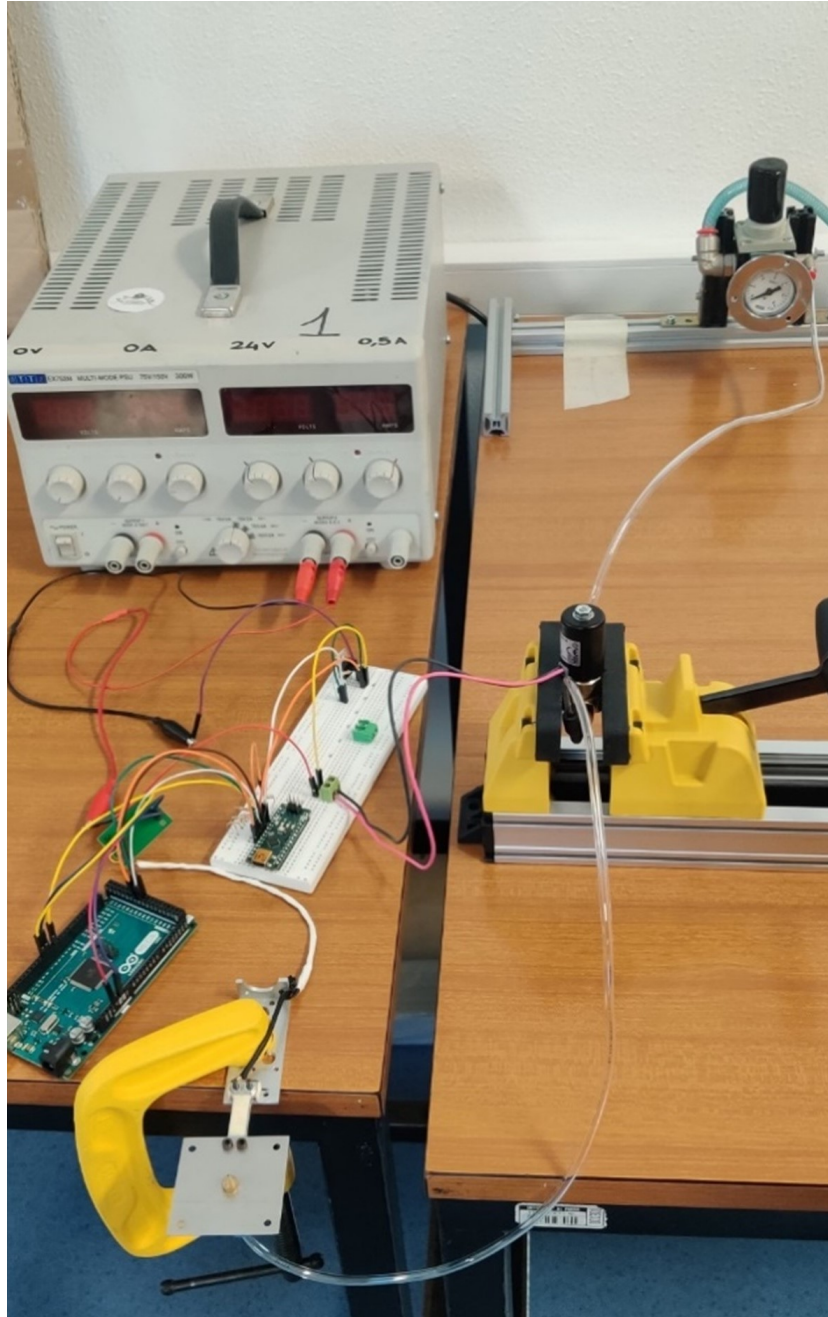


Figure 3.16: Test PWM set-up

Different frequencies have been tested: 30.64 Hz, 122.55 Hz. According to the [21] document those frequencies can be obtained with the use of PWM registers, so it will not interfere with the LLS cycle. Higher frequencies than 122.55 Hz do not work with the EVs chosen. The D value is represented by the discrete values from 0 to 255. This last one is associated with 1, when

the signal is always in its higher state. The results of thrust registered by the load cell are not linear. As consequence of this the Equation 3.24 is not respected, so the output needs to be remapped to have a result similar to the Equation 3.24. In particular, the data acquisition dealt with the 256 steps (8 bit) PWM so that the real trend could be plotted. As expected, the real trend is a sigmoid with the initial values around zero because for low value of duty cycle the electrovalve does not have the time needed to react to the signal and consequently completely open.

Then, the number of steps available has been restricted in a way that makes the trend linear with the respect to the duty cycle values. This choice greatly simplifies the control of the actuators. The simplification lies in the shift from a real sigmoid trend to an ideal linear trend of the thrust. In fact in the Chaser 16 different levels of thrust are sufficient to have a fairly fine division of the maximum thrust. In particular values such that: 0 is 0, 15 is the maximum (in this case 0.135 N), and the remaining once are assigned so that the response is proportional. Finally, this new fictitious 16 steps PWM has been tested by sending random step-like commands of duty cycle to the electrovalves. In conclusion the chosen frequency is 30.64 Hz, because the response seems more stable.

In general, given N bit of PWM precision, a PWM step is:

$$F_{step} = \frac{F_{max}}{2^N - 1} \quad (3.25)$$

and the general force of the PWM value p , $0 \leq p \leq 2^N - 1$:

$$F_{PWM}(p) = F_{step} \cdot p \quad (3.26)$$

Starting from a PWM accuracy in bits greater than N , called \hat{N} in the non linear PWM response \hat{f} , for each p from 0 to $2^N - 1$ it is necessary to find the values of \hat{p} such that:

$$\hat{f}(\hat{p}) = F_{step} \cdot p \quad (3.27)$$

The pair p, \hat{p} represents the map for a linear response with respect to p . In the implementation, an array of 16 elements contains the corresponding \hat{p} -values. The index of the array is p and it is the access key to the map. The final run of the experiment, with 16 different levels, outputs the graphs shown in the following figures. The values of \hat{p} are represented by the dotted lines.

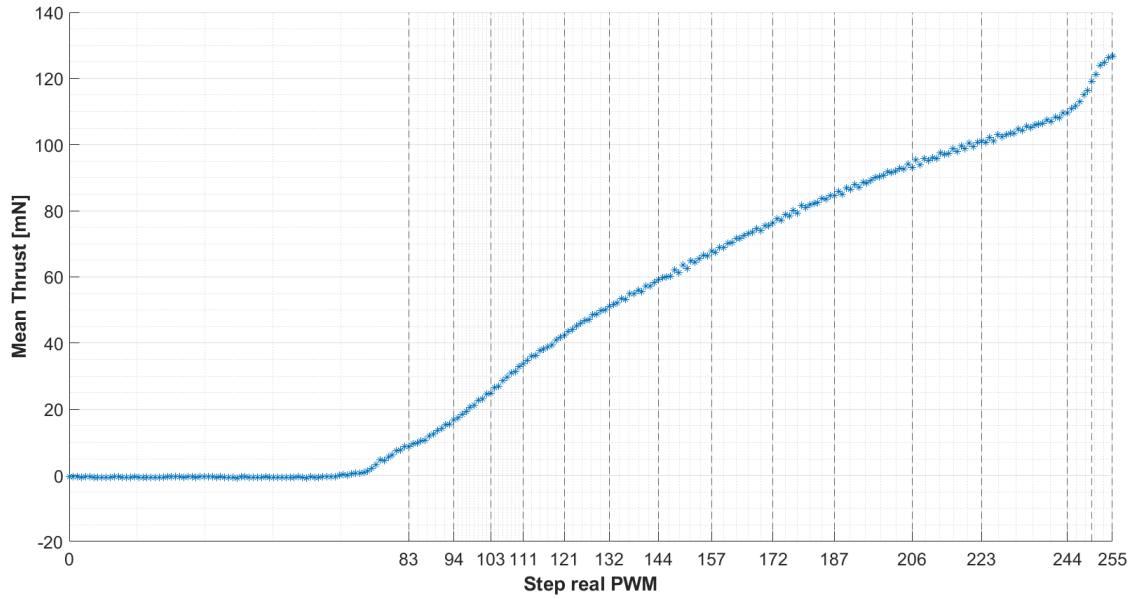


Figure 3.17: Response in thrust of one nozzle if controlled with a 30Hz PWM. The dotted lines represent the values of the PWM to map to the 16-levels values

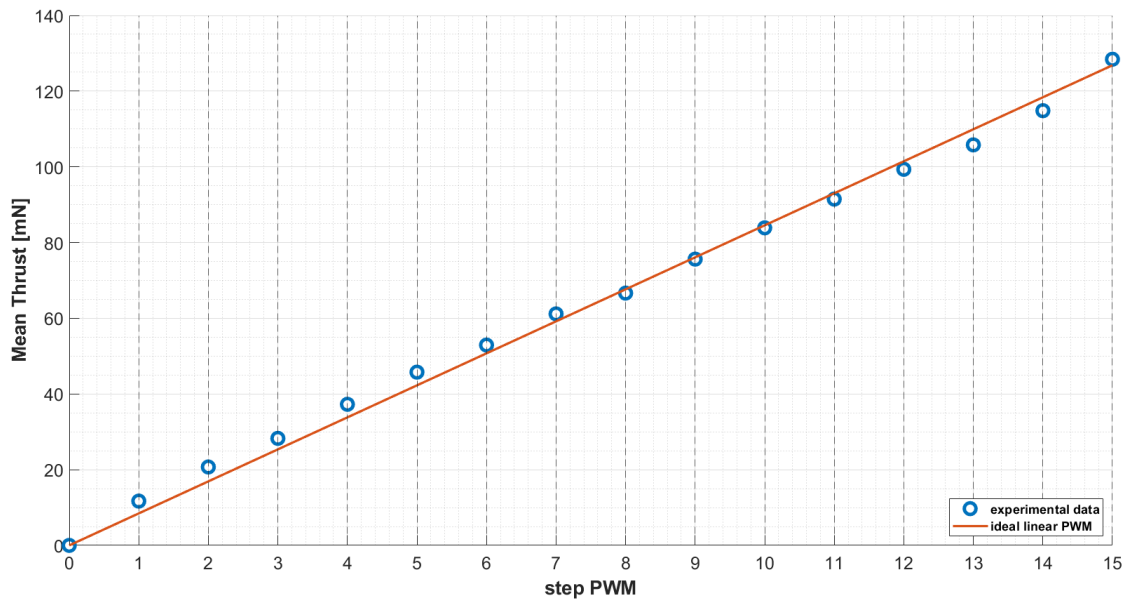


Figure 3.18: Relation between the thrust of the nozzle and the PWM with the remapped values: step number 15 is $D = 1$

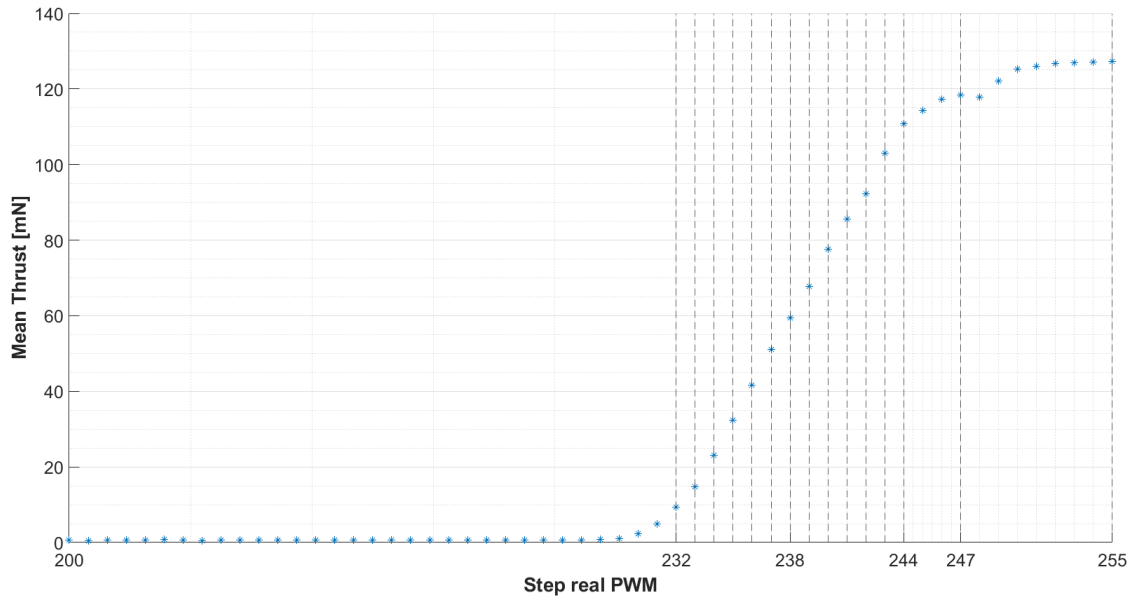


Figure 3.19: Response in thrust of one nozzle if controlled with a 122Hz PWM. The dotted lines represent the values of the PWM to map to the 16-levels values. The points between 0 and 200 are omitted because the mean thrust for them is 0

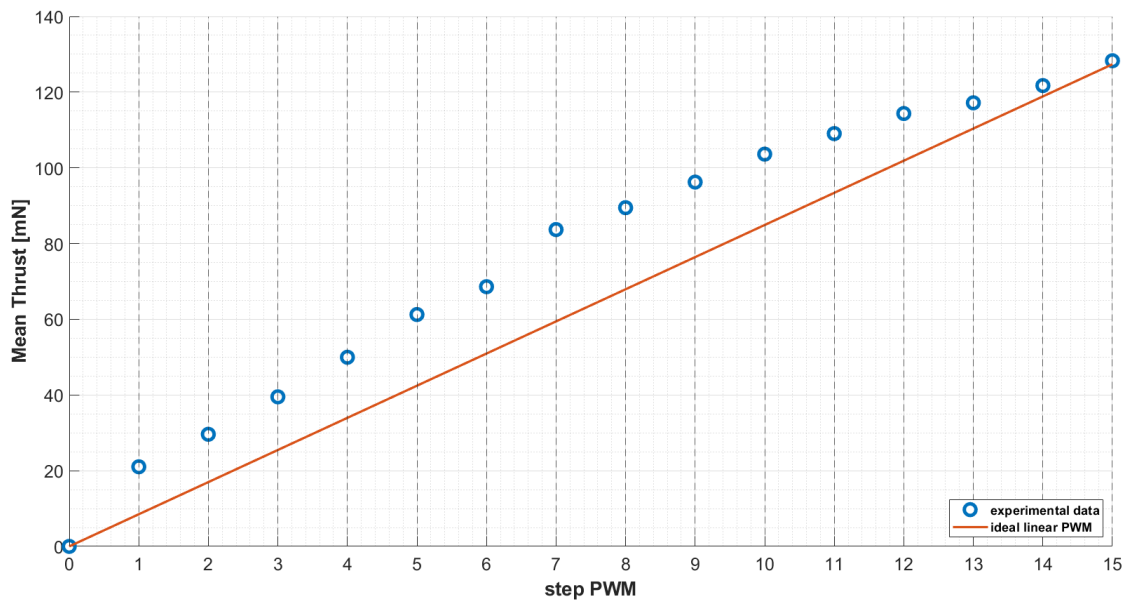


Figure 3.20: Relation between the thrust of the nozzle and the PWM with the remapped values: step number 15 is $D = 1$

3.4.2 Authority of the thrusters

To verify the functions of the experiment system, functional testing are conducted on a low friction table. In these tests the Chaser and the Target have been placed over a sled with IBS flat round air bearings to levitate (Figure 3.21). In particular the simulations focused not only on docking manoeuvre but also on performing basic command-based manoeuvring in order to evaluate the precision as well as the capabilities of the GNC and for the Chaser. The tests have been carried out implementing a Motion Capture technology, in particular the Optitrack technology [22] has been used, that helped tracking the Chaser so that the predicted manoeuvre with the actual one could be compared.

