UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Centro di Ateneo di Studi ed Attività Spaziali "Giuseppe Colombo" (CISAS)

SCUOLA DI DOTTORATO DI RICERCA IN: SCIENZE TECNOLOGIE E MISURE SPAZIALI

INDIRIZZO: MISURE MECCANICHE PER L'INGEGNERIA E LO SPAZIO

CICLO XXV

DEVELOPMENT AND TESTING OF MODEL PREDICTIVE CONTROL STRATEGIES FOR SPACECRAFT FORMATION FLYING

Direttore della Scuola: Ch.mo Prof. Giampiero Naletto

Coordinatore d'indirizzo: Ch.mo Prof. Stefano Debei

Supervisore: Ch.mo Prof. Enrico Lorenzini

Co-Supervisore: Ing. Gianfranco Sechi

Dottorando: Andrea Valmorbida

"Simplicity is the ultimate sophistication"

"La semplicità è l'ultima forma di sofisticazione"

Leonardo da Vinci

Alla mia Famiglia e agli Amici

Abstract

Satellite Formation Flying (SFF) is a key technology for several future missions, since, with respect to a single spacecraft, it allows better performances, new capabilities, more flexibility and robustness to failure and cost reduction. Despite these benefits, however, this new concept poses several significant design challenges and requires new technologies. The Guidance, Navigation and Control (GNC) system is a key element in the SFF concept since it must be reliable in coordinating all the satellites flying in formation during each mission phase, guaranteeing formation integrity and preventing from formation evaporation, and, at the same time, efficient in using the limited on board resources. Model Predictive Control (MPC), also referred to as Receding Horizon Control, is a modern optimal control technique that seems to be suitable for these purposes because of its three main features: model-based control scheme, constraints handling ability and replanning nature.

The final aim of my Ph.D. activities was to develop and test MPC strategies for SFF applications. This task was accomplished by means of both computer simulations and experimental tests conducted on both the MIT Synchronized Position Hold Engage & Reorient Experimental Satellites (SPHERES) testbed and the SFF Hardware Simulator under development at the Center of Studies and Activities for Space "Giuseppe Colombo" (CISAS), University of Padova.

MPC capabilities were first tested in computer simulations in carrying out a formation acquisition maneuver for two space vehicles, taking into account two scenarios: a Leader-Follower (LF) formation and Projected Circular Orbit (PCO) formation. The performances of the MPC-based controller were compared with those of a Linear Quadratic Regulator (LQR) based controller in the presence of active constraints on the maximum control acceleration, evaluating also the effects of the gravitational harmonics J_2 and J_3 and atmospheric drag perturbations on the proposed maneuvers. Simulation results of both scenarios showed that, with similar performances in tracking the same reference state trajectory in terms of settling time, the MPC controller is more efficient (less Δv requirement) than the LQR controller also in the perturbed cases, allowing a Δv requirement reduction by 40 % in the LF formation scenario and by 30 % in the PCO formation scenario.

The next activity concerned the development of some guidance and control strategies for a Collision-Avoidance scenario in which a free-flying chief spacecraft follows temporary off-nominal

conditions and a controlled deputy spacecraft performs a collision avoidance maneuver. The proposed strategy consists on a first Separation Guidance that, using a computationally simple, deterministic and closed-form algorithm, takes charge of avoiding a predicted collision. When some safe conditions on the relative state vector (position and velocity) are met, a subsequent Nominal Guidance takes over. Genetic Algorithms are used to compute a pair of reference state trajectories in order to place the deputy spacecraft in a bounded safe or "parking" trajectory, while minimizing the propellant consumption and avoiding the formation evaporation. The performances of a LQR and a MPC in tracking these reference trajectories were compared, showing how a MPC controller can reduces the total Δv requirement by $5 - 10\,\%$ with respect to a LQR controller.

MPC capabilities were then evaluated on the MIT SPHERES testbed in simulating the close-proximity phase of the rendez-vous and capture maneuver for the Mars Orbital Sample Return (MOSR) scenario. Better performances of MPC with respect to PD in executing this maneuver were confirmed both in a Matlab simulator and in the MIT SPHERES software simulator, with a total Δv requirement reduction by $10 - 15\,\%$. The proposed MPC control strategy was then tested using the SPHERES Flat Floor facility at the MIT Space System Laboratory.

The last part of my research activities was devoted to the SFF Hardware Simulator of the University of Padova. My contributions to this project dealt with: (a) conclusion of the designing, building and testing of the five main subsystems of the hardware simulator; (b) software development for the hardware simulator and its Matlab software simulator; (c) preparatory experimental activities aimed at characterizing the thrust force performed by the on board thrusters and estimating the hardware simulator inertia properties; and (d) test of attitude control maneuvers with the use of predictive controllers.

In particular, three main tests were carried out with the hardware simulator moving at one degree of freedom about the yaw axis. The first one aimed at tuning a Kalman Filter to properly estimate the yaw axis angular velocity using a double-integrator as dynamic model and angular position measurements provided by the yaw quadrature encoder. With the use of a simple Kalman Filter, the yaw angular position and velocity could be estimated with an error less than $0.1\,^\circ$ and $0.1^\circ/s$, respectively. In the second test, an explicit MPC was used to perform a 170° slew maneuver of the hardware simulator attitude module about the yaw axis. The final target angular position was reached with an error less than 0.5° in $20\,s$. In the third test, a 3 degrees of freedom attitude reference trajectory was first computed using pseudospectral optimization methods for a repointing maneuver with active constraints on the attitude trajectory. The state trajectory was then projected along the satellite z-Body axis and tracked in the hardware simulator using an explicit MPC. Experimental results showed that with an explicit MPC the reference trajectories can be tracked with an error less that 1.5° for the angular position and less than $1^\circ/s$ for the angular

velocity, both in dynamic conditions. The final target state was reached with an error less than the estimation accuracy.

The SFF Hardware Simulator is a ground-based testbed for the development and verification of GNC algorithms that in the present configuration allows the development and testing of advanced controls for attitude motion and in its final form will enable the derivation of control strategies for Formation Flight and Automated Rendezvous and Docking.

Sommario

Il volo in formazione tra satelliti è una tecnologia fondamentale per molte missioni future, poiché, rispetto ad un satellite singolo, permette migliori prestazioni, nuove capacità, maggiore flessibilità e robustezza alle avarie e riduzione dei costi. Nonostante questi benefici, tuttavia, questo nuovo concetto pone svariate sfide progettuali e richiede nuove tecnologie. Il sistema di Guida, Navigazione e Controllo (GNC) è un elemento chiave per il volo in formazione, poiché deve essere affidabile nel coordinare tutti i satelliti che volano in formazione durante ciascuna fase della missione, garantendo l'integrità della formazione e prevenendo l'evaporazione della stessa, e, allo stesso tempo, efficiente nell'utilizzo delle limitate risorse di bordo. Il *Model Predictive Control* (MPC), chiamato anche *Receding Horizon Control*, è una moderna tecnica di controllo ottimo che sembra essere adeguata a queste finalità per le sue tre principali caratteristiche: schema di controllo basato su modello, abilità nel gestire i vincoli e ripianificazione.

L'obbiettivo finale delle mie attività di dottorato è stato quello di sviluppare e testare strategie di controllo MPC per applicazioni di volo in formazione. Questo obiettivo è stato raggiunto sia mediante simulazioni al computer sia attraverso test sperimentali condotti e sul sistema *Synchronized Position Hold Engage & Reorient Experimental Satellites* (SPHERES) del MIT e sul simulatore hardware per volo in formazione che è in fase di sviluppo al Centro di Ateneo di Studi ed Attività Spaziali "Giuseppe Colombo" (CISAS) dell'Università di Padova.

Le capacità del controllo MPC sono state dapprima testate mediante simulazioni al computer nell'eseguire una manovra di acquisizione di formazione per due veicoli spaziali, prendendo in considerazione due scenari: una formazione *Leader-Follower* (LF) e una formazione *Projected Circular Orbit* (PCO). Le prestazioni del controllore MPC sono state confrontate con quelle di un controllore LQR in presenza di vincoli attivi sulla massima accelerazione di controllo, valutando inoltre gli effetti perturbativi delle armoniche gravitazionali J_2 e J_3 e dell'attrito atmosferico sulle manovre proposte. I risultati delle simulazioni per entrambi gli scenari hanno mostrato che, per simili prestazioni nel seguire la stessa traiettoria di stato di riferimento in termini di tempo di assestamento, il controllore MPC è più efficiente (minor requisito di Δv) rispetto al controllore LQR anche nei casi con perturbazioni, permettendo una riduzione del requisito di Δv totale del 40 % nello scenario

LF e del 30 % in quello PCO.

L'attività successiva ha riguardato lo sviluppo di alcune strategie di guida e controllo per uno scenario di *Collision-Avoidance* in cui un satellite *chief* non controllato segue temporaneamente condizioni non nominali e un satellite controllato *deputy* esegue una manovra di anti-collisione. La strategia proposta consiste in una prima *Separation Guidance* che, utilizzando un algoritmo semplice, deterministico e in forma chiusa, ha lo scopo di evitare una collisione prevista. Quando vengono soddisfatte alcune condizioni di sicurezza sullo stato relativo (posizione e velocità), subentra una successiva *Nominal Guidance*. Gli Algoritmi Genetici sono usati per calcolare una coppia di traiettorie di stato di riferimento al fine di collocare il satellite *deputy* in una traiettoria chiusa "di parcheggio", minimizzando il consumo di carburante ed evitando l'evaporazione della formazione. Le prestazioni di un controllo LQR e di uno MPC nel seguire queste traiettorie di riferimento sono state messe a confronto, dimostrando come un controllo MPC può ridurre il requisito totale di Δv del $5 - 10\,\%$ rispetto ad un controllo LQR.

Le capacità del controllo MPC sono state valutate anche nel sistema SPHERES del MIT nel simulare la fase di prossimità della manovra di *rendez-vous and capture* per lo scenario *Mars Orbital Sample Return* (MOSR). Migliori prestazioni del controllo MPC rispetto al controllo PD nell'eseguire questa manovra sono state confermate sia in un simulatore Matlab che nel simulatore software di SPHERES del MIT, con una riduzione del requisito totale di Δv del $10 - 15\%$. La strategia di controllo MPC proposta è stata poi testata nella *SPHERES Flat Floor facility* presso lo *Space System Laboratory* del MIT.

L'ultima parte dell'attività di ricerca si è concentrata sul simulatore hardware per il volo in formazione dell'Università di Padova. Il mio contributo a questo progetto ha riguardato: (a) la conclusione delle fasi di progettazione, costruzione e test dei cinque principali sottosistemi del simulatore hardware; (b) lo sviluppo di software per il simulatore hardware e del suo simulatore software in Matlab; (c) alcune attività sperimentali preparatorie finalizzate a caratterizzare la spinta prodotta dai razzetti di bordo e stimare le proprietà d'inerzia del simulatore hardware; e (d) il test di manovre di controllo d'assetto con l'utilizzo del controllo predittivo.

In particolare, sono stati eseguiti tre principali test con il simulatore hardware in moto ad un grado di libertà attorno all'asse di *yaw*. Il primo test è stato finalizzato al *tuning* di un Filtro di Kalman per stimare in modo opportuno la velocità angolare di *yaw* usando un doppio integratore come modello dinamico e misure della posizione angolare fornite dall'encoder di *yaw*. Utilizzando un semplice Filtro di Kalman, è stato possibile stimare la posizione e la velocità angolare con un errore inferiore a $0.1°$ e $0.1°/s$, rispettivamente. Nel secondo test, è stato utilizzato un controllo MPC esplicito per eseguire una manovra di ri-orientazione di $170°$ del modulo d'assetto del simulatore hardware attorno all'asse di *yaw*. La posizione angolare obiettivo è stata raggiunta con un errore

inferiore a $0.5°$ in $20\,s$. Nel terzo test, una traiettoria d'asseto di riferimento è state dapprima calcolata utilizzando metodi di ottimizzazione *pseudospectral* per una manovra di ripuntamento con vincoli attivi sulla traiettoria di stato. La traiettoria di stato è stata poi proiettata lungo l'asse *z-Body* del satellite ed inseguita nel simulatore hardware utilizzando un controllo MPC esplicito. I risultati sperimentali hanno dimostrato che con un controllo predittivo esplicito le traiettorie di riferimento possono essere inseguite con un errore inferiore a $1.5°$ per la posizione angolare e inferiore a $1°/s$ per la velocità angolare, entrambi in condizioni dinamiche. Lo stato finale obiettivo è stato raggiunto con un errore inferiore all'accuratezza di stima.

Il Simulatore Meccanico per il volo in formazione costituisce un banco di prova per lo sviluppo e la verifica in laboratorio di algoritmi di GNC; nella configurazione attuale il simulatore permette lo sviluppo ed il test di controlli avanzati per il moto d'assetto, mentre nella sua configurazione finale consentirà di sviluppare strategie di controllo per *Formation Flight* e *Automated Rendezvous and Docking*.

Acknowledgements / *Ringraziamenti*

I would like to express my deepest gratitude to my supervisor Prof. Enrico Lorenzini for his continuous and strong guide all the time of my Ph.D. I would like to express my sincere gratitude to Ing. Gianfranco Sechi for his encouragement and invaluable suggestions during this work.

A sincere and very special thank to Prof. David Miller and Dr. Alvar Saenz-Otero for the possibility to spend a very formative research period at the MIT Space System Laboratory. A warm thank to Vishnu Jyothindran for out work on MOSR and for his help and patience in debugging my MPC code via Skype at thousand of miles away.

I would like to thank my colleagues and friends who worked with me during these years of Ph.D. and shared this experience: Denis, Francesca, Michele, Giulio, Andrea, Marco, Riccardo and all other guys of the Measures Team. My gratitude also to Alessio and Michele for their constant help with micro-controllers, and to Francesco and Mattia for their important contributions to the Satellite Formation Flying Hardware Simulator. Special thanks to the Measures Laboratory guys, Luca, Filippo and Paolo, and to the DII Mechanical Workshop guys for their practical support.

Un caro ringraziamento alla mia famiglia, Maurizio, Stefania e Davide, per avermi sempre sostenuto (e sopportato). Un caloroso grazie anche agli Amici di Casette e dintorni, agli Amici del Liceo e all' "Allegra Compagnia" con i quali condivido momenti di divertimento e allegria.