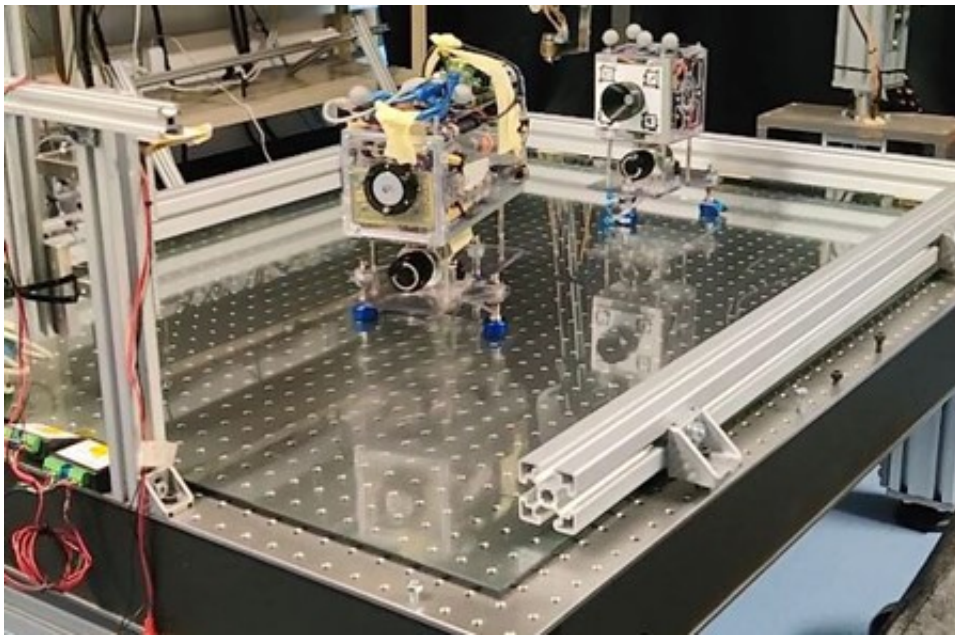


Figure 3.21: Test Chaser and Target on the frictionless table

The Chaser has been placed in front of a fixed Target and it tries to reach it by computing a trajectory to follow. The main focus is to evaluate the precision of the control. In particular the interest lies in the trajectory followed by the Chaser, based on what is calculated by the software and the type of command sent. Although the trajectory computed is correct, the commands sent to the Low-Level can be too strong, too inefficient or simply wrong. To evaluate the commands, the difference between the real corrections of attitude or position and the one computed by the Chaser are compared. Then some parameters that are relevant for the control are changed and also the test is repeated. The following parameters give a clear idea regarding the translation of the software language to propulsive and mechanical setting of the experimental layout:

- **Authority for trajectory corrections:** this macro parameter is further divided into three, each referring to a specific axis, x , y and z . These three parameters refer to how much strong a certain correction along one axis should be. It can be seen as 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, that is indeed different due to the particular thrusters configuration, but they have also been tuned to get desired results. In particular the x axis has the biggest authority usually because it has to cover, eventually, a larger space, while theoretically the y axis and z axis require small and precise corrections.

- **Weight of trajectory corrections:** this macro parameter is further divided into three, each referring to a specific axis, x , y and z . These three parameters refer to how much important is considered correcting one axis with the respect to the others possible corrections, including attitude. Therefore, these parameters give the likelihood of correcting a certain axis given the same percentual misalignment with the respect to the perfect alignment. For example, the x axis has the biggest weight because otherwise the Chaser would have not initiated the approaching, rather it would have been stuck into correcting just misalignment of attitude or trajectory along y and z axis. Moreover, to y and z axis the same weight is given due to the symmetry of the approach, where there are no preferred directions.

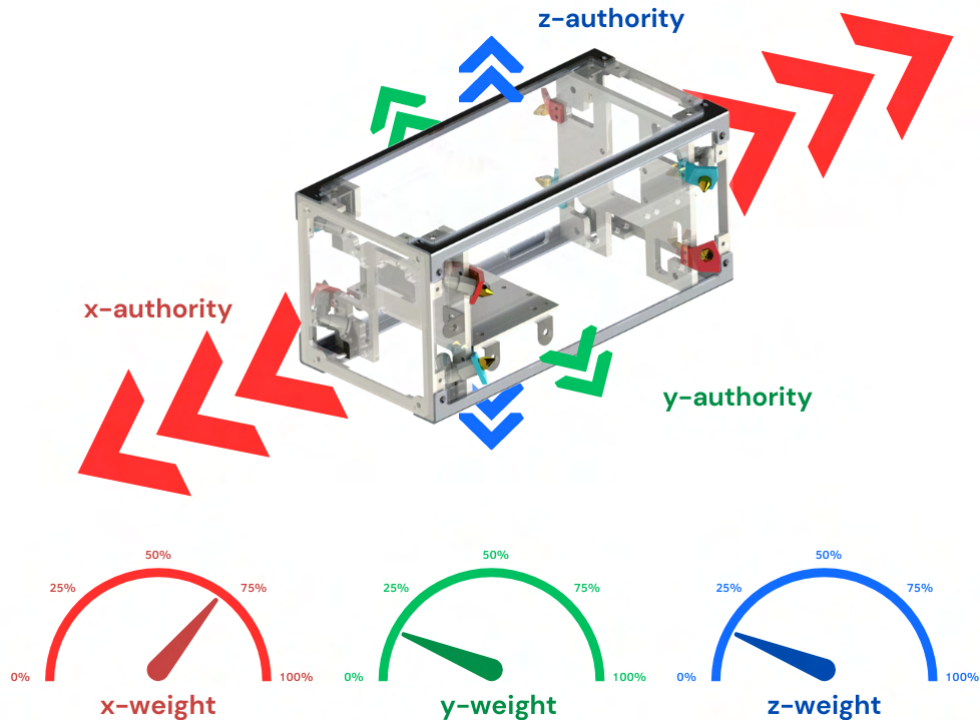


Figure 3.22: Authority and weight for trajectory corrections

- **Authority for attitude corrections:** this macro parameter is further divided into three, each referring to a specific axis, x , y and z . These three parameters refer to how much strong a certain correction along one

axis should be. It can be seen as an empirical proportional coefficient to apply to the theoretical number of cycles used to correct a certain attitude misalignment. This coefficient depends mostly on the value of thrust along each axis, that is indeed different due to the particular thrusters configuration, but they have also been tuned to get the desired results. The x axis (roll) and z axis (yaw) have the lower value with the respect to the y axis (pitch) because of this difference in thrust and consequently in torque.

- **Weight of attitude corrections:** this macro parameter is further divided into three, each referring to a specific axis, x , y and z . These three parameters refer to how much important is considered correcting one axis with the respect to the others possible corrections, including trajectory. Therefore, these parameters give the likelihood of correcting a certain axis given the same percentual misalignment with the respect to the perfect alignment. In particular, to y axis (pitch) and z axis (yaw) the same weight is given due to the symmetry of the approach, where there are no preferred axis of rotation. Instead to the x axis (roll) the least weight is given because the symmetry of the interface allows to withstand a biggest misalignment.

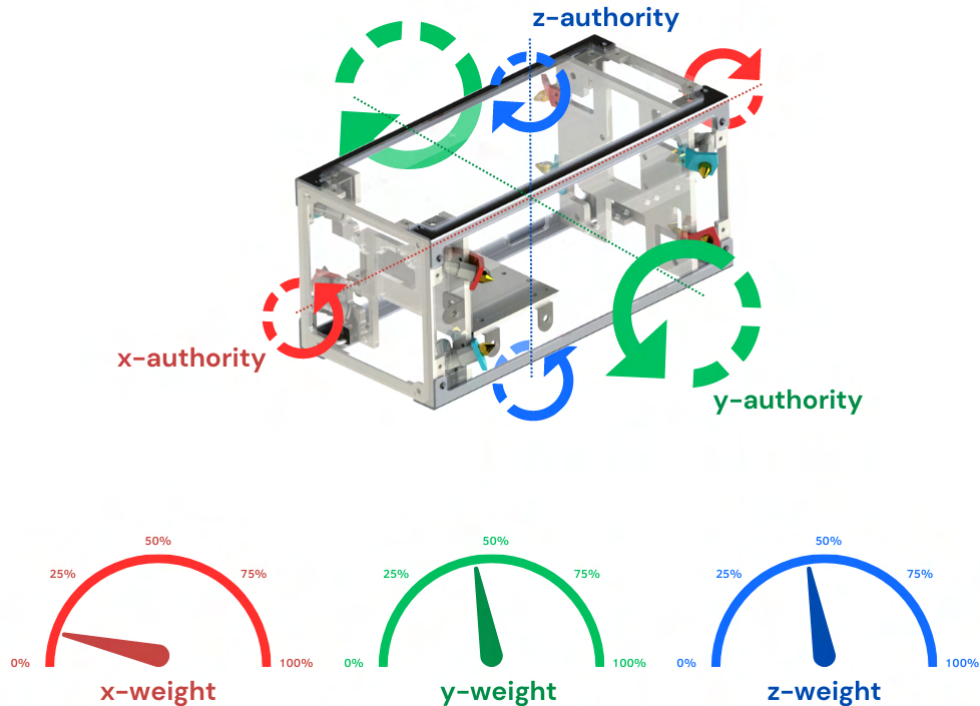


Figure 3.23: Authority and weight for attitude corrections

- **Approaching threshold:** This parameter refers to the minimum distance at which the Chaser engages a faster control to dock more precisely. When the distance is lower than the chosen value the Chaser actively increase the velocity of the control and commands performing more precise

manoeuvring. Although it is mandatory to perform a faster control during this phase, the Chaser can accelerate the control even before if a high misalignment is detected. Including this eventuality was also needed because testing on the frictionless table poses less problems, in terms of DoF to correct, than testing on the parabolic flights. This distance has been set to be around 70 mm face-to-face.

- ***Distance at which tolerances decrease linearly:*** This parameter refers to the distance at which the tolerances of the approach phase are reduced linearly up to the docking ones, that are clearly stricter. This linear decreasing trend describes a 3D cone of approach, that 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 of 150 mm face-to-face.

3.4.3 Complete docking manoeuvre

In these tests Chaser was placed in front of the Target and it tries to reach it by computing a trajectory to follow. The tests are indeed simulations of the docking manoeuvre the Chaser has to perform on the parabolic flight. In Figure 3.24 snapshots from a test performed on the frictionless table are reported.

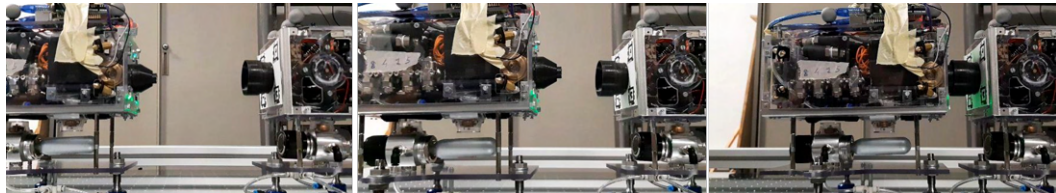


Figure 3.24: Snapshots of the complete docking manoeuvre on the frictionless table

Before proceeding, it should be noted that roll control (rotation along x) has not been tested properly as for the other DoF due to the fact that on the frictionless table it was difficult to test it. In fact, in the case of z axis trajectory corrections and y axis attitude (pitch) corrections simply turning the chaser 90 degree of roll would have inverted the y and z axis allowing us to test the Chaser in a similar condition with the respect to the flight one. In contrary, with the roll there is no direction in which to put the mock-up so that we could have tested it. Hence, to avoid unwanted loops in roll control, the tolerances on this angle have been removed or extremely increase while approaching very close.

Let now consider the above-mentioned test performed on the frictionless table. The plots reported in Figure 3.25 and 3.26 show the relative trajectory followed by the Chaser and the relative velocity with respect to the Target during the tests on the frictionless table. It is presented the trajectory along x axis, represented as face-to-face distance so that it is clear at which distance

the Chaser started the manoeuvre, the trajectory along the y axis and the rotation along z axis (yaw).

First of all, the layout of the graphs must be described:

- The red line with blue dots is the trajectory computed by the Chaser. It takes into consideration all the calculation derived by the various sensors of the Chaser made by the Medium-Level. It is important to underline that it is not the trajectory path planned by the Chaser, it is the one that it has followed according to its sensors. Moreover, every dot refers to a certain command sent. Every time that the Chaser choses to correct some axis it saves its pose and calculation, that are then recovered by the operators as a single file containing this particular track.
- The black line, instead, represent the reconstruction of the trajectory with the Motion Capture that is used as a reference. During the flights the OptiTrack is not available, therefore a software has been prepared and takes into consideration the video from one or more external camera to reconstruct the trajectory.
- The grey dotted lines are the tolerances conditions of maximum misalignment related to the y and z axis, θ and ψ . The roll has not a misalignment tolerance firstly because the interfaces are both symmetric, and secondly because it was more difficult to test this rotation on the frictionless table than the other rotations as explained previously.
- The horizontal grey dotted line is the “approaching threshold”, that is present only on the x axis. This threshold represents the moment in which the face-to-face distance of the mock-ups is around 70 mm. From that moment the tolerances on the other DoF becomes linearly stricter up to the “docked conditions”.
- The commands are represented vertical dotted lines: green cycle means that the necessary correction is along the positive axis, red cycle along the negative axis. Moreover, the number of cycles is labelled alongside the dotted line. The cycle for which the command is active represents the intensity of the command needed for the correction as explained previously.

The tests aimed generally to have a manoeuvre around 8 or less seconds with an around 300 mm initial distance. The reason being that although the low gravity phase is theoretically longer, it is important to have margins.

It must be underlined that the starting point for the manoeuvre is considered the first command sent, therefore the time taken for localizing the Target and compute the path is not reported in these graphs (but it is very short, under 1s). During this time the Chaser has drifted a little bit, causing a slight initial drift velocity in all three DoF considered. This was also due to the fact that it was released manually and therefore a small residual velocity was an nearly inevitable effect after the release. Testing it with the magnetic release with the arm was a problematic solution as explained earlier, therefore it has

been chosen to release them manually far from the corners so that they would not eventually bump into them.

Initially there is a clear misalignment in every axis. That is a wanted condition, in fact most of the tests were performed with an initial misalignment with the respect to the docking conditions. The misalignment was always chosen to be around the tolerances of the software because the focus was seeing if it could correct it and stabilize the approach. Other tests were performed with a nearly perfect alignment (or small errors due to small disturbances that could cause drafts-like effects). In this case, the control would have been just a sequence of forward thrusts without exceeding the maximum velocity of impact, that is a requirement of the experiment, with a series of small adjustment in lateral misalignment and attitude. In these cases, the frequency and number of the commands sent is less than the presented case because the Chaser finds himself already aligned. However, our concerns were mostly regarding if it was able to correct misalignment while approaching the Target rather than if it was able just to reach it.

It is possible to note that generally the software acts with low cycles, in fact most of the cycle have a value of just 1, that means an opening of the valves for around 35 ms (nearly $1/30$ s, with 30 being the frequency of opening of the valves). This behaviour is due to the tests about the tuning of the authority parameters of the Medium-Level. In fact, it has been established that a high frequency with low thrust commands was more efficient than a lower frequency with highest thrust commands due to the possibility of correcting more frequently eventual misalignment or miscalculations. The only difference is in the x axis where the commands have higher thrust because the Chaser have to cover more space. From the tests it has been expected that during the parabolas, if the release is correct the Chaser would perform similarly to this test regarding type of actuation. In fact, the only difference expected is an increase of the frequency of the commands, because of the increase in the number of DoF to control.

The approaching threshold is reached around 7 seconds. However, the increasing in the frequency of the commands starts around 2 seconds before (5 s from starting). The reason being that the Chaser wanted to stabilize the y axis that was still decreasing. Then, when it reaches the approaching threshold it clearly prefers attitude control over position. Both in yaw and y axis, the misalignment decreases mostly linearly during the approaching phase. However, during the docking phase the frequency of commands increase to maintain it inside the tolerance gap and to decrease as possible the velocity along those axes.

As it can be seen by the corrections applied, by how the Chaser was able to correct the initial misalignment of the yaw angle and of the axis, the authority of the command was enough to 2D manoeuvring on the frictionless table with consistency by performing small adjustment. Moreover, the increase in frequency of the commands during the approaching phase improves the control by adding layers of precision.

For what concerns the precision of the computed manoeuvre it can be stated that the errors remains inside the requirements of the localization system.

The error decreases with a decreasing face-to-face distance. The minimum is when the camera is at the nominal focus distance, that is around 150 mm. When too close the errors increase a bit but still under the requirement of the localization. However, 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 most probable reason is that since it was moving slower the pose computing was precise. In fact, during the tests of the localization we found a dependence between the velocity of the approach and the precision of the localization. However, this trend was not explored deeply because, in the range of distances and velocity relevant for the experiment, the localization system was characterized by acceptable errors with the respect to the requirements.

Other tests were performed similarly to the one presented. The results showed that the system was working as expected. Moreover, the simplicity of the software not only did not impact the results of the tests deeply, but also has been confirmed to be optimal for finding and analysing possible software errors.

Regarding the strategy of the manoeuvre is important to underline that when the Chaser was tested imposing fast manoeuvre approaches or inducing a strong initial velocity the precision of the docking clearly decreased. These fast approaches were also characterized by higher thrusts and high frequency commands throughout the whole manoeuvre. In general, the conic shape of the docking interfaces allowed to correct eventual small misalignments other than when the approach was too fast or near the limit imposed by the requirements. This scenario of fast manoeuvre will be found during the campaign, as it will be presented in next Chapter.

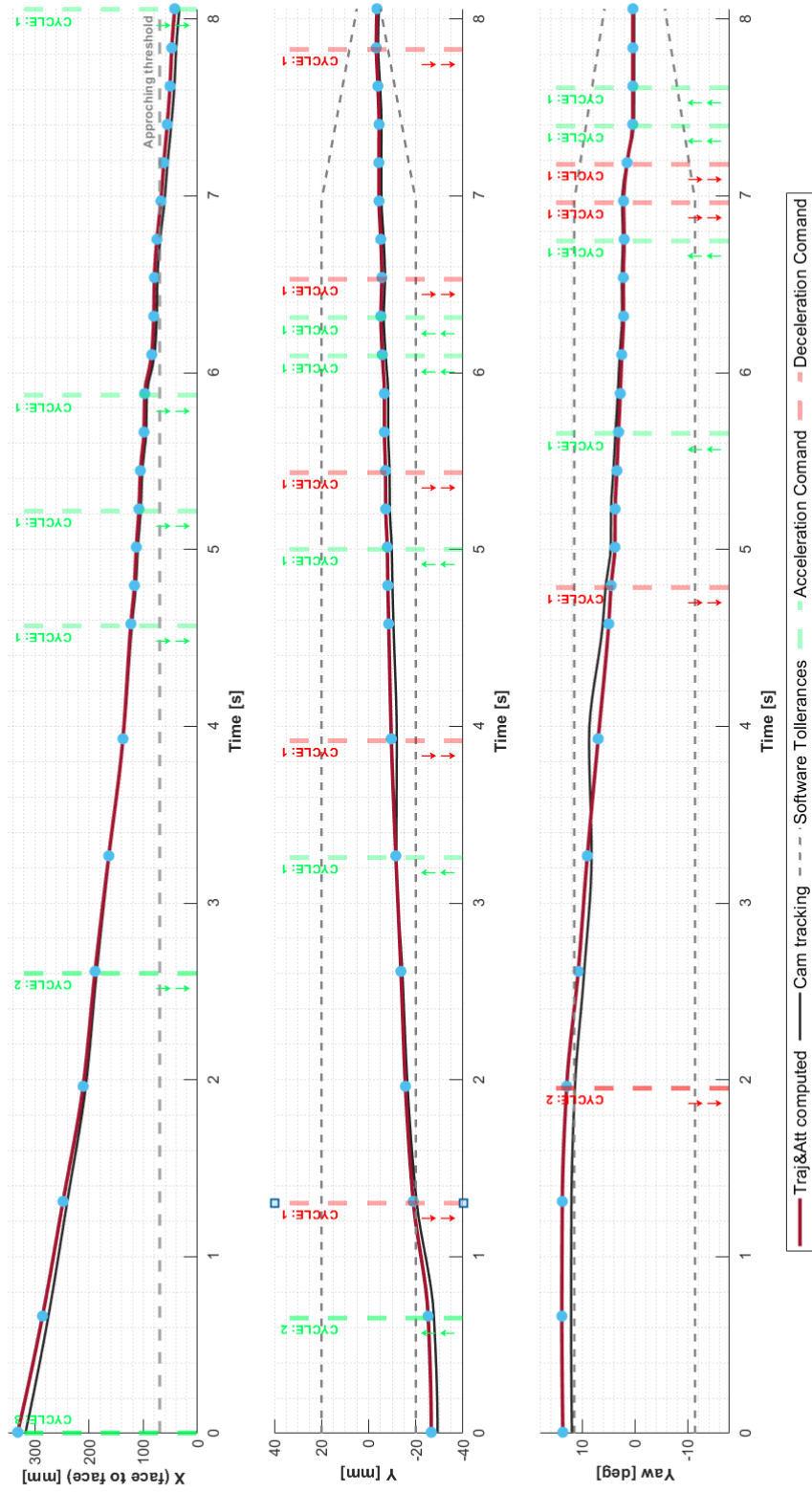


Figure 3.25: Testing on frictionless table - Relative trajectory of the Chaser with the respect to the Target with reference trajectory from external camera tracking (OptiTrack)

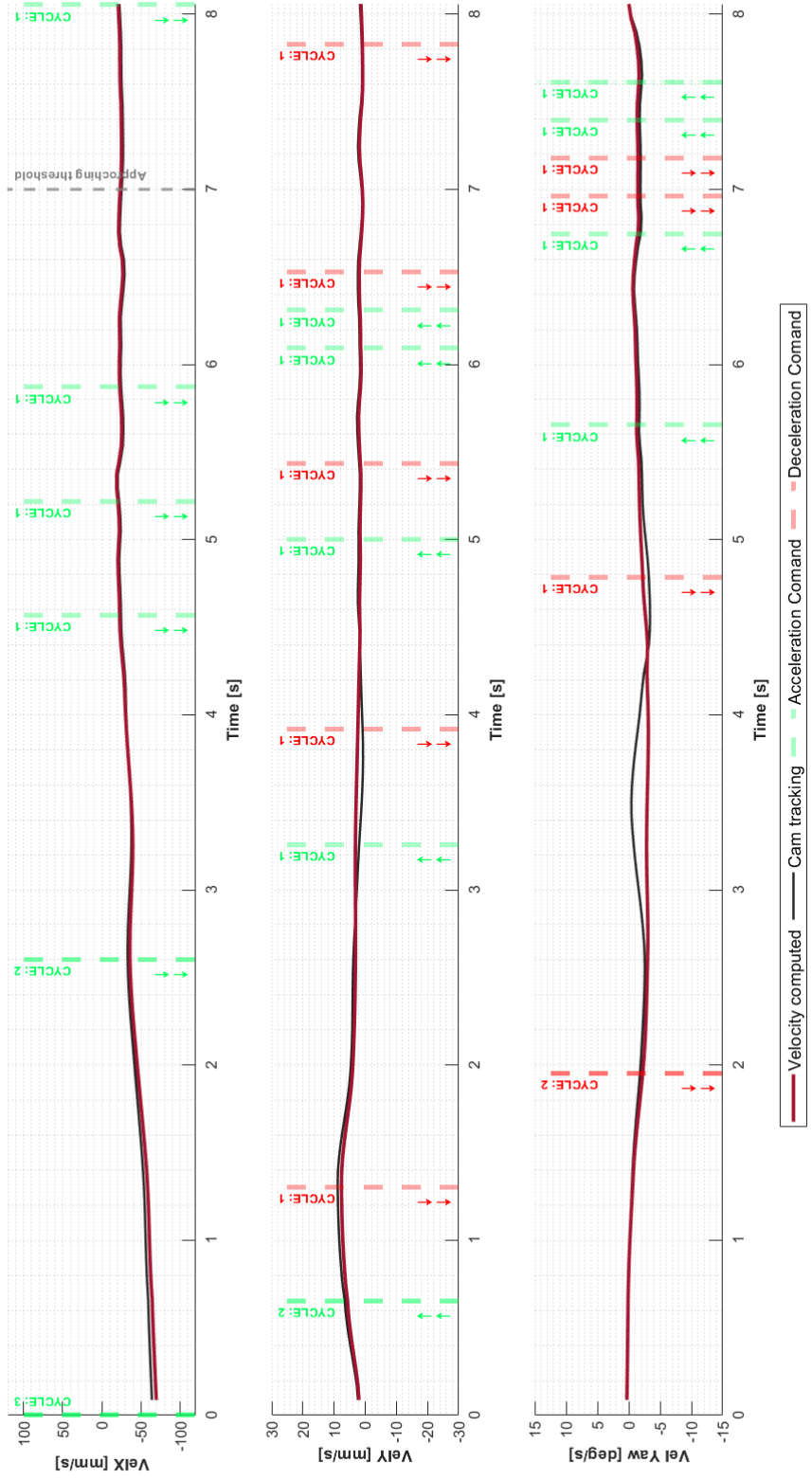


Figure 3.26: Testing on frictionless table - Relative velocity of the Chaser with the respect to the Target with reference trajectory from external camera tracking (OptiTrack)

4 Experiment Results

The flight campaign has been very productive, but not all the parabolas provided the desired results.

The data collected during the campaign are taken into account to conduct analysis regarding the behaviour of the Chaser during the manoeuvres and draw conclusion about the integration between On board Computer System and Propulsive System.

The data stored include:

- Sensor acquisition that include the information derived from data of IMUs and Proximity sensors;
- Apriltag Pose that include the information derived from camera data acquisition used to find the relative pose between Target and Chaser;
- Twist calculation that include the information derived from the Medium level software used to find the relative twist between Target and Chaser;
- Absolute pose relative to Target that include the information derived from the Extended Kalman Filter. These are useful information to reconstruct the trajectory computed by the Chaser and to understand better the commands it chooses;
- Absolute pose relative to odometry that include the information derived from the Extended Kalman Filter. These are useful information to reconstruct the trajectory computed by the Chaser and to understand better the commands it chooses;
- Medium-level to Low-level commands that include a recollection of all the commands sent by the Medium level to the Low level of the Chaser to be actuated. These are stored with an absolute time reference to be in accordance with the video recordings that follow the same format;
- Data from the external cameras that include a recollection of all the video recordings from the 3 available cameras about the 93 parabolas. These are stored with an absolute time reference to be in accordance with the Medium level to Low level commands that follow the same format;
- Data regarding the g-levels that include the information provided by Novespace regarding the g-levels and trends in every parabola of every flight.

These data are published via ROS by the Chaser and can be stored by an operator in a hard disk. Only the data from the external camera do not get published via ROS, instead they are stored in the camera SD.

A selection of "*successful*" parabolas, according to the detection by the software, are taken into account to perform the analysis. In particular the post processing is based on:

- **Medium Level analysis**, that includes all the handling of the data from the mockups (sensors, camera, software etc). It concerns the study of the trajectory computed and followed by the Chaser, including the discussion of the choice of the commands chosen to perform it;
- **Camera Recording analysis** that includes all the handling of the data from the external reference camera. They are used to have an external reference to the Chaser readings and evaluate its efficiency in computing its relative position to the Target.

From the data acquired during the tests a lot of information can be derived. The first part of the post processing has the main objective to investigate the operation of the **Medium Level Software**. In particular, the manoeuvres of every parabola can be recreated and analysed singularly thanks to the above-mentioned ROS poses. Therefore, the first task is to interface the data between the poses, the twists and the sensors to recreate the relative motion between the mock-ups. Moreover, to further analyse the Apriltags implementation, the data from the sensors and the Poses of the Target from the Apriltag visual system can be compared underlining the strengths and capabilities of such technology. Then the Medium-level to Low-level commands are introduced, so that it can be discussed how the software tried to perform the manoeuvre by highlighting the effect of each command on the attitude and momentum of the Chaser, and by comparing its choices to different kind of manoeuvring.

The second part of the post processing has the objective to compare data collected from the On board Computer System with the reconstruction of the manoeuvre from the external view guaranteed by video recording.

Finally, the interest of the analysis lies on evaluate how much effective the shape of the interface is in terms of helping the Chaser to centre during the final moment of the approach, that are the most critical. To do that first of all the last commands sent by the software is compared, trying to find any differences in the type of approach. In particular it is noted if relatively great misalignment can be corrected by the interfaces themselves reporting from the command data a faster and more precise correction, due to the fact that the contact forces generated in that case are immediately reacted by the software commands computation.

Moreover, another interesting topic is the Apriltag visibility, in fact it is evaluated how the shape of the interface affects the Apriltag visual recognition in a mission-like environment, comparing the errors in evaluating the pose with the respect to the ToF readings.