Andrea Valmorbida

Contents

List of Acronyms

ADCS	Attitude Determination and Control Subsystem
AM	Attitude Module
AR	Avoidance Region
CANG	Collision Avoidance Nominal Guidance
CASG	Collision Avoidance Separation Guidance
DOF	Degree(s) Of Freedom
DSS	Distributed Space System
ECI	Earth Centered Inertial
EMFF	Electromagnetic Formation Flight
ERT	Evasive Relative Trajectory
FFC	Formation Flying Control
FFG	Formation Flying Guidance
FOV	Field Of View
GAs	Genetic Algorithms
GCO	General Circular Orbit
GNC	Guidance Navigation and Control
GVEs	Gauss Variational Equations
HCW	Hill-Clohessy-Wiltshire
IC	Integrated Circuit

IR	Incremental Ratio
KF	Kalman Filter
LF	Leader-Follower
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
LVLH	Local-Vertical Local-Horizontal
MIMO	Multiple-Input Multiple-Output
MOSR	Mars Orbital Sample Return
MPC	Model Predictive Control
NB	Nominal Boundary
NL	Non-Linear
OOP	Object Oriented Programming
PCO	Projected Circular Orbit
PD	Proportional Derivative
PID	Proportional Integral Derivative
PRO	Passive Relative Orbit
PWA	Piece Wise Affine
PWC	Piece Wise Constant
PWM	Pulse Width Modulation
QP	Quadratic Programming
mpQP	multi-parametric Quadratic Program
RAAN	Right Ascension of the Ascending Node
RF	Radio Frequency

RF	Reference Frame
RHC	Receding Horizon Control
SAR	Synthetic Aperture Radar
SFF	Satellite Formation Flying
SRT	Safe Relative Trajectory
STM	State Transition Matrix
TM	Translational Module
US	Ultra-Sound

Chapter 1

INTRODUCTION

The foundations of satellite formation flying can be traced to George William Hill, who published in 1878 a set of equations which characterized the motion of the Moon with respect to the Earth [1]. In the 1960s, W. Clohessy and R. Wiltshire applied Hill's equations to the case of the rendezvous and docking of two orbital vehicles [2], leading to the development of the Hill-Clohessy-Wiltshire (HCW) equations of relative orbital motion. The HCW dynamics played a critical role in the Apollo missions where, in Low Earth Orbit (LEO), the main Command and Service Module (CSM) would disengage from the final stage of the Saturn V launch vehicle, turn around and dock with the Lunar Excursion Module (LEM). For the first time, two space vehicles were maneuvered to adjust their relative orbit rather than their absolute motion with respect to the Earth. The same technique is expected to be employed with the Orion spacecraft in the Constellation program by NASA.

Research into satellite formation flying has gained in popularity during the 1990's with the idea that missions based on formations of multiple cooperative small satellites offer a more robust, flexible, performing and cost-effective alternative to expensive and risky single satellite missions. From the first artificial satellite Sputnik 1 to the International Space Station (ISS), the largest man-made satellite in Earth orbit, requirements for spacecrafts have increased extensively and dramatically. In order to satisfy multi-task requirements, complex large spacecrafts are designed and manufactured with various kinds of payloads and instruments on board. Consequently, single large satellite is common in missions such as scientific observation, weather monitoring, global navigation and civil relay communication. At the same time, the cost of design and developing single large satellites has become a critical problem facing space organizers and engineers, as few nations in the words could afford to build rockets and launch satellites. Moreover, with the increasing complexity, reliability of large spacecraft decreases and it is possible that minor mistakes lead to the failure of the whole mission and causes a catastrophe, which was the case with space shuttle Challenger (1986). On one hand, risk rises with increasing of the size of the spacecraft and with

the overlap of orbits[1]; on the other hand, in mission operation large spacecraft could generate more space debris.

To be precise, formation flying means two or more satellites whose positions and attitudes are mutually controlled and permanently assessed, so that the distributed payload over these satellites is equivalent to a very large dimension, single instrument in space. Formation flying relies on sensors and actuators able to continually assess and maintain relative distances and attitudes with very high accuracy, and a system architecture ensuring that instrument performance is achieved by each individual satellite in the loop. Some main advantages of a satellite formation with respect to a single spacecraft are listed below.

- *Better performances.* A single satellite is indeed limited by both the launchers capacity to deliver the whole satellite mass into a proper orbit, and the launcher fairing's available volume. European launchers are limited to payloads of 2 - 7 tons, depending on the final orbit, and the available volume inside their fairing are around 4 meter in diameter and 5 - 7.5 meter in height. This constrain the capabilities of telescopes, interferometers, coronographs, etc., since they are strongly related to focal length, mirror surfaces, aperture sizes and mass. Overcoming these mass and volume constraints, a satellite formation can has significant better performances through proper instrument and task distribution.

- *New capabilities.* Several cooperating space vehicles can solve assignments which are more difficult and expensive, or even impossible to do with a single spacecraft; e.g. the capability of obtaining multiple simultaneous measurements along a large baseline.

- *Flexibility.* Reconfiguring the relative positions of the constituent satellites in situ would permit a satellite formation to engage in multiple mission objectives.

- *Robustness to failures.* Using multiple cooperating satellites flying in formation, redundancy is added to the system and the risk of total mission failure is minimized. The system would exhibit gradual performance degradation as individual satellites fail and tasks are redistributed, rather than the total mission termination due to failures on a single spacecraft. By adding replacement satellites, the formation could be returned to full mission specifications.

- *Cost reduction.* Simpler and smaller satellites are cheaper to produce (series production); furthermore, launch costs may be reduced since the spacecraft of the formation may be distributed on more inexpensive launch vehicles.

[1] e.g. the collision between the Iridium Satellite LLC-operated satellite of the USA and Russian Cosmos-2251 military satellite on 11th of February 2009, at about 17:00 GMT.

Despite the benefits, however, the formation flying concept poses several significant design challenges and requires new technologies. Some of these challenges are listed below.

- *Guidance Navigation and Control system complexity.* A high level of formation autonomy results in stringent requirements on the Guidance Navigation and Control (GNC) system: it has indeed to menage and coordinate all the satellites flying in formation during each mission phase.

- *New Sensors and Actuators.* Mastering formation flying missions requires the development of specific technologies beyond the present state-of-the-art in fields of relative sensors and actuators. Direct sensing of relative range and bearing is essential to enable position and attitude control of each spacecraft in formation, especially in deep-space scenarios, where the benefit of the GPS is not available. Such relative sensors need to have an operating range from few meters to at least few kilometers. Furthermore, precision spacecraft positioning and formation maneuvers require the development of very precisely controllable microthrusters.

- *Propellant consumption.* Uncontrolled formations quickly evaporate or lose their performances because of a number of secular differential orbital perturbations, including gravitational forces produced by the non-spherical shape of the Earth, differential atmospheric drag, solar radiation pressure, etc. To compensate for these effects, the satellites must thrust at regular intervals, thus leading to need a more complex GNC system. The GNC system has also to be more efficient as possible because the life of the mission is strongly dependent on the fuel availability.

- *Inter-satellite communication.* Formation of spacecraft need an higher level of inter-satellite communication, especially for a decentralized control architecture, in which the formation control actions computation is performed jointly between all the satellites in formation.

1.1 Formation Flight Overview

Formation (i.e. spacecraft flying in formation) is a subset of a more general category of *Distributed Space System* (DSS). Figure 1.1 shows a relationship among a number of common terms used relative to distributed spacecraft and formation flying, including the concept for sensor webs, which may involve many non-space elements [3]. A *Distributed Space System* (DSS) consists of multiple satellites dispersed in space forming a cooperative infrastructure for measurement, data acquisition, processing, analysis and distribution. Each member of a DSS can make independent observations without linking directly to companion satellites. A *sensor web* can be defined as a system of

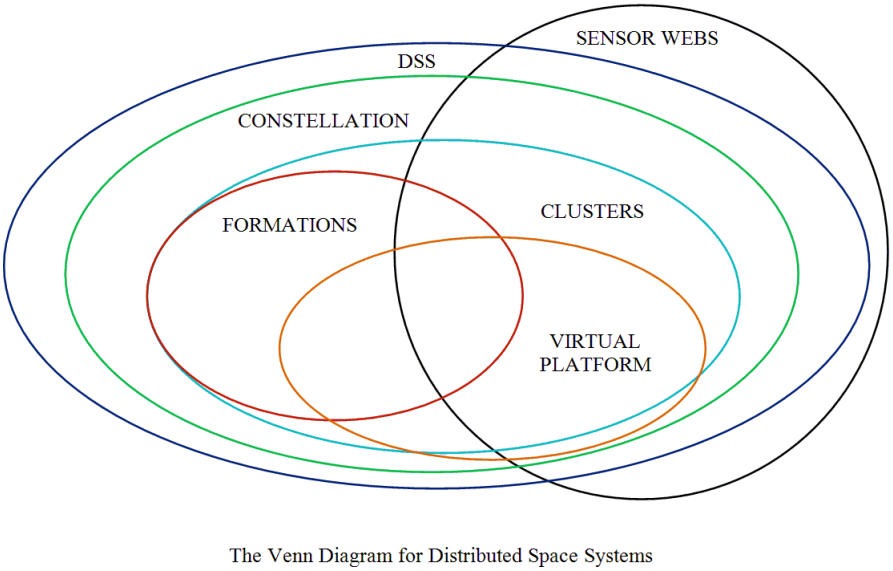


Figure 1.1: Relationship of common terms associated with formation flying [3].

intra-communicating sensor vehicles distributed spatially that may be used for environmental observations. Since sensor webs may include several non-space elements, they are not entirely covered by a DSS. A *satellite constellation* is, on the other hand, a group of spacecraft operating under shared control and reinforcing each other's coverage by mutual coordination. They are usually separated by large distances and dispersed across various orbital planes. A *cluster* is an assemblage of satellites, formations or virtual satellites sharing the same functionality; the inter-vehicle distance in a cluster is usually smaller than that of a constellation. According to [3] *formations* are a special case of constellation that involves the use of an active control scheme to maintain specified relative position and/or attitude.

In [4] and [5], Scharf and Hadaegh define *Spacecraft Formation Flying* as *a set of more than one spacecraft in which any of the spacecraft dynamic states are coupled through a common control law.* This coupling can be in translational and/or rotational degrees of freedom and in position and/or velocity. In particular:

- at least one member of the set must track a desired state relative to another member;
- the tracking control law must at the minimum depend upon the state of this other member.

The second point is critical. For example, even though specific relative positions are actively maintained, the GPS satellites constitute a *constellation* since their orbit corrections only require an individual satellite's position and velocity (state). A *constellation* is actually *a set of spacecraft whose states are not dynamically coupled in any way* (i.e., the change of state of one spacecraft does not impact the state of another).

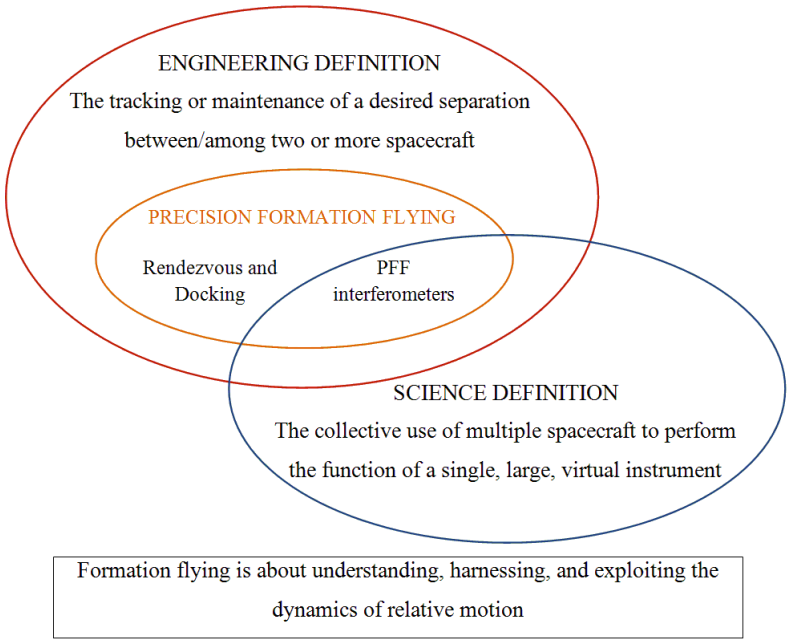


Figure 1.2: Science vs. Engineering definition for Formation Flying [3].

Across the Formation Flying community there exists however a wide range of definitions for formation flying and related terms, each set generally geared towards its own purpose. As suggested in [3], the most distinct differences in definition occur between the science or instrument/sensor community and the engineering or technology community, where for science the interest is in the collection of data, and for engineering the concern is how to collect the data that meet specifications for quality. Figure 1.2 portrays the relationship between the science and engineering definitions, including where they overlap and how "precision" formation flying (PFF) and Rendezvous and Docking fit into the picture. The engineering definition is convenient to employ for the purposes of developing technology plans because missions in that class can have related bins of technologies (at various performance levels), while missions that meet the science definition may have no related technologies at all.

From and engineering perspective, formation flying satellites consists of the fuel-constrained design of satellite formation geometry to meet science requirements, and the measurement and control of relative vehicle states (position, velocity, attitude, etc.) implemented through inter-spacecraft communication links to maintain that geometry.

The following four elements may be considered specific of a formation flying mission.

- *Formation design*: it means design the desired relative motion of the vehicles, the formation geometry, the satellite orbit and collective guidance problem to best meet mission requirements without prohibitive fuel consumption.

- *Relative navigation*: it is the estimation of relative position based on measurements between spacecraft. It includes the sensors, metrology systems and algorithms needed to determine relative position and attitude, either for science use or for feedback control. The requirements for the relative navigation can be driven directly by science requirements (e.g. based on a dynamic range limitation for post-processing) or indirectly by other engineering requirements, such as formation control. For what concern sensor system, two levels of metrology are generally needed to measure relative positions and attitude for control: a RF sensor that is used for coarse formation deployment after launcher exit or after failure recovery, and that allows relative position determination in a range up to 30 km, with an accuracy of a few centimeters in distance and one degree in azimuth and elevation; lateral and longitudinal optical sensors, based on laser and interferometry techniques, that are used to perform a fine position determination with sub-millimeter accuracy.

- *Inter-satellite communication*: it is the data bus of the formation, than robustness and continuity are essential. Since formation control laws are implemented through inter-satellite communication, critical parameters are related to communication bandwidth (data rate) and time synchronization.

- *Formation control*: generally, it is in charge of the formation geometry acquisition, the formation keeping and maneuvers such as collision-avoidance or formation reconfiguration. It requires the control for relative position and attitude, guarantying formation stability and integrity. The formation control problem includes formation control architecture (coordination scheme) selection and formation and satellite modes definition starting from mission phases, actuators selection, sensors selection, relative navigation requirement definition, control algorithm design, together with autonomy and high-level command and control.

1.2 Formation Flight Applications

The satellite formation flying concept enables several applications that would not have been possible or that are enhanced when compared to using a single large spacecraft. Some possible applications are listed below.

- *Synthetic Aperture Radar* (SAR) missions for Earth imaging and remote sensing. SAR is an implementation of radar technology that uses many small antennas distributed among two or several spacecrafts instead of using a single rotating antenna to sense the reflection of electromagnetic wavefronts off the Earth. An improvement of the overall performance can be achieved by implementing large and variable baselines between antennas.

- *Space and Earth Science* missions. Formation flight enables the acquisition of simultaneous measurements of the same scientific target using instruments located on multiple vehicles separated by a few kilometers. Possible applications in Earth science field include gravitational field mapping, magnetic field measurements and sampling of atmospheric data. Formation flight can also be used in missions of fundamental physics, e.g. the gravitational waves detection through the space-time stretch measures.