Concerning the propulsive system it is possible to investigate how it works during the manoeuvre in terms of response to the commands sent by the software. In particular based on the number of corrections needed, the timing and the frequency of the opening of the lines, and consequently the precision of the manoeuvre.

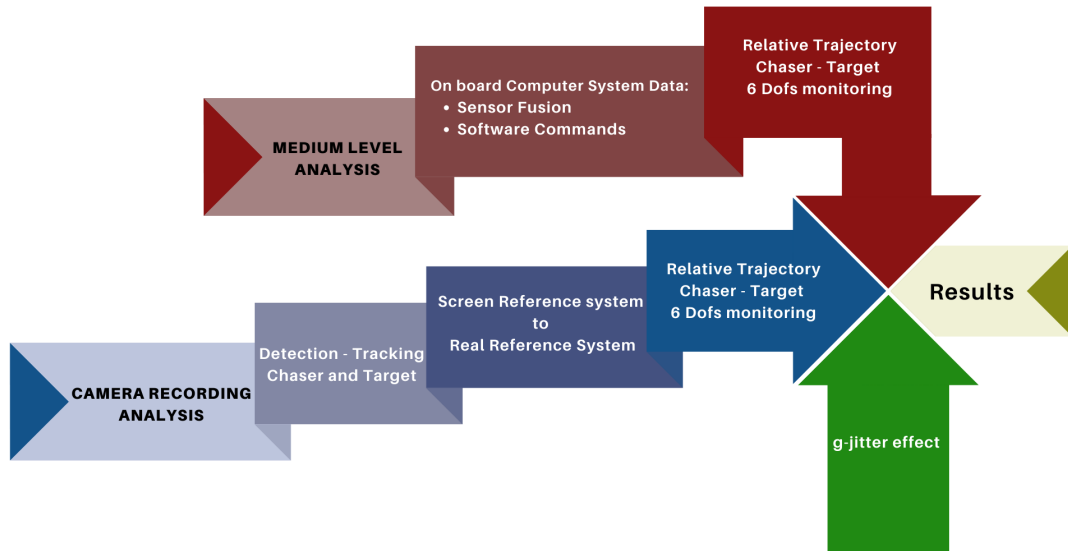


Figure 4.1: ERMES post-processing flowchart

Conclusions can be drawn about the Chaser actuators behaviour, the software operation during the flight campaign, and all the aspects that could be improved in order to obtain succesful docking parabolas.

The parabolas that can be defined as “successful” are:

- **Flight #2 - Parabola #27 - #2/27**
- **Flight #3 - Parabola #17 - #3/17**

These two parabolas shared similarities in terms of sequence of events, precision of the manoeuvres and gravitational effects. In fact, in both of them, the Chaser was able to enter the docking interface inside the desired tolerances but, as a consequence of the impact, the mock-ups hit the external structure. However, the two parabola differ just on the quantity of commands sent, because the #2/27 has a larger quantity than the #3/17. This difference is linked to the fact that, during the #3/17, the disturbances due to various factors (gravity level, release etc) have been very low. In the following sections it is presented the post processing regarding parabola #2/27, for which a large quantity of interest topics can be discussed.

During the low gravity phase after the injection, the initialization of the experiment is autonomous. In fact, the experiment will start while whenever the timer reaches a certain moment. Since the parabola lasts 22 seconds and since the gravity is at a minimum in the middle, the experiment will start at 6 seconds, considering a total duration of 10 seconds, 2 seconds of release and 8 seconds of free flying manoeuvring. In the meanwhile, each operator shall sit near the laptop, both anchored in place with straps.

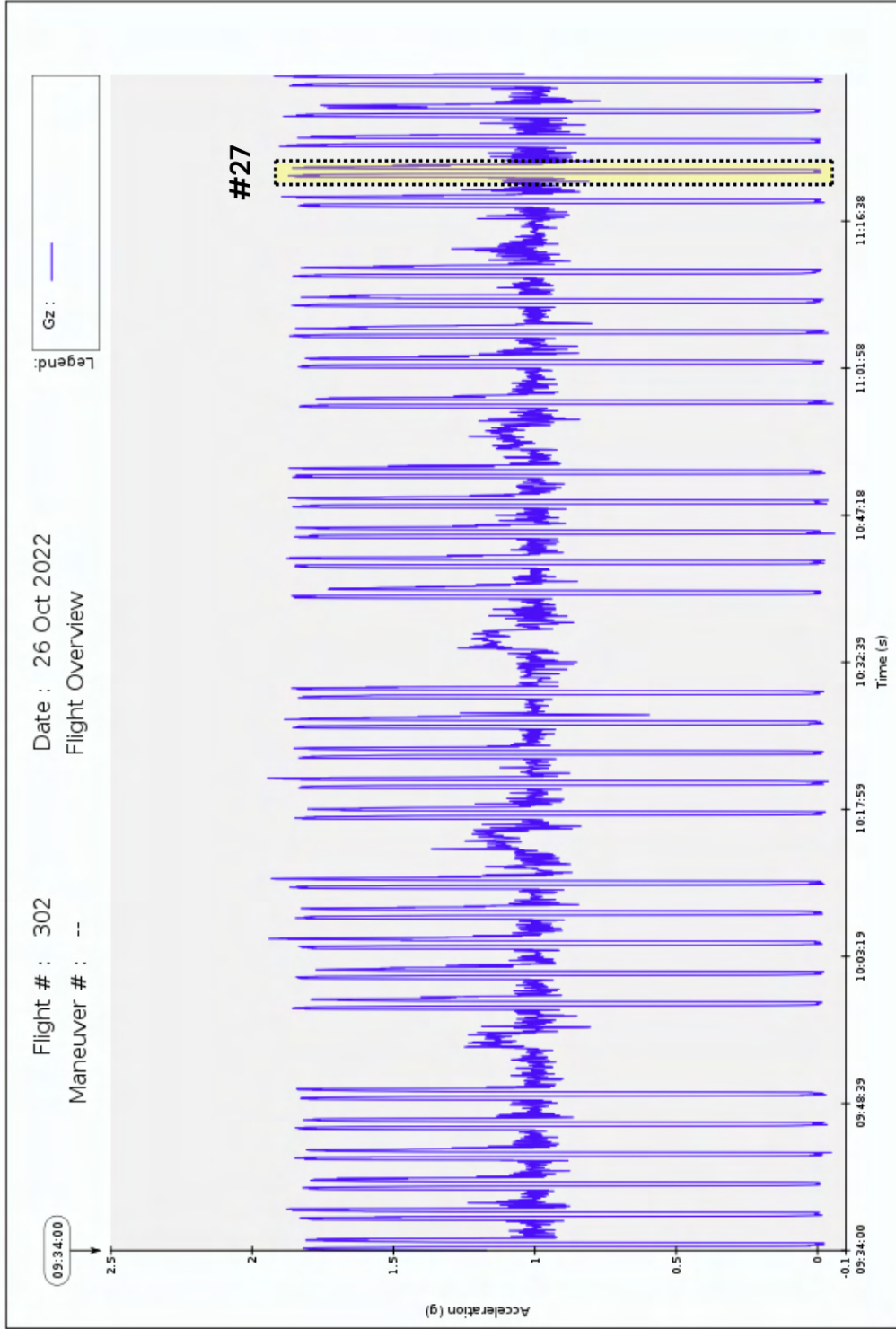


Figure 4.2: Acceleration profile of the Flight #2 (credits: Novespace)

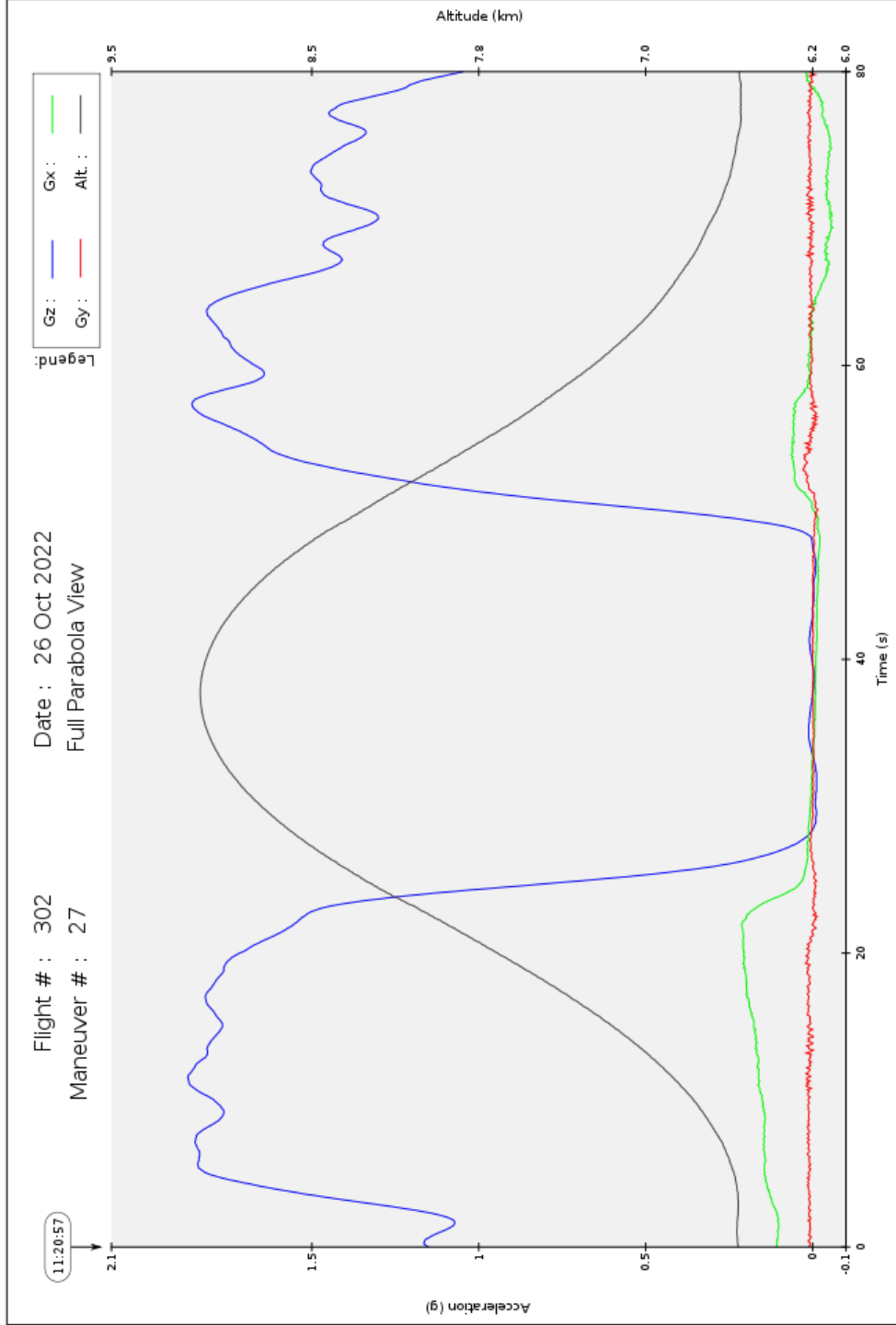


Figure 4.3: Acceleration profile of the Flight #2 - Parabola #27 (credits: Novespace)

4.1 Medium Level Analysis

The objective of this first part is the reconstruction of the manoeuvre from the data collected by the On board Computer System of the Chaser mock-up. As mentioned, the collected data for each manoeuvre consist of output of the sensor fusion process, that involves data from the IMU, ToF and Pi Camera. In particular the information provided consist of position and attitude of the Chaser mock-up relative to the Target. It is important to underline that these data are defined in respect of a reference system centered in the center of mass of the Chaser, with x axis along the conjunction Chaser-Target, y axis on the horizontal plane and z axis to close the left-handed triad (UCS_C).

Considering a same constructed reference system for the Target mock-up (UCS_T)., the Chaser software command has the objective to make the two reference systems aligned, with a distance in the conjunction in x axis equal to 190 mm, that is the distance between the two center of mass in order to have a face to face distance (d_{FtoF}) of the dimension of the probe-drogue docking system, that is equal to 40 mm. Along y and z axis, the Chaser has to be inside the approach margins that have been set to ± 5 mm, and regarding the relative attitude it has to be inside the approach margins that have been set to $\pm 5^\circ$ for pitch and yaw. No final constraint is set for the roll due to the symmetry of the interfaces. The software recognize "**docking**", and also a "*successful*" parabola, when these conditions have been satisfied.

Position Tolerance			Attitude Tolerance		
x	y	z	ϕ	θ	ψ
40 mm	± 5 mm	± 5 mm	no data	± 5 deg	± 5 deg

Table 4.1: Margins of tolerance

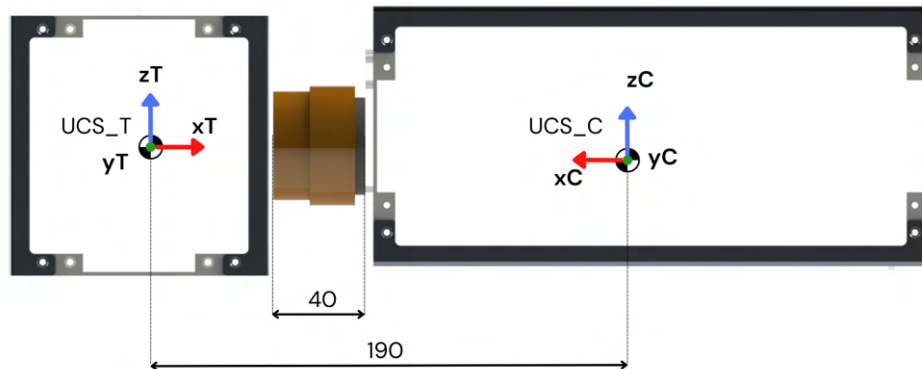


Figure 4.4: Rendering of the docked configuration



Figure 4.5: Target and Chaser in docked configuration

The manoeuvre reconstruction starts with input data represented by the information relative to the Chaser mock-up, and that create a structure of data necessary for the processing of the collected one. This structure is composed by:

- *Inertial data* that consists of mass, moment of inertia;
- *Shape* that consists of shape of the Chaser mock-up not translated or rotated;
- *CamOffset* that consists of offset of position and attitude of the Camera with the respect to the reference system centered in the center of mass of the Chaser mock-up;
- *Thruster data* that contains shape, position and axis of the commands. It consists of a matrix that links the ordered number of thrusters opened with each command;
- *Software data* that contains the duration of one cycle, the thrust of one nozzle and the tolerances of misalignment considered by the software. The tolerances have been experimentally determined, and consist of cylindrical range for x face to face distances above the approaching threshold, around 7 cm, while become conical range for distances under this threshold.

Thruster data and Software data structures are processed from the Raspberry Pi saved readings regarding sensors acquisitions and commands sent. These are located in two different Arrays:

- the first one consists of (30×1) Array. Each of the 30 lines is a matrix that contains the relative trajectory and attitude of the Chaser mock-up with the respect to the Target, referring to the mean time of the sensor readings. Each line is: $[time; x; y; z; roll; pitch; yaw]$. The time in the vector is the mean time of the reading and processing of the data from the sensors.
- the second one consists of (30×1) Array too. Each of the 30 lines is a matrix that contains the commands sent by the Raspberry Pi to the electrovalves of the Chaser mock-up. Each line is: $[time; axis; cycle]$

The analysis provides the interpolation of the collected data regarding relative position and attitude of the Chaser with respect to the Target, hence the six Degrees of Freedom of the mock-up, reported in the graphs of Figure 4.6. Furthermore, sending command of the software to the actuators, for each axis, are expressed in cycle. The cycle is the period of opening of the valves, linked to the PWM frequency selected for the control, $1/f_{PWM}$. As shown in the graphs, the Chaser mock-up has to be inside the gap of tolerances related to the y and z axis, while along x axis it has to reach the approaching threshold up to a face to face distance of 40 mm in order to detect the "*docking*".

To cover these two cited conditions the software sends correction commands to the system by activating the actuators. As mentioned before, from the graphs, the commands are represented by the cycle: green cycle means that the necessary correction is along the positive axis, red cycle along the negative axis. The cycle for which the command is active represents the intensity of the command needed for the correction. It is possible to note that generally the software acts with low cycles (period of opening very short) and high frequency of commands (sequence of opening very fast), as deduced from the Test Intergration. In fact, also during the campaign, most of the cycle are type 1, that means an opening of the valves of 35 ms, but they are repeated in sequence in a closed gap time. This behaviour is clear especially along the x axis, so the approaching of the Chaser mock-up to the Target along their conjunction is obtained from a dense sequence of thrusters activation. As consequence of this, the approaching threshold is reached in a time of 4 seconds from the instant in which the two mock-ups are released. At this time the initial contact between the two docking ports is registered, and this is translated in disturbances especially in the relative attitude related to the lack of authority of the Target control rather than anomalies of the Chaser mock-up.

These acting thruster behaviour is translated in fast approach along the x axis, that is due to the necessity of reducing the time needed for the manoeuvre. The reason why the time of manoeuvre was that short is due to the combined effect of g-jitter effect and "imperfect" release. Therefore, the necessity to have a certain release velocity was needed. Hence, the approach along x -axis is faster than expected. However, the Chaser did not choose to slow down, that could have been achieved by sending a deceleration command towards the x -axis. This behaviour is not a random occurrence, because the Medium Level

control would also take into consideration the type of release in order to avoid that the Chaser would decelerate while a pushing release has been deliberately chosen.

Furthermore, during the manoeuvre important influences on the position along the y and z axis are registered, moreover it can be noted that the attitude during the approaching phase is constantly corrected. As consequence of this the Chaser mock-up is approaching the Target with parallel face to face condition but with a not precise alignment along the axis. In this case an important role for the docking is played by the conical design of the docking port. In fact the disalignment can be recovered with sliding between the two doors contact surfaces and than the attraction between the two magnetic interfaces. Nevertheless, during the campaign it has been registered in most of the parabola that between the probe and the drogue destabilising contact forces are generated, causing the failure of the manoeuvre. The reason of this uncontrolled proximity navigation are first of all the insufficient gyroscopic stiffness of the Target, that is not able to stabilize itself at the moment of the contact. Then the fact that after the impact all the momentum is shared between the two mock-ups resulting in a relatively strong impact. The pushing back effect, also caused by the contact, is one of the major causes, in addition to g-jitter fluctuation, for the impact with a surface of the plane or of the experiment setup that ends completely the experiment.

In this analysed parabola critical behaviors are found in the cited y and z axis, and also in yaw angle. In fact, yaw angle is near zero during the initial navigation, while increases during the approaching phase.

On the other hand, at the beginning of the manoeuvre only the pitch is not near zero. This initial misalignment in pitch value could have been due to the magnetic constraints that kept the mock-ups in place. In fact, the initial placement of the mock-ups was not perfect. In particular, the increase in the pitch misalignment could be due to two different factors coexisting, in particular it could be due to a disturbance in the release that made the Chaser nose down a little, or it could be due 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 it immediately corrects it. It should be noted that pitch manoeuvre has not been tested before. For this reason the software compute a frequent sequence of commands; in particular tuning data related the other axis control are provided during tests, giving a premultiplicative coefficient as function of the thrust forces F_y/F_z . These frequent commands are translated in a good response of the system, with precise corrections that noted the fact that it is designed for short and consecutive impulse thrusters operation.

Nevertheless, the misalignment in different axis did not happen at the same time, therefore the Chaser was able to correct them individually. The pitch and z -axis misalignment happened at the same time. However the combine effect toned down each of them because the nose down (pitch negative) has compensated the increase in height with the respect to the ground so that the Target remained visible throughout the manoeuvre.

Furthermore, from the obtained data it is possible to estimate the relative

velocity of the Chaser mock-up with the respect to the Target. The results show that the Chaser cared more about the alignment than to slow down. In fact, it focused a lot on reducing the angular pitch velocity during the docking phase even though the misalignment along the y and z-axis could have been improved too. One reason for this behaviour is derived from the tests on the frictionless table, where a prevalence in attitude control over trajectory during the docking phase was given.

As it can be noticed studying the relative motion, the effect of the residual gravity during the parabola is the same for both Chaser and Target, and so they are affected by the same g-jitter effect. This is true for a coordinate release of the two mock-ups. The timing of the release of Target and Chaser differs of under a thousandth of a second and since the frequency is lower for the major harmonics that describes 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. However, in some parabolas a “late” release made the mock-up free-float very near the Release Structures or even in between, and the effect of the g-jitter was to cause one of the mock-ups to hit some surface due to disturbances to the trajectory relative to the plane. In that case the parabola is completely lost.

As it can be seen by the corrections applied, by how the Chaser was able to correct the initial misalignment of the pitch angle and of the axis, the authority of the command was enough to 3D manoeuvring with consistency by performing small adjustments. Moreover, 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. It must be recalled that the Chaser did not have an authority so high that it could have corrected the g-jitter effect or any other disturbances, in fact it was designed to perform small and simple corrections.

Due to the experiment conditions and some un-modelled dynamics, the repeatability of the experiment with the developed pushing system is not very high, on the contrary, on each parabola the imposed initial angular velocity has a different magnitude, even keeping unchanged the initial conditions. For this reason, it is very difficult to compare and analyse the obtained data for each parabola, because in each test the actual initial conditions are very different from others.

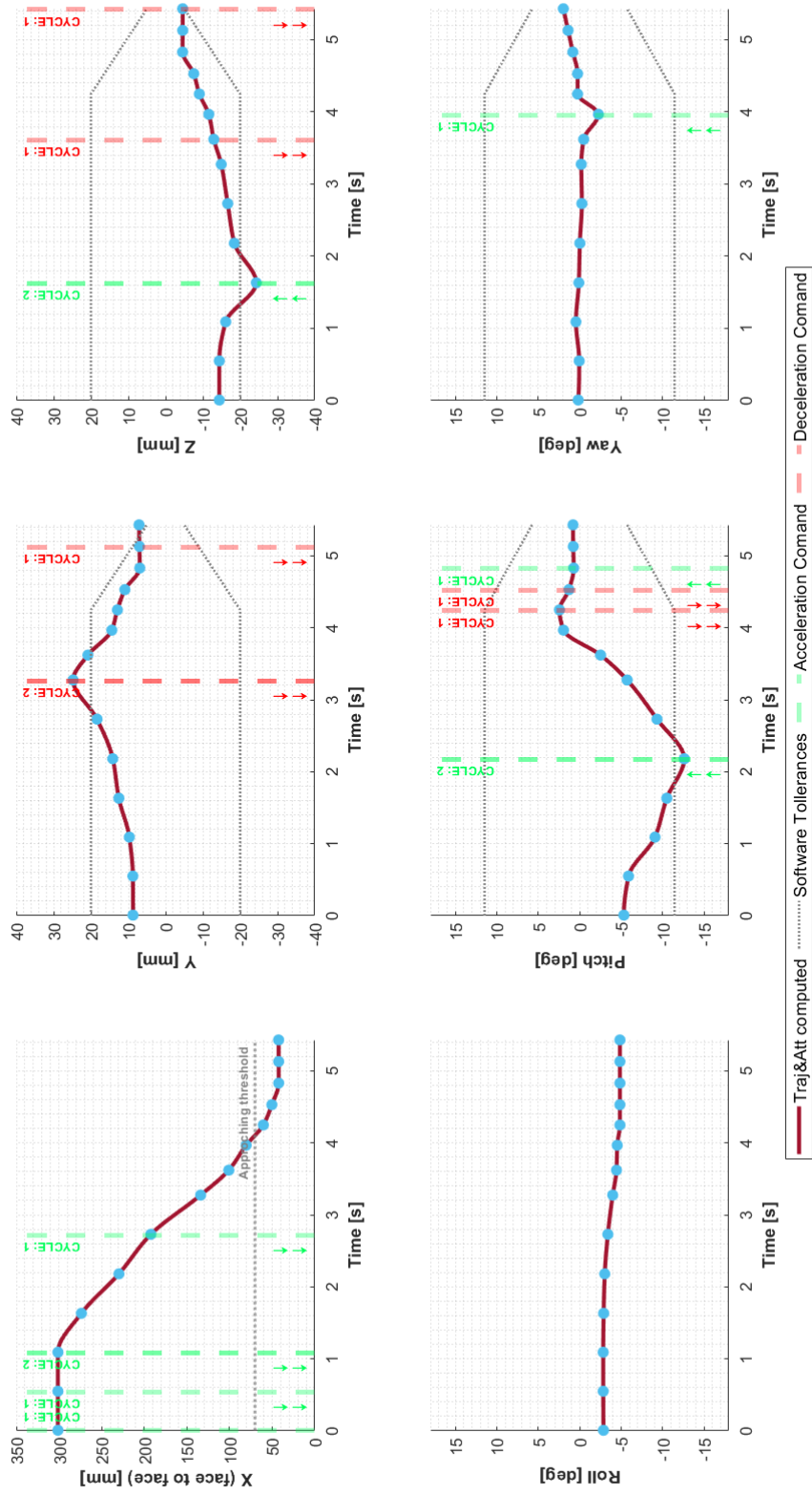


Figure 4.6: Output graphs from the analysis: relative trajectory of the Chaser with the respect to the Target

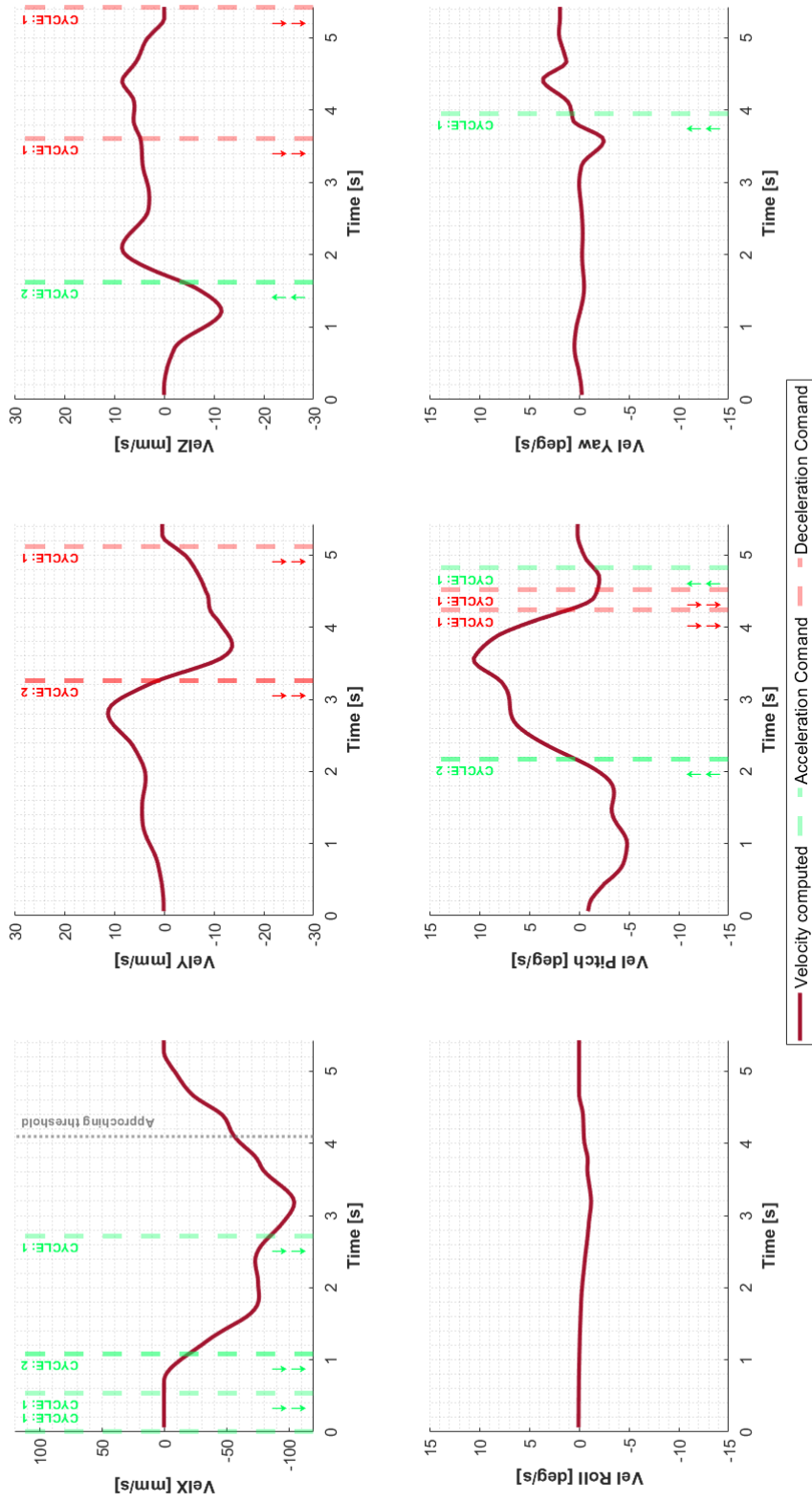


Figure 4.7: Output graphs from the analysis: relative velocity of the Chaser with the respect to the Target

4.2 Camera Recording Analysis

4.2.1 Detection and Tracking

The focus of this section is moved on the analysis of the video recording of the manoeuvre. In particular the objective is to track reference points of the geometry of the mock-up in order to obtain their position in every frame of the video, using the Motion-Based Multiple Object Tracking method.

Tracking is the process of locating a moving object or multiple objects over time in a video stream. Tracking an object is not the same as object detection. Object detection is the process of locating an object of interest in a single frame. Tracking associates detections of an object across multiple frames. Tracking multiple objects requires detection, prediction, and data association.