- *Interferometric* missions. The advantages that formation flying of spacecraft offers to this type of missions is not so much in cost and assembly line production, but rather in increased accuracy. In these missions interferometry will be applied in the search of extra-solar planets and signs of life, such as ozone, in Earth-like planets. Referring to Young's two-slit experiment, the two light splitters are replaced by two or more space-based telescopes or collector spacecrafts. The observed waves are then transferred to the combiner spacecraft, where the waves interfere to make fringes. The accuracy is determined by the baseline, i.e. the distance or separation between the spacecrafts. Logically with satellite formation flying it is possible to increase the separation far more than with a structurally connected craft. The drawback is of course the difficulty of keeping inter-spacecraft position and orientation.

- *Multi-aperture telescopes.* In this approach, a set of vehicles achieve improved optical performance and better coverage of multiple targets by providing a flexible reconfigurable system. Additionally, enhanced upgradeability is considered as a projected advantage as modules could be changed or added and an assembly mission could be performed in space achieving total apertures otherwise impossible to launch as a monolithic unit.

- *Electromagnetic Formation Flight* (EMFF) technology. EMFF is a concept developed by the Space Systems Laboratory at MIT, also independently envisioned by Boeing (Formerly Hughes Aerospace) and a Japanese research group at the university of Tokyo. Its principle of operation is the force created by the interaction of magnetic fields generated by current running through High Temperature Superconducting (HTS) coils. These electromagnetic coils generate fields equivalent at long distance to magnetic dipoles, which can be steered in any three dimensional direction by the combination of currents running through three orthogonal coils.

There are several current projects which are dealing with the formation flying and coordinated control of satellites.

- *TanDEM-X* [6]. TanDEM-X is a project realized by DLR, EADS Astrium GmbH and GSOC, consisting of two satellites equipped with synthetic aperture radar (SAR). By flying in close

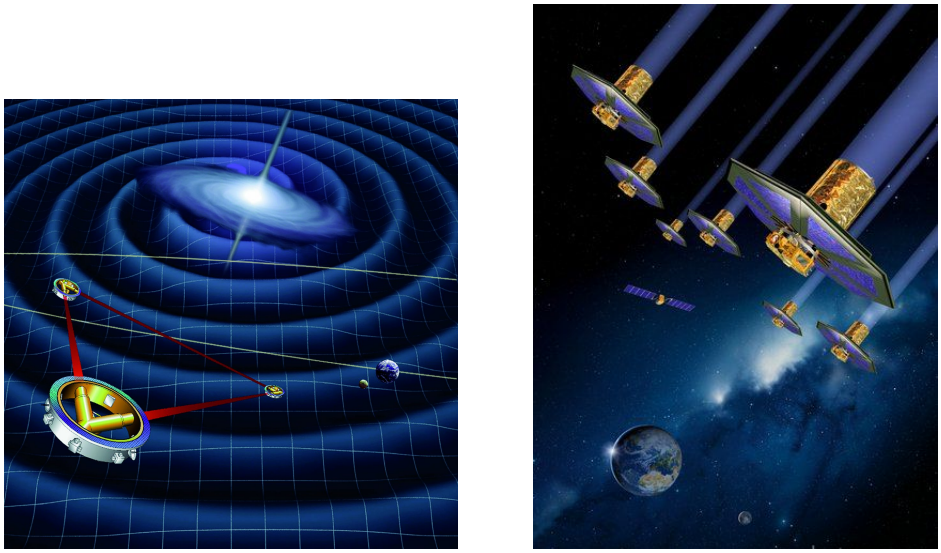


Figure 1.3: LISA (left) and DARWIN (right).

and accurate formation the two spacecraft form a radar interferometer with a baseline of 1 km. This allows for much higher resolution than any earlier SAR mission, and can deliver digital elevation models with unrivaled accuracy.

- *PRISMA*. The PRISMA program is a cooperative effort between Swedish National Space Board (SNSB), Centre National d'Etudes Spatiales (CNES), German aerospace research center (DLR) and the Danish Technical University (DTU). The project constitutes an in-orbit test bed for Guidance, Navigation and Control (GNC) algorithms and sensors for advanced closed-loop formation flying and rendezvous. The two PRISMA satellites were launched on June 15, 2010 into a 600 kilometers altitude, sun-synchronous, dawn/dusk orbit, with the aim of carrying out a series of maneuvering experiments and sensor experiments during a mission time of 8-10 months.

- *GRACE*. The Gravity Recovery and Climate Experiment (GRACE) is a joint partnership between NASA and DLR. This mission has two twin spacecraft launched in March of 2002, flying in a leader-follower configuration in a polar orbit at 500 km above the Earth, with an intersatellite distance that changes over time from 40 kilometers to 200 kilometers. The main goal is to map the Earth's gravity field by making accurate measurements of the distance between the two satellites (10 micrometers) using GPS and a laser ranging system.

- *DARWIN*. DARWIN was a project proposed by ESA in 1993 as part of Cosmic Vision 2015-2025 call for proposals, with the goal of launching a space-based telescope aiding in the search for possible life-supporting planets (Figure 1.3). The telescope would consist of 4 spacecraft flying in autonomous formation: one of them is placed on the center of the symmetric for-

mation and using the nulling interferometry technique combines the light coming from the other satellites to detect the presence of exoplanets around studied stars and their chemical composition.

- *TPF.* Terrestrial Planet Finder (TPF) is a mission concept under study by NASA that would study all aspects of planets outside our solar system. It will be composed of two observatories: a visible-light coronograph and an formation-flying infrared interferometer. Like its European counterpart DARWIN, the mission was postponed indefinitely in 2007.

- *PROBA 3* [7]. Project for On-Board Autonomy (PROBA) 3 is the third satellite mission in the ESA's series of PROBA low-cost satellites that are being used to validate new spacecraft technologies while also carrying scientific instruments. Currently, PROBA-3 is scheduled to be launched in 2015-2016 and it is planned to have a lifetime of around two years. PROBA 3 will verify the metrology, the actuation techniques and the GNC strategies for future formation flying missions. The two satellites that compose the formation will fly in very eccentric orbits and will be controlled only far from the perigee, where the fuel consumption is smaller. Utilizing either cold-gas or electrical thrusters for agile maneuvering, and both radio-frequency and optical (laser-based) metrology techniques for accurate position measurement and control, the combined system is expected to achieve sub-millimeter relative positioning accuracy over a separation range of 25 to 250 meters. An instrument to observe the solar corona will be used for the ongoing design phase to complete the demonstration.

- *XEUS.* X-ray Evolving Universe Spectroscopy (XEUS) is an ESA proposed two-satellite X-Ray observatory, with a detector and a mirror satellite. The mirror satellite is the leader of the formation pointing at the area of interest, the detector tracks the focal point of the mirror satellite. However, in 2008 the XEUS mission was merged with Constellation-X mission (NASA) to form the International X-Ray Observatory (IXO), and the formation flying part was canceled in favor of a large single spacecraft bus.

- *LISA.* Laser Interferometer Space Antenna (LISA) is a future mission issued from a collaboration between NASA and ESA composed of a three satellites interferometer. The orbits of the satellites will be similar to the Earth's one around the Sun, but will trail behind our planet at distances of around 50 million kilometers, equivalent to 20 degrees. Launching date is about 2018 with a mission lifetime of 5 years. The three satellites form an equilateral triangle (5 million kilometers between satellites) facing the Sun, slanting at 60 degrees to the plane of the Earth's orbit and revolving with the Earth around the Sun (Figure 1.3). The main goal of the mission is the detection of gravitational waves. They are predicted by Einstein theory

but they have never been directly detected in spite of very performing experiments. The tiny size of the waves and the number of perturbations on Earth are the two factors that have prevented their detection. Their detection would certainly open another door for the exploration of the Universe. Nowadays, natural relative motion of the formation is not adapted for the goal of the mission. For interferometer purposes, satellites should keep constant interdistance and angle, but natural motion of the satellites introduces a kind of "breathing" of the configuration all along the orbit.

- *NGGM.* Next-Generation Gravimetric Mission (NGGM) [8] is a formation flying mission under study by Thales Alenia Space Italia (TAS-I) whose aim is monitoring of the temporal variations of the Earth's gravity field at high spatial resolution, up to harmonic degree 180-240, as for GOCE but extended also to longer wavelengths, over a long period of time, greater than 6 years, as for GRACE, and with high time resolution, weekly or better, to reduce the level of aliasing of the high frequency phenomena found in the time series of the Earth's gravity field variation provided by GRACE. A resolution equal to 0.1 mm/year of the geoid height variation rate at 200 spherical harmonic degree (corresponding to a 100 km spatial resolution) was preliminarily identified as the mission performance needed for detecting the finest geophysical phenomena of interest. The reference mission scenario consists on two co-orbiting satellites flying for 6 years at a 10 km relative distance on a 325 km altitude sun-synchronous orbit. The distance between the two satellites center of masses is subject to a time variation due both to differential gravitational accelerations, since the spacecraft fly over different zones of the Earth at the same time, and to other non-gravitational acceleration, mainly differential aerodynamic drag (Figure 1.4). The inter-satellite distance variation induced by the non-gravitational component is first measured by a set of accelerometers installed on each spacecraft. A satellite-to-satellite tracking technique is used to measure, with a laser interferometry system, the total (gravitational plus non-gravitational) inter-satellite distance variation. The two measurements are then properly subtracted to obtain the gravitational component of the inter-satellite distance variation and to estimate the shape of the geopotential that has caused it. The formation control for this mission is designed to work in synergy with the drag-free control, necessary for providing quiet operational environment to the accelerometers, to not interfere with the scientific measurement and to minimize the use of the thrusters. Another control system is in charge of maintaining the fine pointing of the interferometer laser beam from one satellite to the other one.

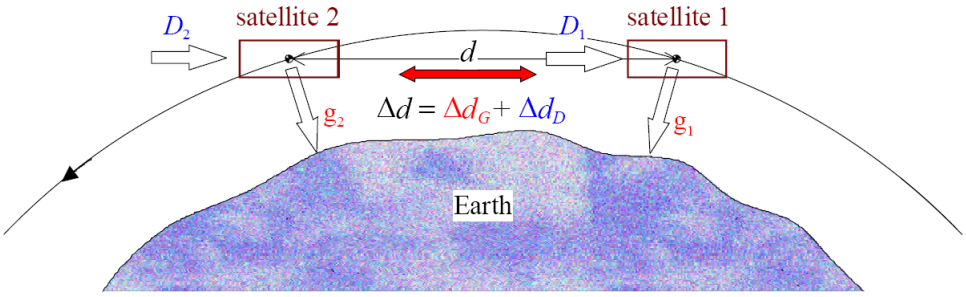


Figure 1.4: Next Generation Gravimetric Mission [8].

1.3 Thesis Outline

In the last five years, a team of Ph.D. students, Master Degree students and some CISAS researchers leaded by prof. Enrico Lorenzini of the Department of Industrial Engineering (DII) at the University of Padova has been and some of them are currently involved in the following research activities concerning SFF:

1. designing, building and testing of a SFF hardware simulator that allows to conduce on ground experimental research on both formation flight and rendez-vous and docking;

2. software development for the SFF hardware simulator, including both the on board software and the control station software and a Matlab software simulator;

3. planning of collision-avoidance and modern optimal control strategies for SFF;

4. test of formation control strategies with the SFF hardware simulator;

5. development of highly accurate and fast propagators for predicting spacecraft relative motion.

With my Ph.D. research activities I gave a contribution mainly to the first four points as described in the following.

As clearly stated in the previous sections, SFF is a key technology for several future missions. It is strongly based on a GNC system that must be: (a) reliable in coordinating all the s/c flying in formation during each mission phase, guaranteeing formation integrity and preventing from formation evaporation; and at the same time (b) efficient in using the on board resources. Model Predictive Control (MPC) is a modern optimal control technique that appears suitable to face these challenges thank to its three main features.

1. **Model-based**. An *explicit model of the system* to be controlled is used to predict the system future behavior over a finite prediction horizon as a function of a control sequence.

2. **Constraints handling**. *Constraints on* both the dynamic system *output variables* and the *control variables* are *inherently taken into account* within the optimal control problem.

3. **Re-planning nature**. An *optimal control sequence* that minimizes a certain cost function for the system over the prediction horizon is *computed on-line for the current state of the system and at each control step.*

Chapter 2 begins with a description of the most commonly used mathematical models that describe the SFF relative dynamics. It is indeed essential to understand the dynamics of s/c flying in formation in order to properly design formation flight Guidance, Navigation and Control (GNC) systems. After an analysis of some formation configurations used in many proposed missions, Chapter 1 concludes with a description of GNC systems for SFF.

In Chapter 3 we present three types of control strategies that can be used in SFF: Proportional-Integral-Derivative (PID), Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC). For each control type, the basic algorithms and the tuning of those parameters that determine the control performances in tracking a reference trajectory and the fuel consumption are analyzed and discussed. A great emphasis has been dedicated to MPC describing in detail the problem formulation and the control action computing algorithm both for an on-line and an explicit solution.

In order to simulate the SFF dynamics and test control strategies, a Matlab SFF software simulator has been developed and in Chapter 4 we describe its main features. The Matlab SFF simulator was used in Chapter 5 to study the MPC application to a formation acquisition maneuver for two space vehicles, taking into account two scenarios: a Leader-Follower (LF) formation and Projected Circular Orbit (PCO) formation. The performances of a MPC-based controller are compared with those of a LQR-based controller in carrying out the same maneuver in the presence of active constraints on the maximum control acceleration, evaluating also the effects of the gravitational harmonics J_2 and J_3 and atmospheric drag perturbations on the proposed maneuvers.

In Chapter 6 we propose some guidance and control strategies for a Collision-Avoidance scenario with a pair of s/c flying in formation. In this scenario, a free-flying chief spacecraft follows temporary off-nominal conditions and a controlled deputy spacecraft performs collision avoidance maneuvers. In particular, the Collision-Avoidance strategy consists in a Separation Guidance and a Nominal Guidance. The Separation Guidance is in charge of the avoidance of a predicted collision soon to occur, and it is based on a computationally simple, deterministic and closed-form algorithm, so that a valid solution is always available without delay. The Nominal Guidance uses Genetic Algorithms and it takes charge of placing the deputy spacecraft in a bounded safe or "parking" trajectory, while minimizing the propellant consumption and avoiding the formation evaporation. The output of the Nominal Guidance is a pair of reference position and velocity trajectories. In

the second part of the chapter, the performances of a LQR and a MPC are compared in tracking these reference trajectories.

Chapter 7 deals with testing the MPC capabilities on the MIT SPHERES testbed in simulating the close-proximity phase of the rendez-vous and capture maneuver for the Mars Orbital Sample Return (MOSR) scenario. In this scenario, a small Orbiting Sample (OS) satellite with some geological samples on board rendezvous with an Earth Return Vehicle (ERV) in Martian orbit. Since the OS will most likely be passive during this maneuver, the ERV must determine the OS location in Martian orbit using a single visual-band camera, and maneuver itself to capture the OS keeping it within the camera field of view. After a description of the MPC formulation for this scenario, we provide a comparison between a predictive controller and a PD controller in performing this maneuver in two simulation environments: a Matlab simulator with non representative SPHERES properties and no sensors/actuators noises, and the MIT SPHERES simulator, which is a more representative environment than the previous one. This chapter ends with the evaluation of MPC performances in the SPHERES Flat Floor facility at the MIT Space System Laboratory.