- **detection** detects objects of interest in a video frame.
- **prediction** predicts the object locations in the next frame.
- **data association** uses the predicted locations to associate detections across frames to form tracks.

To detect objects in motion using a stationary camera it is performed a background subtraction. In particular the foreground detector algorithm, based on Gaussian mixture models, compares a color or grayscale video frame to a background model to determine whether individual pixels are part of the background or the foreground. A Gaussian Mixture Model is a parametric probability density function which is a weighted sum of Gaussian component densities. Morphological operations are applied to the resulting foreground mask to eliminate noise. Finally, blob analysis detects groups of connected pixels, which are likely to correspond to moving objects. The blob analysis is one of the important step for the detection of moving object. Basically it recognize the moving object in a scene. It works on a binary image and stands for “Binary Large Object”. After using this it can be concluded that the large connected moving pixels are the target and the small connected pixels are not of interest because these are noise.

The association of detections to the same object is based solely on motion, and the motion of each track is estimated by a Kalman filter. In particular, to track an object over time means that you must predict its location in the next frame. The simplest method of prediction is to assume that the object will be near its last known location. In other words, the previous detection serves as the next prediction. This method is especially effective for high frame rates. However, using this prediction method can fail when objects move at varying speeds, or when the frame rate is low relative to the speed of the object in motion. A more sophisticated method of prediction is to use the previously observed motion of the object. The 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. The Kalman filter also takes into account process noise and measurement noise. Process noise is the deviation

of the actual motion of the object from the motion model. Measurement noise is the detection error.

Finally data association is the process of associating detections corresponding to the same physical object across frames. The temporal history of a particular object consists of multiple detections, and is called a track. A track representation can include the entire history of the previous locations of the object. Alternatively, it can consist only of the object's last known location and its current velocity. Data association must take into account the fact that new objects can appear in the field of view, or that an object being tracked can leave the field of view. In other words, in any given frame, some number of new tracks might need to be created, and some number of existing tracks might need to be discarded.



Figure 4.8: Flowchart of the algorithm

The application of the method just presented is shown below: starting from the video recording of the manoeuvre during the parabola, a frame discretization has been performed. In particular every manoeuvre is recorded in video of duration of 5 seconds, the duration of the manoeuvre, and in order to obtain a good detection sequence from the blob analysis a snapshot frame for every half second of the recording has been exported, also collecting ten frames for every manoeuvre. Figure 4.9 shows four of these frames in order to visualize the manoeuvre of the analysed parabola.

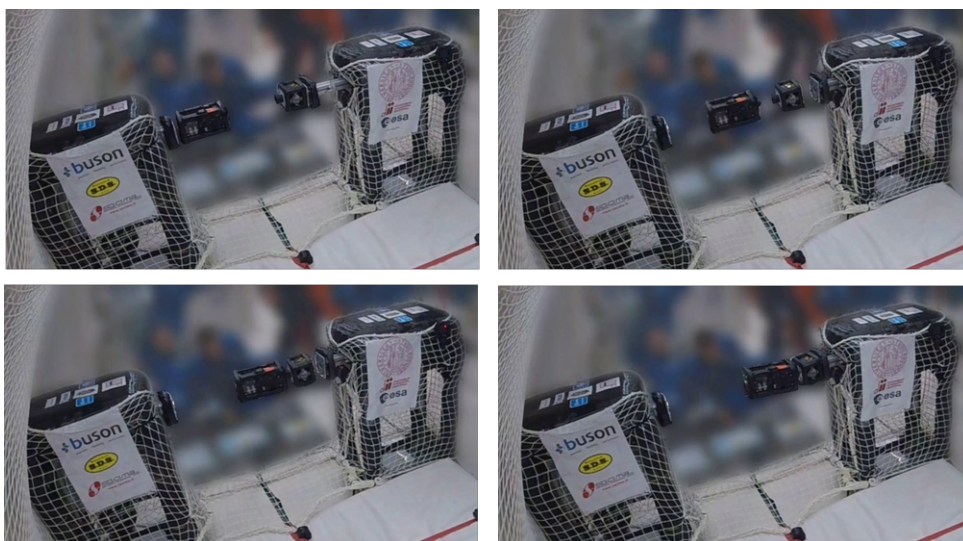


Figure 4.9: Sequence of frame of the manoeuvre of Flight #2 - Parabola #27

The algorithm executes the detection, iteratively frame by frame find the mock-up and save the coordinates in the screen plane, as shown in Figure 4.10. This is possible by the extrema point research from the uniform surface represented by the projection on the screen of the geometry of the mock-up. The principle is to find maxima and minima, also known as extrema, of the domain of a function. Once determined the extrema, the reference points coinciding with the vertices of the mock-up are selected, as reported in Figure 4.11.

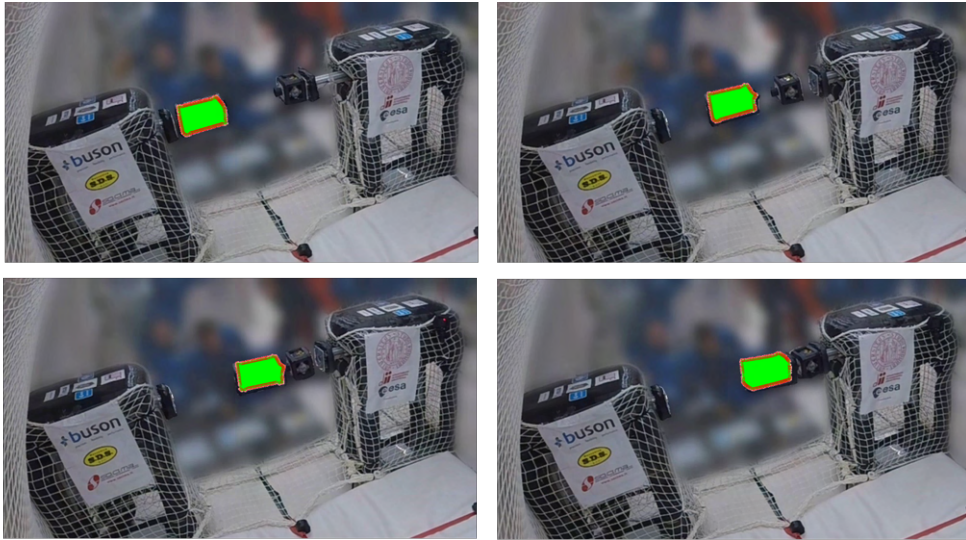


Figure 4.10: Extrema point research of surface of the mock-up

The *Screen Reference System SRS* is defined as $O_{x,y,z}|S$ in which from this processing are completely defined the vertices of the Chaser mock-up. From the projection of the parallelepiped on a plane it is possible to visualize 6 vertices, in fact the rectangle in isometric view is an hexagon.

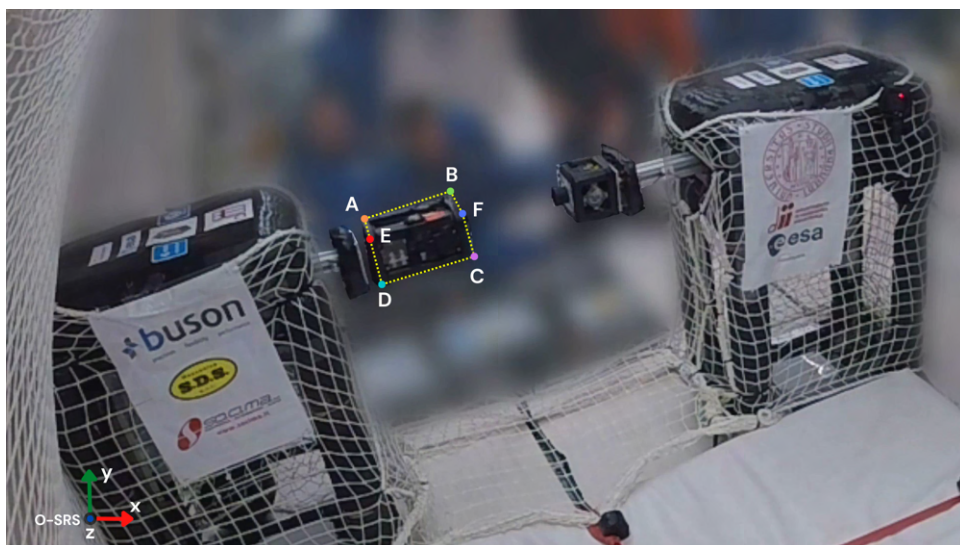


Figure 4.11: Screen Reference System - SRS

4.2.2 Trajectory Computation

Starting from the geometry of the Chaser mock-up, it is possible to identify a reference system centered in one of the vertice of the mock-up, arbitrarily chosen and named R . This reference system is the **Real Reference System RRS**, defined $O_{x,y,z|R}$, with $O \equiv R$ (Figure 4.12).

A geometrical analyses of the mock-up is conducted. Knowing the dimensions a , b , and selecting the following vertices A , B , C , known as:

$$A = \begin{Bmatrix} 0 \\ b \\ -b \end{Bmatrix} \quad B = \begin{Bmatrix} a \\ b \\ -b \end{Bmatrix} \quad C = \begin{Bmatrix} a \\ 0 \\ 0 \end{Bmatrix} \quad (4.1)$$

the following geometric considerations has to be taken into account:

- at each frame are known:

$$\|AR\|, \|BR\|, \|CR\| \quad (4.2)$$

- at each frame is defined:

$$AR + CR = BR \quad (4.3)$$

- at each frame is defined:

$$AR \perp CR \quad \rightarrow \quad AR \cdot CR = 0 \quad (4.4)$$

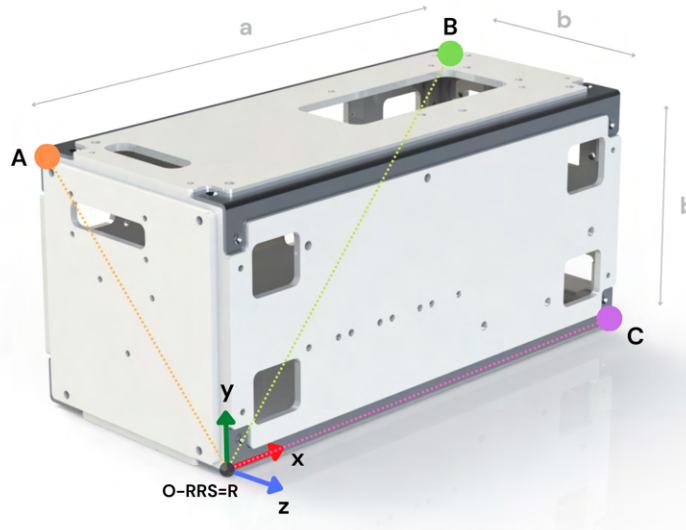


Figure 4.12: Real Reference System - RRS

From Detection and Tracking analysis are derived the data from video recording regarding the position of the previously defined reference point and

the selected vertices for the analysis, processed in every frame in the reference system associated to the screen, **SRS**:

$$R = \left\{ \begin{matrix} x_R \\ y_R \end{matrix} \right\}_{|SRS} \quad A = \left\{ \begin{matrix} x_A \\ y_A \end{matrix} \right\}_{|SRS} \quad B = \left\{ \begin{matrix} x_B \\ y_B \end{matrix} \right\}_{|SRS} \quad C = \left\{ \begin{matrix} x_C \\ y_C \end{matrix} \right\}_{|SRS} \quad (4.5)$$

As consequence it is unknown the z_S coordinate from the Detection and Tracking analysis, but it is possible to state that z_S is defined orthogonal to the plane of the **SRS**.

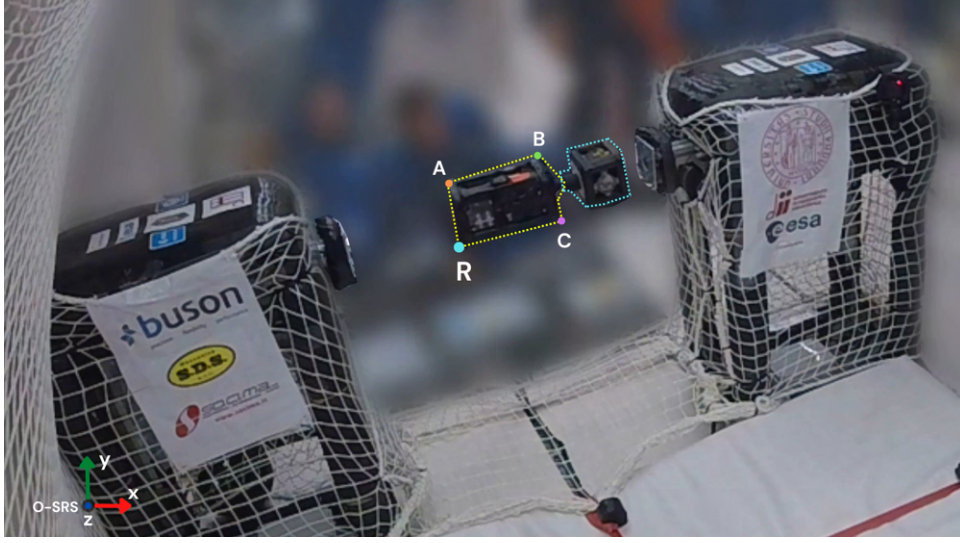


Figure 4.13: Reference Point and Selected Vertices

The starting point of the analysis is represented by these initial input data provided by the Detection and Tracking studies and the geometrical knowledge related to the mock-up. The objective of the first part is to solve a system of equation, built around these instruments, in order to find the rotation matrix from the **RRS** to the **SRS**, $M_{ROT_{R \rightarrow S}}$.

The first step subtracts the reference point R to the screen data. In particular the reference equation for a generic frame is:

$$A_{SRS(i)} = A_{SRS(i)} - R_{SRS(1)} \quad (4.6)$$

where the points are referred to the **SRS**, and with the subscript (i) is indicated the generic frame. From this relation it is obtained that $R_{S(1)} = 0$, and $R_{S(2)}$ represents the displacement of the reference point R . It is important to underline that, as consequence of this, R is used as reference of the trajectory computation replacing the usual centre of mass tracking. In fact the displacement of all vertices of the mock-up is consequence of the displacement of the reference point. Then another important assumption is that **SRS** is centered in R , and also this means that **SRS** and **RRS** have the same origin.

In the **RRS** the initial condition is known, in particular it consists of the position of the Chaser mock-up at the starter of the release, so it is possible

to define $P_{RRS} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{|RRS}$.

The system of equation to find the rotation matrix at the first step is defined as follows:

$$P_{SRS} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{|SRS} = f_{SRS} \cdot M_{ROT_{RRS \rightarrow SRS}} \cdot P_{RRS} \quad (4.7)$$

Note that P_S and P_R have different units, that means $\|P_{SRS}\| \neq \|P_{RRS}\|$, because of the scale of the **SRS** is unknown. For this reason is introduced the scale factor f_{screen} defined as:

$$f_{SRS} = \frac{\|P_{SRS}\|}{\|P_{RRS}\|} \quad (4.8)$$

The rotation matrix is defined as a matrix 3 – 2 – 1:

$$M_{ROT} = M_z(\gamma)M_y(\beta)M_x(\alpha) \quad (4.9)$$

$$M_{ROT} = \begin{Bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{Bmatrix} \quad (4.10)$$

Analysing the Equation 4.7 the system of equation consist of three equations in four unknowns, that are the three angle of the rotation matrix, γ , β , α , and the third coordinate of the **SRS**, z_S . To complete the system in order to find the solutions it is necessary to use at least two vertices of the mock-up. In particular, using the vertices A , B , the system is based on six equations, while the unknowns are: γ , β , α , $z_{S,A}$, $z_{S,B}$. Interest is taken in the three Euler's angles derived by the rotational matrix, because their meaning is the rotations necessary to switch from **RRS** to **SRS**.

Once the rotation matrix $M_{ROT_{RRS \rightarrow SRS}}$ is known, from the propriety of the rotational matrix it is possible to define $M_{ROT_{SRS \rightarrow RRS}}$ as the inverse of the matrix $M_{ROT_{RRS \rightarrow SRS}}$:

$$M_{ROT_{SRS \rightarrow RRS}} = M'_{ROT_{RRS \rightarrow SRS}} \quad (4.11)$$

Finally the system of equation to find P_R in every frame and constructs the trajectory of the mock-up could be defined. Starting from:

$$P_{RRS} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}_{|RRS} = f_{real} \cdot M_{ROT_{SRS \rightarrow RRS}} \cdot P_{SRS} = \frac{1}{f_{SRS}} \cdot M'_{ROT_{RRS \rightarrow SRS}} \cdot P_{SRS} \quad (4.12)$$

noted that it consists of three equations in four unknowns that are $[x, y, z]_R$ and z_S . Adding another vertice it is obtained a system of six equations in eight unknowns, once again unresolvable. This is repeated considering three vertices,

R , A and C , obtaining a system of nine equations and twelve unknowns. So it is necessary consider three more equations to close the system. To do this the geometrical properties mentioned before are used. In particular:

$$\|A_{RRS} - R_{RRS}\| = \|AR\| \quad (4.13)$$

$$\|C_{RRS} - R_{RRS}\| = \|CR\| \quad (4.14)$$

$$(A_{RRS} - R_{RRS}) \perp (C_{RRS} - R_{RRS}) \quad (4.15)$$

all three equations do not add unknowns to the system. In conclusion the system of equation defined consists of twelve equations and twelve unknowns.

Below is the complete system to solve:

$$\left\{ \begin{array}{l} R_{RRS} = \frac{\|R_{RRS}\|}{\|R_{SRS}\|} M_{ROT_{SRS \rightarrow RRS}} R_{SRS}, \quad 3 \text{ eq. and } 4 \text{ unkn.} \\ A_{RRS} = \frac{\|A_{RRS}\|}{\|A_{SRS}\|} M_{ROT_{SRS \rightarrow RRS}} R_{SRS}, \quad 3 \text{ eq. and } 4 \text{ unkn.} \\ C_{RRS} = \frac{\|C_{RRS}\|}{\|C_{SRS}\|} M_{ROT_{SRS \rightarrow RRS}} R_{SRS}, \quad 3 \text{ eq. and } 4 \text{ unkn.} \\ \|A_{RRS} - R_{RRS}\| = \|AR\| \\ \|C_{RRS} - R_{RRS}\| = \|CR\| \\ (A_{RRS} - R_{RRS}) \cdot (C_{RRS} - R_{RRS}) = 0 \end{array} \right. \quad (4.16)$$

This is iterated for every frame to get the trajectory computation.

The last step is to compute the center of mass of the mock-up and its attitude from the obtained data of the system of equation. In particular, referring to the geometry, the center of mass is obtained as half distance between the vertice B and the reference point R :

$$CM = \left\{ \begin{array}{l} x \\ y \\ z \end{array} \right\}_{|CM} = \frac{B + R}{2} \quad (4.17)$$

Known the positions of the vertices of the mock-up, their direction is computed as:

$$\left\{ \begin{array}{l} R \\ A \\ B \\ C \end{array} \right\}_i = M_{ROT} \cdot \left\{ \begin{array}{l} R \\ A \\ B \\ C \end{array} \right\}_1 \quad (4.18)$$

where the unknowns are the Euler's angles γ , β , α .

To conclude the analysis, in the graphs of Figure 4.14 the relative trajectory of the Chaser mock-up are shown with the respect to the Target in all the six Degrees of Freedom, and the comparison between the analysis obtained by the sensor data and by the camera recording post processing can be conducted. In particular it can be underlined that the camera recording analysis has given coherent results with the respect to the data from sensors fusion.

Important considerations regarding the performance of the software can be provided. In particular a great influence on the Medium Level Software is represented by the operation of the Pi Camera; in fact it is clear from the resulting errors between the two analysis that the Camera is out of focus with a face to face distance above 150 mm. The error become lower under this threshold, in which the camera focusing improves, and as consequence the Chaser mock-up has a clear vision of the Apriltag of the Target. The choice of having the focus at 150 mm has been made so that the last commands were more precise due to an increased precision in the sensors and camera readings. In fact, the focus is translated in a better control in position by the software about the translation along the x , y and z -axis. Another layer of perception is given by the data provided by the ToF when the mock-ups are very close. The ToF are also useful when really close to each other because they can overcome possible errors in calculation of distances due to a partial eclipse of the Apriltags by to the docking interfaces.

On the other hand, the attitude keep a constant error under the threshold too. In fact the roll, pitch and yaw are affected by the computation of the software and its authority. In particular the computation of the attitude is based on twisting mode estimation, and this requires the constant vision of the single reference marker, each one is related to an angular relative position of the Chaser mock-up with the respect to the Target. With the approaching of the first one to the second one, the camera has not a good vision of the single marker due to shaded areas caused by the docking port. A constant integration error propagates in the computation of the attitude, and this justifies the disagreement with the data reconstructed by video recording. However, both attitude and trajectory are inside the expected errors and design margins. This leads to the conclusion that the localization system has performed as expected in the plane environment. Moreover, since the effect of the commands seem to replicate the real behaviour of the mock-up, also the Medium-Level showed a discrete accordance between its behaviour in the laboratory and how it performed during the flights.

In conclusion, in the **Parabola #27** the docking manoeuvre was accomplished with an acceptable final alignment between the CubeSats. Pitch angle has been recovered from initial misalignment. Subsequently, the impact between the docking ports, a state of good stability has been registered, and the contact between the surfaces of the probe and drogue is soft enough to guarantee the sliding process. However, the collision pushed back the Target causing a collision with the external structure. This was a major problem for the ERMES experiment, that highly affected the success of all the manoeuvres.

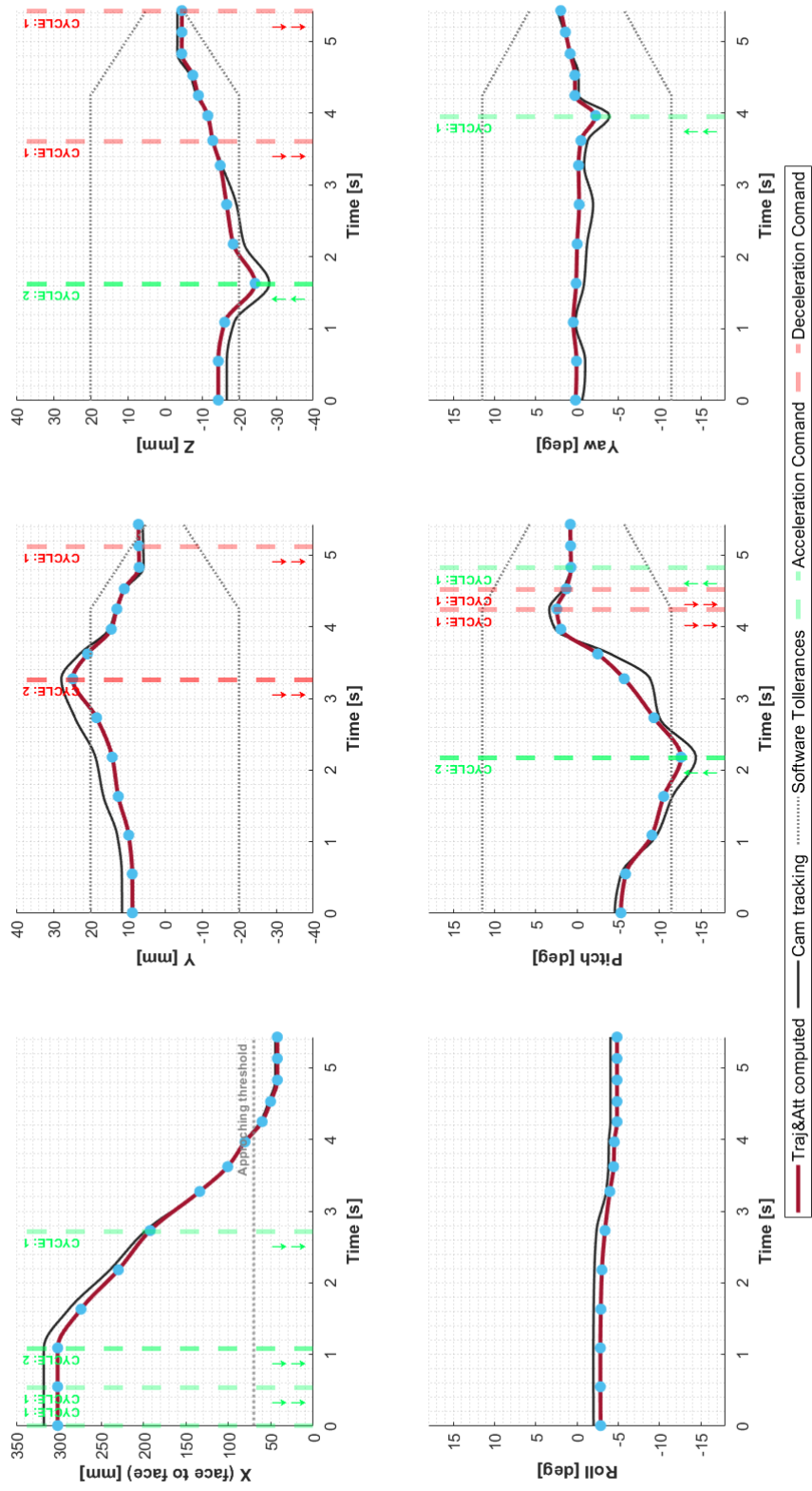


Figure 4.14: Output graphs from the analysis: relative trajectory of the Chaser with the respect to the Target

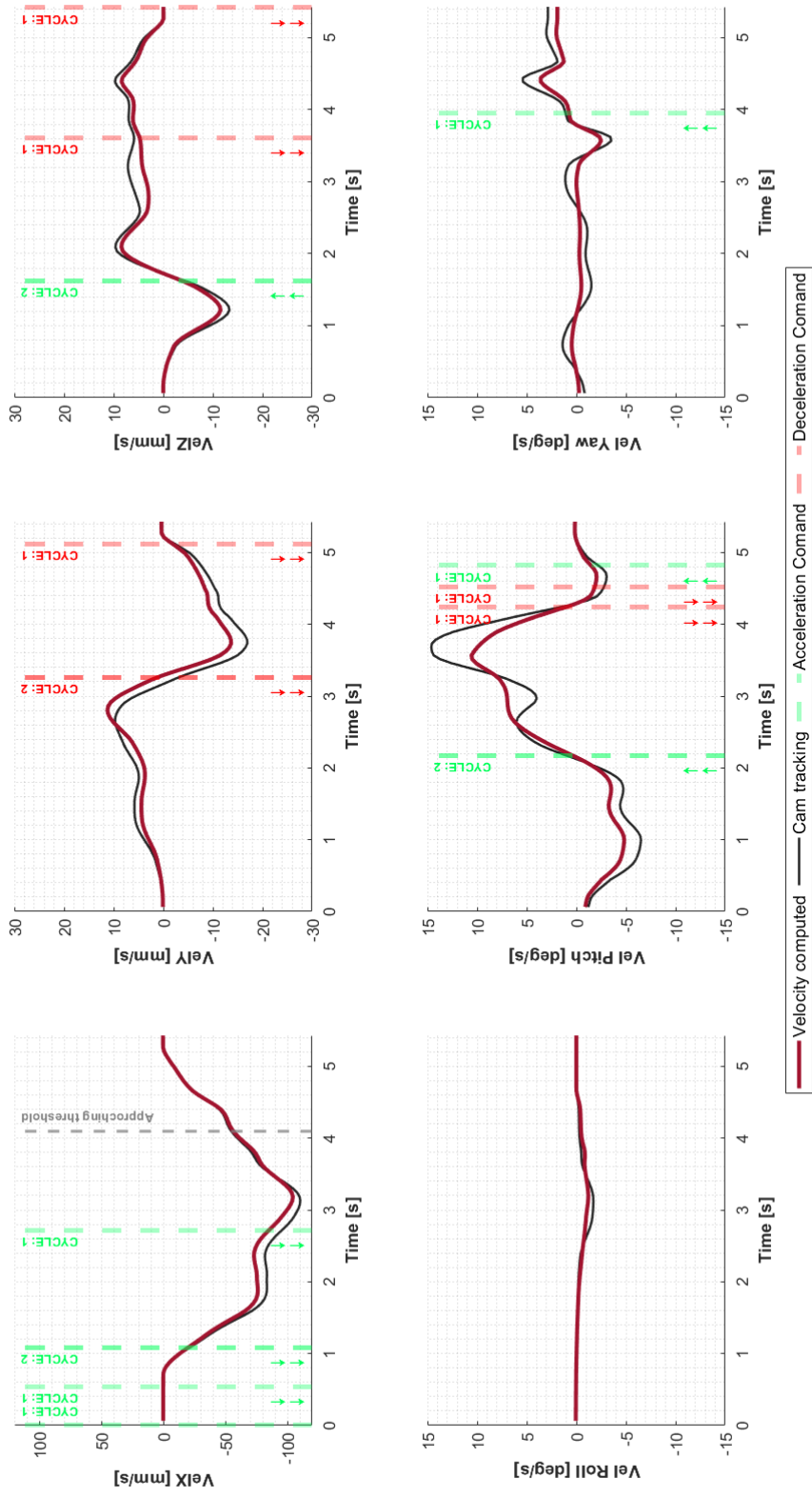


Figure 4.15: Output graphs from the analysis: relative velocity of the Chaser with the respect to the Target

Conclusions

This work examines in detail the propulsive concept of the Chaser mock-up, starting from a general overview of the design, the choices in terms of components, their sizing and testing. Then the important steps of integration of the propulsive system with the software and moreover the safety management for the test environment are presented. Finally the results of the flight campaign are analysed. These data are not entirely negative or positive, and in particular they give the opportunity to make considerations related what went well and what went wrong during the manoeuvres. Infact the experiment worked as intended, and the propulsive system of the active mock-up demonstrated everytime good authorities to complete the manoeuvre in such that microgravity conditions, but at the same time ERMES had to face some problems that partially impacted to the success.