Chapter 8 is devoted to the SFF hardware simulator that we are developing at the University of Padova. The first part of the chapter illustrates the hardware simulator main features providing a synthetic description of its 5 main subsystems. After that, we describe some preparatory experimental activities aimed at: (a) determining the hardware simulator on board thrusters force; and (b) estimating the hardware simulator moment of inertia about its z-Body axis. The characterization of the torsional system used for both experimental activities is also provided. The final part of this chapter shows some experimental results on 1 DOF attitude maneuvers performed with the use of MPC. The chapter concludes with some ideas concerning future activities and perspectives on the SFF hardware simulator.

Chapter 2

SPACECRAFT FORMATION FLYING

2.1 Relative Motion Models

The analysis of spacecraft relative motion and the development of models that accurately describe it constitute an issue of increasing interest due to existing and planned spacecraft formation flying and orbital rendezvous missions. Relative motion models are very useful in the following areas:

- comprehension of the relative motion between satellites flying in formation;
- design of several formation geometries and analysis of their performances in relation to a specific application, e.g. SAR, interferometry, etc. (see Chapter 1);
- analysis of the perturbative effects such as J_2, atmospheric drag or third-body;
- selection of initial conditions that reduce undesired effects, e.g. secular drift that cause formation evaporation or formation geometry alteration;
- design of the GNC system: reference trajectories generation or mode transition (Guidance); optimal relative position and/or attitude estimation (Navigation); and optimal control law synthesis such as Linear Quadratic Control or Model Predictive Control (see Chapter 3).

Satellite relative motion models can be classified according to the following two criteria:

1. linearity – non linearity, time invariance – time variance;
2. coordinate type used, such as Cartesian coordinate or orbital element difference.

A classification based on the former criteria is more useful in GNC system design, since this criteria is based on a control system perspective; the last criteria is mainly used in astrodynamical field to

analyze the effects of perturbations on the relative motion and to find proper initial conditions to reduce their effects.

From a linearity – non linearity, time invariance – time variance point of view, relative motion models can be divided in:

1. Linear Time-Invariant (LTI) models;

2. Linear Time-Varying (LTV) models;

3. Non-Linear (NL) models.

In the following subsections, each relative motion model type is briefly introduced, followed by a short description of those models that are used to obtain numerical results presented in the next chapters.

2.1.1 Linear Time-Invariant models

Linear Time-Invariant models are the most simple satellite relative motion models. In the early 60's Clohessy and Wiltshire first published their celebrated work that utilized a Hill-like rotating Cartesian coordinate system to derive expressions for the relative motion between satellites in the context of a rendezvous problem [2]. The Hill-Clohessy-Wiltshire (HCW) model is based on the assumptions that the orbit of the target spacecraft is circular (eccentricity e ≈ 0), that the distance between the target and the chaser is small compared with the radius of the target's orbit and with no perturbations. HCW model can also be expressed in cylindrical coordinate, providing better accuracy for larger in-track separations, or in terms of differential orbital elements[1] [9]. HCW model has been modified to take into account the averaged effects of perturbations such as J_2 (Earth's oblateness) [10] and/or differential atmospheric drag [11].

The HCW model describe the relative motion of a deputy satellite with respect to a chief satellite in the Hill's Local-Vertical Local-Horizontal (LVLH) rotating Cartesian coordinate system $\mathbf{O}$ with origin at at the osculating Chief satellite position and orientation given by the vector triad $\{\hat{\mathbf{o}}_r, \hat{\mathbf{o}}_\theta, \hat{\mathbf{o}}_h\}$, as shown in Figure 2.1: $\hat{\mathbf{o}}_r$ is in the orbit radius direction, $\hat{\mathbf{o}}_h$ is parallel to the orbit momentum vector in the orbit normal direction and $\hat{\mathbf{o}}_\theta$ completes the right-handed coordinate system. In this reference frame, the Deputy has a relative position $\rho = [\, x,\, y,\, z\,]^T$ and a relative velocity $\dot{\rho} = [\, \dot{x},\, \dot{y},\, \dot{z}\,]^T = [\, u,\, v,\, w\,]^T$.

[1] The equinoctial orbit elements may be used instead of the classical set of elements to avoid singularities in the equations of motion at zero inclination and eccentricity.

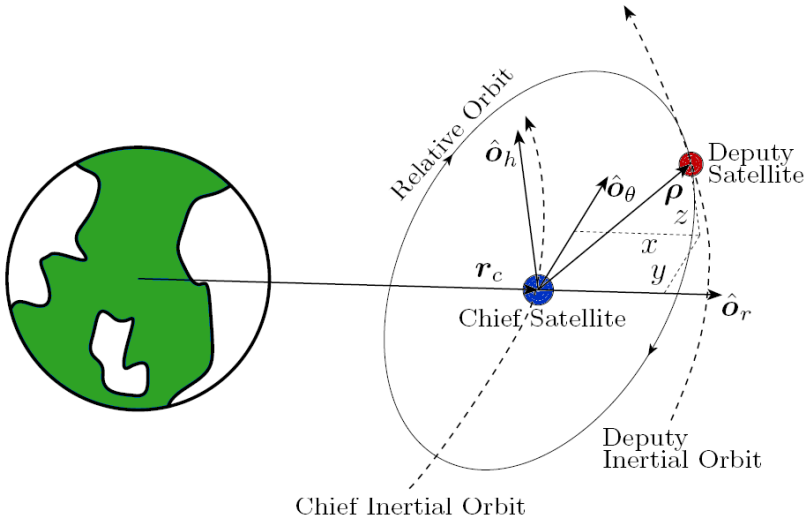


Figure 2.1: Hill's LVLH reference frame [12].

The Hill's differential equations in Cartesian coordinates and in the non homogeneous form can be written as follow [13] [12]:

$$\begin{cases} \ddot{x} - 3\,n^2\,x - 2\,n\,\dot{y} &= f_x \\ \ddot{y} + 2\,n\,\dot{x} &= f_y \\ \ddot{z} + n^2 z &= f_z \end{cases} \tag{2.1}$$

Hill's equations are a set of coupled second order differential equations with constant coefficient. Note also the types of acceleration in these equations: in the x equation, from left to right, they are total, centripetal and Coriolis acceleration. f_x, f_y and f_z denote the components of other accelerations, i.e. accelerations due to the propulsion system or accelerations due to perturbations, expressed in the Hill's frame. One interesting property is that, although the equations describing the in-plane motion are coupled, the out-of-plane motion is uncoupled. The velocity dependent terms $2\,n\,\dot{x}$ and $2\,n\,\dot{y}$ represent a damping term in the system that is non-dissipative: it is present only because the motion is described in a rotating coordinate frame.

The HCW equations can be obtained solving the Hill's differential unforced equations through a standard Laplace transformation, form example:

$$x(t) = \left(4\,x_0 + 2\,\frac{v_0}{n}\right) + \frac{u_0}{n}\,\sin\,nt - \left(3x_0 + 2\,\frac{v_0}{n}\right)\,\cos\,nt \tag{2.2}$$

$$y(t) = \left(y_0 - 2\,\frac{u_0}{n}\right) + \left(6\,x_0 + 4\,\frac{v_0}{n}\right)\,\sin\,nt + 2\,\frac{u_0}{n}\,\cos\,nt - (6\,n\,t + 3\,v_0)\,t \tag{2.3}$$

$$z(t) = z_0\,\cos\,nt + \frac{w_0}{n}\,\sin\,nt \tag{2.4}$$

$$u(t) = u_0\,\cos\,nt + (3\,n\,x_0 + 2\,v_0)\,\sin\,nt \tag{2.5}$$

$$v(t) = (6\,n\,x_0 + 4\,v_0)\,\cos\,nt - 2\,u_0\,\sin\,nt - (6\,n\,t + 3\,v_0) \tag{2.6}$$

$$w(t) = -\,z_0\,n\,\sin\,nt + w_0\,\cos\,nt \tag{2.7}$$

Although the deputy doesn't actually "orbit" the chief satellite, the instantaneous motion is somewhat elliptical [14]. The term multiplied by time in the y equation, i.e. $(6\,n\,t + v_0)\,t$, represents the drift of the deputy with respect to the chief and it is the reason the deputy's path is not truly elliptical, due mainly to small differences in the semi-major axis of the two spacecrafts. The first terms in the x and y equations, i.e. $4\,x_0 + 2\,\frac{v_0}{n}$ and $y_0 - 2\,\frac{u_0}{n}$ respectively, represent the initial displacement of the deputy average position from the chief.

In state space form, the Hill's differential equations assume the form:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) \tag{2.8}$$

where $\mathbf{x} = [\,x,\,y,\,z,\,\dot{x},\,\dot{y},\,\dot{z}\,]^T$ is the state vector and $\mathbf{A}$ is the state matrix given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \tag{2.9}$$

The HCW model in state space form can be written as:

$$\mathbf{x}(t) = \mathbf{\Phi}(t - t_0)\,\mathbf{x}(t_0) \tag{2.10}$$

where $\mathbf{\Phi}$ is the State Transition Matrix (STM):

$$\mathbf{\Phi}(\Delta t) = e^{\mathbf{A}\Delta t} = \begin{bmatrix} 4-3\mathrm{c} & 0 & 0 & \frac{s}{n} & 2\frac{1-c}{n} & 0 \\ 6\,(\mathrm{s}-n\Delta t) & 1 & 0 & 2\frac{c-1}{n} & 4\frac{s}{n}-3\Delta t & 0 \\ 0 & 0 & \mathrm{c} & 0 & 0 & \frac{s}{n} \\ 3\,n\,\mathrm{s} & 0 & 0 & \mathrm{c} & 2\,\mathrm{s} & 0 \\ 6\,n\,(\mathrm{c}-1) & 0 & 0 & -2\,\mathrm{s} & 4\,\mathrm{c}-3 & 0 \\ 0 & 0 & -n\,\mathrm{s} & 0 & 0 & \mathrm{c} \end{bmatrix} = \begin{bmatrix} \Phi_{rr} & \Phi_{rv} \\ \Phi_{vr} & \Phi_{vv} \end{bmatrix} \tag{2.11}$$

with $\Delta t = t - t_0$, $\mathrm{c} = \cos(n\,\Delta t)$ and $\mathrm{s} = \sin(n\,\Delta t)$.

If the relative orbit initial conditions satisfy the constraint:

$$\dot{y}_0 + 2\,n\,x_0 = 0 \tag{2.12}$$

then a bounded relative motion to *first order*, i.e. only for the HCW model, will occur. Assuming this constraint is satisfied, then the HCW equations can be rewritten as follow:

$$x(t) = \rho_x\,\sin\left(nt + \varphi_{xy}\right) \tag{2.13}$$

$$y(t) = \rho_y\,\cos\left(nt + \varphi_{xy}\right) + \Delta y \tag{2.14}$$

$$z(t) = \rho_z\,\sin\left(nt + \varphi_z\right) \tag{2.15}$$

with $\rho_x = \rho_y/2$. Parameters ρ_y, Δy, ρ_z, φ_{xy} and φ_z are determined through the relative orbit initial conditions:

$$\rho_x = \sqrt{x_0^2 + \frac{\dot{x}_0^2}{n^2}} \tag{2.16}$$

$$\Delta y = y_0 - 2\,\frac{\dot{x}_0}{n} \tag{2.17}$$

$$\rho_z = \sqrt{z_0^2 + \frac{\dot{z}_0^2}{n^2}} \tag{2.18}$$

$$\varphi_{xy} = \arctan\left(\frac{n\,x_0}{\dot{x}_0}\right) \tag{2.19}$$

$$\varphi_z = \arctan\left(\frac{n\,z_0}{\dot{z}_0}\right) \tag{2.20}$$

An example of a first order bounded relative motion is shown in Figure 2.2.

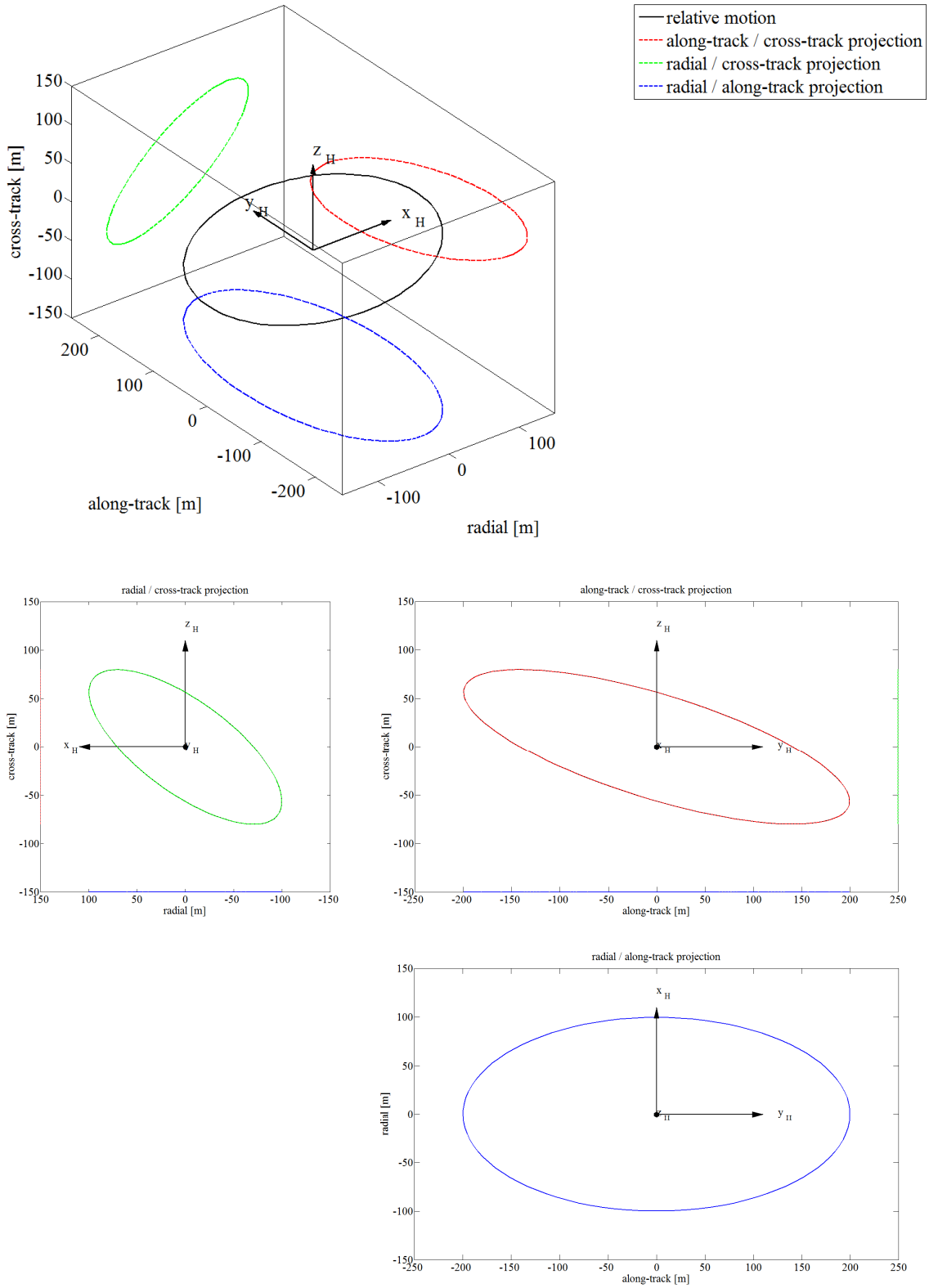


Figure 2.2: First order bounded relative motion obtained with $\rho_x = 100\,m$, $\rho_y = 200\,m$, $\Delta y = 0\,m$, $\varphi_{xy} = \pi/2$, $\rho_z = 80\,m$ and $\varphi_z = \pi/4$. The Hill's reference frame is shown in black with x_H radial, y_H along-track and z_H cross-track.