As mentioned, in such cases it is difficult to accurately rate the results as positive or negative, therefore the experiment must be deconstructed to identify exactly the strengths of the design and the errors it had. For this reason to make interesting conclusions the objectives of the experiment should be divided into small and simpler objectives so that it is possible to evaluate the grade of success of ERMES. In particular for this work the keypoints related the propulsive system design, safety, test and integration are taken into account. Following a system engineer approach, three macrocategories can therefore be identified and for each one the most relevant requirements of success of the system are listed:

- *Design and Performance* [DP]: it concerns all the tasks of the design step of the project, including the manufacturing, assembly and testing, and also the performance of the realised system. All these activities carried out prior to the campaign.
 - 1 Design, manufacturing and assembly of a cold gas propulsive system for the Chaser mock-up accordingly to Novespace Guidelines, based on **FUNCTIONAL REQUIREMENTS**;
 - 2 Assembly, integration and and testing of the selected commercial components of the cold gas propulsive system, based on **PERFORMANCE REQUIREMENTS**;
 - 3 Assembly, integration and testing the connections between the selected commercial components of the cold gas propulsive system, based on **PERFORMANCE REQUIREMENTS**;
 - 4 Integration and testing of a system of control for the electrovalves, based on **OPERATION REQUIREMENTS**;
 - 5 Integration and testing of the complete software in order to define the authority of the cold gas propulsive system, based on **OPERATION REQUIREMENTS**;
 - 6 Testing of the entire system on the ground, based on **OPERATION REQUIREMENTS**.

- *Safety and Management* [SM]: it concerns all the tasks relative to the risks and hazards management.
 - 1 Preparation of a complete and precise risk and hazard analysis accordingly to Novespace Guidelines, based on **FUNCTIONAL REQUIREMENTS**;
 - 2 Tuning of safety regimes of the cold gas propulsive system, based on **PERFORMANCE REQUIREMENTS**;
 - 3 Implementation of safety components of the cold gas propulsive system, based on **PERFORMANCE REQUIREMENTS**;
 - 4 Testing of the pressurized system in standard-stressed operation conditions, based on **PERFORMANCE REQUIREMENTS**;
 - 5 Testing of the pressurized system in over-stressed operation conditions, based on **PERFORMANCE REQUIREMENTS**;
 - 6 Certify the class of security of the components of the cold gas propulsive system, based on **OPERATION REQUIREMENTS**.
- *Operation and Post-processing* [OP]: it concerns all the tasks that characterized the preparation for the campaign and the flights themselves. Moreover it includes the post flight analysis and the related discussion of results in relation to the benchtest data collection.
 - 1 Acceptance of the system accordingly to Novespace Guidelines, based on **DESIGN REQUIREMENTS**;
 - 2 Performing of the procedures of use of the system, based on **OPERATION REQUIREMENTS**;
 - 3 Performing of the trajectory computed in microgravity by the system, based on **OPERATION REQUIREMENTS**;
 - 4 Collection of the sensors data and camera recording, based on **PERFORMANCE REQUIREMENTS**;
 - 5 Detection of the docking accomplishment, based on **PERFORMANCE REQUIREMENTS**;
 - 6 Implementation of the data for the post-flights analysis, based on **PERFORMANCE REQUIREMENTS**.

As reported in the Table 4.2, the ERMES experiment achieved the 85% of all the requirements of success of the propulsive system. The only objective that was not fully achieved was the one regarding the accomplished docking manoeuvre. In fact it has been considered partially achieved because the magnetic connection has not been established, but the precision needed to dock has been confirmed to have been reached in the parabola analysed. Important results for the propulsive system are the success in all the design and safety tasks related to the functional and performance requirements. The main strengths highlighted by the propulsive system are its easy construction and availability of materials, that helps to guarantee the perfect operation under the working

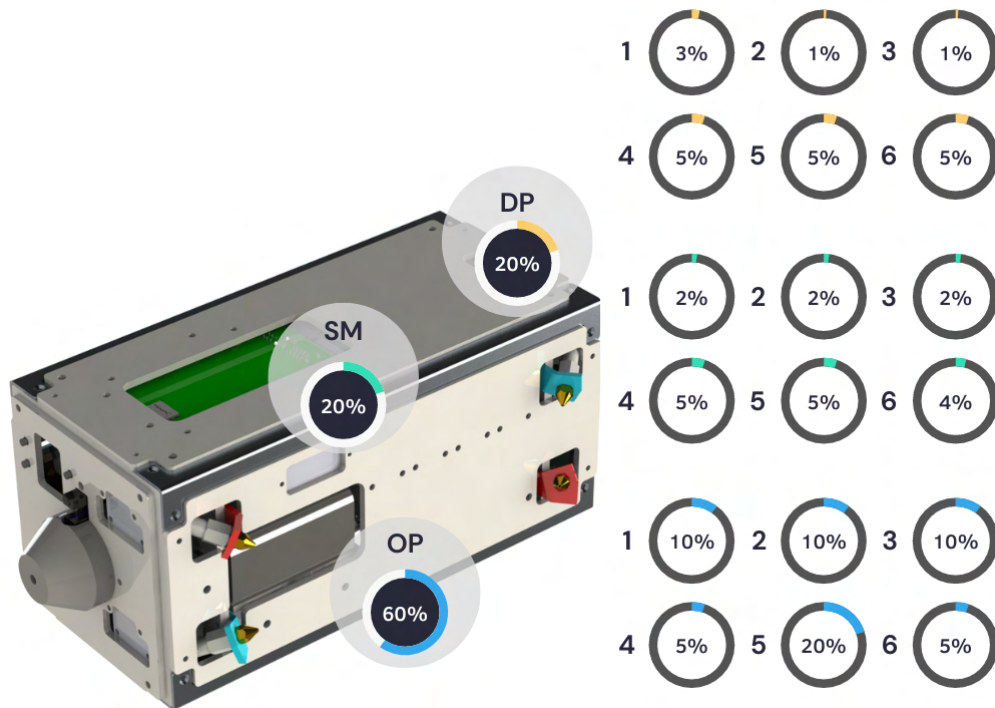


Figure 4.16: Weight for global success given to each objective

conditions of the experiment. Then the control of the actuators, with linear PWM trend, makes it easy to manage the propulsive power of the system, allowing coverage of the entire operating range under the working conditions of the experiment. From a safety point of view, the system has never encountered problems, thanks to its low operating pressures. Finally the implementation of the system with the software navigator control gives good feedback in terms of choice of the commands; in particular the designed low cycle implementation covers the small and simple corrections required in the operating range under the working conditions of the experiment.

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 and the high level of grade of success, the ERMES experiment can be considered a partial success. Moreover, one of the interesting features of ERMES is the simplicity of the system, that is one of the key points of its designing phase. Nevertheless, the simplicity of the entire system, in terms of software and general design, has shown still to work efficiently and accordingly to the tests performed in the laboratory. Finally, it can be stated that ERMES, as a technological demonstrator, has been able to confirm the validity of the integration of different technologies and subsystems in order to perform such complicated manoeuvre. In fact, all the subsystems involved in both Target and Chaser worked well together and performed as planned without showing any major flaws.

N.	Category	Achieved (%)	Weight (%)
1	DP	3	3
2	DP	1	1
3	DP	1	1
4	DP	5	5
5	DP	5	5
6	DP	5	5
		20	20
1	SM	2	2
2	SM	2	2
3	SM	2	2
4	SM	5	5
5	SM	5	5
6	SM	4	4
		20	20
1	OP	10	10
2	OP	10	10
3	OP	10	10
4	OP	5	5
5	OP	5	20
6	OP	5	5
		45	60
TOTAL		85	100

Table 4.2: Achieved requirements of success of the propulsive system

Finally the experience related to this project has been amazing and rewarding: after two years of hard work, through design, prototyping and testing, the three days of Parabolic Flight have given all the results of the process. Working in such a professional environment, alongside researchers, technicians and ESA and Novespace engineers, made this experience invaluable and unforgettable.

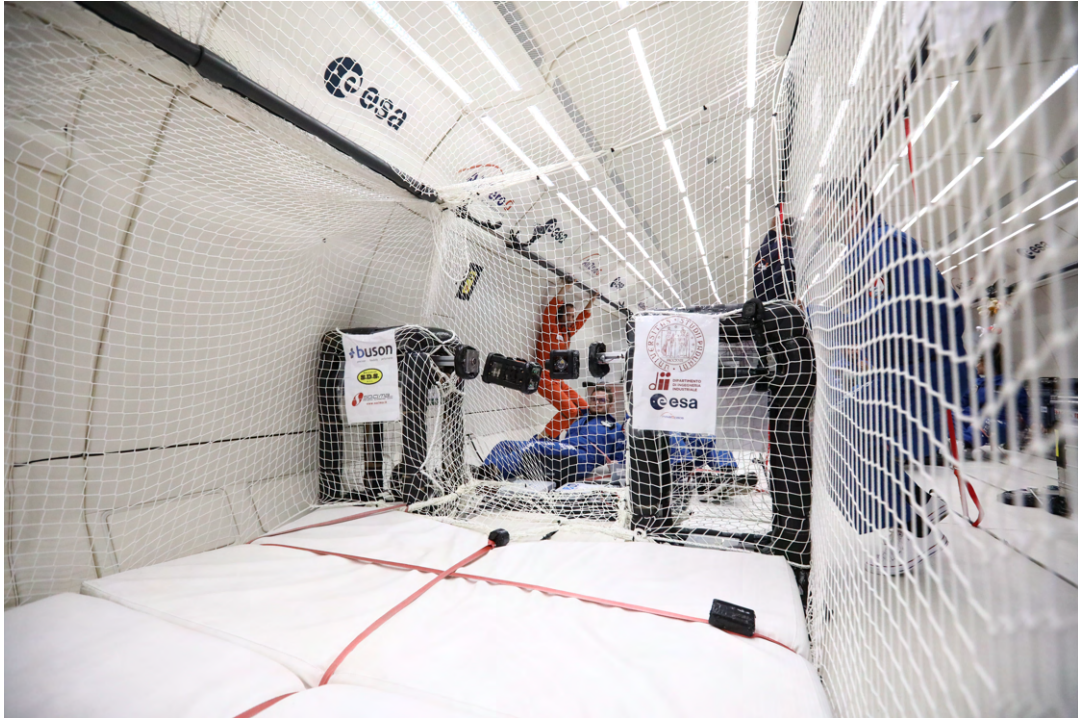


Figure 4.17: ERMES Experiment operating in microgravity - ©Novespace

Acknowledgments

I wish to thank the ESA Education Office, that supported the ERMES project as part of the Fly Your Thesis! Programme 2022, in particular our supervisors Mr. Nigel Savage, Jeffrey Gorissen and Felix Scharnhölz for all the help both prior to and during the campaign. Moreover, I would like to thank Novespace and its staff that owns the Airbus A310 Zero G and operates the flights, in particular our supervisor Alexis Jolion, who has always shown himself kind and helpful and has proven to have interest in our project since the beginning.

Then I wish to thank Università degli Studi di Padova and in particular Dipartimento di Ingegneria Industriale for believing in our student project and for funding it. In particular, I would like to thank firstly our endorsing professors Prof. Francesco Branz and Prof. Alessandro Francesconi for their availability and mentorship throughout the entire journey started in November 2020. In addition, I would like to thank Ing. Lorenzo Olivieri for his constant presence, friendliness and help throughout all the phases of the project. Last but not least I wish to thank my colleagues and also friends Alessandro Bortotto, Mattia Dignani, Fabio Mattiazzi, Nicola Pozzato, Federico Favotto and Miroljub Mihailovic, with whom I have shared good times but also difficult times, with the sole goal of making our ambitions a reality.



Bibliography

- [1] Automated Rendezvous and Docking of Spacecraft - Wigbert Fehse, Cambridge Aerospace Series 16
- [2] P. Amadiou and J. Heloret, "The automated transfer vehicle, Air & Space Europe, vol. 1, no. 1, pp. 76–80, 1999.
- [3] J. Pavlich et al. Autonomous satellite docking system, 2001.
- [4] SpaceX, 24 April 2021,
< <https://www.spacex.com/updates/crew-2-mission/index.html> >
- [5] A. Fejzic et al. Results of spheres microgravity autonomous docking experiments in the presence of anomalies, 2008.
- [6] J. Bowen et al. Cubesat proximity operations demonstration (cpod) mission update, in 2015 IEEE Aerospace Conference, 2015.
- [7] Lorenzo Fluckiger et al.: Astrobeer Robot Software: A Modern Software System for Space, iSAIRAS (International Symposium on Artificial Intelligence, Robotics and Automation in Space), Madrid, 4 June 2018
- [8] Petrillo D et al.: Flexible Electromagnetic Leash Docking system (FELDs) experiment from design to microgravity testing, 66th IAC, Jerusalem, 12-16 October 2015.
- [9] Barbetta et al. AUTONOMOUS RENDEZVOUS, CONTROL AND DOCKING EXPERIMENT-REFLIGHT 2. ESA/CNES Small Satellites Systems and Services Symposium, 2014
- [10] Duzzi, et al. Electromagnetic position and attitude control for PACMAN experiment. In Proceedings of the 10th International ESA Conference on Guidance, Navigation & Control Systems, 2017.
- [11] F. Branz, L. Olivieri, F. Sansone, and A. Francesconi, "Miniature docking mechanism for cubesats," Acta Astronautica, 2020.
- [12] Lorenzo Olivieri and Alessandro Francesconi. Design and test of a semian-drogynous docking mechanism for small satellites. Acta Astronautica, 122:219-230, 2016.
- [13] L Olivieri, A Antonello, M Duzzi, F Sansone, and A Francesconi. Semi-androgynous multifunctional interface for expandable space structures. In 66Th International Astronautical Congress, IAC-15.
- [14] ESA: "Fly your thesis! programme". Available at:
http://www.esa.int/Education/FlyYourThesis/FlyYourThesis!_programme
- [15] UG-2019-01-User Guide-EN, Novespace

- [16] Fly Your Thesis! 2017 Experiment Report - PACMAN experiment
- [17] M. Krogus, A. Haggemiller, and E. Olson, “Flexible layouts for fiducial tags,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2019.
- [18] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 3400–3407, IEEE, May 2011.
- [19] J. Wang and E. Olson, “AprilTag 2: Efficient and robust fiducial detection,” in Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2016.
- [20] R. E. Kalman, “A new approach to linear filtering and prediction problems,” Transactions of the ASME-Journal of Basic Engineering, vol. 82, no. Series D, pp. 35–45, 1960.
- [21] Arduino, “Secrets of Arduino PWM.”
[https : //docs.arduino.cc/tutorials/generic/secrets - of - arduino - pwm](https://docs.arduino.cc/tutorials/generic/secrets-of-arduino-pwm), 2022.
- [22] *[https : //optitrack.com/applications/robotics/](https://optitrack.com/applications/robotics/)*