The HCW model can be written in terms of differential orbital elements as follow [9]:

$$x(f) = \delta a - a\,\delta e\,\cos\,f \tag{2.21}$$

$$y(f) = a\,(\delta\omega + \delta M + \delta\Omega\,cos\,i) + 2\,a\,\delta e\,\sin\,f \tag{2.22}$$

$$z(f) = a\,\sqrt{\delta i^2 + \delta\Omega^2\,sin^2 i}\,\cos\,(\theta - \theta_z) \tag{2.23}$$

where a is the semi-major axis, f is the true anomaly, ω is the argument of perigee, $\theta = \omega + f$ is the true latitude angle, e is the eccentricity, i is the orbit inclination angle, Ω is the argument of the ascending node and θ_z is a phase angle which is function of the relative initial conditions.

2.1.2 Linear Time-Varying models

For non-circular orbits, several LTV models are available and differ according to linearization assumptions, how J_2 is considered, and the coordinate frame used. The most used ones are listed in the follow.

Lawden or Tschauner-Hempel model

Lawden [15] and Tschauner and Hempel [16] independently developed a set of linearized equations describing spacecraft relative dynamics in elliptical orbits. These equations are similar to the HCW equations, in that the linearization assumes that the distance between the target and the chaser is small compared with the radius of the target's orbit, but the solution is a function of the true anomaly, which itself is a function of time (Euler's equation), given an eccentric chief orbit. Lawden's equations has been adjusted to prevent singularities at $e = 0$ in [17]. This model can be expressed in Cartesian or curvilinear coordinates and can be modified to include J_2 effects [18].

Yamanaka-Ankersen State Transition Matrix

State Transition Matrices provide a very useful tool for mission planning purposes, since they allow for the determination of the state at a time t_1 given the state at time t_0. The work by Yamanaka and Ankersen [19] provided a simple State Transition Matrix (STM) that functions well in arbitrary elliptical orbits $0 \leq e < 1$. The relative motion is expressed in a Cartesian frame, and because the STM is derived from the solution of the unforced linearized relative dynamics, the inclusion of J_2 is not permitted.

Melton State Transition Matrix

Melton [20] developed a time-dependent solution for an elliptical reference orbit by generating a STM which is expanded in powers of eccentricity. The solution includes separate matrix elements for first- and second-order terms in eccentricity and for both Cartesian and cylindrical coordinates. Melton also shows that the solution is accurate for practical purposes with eccentricity in the range $0 - 0.3$.

Gim-Alfriend State Transition Matrix

Gim and Alfriend [21] obtained a complex STM for the linearized equations of motion of the deputy in an elliptical orbit that included the first-order absolute and differential J_2 effects. In particular, this STM was obtained using a linearized geometric mapping between relative curvilinear coordinates and differences in a non singular set of orbital elements (geometric method). A less complex version for low eccentricity is discussed in [22].

Gauss's Variational Equations

Models based on Gauss's Variational Equations [23] (GVEs) use Gauss's expressions to relate an acceleration vector expressed in a local frame (radial, along-track and cross-track directions) to changes in classical orbital elements. Linear models can be obtained by linearizing differences in these elements between a deputy and a reference satellite. GVEs are convenient for specifying and controlling widely separated formations because they are linearized about orbital elements, which are expressed in a curvilinear frame in which large rectilinear distances can be captured by small element perturbations. In addition, the GVEs provide a computationally simple way (no frame rotations are required) to obtain linearized dynamics about the orbits of each spacecraft in the formation. This bypasses the linearization error created by representing the entire formation in a single rectilinear frame. Breger and How [24] showed how the Gim-Alfriend approach can be used with GVEs to include the effects of J_2.

2.1.3 Non-Linear models

Vaddi, Vadali and Alfriend [25] have extended the HCW equations to include nonlinear terms and eccentricity. Richardson and Mitchell [26] follow a Lagrangian approach to develop the nonlinear Hill's equations and show that the effect of a spherical primary mass (Earth) can be interpreted as a third body perturbation in the relative motion frame. They expanded the relative motion equations through third-order in the local Hill's coordinates and they also developed an accurate successive approximations solution to describe nonlinear periodic motions.

2.2 Formation Geometries and Configurations

This section describes the geometry of a bounded relative motion obtained if the 1st order condition expressed in Equation 2.12 is satisfied. Equations 2.13 - 2.15 constitute a parametric representation of an *elliptic cylinder*. The in-plane motion is a 1:2 ellipse with semi-minor axis ρ_x in the radial direction, semi-major axis $\rho_y = 2\,\rho_x$ in the along-track direction and constant eccentricity of $\sqrt{3}/2$. The center of motion is located in the along-track direction with an offset of Δy from the origin. The out-of-plane motion is an oscillation with semiamplitude ρ_z. φ_{xy} and φ_z are the initial phase angles for the in-plane and out-of-plane motions, respectively.

Different formation geometries can be obtained through a proper selection of parameters ρ_x, Δy, ρ_z, φ_{xy} and φ_z, as summarized in Table 2.1. In particular, in PCO formations (see Figure 2.3), the projection of the relative motion on the along-track / cross-track plane is a circle; in other words, the deputy satellite appears to be orbiting the chief satellite describing a circle as viewed from the Earth. Also, a PCO formation lies on one of the two possible planes inclined by $\pm 63.4^\circ = \pm atan(2)$ with respect to the radial / along-track plane. In GCO formations (see Figure 2.3), the relative motion is a 3D circle, i.e. the distance between the deputy and the chief satellites is constant. GCO formations lie on two possible planes forming an angle of $\pm 60^\circ = \pm atan(3)$ with the radial / along-track plane.

The projection of the relative motion in the x-z plane is, in general, a rotated ellipse whose semi-major axis a and semi-minor axis b are:

$$a = \sqrt{\frac{\rho_x^2 + \rho_z^2 + \sqrt{\rho_x^4 + \rho_z^4 + 2\,\rho_x^2\,\rho_z^2\,\cos\,2\alpha}}{2}} \tag{2.24}$$

$$b = \sqrt{\frac{\rho_x^2 + \rho_z^2 - \sqrt{\rho_x^4 + \rho_z^4 + 2\,\rho_x^2\,\rho_z^2\,\cos\,2\alpha}}{2}} \tag{2.25}$$

where $\alpha = \varphi_{xy} - \varphi_z$.

The study of satellite formation configurations is another important research area in satellite formation flying with several interesting results and applications. Using the HCW model, both in cartesian coordinate and in orbital elements difference formulation, the following five formation configurations can be obtained: In-line, Pendulum, Cartwheel, Car-Pe and LISA-like.

Table 2.1: Formation geometries as a function of parameters ρ_x, Δy, ρ_z, φ_{xy} and φ_z.

projection plane	parameters selection	formation geometry
x - z	/	rotated ellipse
	$\varphi_z = \varphi_{xy} + (\frac{\pi}{2} + k\,\pi)$	ellipse
	$\varphi_z = \varphi_{xy} + (\frac{\pi}{2} + k\,\pi);\ \rho_z = \rho_x$	circle
	$\varphi_z = \varphi_{xy}$	line
y - z	/	rotated ellipse
	$\varphi_z = \varphi_{xy}$	ellipse
	$\varphi_z = \varphi_{xy};\ \rho_z = 2\,\rho_x$	circle (PCO)
	$\varphi_z = \varphi_{xy} + (\frac{\pi}{2} + k\,\pi)$	line
/	$\varphi_z = \varphi_{xy};\ \rho_z = \sqrt{3}\,\rho_x$	3D circle with radius $2\,\rho_x$ (GCO)

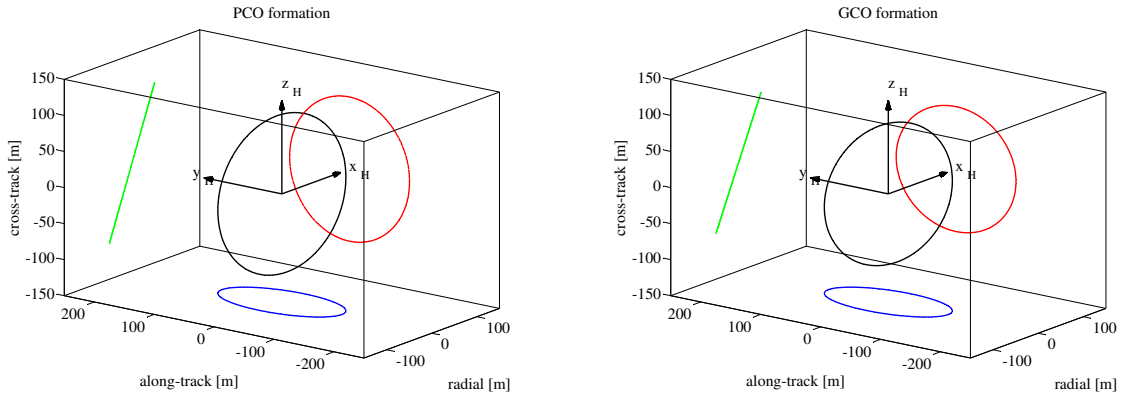


Figure 2.3: PCO formation (left) with $\rho_x = 50\,m$, $\rho_y = 100\,m$, $\Delta y = 0\,m$, $\rho_z = 100\,m$ and $\varphi_{xy} = \varphi_z = 0$; GCO formation (right) with $\rho_x = 50\,m$, $\rho_y = 100\,m$, $\Delta y = 0\,m$, $\rho_z = 50 \cdot \sqrt{3}\,m$ and $\varphi_{xy} = \varphi_z = 0$.

In-line

In the In-line formation, also referred to as Leader-Follower formation, two or more spacecraft follow the same orbit and are separated in the along-track direction. All the satellites have the same orbital elements except for the true anomaly, whose difference determines the along-track separation between them.

Pendulum

The cross-track Pendulum configuration consists of two or more spacecraft in circular orbits with different RAAN and optionally[2] inclination to produce a stable cross-track "swinging" motion between the satellites, resulting in a cross-track baseline[3] that varies between near-zero and a maximum (desired) value. The along-track baseline remains nearly constant at a predetermined distance, and the vertical baseline is near-zero. Also, the along-track position of each satellite relative to the others can be adjusted independently of the cross-track motion.

Cartwheel

In the radial–along-track Cartwheel formation, all satellites follow the same relative orbit, which ideally is a $2:1$ ellipse in the radial–along-track plane with semi-major axis $2\,a\,e$ and semi-minor axis $a\,e$. Like the wheel of a cart, the satellites move around a reference point that follows a circular orbit with radius a and coplanar with the formation. To obtain a Cartwheel formation, all the satellites must have the same inclination i and RAAN Ω, which means coplanar orbital planes, and the same semi-major axis a and eccentricity e, which means same mean motion and relative ellipse dimensions. Design parameters are therefore the orbital radius a, which also defines the formation altitude, the eccentricity e, which defines the relative ellipse dimensions, and the number of satellites N flying in formation.

A more general type of Cartwheel formation presents all the satellites following the same relative orbit that lays on a general-orientated plane. In this kind of Cartwheel, the baselines in the radial, along-track and cross-track directions are coupled and, unlike in the Pendulum configuration, they can not be independently adjusted.

Car-Pe

The Car-Pe configuration consists of two or more satellites in a Pendulum configuration, combined with one or more satellites in a Cartwheel formation. The Car-Pe configuration combines the advantages of both the Pendulum and Cartwheel configurations, including the decoupled baselines of the Pendulum configuration and the baseline envelopes of the Cartwheel. The along-track baseline of the Pendulum satellites can be adjusted for a minimum along-track baseline, and the larger along-track baseline of the Cartwheel satellite(s) can be used if a longer along-track baseline is desired.

[2]Differential inclinations used to contribute to the cross-track motion cause secular drifts of the ascending nodes of the orbits.

[3]The baseline is the distance between each pair of satellites flying in formation. This is an important parameter for applications such as SAR.

LISA-like

This kind of formation geometry takes its name from the Laser Interferometer Space Antenna (LISA) mission. Three satellites are placed at the vertices of an equilateral triangle and, in the absence of perturbations, the separation between them remains constant. Also, the relative orbit plane forms an angle of 60° with respect to the absolute orbital plane of the formation center of mass.

The previous formation configurations were selected for several space applications, such as interferometric Synthetic Aperture Radar (SAR) or Earth gravity field recovery.

In [27, 28], Massonnet proposed a Cartwheel formation for SAR applications. In the proposed scenario, a conventional radar s/c transmits radar pulses toward the Earth surface. Radar signals, after reflection off the ground, are acquired quasi-simultaneously by a set of several (typically three) receive-only s/c flying in a Cartwheel configuration. The Cartwheel formation flies at a distance (typically several kilometers) from the radar s/c in an In-line configuration. Advantages that this kind of formation geometry offers include vertical and horizontal baseline geometric stability (less of 8% variation during the orbit) and the possibility to systematically produce both cross-track and along-track interferometric data.

In [29], Krieger et al. analyzed the interferometric performances of Cartwheel, cross-track Pendulum and Car-Pe configurations in realizing a low-cost system for high-quality along-track and cross-track SAR interferometry. Moccia and Fasano [30] investigated potential formation configurations for the Italian COSMO-BISSAT formation, describing achievable baseline components, percentage of the orbit adequate for interferometry, and covered latitude intervals.

In gradiometry from satellite formations [31], the gravity field recovery is based on the observation of ranges ρ and range rates $\dot{\rho}$ between couples of satellites obtained by means of laser interferometry based technology. From ρ and $\dot{\rho}$, it is possible first to calculate both the scalar range acceleration $\ddot{\rho}$ by numerical differentiation and the vectorial acceleration difference $\ddot{\boldsymbol{\rho}}$ by Newton's equations, and then to obtain the gravity gradient in the base-line direction. With sufficiently many satellites linked together in a strategic way, one can even achieve full-tensor gravity gradiometry. For example, it is possible to determine the three instantaneous in-plane components of the gravity gradient tensor with two possible coplanar satellite configurations: a Cartwheel of three satellites measuring in a triangle, or a Cartwheel of six satellites measuring along the spokes of the wheel. Gradiometry of out-of-plane components can only be achieved through non-coplanar satellite configurations in which six instantaneous intersatellite distances have to be measured.

In [32], Wiese et al. compare the performances of four candidate mission architectures in recovering the gravity field: a traditional two-satellite In-line formation, a four-satellite In-line con-

figuration using two collinear satellite pairs, a two-satellite Cartwheel formation and a four-satellite Cartwheel formation. The authors showed that, with respect to an In-line formation, a Cartwheel formation adds radial information to the observable in addition to the along-track information, resulting in lower uncertainties in the estimation of the geopotential coefficients, with a reduced longitudinal striping which is seen in the GRACE In-line mission, and a more isotropic error spectrum. The Cartwheel architecture is however more demanding in terms of mission implementation, since station-keeping maneuvers are needed to avoid the formation evaporation mainly caused by constantly changing drag coefficients between the satellites.

The gravity recovery capability of four formation flying configurations was also studied by Sharifi et al. in [33].

2.3 Guidance Navigation and Control

Guidance, Navigation and Control (GNC) represent the three different logical steps, good for every aerospace mission concepts, both for single vehicle or multiple agent systems. *Navigation* refers to the determination, at a given time and in an ad-hoc reference system, of the kinematic state of the vehicle(s), given by position and velocity, and often including attitude information. *Guidance* refers to the determination of the desired trajectory from the vehicle's current location to a designated target, as well as desired changes in velocity, rotation and acceleration for following that path. Trajectory selection is performed before the mission begins or during the flight (real time), acknowledging and handling different environmental conditions, system status and mission scenarios. *Control* refers to exploit forces and/or torques to track guidance commands, aiming to have the current state, given by navigation function, (almost) equal to the desired one, while maintaining vehicle stability.

Guidance is the brain of the loop and it should be performed taking into account the limits of the system, but (ideally) following a receipt which could not vary depending on the state, but just timely re-evaluated with state information. Hardware constraints for Guidance are related to the computing capability of the on board computer. Navigation and Control interact with the real world, and therefore need some agents to exploit such an interaction, i.e. sensors for navigation and actuators for control. Constraints due to the hardware limitations clearly appear.

The GNC Loop is presented in Figure 2.4. It is possible to identify two types of GNC systems: Open Loop and Close Loop systems. In Open Loop systems, Guidance is a priori defined and does not change, Control follows a time schedule and exploit previously computed maneuvers, and Navigation is intended only to make aware of the system status. It has been used even for sophisticated aerospace system, but it is constrained by our poor knowledge of the environment

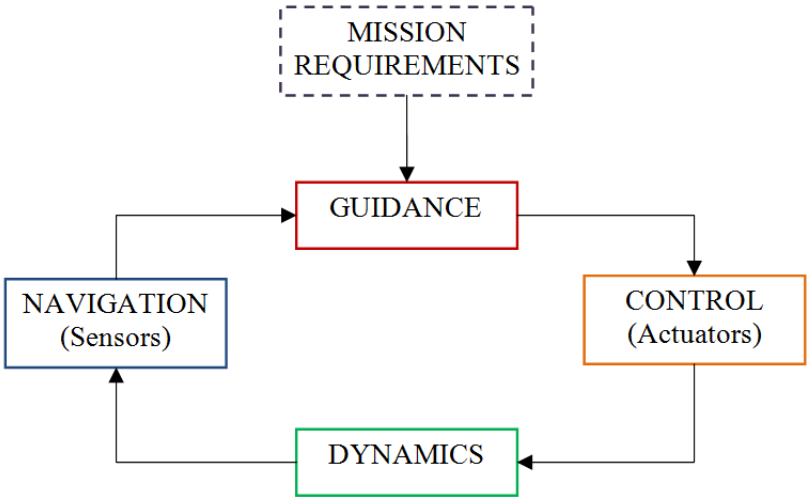


Figure 2.4: Guidance Navigation and Control Loop.

and by our limitation in manufacturing and operating the system. For these reason, an Open Loop system has poor performances. On the other hand, in Closed Loop systems the system behavior is taken into account, and differences from nominal, designed evolution are considered, resulting thus in better performances. It is quite obvious that satellite formation flying require Close Loop systems.

2.3.1 Guidance

In [34], Scharf et al. present a comprehensive survey of the guidance aspects of spacecraft formation flying. *Formation Flying Guidance* (FFG) aims at generating any reference trajectories, e.g. those for the translational and the attitude motion, used as an input for the relative state tracking control law of each member of the formation. Tacking into account the ambient dynamic environment in which a satellite formation operates, the FFG literature can be divided into two main categories.

The first main category is *Deep Space* (DS), where relative spacecraft dynamics reduce to double integrator form. As a consequence, any arbitrary rigid formations, e.g. those with a constant inter-spacecraft distances, can be maintained with no or poor fuel consumption. Optimization methods, aiming at both reducing the fuel consumption and, at the same time, guaranteeing the formation safety, are generally used for formation reconfiguration, rotating a rigid formation and planning u,v-coverages for *Multiple Spacecraft Interferometers* (MSIs).

The second main category is *Planetary Orbital Environments* (POE), where spacecraft are subjected to significant orbital dynamics and environmental disturbances. Since tracking arbitrary trajectories requires a continuous usage of the on board propulsion subsystem resulting in a prohibitive fuel consumption, the POE literature focuses on developing periodic, thrust-free relative

trajectories called *Passive Relative Orbits* (PROs). The effectiveness of a PRO, however, is dependent on the fidelity of the model used for its design. For example, if the main perturbative effects are not taken into account when designing PROs that the satellites must track for a long time period, extra fuel is consumed to correct the model errors.

For circular orbits, the most common linear PROs are obtained from the HCW equations, as described in Section 2.2. For non linear models, there are some similar initial condition constraints that allow to obtain a PRO about an eccentric reference orbit. Some approaches used in nonlinear PRO design are: numerically search of PROs either imposing an energy-matching condition, or introducing a formation performance metric, or using purely geometrical considerations for some SFF types. Another approach consists on first take the expansion of the formation geometry parameters in a series of eccentricity and then select relative orbital elements to eliminate those terms that cause the formation evaporation, i.e. first order terms.

Given a PRO, the next step is to study its robustness in the presence of disturbances, such as electric forces due to spacecraft charges, luni-solar gravitational perturbations or atmospheric drag for low formations. Two strategies that do yield a PRO when J_2 effects are included are: (i) to set the J_2-induced secular drifts of two orbits equal and derive constraints on the orbital elements, and (ii) to use dynamical system theory to select appropriate initial conditions. To improve the robustness of PROs designed using linear models, the HCW equations have been modified to include the effect of drag and J_2. However, for an eccentricity of 0.005, the error induced in the HCW equations due to ignoring eccentricity dominates the error due to ignoring J_2. The primary approach for incorporating both J_2 and reference orbit eccentricity is to express the relative motion in the local-vertical, local-horizontal frame as a function of the known solutions to the differential mean orbital elements. Osculating solutions require an eccentricity series-based approximation.

2.3.2 Navigation

In [35], Ferguson and How analyze and compare three basic estimation architectures for large satellite formation flying missions: Centralized, Decentralized and Hierarchic.

In the *Centralized Architecture*, one satellite is the master while the other satellites are slaves. Each slave vehicle sends its local measurements to the master satellite who process them in a centralized filter (Kalman or Information filters). Depending on the fleet mission requirements, the master may need to broadcast the estimation solutions to the slave vehicles for control and/or science use. Communication, computational and synchronization requirements of this type of architecture become prohibitive for large satellite fleets.

In a *Decentralized Architecture* the computational effort of the estimation is distributed more uniformly across the fleet thus reducing the communication requirements. Decentralized estimation

architectures can be divided into two classes: Full-Order and Reduced-Order filters. In a *Full-Order Decentralized Architecture*, each vehicle estimates the entire fleet state, either using measurements from every other vehicle in the fleet (Decentralized Information Filter), or using the locally available measurements (Iterative Cascade Filter based on a standard Kalman Filter). The former solution require high communication and computational efforts, while the last one is sub-optimal and don't provide a good balance between estimation accuracy and computational effort. In a *Reduced-Order Decentralized Architecture* (Cascade Filter or Schmidt-Kalman Filter), each vehicle estimates only its local state thereby substantially reducing the computational demands on each vehicle at the cost of sub-optimality and increased synchronization requirements.

The *Hierarchic Architecture* performs the detailed estimation for smaller groups of vehicles inside the same formation and then assimilates partial results at a higher level. In a two layers architecture, for instance, the fleet is divided into smaller clusters that perform their ranging and navigation independently with the exception of one master vehicle in each cluster. To link the estimates of each cluster to one another, each cluster master joins together to form a "super-cluster". Such a type of architecture does not need to have a high degree of synchronization between clusters and super-clusters. Cluster sizing and selection could be done based on several different criteria, including geographic separation, common GPS visibility or even existing communication connectivity from science experiments. The type of filter for the cluster estimators is chosen based on the cluster sizes, available communication bandwidth, CPU loading and required accuracy.

A viable relative estimation approach for large fleet future missions using augmented measurements would be comprised of reduced-order estimators implemented within a hierarchic architecture.

A commonly-used, highly accurate sensor for formation relative state estimation in LEO is the Global Positioning System (GPS). Recent works on Carrier-Phase Differential GPS for these applications [36] demonstrated 2 cm accuracy in relative position and better than 0.5 mm/s in relative velocity.

2.3.3 Control

Many different control strategies, schemes and applications of multiple vehicle control can be found in literature. While the applications are different, e.g. multiple robots, unmanned air vehicles, unmanned underwater vehicles and spacecrafts, the fundamental approaches for formation control are similar being the objective the *coordination of multiple vehicles*. As mentioned in Section 1.1, at least one spacecraft of the formation must track a desired state profile relative to another spacecraft and the associated tracking control law must at the minimum depend upon the state of this other spacecraft. A control law satisfying the last condition is called a *formation tracking control law*.

Based on the above definition of FF, Scharf et al. [37] present a comprehensive survey of the spacecraft Formation Flying Control (FFC) literature. Specifically, FFC refers to design techniques and associated stability results for formation tracking control laws. In [38], Lawton et al. overview the FFC literature up to 2000 and define three main FFC architectures. In [37], Scharf et al. examine these architectures in more detail and added two new architectures, including the prolific research of the last few years and emphasize theoretical developments.

2.3.3.1 Control Architectures for Satellite Formation Flying

The primary distinction in the FFC literature is the type of FFC architecture used. A formation control architecture can be in general defined as a coordination scheme. In [4], Scharf et al. describe five basic formation architectures, briefly described in the following.

Leader/Follower Architecture. In the Leader/Follower (L/F) architectureone satellite is designed as leader tracking a predefined trajectory while the other satellites are designated as followers following transformed versions of the states of their nearest neighbors according to formation geometry. This kind of architecture has also been referred to as Chief/Deputy, Master/Slave and Target/Chase[4]. The L/F has therefore a hierarchical organization in which only the leader vehicle knows the formation shape and goal.

The implementation of the L/F architecture is very simple, since individual spacecraft controllers reduces to tracking problems. However, there is no explicit feedback from the followers to the leader and the leader is a single point failure.

Multiple-Input, Multiple-Output Architecture. The formation is treated as a single multiple-input, multiple-output plant and all the methods of modem control may be applied to formation control; e.g. LQR controller, Directed Graphs and, more recently, nonlinear and constrained Model Predictive Control (MPC) strategies.

The primary advantages of the MIMO architecture are optimality and stability. However, the main disadvantages of this type of architecture are due to its high information requirement, since the entire state is used, and its poor robustness to local failure, since a local failure can have a global effect.

Cyclic. A formation controller in the Cyclic architecture is formed by connecting individual spacecraft controllers in a non-hierarchical architecture resulting in a *cyclic control dependency directed graph.*

[4]From the traditional terminology from two-spacecraft rendezvous.

The cyclic architecture basically lays between the L/F and the MIMO architectures and can perform better that L/F algorithms and distribute control effort more equally, since individual controllers are not hierarchically connected.

Virtual Structure Architecture. In Virtual Structure Architecture, the entire formation is treated as a single structure. For example, in an interferometry mission it may be desirable to have a constellation of spacecraft act as a single rigid body. In the virtual structure approach, the control is derived in three steps: first, the reference state trajectory of the structure is defined according to mission objectives; second, the motion of the virtual structure is translated into the reference motion for each satellite of the formation; and third, tracking control for each spacecraft is actuated.

Advantages of this formation control approach are that it is easy to prescribe the coordinated behavior for the group, and that the virtual structure can maintain the formation very well during the maneuvers in the sense that the virtual structure can evolve as a whole in a given direction with some orientation and maintain a rigid geometric relationship among multiple spacecrafts. Disadvantage is essentially related to the cases where formation shape is time-varying or it needs to be reconfigured frequently.

Behavioral Architecture. In the *Behavioral Architecture* all units work together to reach formation shape and goal on the basis of a given set of behavioral schemes. Possible behaviors include collision avoidance, obstacle avoidance, goal seeking and formation keeping.

Advantages of this formation control approach are that it is natural to derive control strategy when spacecraft have multiple competing objectives, explicit feedback is included through communication between neighbors. Disadvantages are that the group behavior cannot be explicitly defined, and it is hard to analyze the behavioral approach mathematically and guarantee its group stability.

As clearly stated in [38], these five control architectures may be viewed as special cases of a more general control architecture (block diagram is given in Figure 2.5), motivated by the existence of several levels of control in formation flying.

- *Spacecraft local control* (K_i, i index of spacecraft). The system S_i represents the i^{th} spacecraft, with control input vector u_i representing control forces and torques, and output vector y_i representing the measurable output of the spacecraft, most likely position and attitude vectors. The inputs to K_i are the output of the i^{th} spacecraft y_i and the coordination variable ξ. The outputs of K_i are the control vector u_i, and the performance variable z_i. Starting from coordinate variables ξ (i.e. reference state trajectory, commands) and measured spacecraft

output y (i.e. absolute and/or relative position and attitude) , it computes the command for its actuators u.

- *Formation control* (F): it represents the primary coordination mechanism in the system. The formation control block outputs the coordination variable ξ which is broadcast to all spacecraft. In addition, the formation control block outputs z_F , which encapsulates the performance of the formation, to the supervisor. The inputs to F are the performance variables from each spacecraft z_i, and the output of the supervisor u_G. Starting from supervisor outputs u_G and measured i spacecraft local control output z_i (performance variable, operating states), it computes the coordinate variables ξ.

- *Supervisor* (G): it is discrete-event control that uses the measured formation control output z_F to determine the input to the formation control u_G.

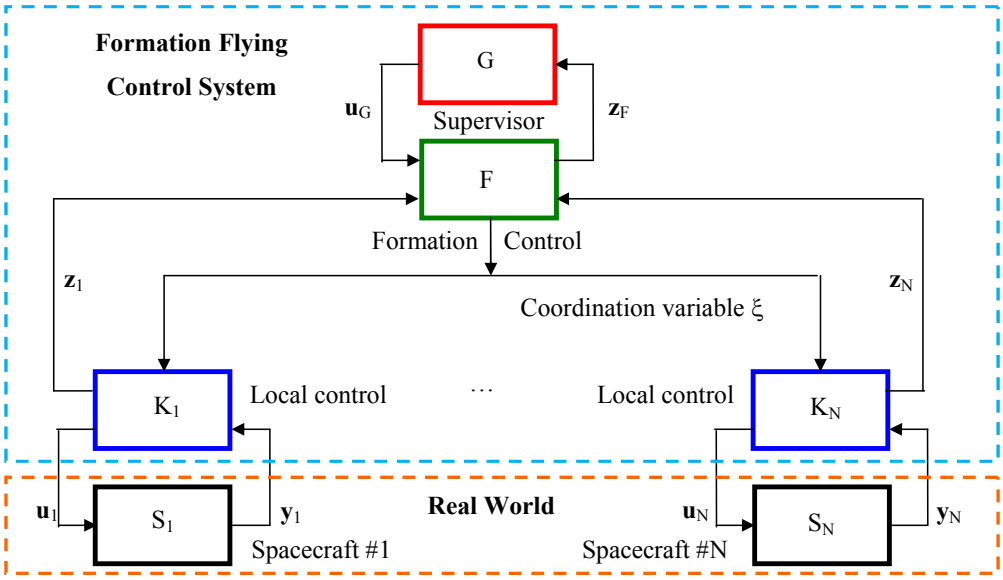


Figure 2.5: General Architecture for Spacecraft Formation Flying [38].

The formation control F is in charge of the coordination of each spacecraft present inside the formation. In general, it is in charge of the control of the absolute position control, the spacecraft relative control, the absolute attitude control and the relative attitude control in order to permits formation acquisition, formation keeping, formation maneuvers (nominal observation and contingency). The local control K_i is a slave of the formation control. It takes charge the spacecraft attitude and position control according to the outputs given by the formation control. The local control may be also a simple feed-through toward the actuators of the formation control outputs. The supervisor G is in charge of the mode transition according to the received tele-commands

coming from ground-segment, and events generated by formation control (failure detection events, convergence achieved, etc.).

2.3.3.2 Approaches for Formation Flying Control

There are basically three different ways or approaches[5] to control formations in space [39]:

- ground-based control;
- ground-in-the-loop control;
- fully space-borne autonomous control.

A *ground-based control* is primarily restricted to formations with a large separation and control windows. GRACE may be a good example of this way (nominal separation 200 km, control window about 50 km), where the required formation keeping maneuvers take place every few weeks. Avoiding the complexity of on-board autonomous systems, such maneuvers are efficiently executed by the satellite operations center.

Ground-in-the-loop control of formation might be necessary when high safety requirements have to be met, such as when a docking takes place involving manned missions, as for example the STS docking to the ISS.

A *fully autonomous space-borne control* of formation is a must when close formations (separation distance $<$ 1km) with tight control windows, related to formation reconfiguration and/or collision avoidance. Fully autonomous control is required also when the required relative position control accuracy requires real-time.

[5] Levels of control autonomy.

Chapter 3

CONTROL STRATEGIES

3.1 Introduction

Many approaches for controlling satellites flying in formation exist in the formation flying literature. In this chapter we present the main features of three types of controllers: Proportional-Integral-Derivative (PID), Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC). For each control type, the basic algorithms and the tuning of those parameters that determine the reference trajectory tracking performances and the fuel consumption are analyzed.

The following relative dynamics model is used in both LQR and MPC synthesis[1]. The system dynamics, which are described by the Hill's equations, can be written in state-space form as follows:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) \\ \\ \mathbf{y}(t) = \mathbf{C}\,\mathbf{x}(t) \end{cases} \tag{3.1}$$

where $\dot{\mathbf{x}} = [\,x,\, y,\, z,\, \dot{x},\, \dot{y},\, \dot{z}\,]^T \in \mathbb{R}^6$ is the state vector, composed by the three components of the relative position vector and the three components of the relative velocity vector, $\mathbf{u}(t) \in \mathbb{R}^3$ is the control input vector, $\mathbf{y}(t) \in \mathbb{R}^6$ is the output or controlled vector, $\mathbf{A} \in \mathbb{R}^{6\text{x}6}$ is the state matrix (Equation 2.9), and $\mathbf{B} \in \mathbb{R}^{6\text{x}3}$ and $\mathbf{C} \in \mathbb{R}^{6\text{x}6}$ are the control matrix and the output matrix, respectively:

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\text{x}3} \\ \\ \mathbf{I}_{3\text{x}3} \end{bmatrix} \tag{3.2}$$

$$\mathbf{C} = \mathbf{I}_{6\text{x}6} \tag{3.3}$$

[1] A PID controller does not use a model of the plant to be controlled.

The previous model is a Continuous-time Linear Time Invariant model (CLTI). It is supposed that the complete state vector is available for feedback purposes, although in real applications this is generally not true and a state observer, based on Kalman filtering techniques for instance, has to be used.

In real applications, a discrete-time version of the above model has to be used. If T_s is the sampling time of the control, the Discrete-time Linear Time Invariant system (DLTI) is therefore:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}_d\,\mathbf{x}_k + \mathbf{B}_d\,\mathbf{u}_k \\ \mathbf{y}_k = \mathbf{C}_d\,\mathbf{x}_k \end{cases} \tag{3.4}$$

where $\mathbf{A}_d$, $\mathbf{B}_d$ and $\mathbf{C}_d$ are the discrete-time matrices of the system obtained from the continuous ones as follows[2]:

$$\mathbf{A}_d = \mathbf{\Phi}(t)|_{t=T_s} \tag{3.5}$$

$$\mathbf{B}_d = \int_0^{T_s} \mathbf{\Phi}(t)\,\mathbf{B}(t)\,dt \tag{3.6}$$

where $\mathbf{\Phi}$ is the HCW continuous-time state transition matrix (Equation 2.11).

In general, optimal control schemes can be used to deal with either tracking problems or regulation problems. A *tracking problem* consists of designing a set of plant inputs $\mathbf{u}(t)$ that will force the system state, i.e. both position and velocity, to track a prescribed reference trajectory, which is often an unforced (or natural) motion to reduce the fuel consumption. On the other hand, a *regulator problem* is one that computes an input vector $\mathbf{u}(t)$ to transfer the system from a nonzero state to the zero state. Although the problem at hand is one of trajectory tracking, it can be cast as a regulator problem which is easier to deal with. The trajectory tracking problem can be seen in the light of a regulation problem for the instantaneous deviation between the actual and the desired trajectories. In other words, the objective is to drive the error between the desired and actual states to zero in an optimal fashion. The state dynamics of desired or reference trajectory $\mathbf{x}_r(t)$ can be expressed as follows:

$$\dot{\mathbf{x}}_r(t) = \mathbf{A}\,\mathbf{x}_r(t) \tag{3.7}$$

The state error, which is the difference between the actual state and the desired one, is defined as:

[2] A zero-order hold approach is adopted.

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}_r(t) \tag{3.8}$$

and the instantaneous error dynamics is then obtained:

$$\dot{\mathbf{e}}(t) = \mathbf{A}\,\mathbf{e}(t) - \mathbf{B}\,\mathbf{u}(t) \tag{3.9}$$

The trajectory tracking problem is therefore reduced to a regulation problem for the error dynamics. The aim is to reduce the state error to zero over an infinite horizon in an optimal manner. It is to note that the control is setup to tracking both position and velocity reference trajectory.

The conceptual structure of both LQR-based and MPC-based controllers is depicted in Figure 3.1. Both controllers use a state feedback in the optimal control low computation (optimizer in figure), but, for the problem on hand, while the LQR control law is computed off-line, the MPC control law is obtained, on the other hand, by solving on-line, at each sampling instant, a finite horizon open-loop optimal control problem with control, state and terminal constraints, in its general formulation.

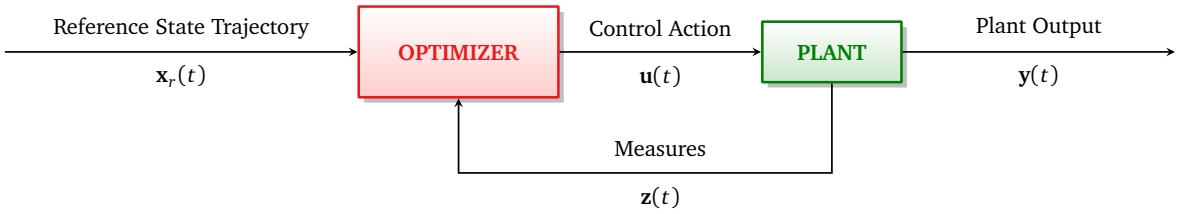


Figure 3.1: Basic structure of LQR and MPC controllers.

3.2 PID Control

For a sake of simplicity, the PID formulation presented in this section refers to a 1 degree of freedom (DOF) system. A more general multi degrees of freedom PID formulation can be obtained in a matrix form with the same basic structure.

A basic PID control law consists of the sum of three types of control actions: an action $u_p(t)$ proportional to the current control tracking error $e(t)$, an action $u_i(t)$ proportional to the integral of $e(t)$ and an action $u_d(t)$ proportional to the time derivative of $e(t)$:

$$u_{PID}(t) = u_p(t) + u_i(t) + u_d(t) = K_p\, e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d\, e(t)}{dt} \tag{3.10}$$

where $e(t) = r(t) - y(t)$ is the control tracking error, $y(t)$ is the process (or controlled) variable and $r(t)$ is the reference or set-point signal to track. K_p, K_i and K_d are the proportional, integral

and derivative gains, respectively. Thank to the proportional term, the control variable is increased when the control error is large. The integral or automatic reset term introduces a pole at the origin of the complex plane, allowing the reduction to zero of the steady-state error when a step reference signal is applied to the system or a step disturbance occurs. The derivative or anticipatory action is based on the predicted future values of the control error and can therefore anticipate an incorrect trend and counteract for it.

In practical cases, however, the previous basic PID formulation has some critical issues concerning proportional and derivative kick and wind up phenomena. An improved and more robust version of the PID control law in continuous-time form is (see Figure 3.2):

$$u(t) = K_p \left\{ \beta\, r(t) - y(t) + \frac{1}{T_i} \int_0^t \left[e(\tau) + \frac{T_i}{K_p T_t} \left(u'(t) - u(t) \right) \right] d\tau + T_d \left[\frac{d\left(\gamma\, r(t) - y_f(t)\right)}{dt} \right] \right\} \tag{3.11}$$

$$\frac{T_d}{N} \frac{dy_f(t)}{t} = y(t) - y_f(t) \tag{3.12}$$

where in this case $u(t)$ is the computed control variable and $u'(t)$ is the process control input. Also, $T_i = K_p/K_i$, $T_d = K_d/K_p$. In this PID formulation, a proportional set-point weight β and a derivative set-point weight γ are used to reduce the proportional and the derivative kick phenomena, respectively. In general $0 \leq \beta \leq 1$ and $0 \leq \gamma \leq 1$, even if γ is usually either 0 or 1 and $\beta = 0$ to avoid the proportional kick. A first-order low pass filer is used in order to reduce the influence of the measurement noise in the control variable.

A saturation model of the actuators is also used in an anti-reset windup feedback scheme to properly manage the actuators nonlinear behaviors introduces by saturation [40]. When the control variable attains the actuator limit during the transient response, the system operates in open-loop, since the actuator performs its maximum action independently of the process output value y. As a consequence, the control error decreases more slowly as in the ideal case and the integral term becomes large (it *winds up*). For this reason, the controller still saturates even when the process output matches the reference signal r, leading to large overshoots and settling times. Windup phenomenon can be avoided feeding back the difference of saturated and unsaturated control signal, so that the correct state of the controller u is estimated when it does not match the process input u'.

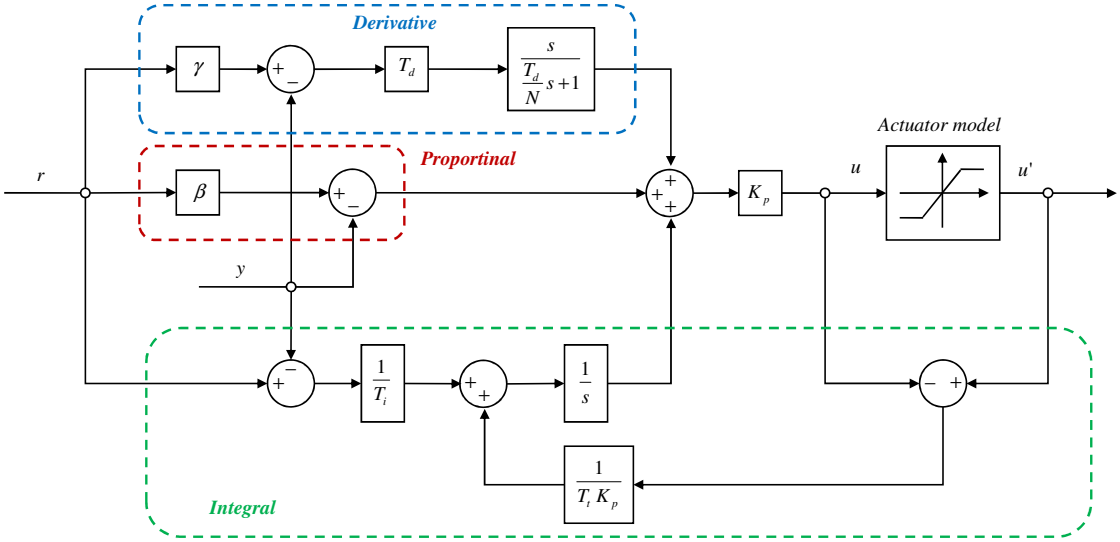


Figure 3.2: Block diagram of a 1 DOF PID control with proportional and derivative kick avoidance and anti-reset wind up.

3.3 Linear Quadratic Control

The purpose of LQR synthesis is to find a state feedback control law in the form $\mathbf{u}(t) = -\mathbf{K}\,\mathbf{e}(t)$, where $\mathbf{K}$ is the gain matrix, in order to minimize the following quadratic performance index:

$$J = \int_0^\infty \left[\mathbf{e}^T(\tau)\,\mathbf{Q}(\tau)\,\mathbf{e}(\tau) + \mathbf{u}^T(\tau)\,\mathbf{R}(\tau)\,\mathbf{u}(\tau)\right]\,d\tau, \tag{3.13}$$

where $\mathbf{Q}$ is a positive-definite matrix that represents the state error weighting matrix, and $\mathbf{R}$ is a positive-definite matrix representing the control weighting matrix. In our problem, $\mathbf{Q} \in \mathbb{R}^{6\times6}$ and $\mathbf{R} \in \mathbb{R}^{3\times3}$. The performance index J represents the global energy of the system and is the sum of two terms representing the error system energy and the control energy, respectively. Since (A, B) is controllable and (A, C) is observable, there is one and only one optimal controller, whose gain matrix is:

$$\mathbf{K}(t) = \mathbf{R}^{-1}(t)\,\mathbf{B}^T\,\mathbf{P}(t), \tag{3.14}$$

where the matrix $\mathbf{P}(t)$ is the solution of the continuous time Algebraic Riccati[3] Equation (ARE):

$$\mathbf{P}(t)\,\mathbf{A} + \mathbf{A}^T\mathbf{P}(t) - \mathbf{P}(t)\,\mathbf{B}\,\mathbf{R}^{-1}(t)\,\mathbf{B}^T\mathbf{P}(t) + \mathbf{Q}(t) = \mathbf{0}. \tag{3.15}$$

The discrete-time version of the previous equations that describe a LQR controller are as follow:

[3]Venice born Italian mathematician Jacopo Francesco Riccati (1676 - 1754)

$$J_d = \sum_{k=0}^{\infty} \left[\mathbf{e}_k^T \, \mathbf{Q}_k \, \mathbf{e}_k + \mathbf{u}_k \, \mathbf{R}_k \, \mathbf{u}_k \right] \tag{3.16}$$

$$\mathbf{u}_k = -\mathbf{K}_{d,k} \, \mathbf{e}_k \tag{3.17}$$

$$\mathbf{K}_{d,k} = \left(\mathbf{R}_k + \mathbf{B}^T \, \mathbf{P}_k \, \mathbf{B} \right)^{-1} \mathbf{B}^T \, \mathbf{P}_k \, \mathbf{A} \tag{3.18}$$

where $\mathbf{P}_k$ is the unique positive definite solution to the Discrete time Algebraic Riccati Equation (DARE):

$$\mathbf{A}^T \mathbf{P}_k \mathbf{A} - \mathbf{P}_k - \left(\mathbf{A}^T \mathbf{P}_k \mathbf{B} \right) \left(\mathbf{R}_k + \mathbf{B}^T \mathbf{P}_k \mathbf{B} \right)^{-1} \left(\mathbf{B}^T \mathbf{P}_k \mathbf{A} \right) + \mathbf{Q}_k = \mathbf{0} \tag{3.19}$$

Both $\mathbf{Q}$ matrix and $\mathbf{R}$ matrix are arbitrary parameters that have to be adjusted related to the system on hand and to the performances one would like to obtain in tracking the desired trajectory, also taking into account the scenario one is studying and the among of control action used to obtain the desired result, that means fuel consumption in our case. In particular, the values of $\mathbf{Q}$ elements in relation to those of $\mathbf{R}$ is important: higher values of $\mathbf{Q}$ elements with respect to $\mathbf{R}$ elements means an higher weight on state error with respect to weight on control action, and this lead to a more aggressive control, a better performance in tracking the desired trajectory but a more consumption of fuel. Furthermore, a LQR controller would be expected to give a continuous control action in response to system uncertainty, incurring in fuel penalty over maneuver-planning controllers. To prevent continuous firing, an LQR controller would likely be combined with a dead band. For example, in collision-avoidance applications, such as those presented in Chapter 6, the reference trajectories have not to be always tracked with high accuracy. An higher tracking accuracy may be imposed near delta-v applications: if relative position and velocity are close to the desired ones, then the deputy spacecraft tends to naturally follow the reference trajectory.

3.4 Model Predictive Control

3.4.1 Introduction

Model Predictive Control (MPC) [41, 42, 43], often referred to as Receding Horizon Control (RHC), is a modern optimal control technique that has been widely adopted in industry as an effective means to deal with multivariable constrained control problems [44, 45, 46].

MPC is based on the idea, illustrated in Figure 3.3, of employing an explicit model of the plant to be controlled and a sample of the current state $\widehat{\mathbf{x}}_{t_k}$ as the initial state to predict the

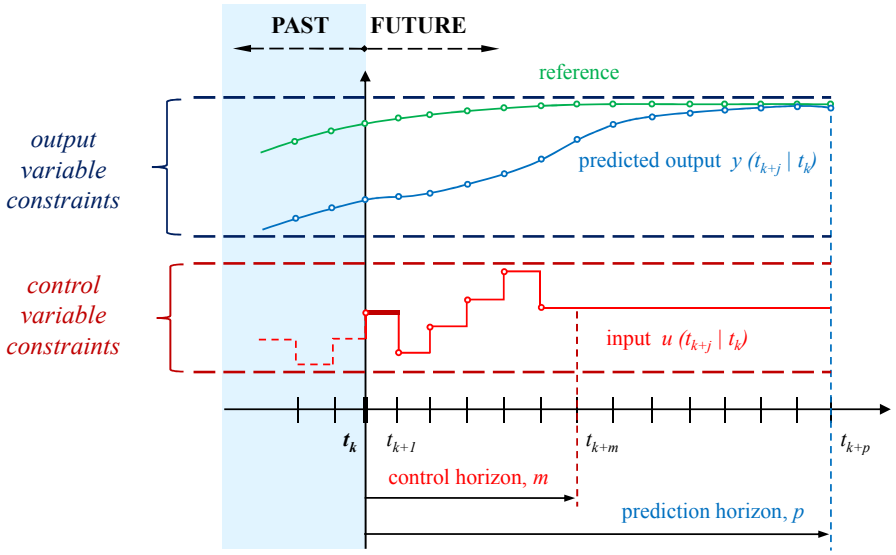


Figure 3.3: Basic idea behind Model Predictive Control.

future output behavior $\mathbf{y}(t)$ over a finite horizon p as a function of a control sequence. The control signal $\mathbf{u}(t)$ is assumed to change only within the interval $[t_k, t_{k+m}]$, remaining constant for the next $p - m$ control steps, with $m \leq p$. p and m are referred to as the *prediction horizon* and the *control horizon*, respectively. Using this prediction capability, the current control action to be applied to a dynamic system is obtained by solving *on-line*, at *each* sampling instant, a finite horizon open-loop optimal control problem. The optimization process yields an optimal control sequence $\mathcal{U}_k^\star(\widehat{\mathbf{x}}_{t_k}) \doteq \{\mathbf{u}_{t_k}^\star, \mathbf{u}_{t_{k+1}}^\star, ..., \mathbf{u}_{t_{k+m}}^\star\}$ that minimizes a certain cost measure for the system over the prediction horizon. According to a receding horizon philosophy, only the first of those control actions $\mathbf{u}_{t_k}^\star$ is applied to the plant while the remaining optimal inputs are discarded. In order to compensate for possible modeling errors or disturbances acting on the system, a new state estimate $\widehat{\mathbf{x}}_{t_{k+1}}$ is taken at the next sampling instant and a new optimal control problem is solved at the next sampling instant, thereby introducing feedback into the system.

From the above explanations, it is clear that a fixed prediction horizon is shifted or receded over time, hence the name Receding Horizon Control. The procedure of this on-line optimal control technique is summarized in the Algorithm 3.1.

The key difference between MPC and classical control techniques is its ability to handle constraints, which are inherent in nearly every real application. For example, actuators are naturally limited in the force (or equivalent slew rates) they can apply. Alternatively, operating limits may be imposed for safety or efficiency reasons. The presence of such constraints can render the off-line determination of a control law very difficult, whereas the MPC on-line nature is more adept at handling such problems, as constraints are handled naturally within the optimization framework.

Algorithm 3.1 On-line Receding Horizon Control Algorithm.

1. Estimate the state vector $\widehat{\mathbf{x}}_{t_k}$ at time instant t_k;
2. Obtain $\mathcal{U}_k^\star(\widehat{\mathbf{x}}_{t_k})$ by solving an optimization problem over the horizon p;
   IF $\mathcal{U}_k^\star(\widehat{\mathbf{x}}_{t_k}) = \emptyset$, THEN problem infeasible and STOP.
3. Apply the first element $\mathbf{u}^\star(t_k)$ of $\mathcal{U}_p^\star(\widehat{\mathbf{x}}_{t_k})$ to the system;
4. Wait for the new sampling time t_{k+1}, goto (1.).

It is this re-planning nature, i.e. the explicit consideration of the system dynamics, and constraint-handling ability that makes MPC well-suited also to aerospace problems, allowing fuel-efficient, feasible plans to be determined autonomously and on-line.

In the following, after a brief review of MPC applications to SFF for both translational and attitude motions (Section 3.4.2), the basic formulation of a MPC problem is presented (Section 3.4.3). The next Sections 3.4.4 and 3.4.5 describe how a MPC problem can be solved on-line in terms of a Quadratic Programming (QP) problem. The last Section 3.4.6 deals with an explicit solution to an MPC problem. The difference between an on-line MPC solution and an explicit MPC solution is as follows. An on-line MPC solution involves the use of an iterative algorithm at each control step to compute the MPC control action. On the other hand, an explicit MPC solution means that the MPC problem is first solved off-line and the MPC control action is computed in real time at each control step by means of a simple table-lookup whose data are stored in the on board computer.

3.4.2 Review of MPC applications for formation flight

Many papers on MPC applications for SFF consider formations flying in a circular reference orbit ($e = 0$) and the Hill-Clohessy-Wiltshire equations, i.e a LTI dynamic model, are used to model the formation relative dynamics. Several contributions in the field of formation control come from the Aerospace Control Laboratory at MIT. Papers such as Inalhan et al. [17], Tillerson et al. [47], Richards et al. [48], Tillerson [49], Richards and How [50, 51, 52] and Breger et al. [53, 54] deals with optimization-based trajectory planning for formation flight and autonomous rendez-vous and constitute relevant examples of how to formulate objectives, models and constraints for MPC in these research fields. Other contributions come from Rossi and Lovera [55] and Prieto and Ahmed [56].

For what concern general eccentric orbits ($0 \leq e \leq 1$), Tillerson et al. [47] used the Lawden equations to compute at each sampling time instant the dynamics model as function of the true anomaly ϑ. The resulting dynamics is then discretized assuming ϑ as constant over a certain period

an performing on-line the conversion between the true anomaly domain and the time domain. To reduce the computational effort of this approach, Larsson et al. [57] proposed an approach based on the Yamanaka-Ankersen STM [19]. Also, in [58], the relative dynamics is written with explicit time-dependence by using the Kepler's equation to create a true anomaly – time map.

Breger and How proposed in [24] a MPC formulation for formation control with a system model based on LTV Gauss Variational Equations (GVE's) extended to include the effects of J_2, showing that the resulting control is more fuel-efficient than the Lawden equation-based control scheme in highly eccentric orbits with large rectilinear separations. In fact, in GVE's the motion is linearized about orbital elements, which are expressed in a curvilinear frame. As a consequence, large rectilinear distances are expressed as small perturbations to orbital elements, avoiding the linearization errors introduced in the Hill and Lawden approaches at large separations.

In Saponara et al. [59] and Hartley et al. [60], the authors described the design and implementation of a MPC system for the rendez-vous and capture associated with the Mars Sample Return (MSR) mission, including the implementation of the developed solution in the space representative avionic architecture system ORCSAT (On-line Reconfiguration Control System and Avionics Technologies). The MPC control system is designed to be used from the point of target detection to the point of target capture and to function in both circular and elliptical orbits. To achieve an efficient system design, the rendez-vous maneuver has been partitioned into three main phases based on the range of operation. For the MPC design, the following relative dynamics prediction models were used: J_2 modified GVE's for long range dynamics and the Yamanaka-Ankersen STM for short range dynamics. The authors showed how MPC significantly improves the performances both in trajectory generation and in propellant save.

MPC was also used for attitude control of spacecraft. Some authors, such as Manikonda et al. [61], Soest et al. [62], Oort et al. [63] and Shi and Kelkar [64], combined a linear MPC and a local feedback linearization that is used to provide a linear model around a reference state. Also, Crassidis et al. [65] apply a form of non-linear MPC to three-axis attitude control for large-angle maneuvers of a rigid-body spacecraft. Hegrenaes et al. [66] used an explicit solution to constrained MPC to control the attitude of the SSETI/ESEO[4] micro-satellite. The nonlinear attitude dynamics were linearized about an equilibrium point to obtain a linear model for MPC and a cold thruster system plus reaction wheels were used to control the satellite attitude.

MPC was also proposed as a technique for attitude control by magnetic-torque actuators in Silani and Lovera [67] and Wood and Chen [68, 69]. In such applications, we have time-varying actuation limits coming from the time variation of the Earth magnetic field over the satellite orbit.

[4]Student Space Exploration and Technology Initiative (SSET), European Student Earth Orbiter (ESEO).

3.4.3 Basic MPC problem formulation

To make predictions as part of the control decision-making process, the basic MPC formulation shall use the following linear time-invariant model of the system:

$$x_{k+1|k} = A\,x_{k|k} + B\,u_k \tag{3.20}$$

$$y_{k|k} = C\,x_{k|k} \tag{3.21}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_y}$. $a_{k+j|k}$ denotes the value of a predicted for time t_{k+j} based on the information available at time t_k. We define $\Delta u_{k+j|k} = u_{k+j|k} - u_{k+j-1|k}$ as the difference between two consecutive control vectors, that is the control variable increment between the previous step and the next one, $e_{y,k+j|k} = y_{k+j|k} - y_{r,k+j|k}$ as the output variable tracking error with respect to a reference output y_r, if any, and $e_{u,k+j|k} = u_{k+j|k} - u_{r,k+j|k}$ as the difference between the input and the reference control, if any.

MPC solves on-line the following optimization problem for a given sample of the state $\widehat{x}_k$ at time t_k:

$$V^{\star}(x_{k|k}) = \min_{\Delta\mathcal{U}_k} \; \mathcal{M}(e_{y,k+p|k}) + \sum_{j=0}^{p-1} \mathcal{L}_j(e_{y,k+j|k}, e_{u,k+j|k}, \Delta u_{k+j|k}) \tag{3.22}$$

subject to, $\forall j = 0, 1, ..., p-1$:

$$x_{k+j+1|k} = A\,x_{k+j|k} + B\,\Delta u_{k+j|k} \tag{3.23}$$

$$y_{k+j|k} = C\,x_{k+j|k} \tag{3.24}$$

$$x_{k|k} = \widehat{x}_k \tag{3.25}$$

$$\begin{bmatrix} E & G \end{bmatrix} \begin{bmatrix} y_{k+j|k} \\ \\ u_{k+j|k} \end{bmatrix} \leq L \tag{3.26}$$

$$H\,\Delta u_{k+j|k} \leq M \tag{3.27}$$

$$\Delta u_{k+h} = 0\,, \quad \forall h = m, ..., p-1 \tag{3.28}$$

with an optional terminal constraint:

$$E_f\,y_{k+p|k} \leq L_f \tag{3.29}$$

The objective function is quadratic with:

$$\mathcal{M}(e_y) = e_y^T \, P \, e_y \tag{3.30}$$

$$\mathcal{L}_j(e_y, e_u, \Delta u) = e_y^T \, Q_j \, e_y + e_u^T \, R_{u,j} \, e_u + \Delta u^T \, R_{\Delta u,j} \, \Delta u \tag{3.31}$$

The solution is the optimal control sequence:

$$\Delta \mathcal{U}_k^\star \doteq \{\Delta u_{k|k}^\star, \Delta u_{k+1|k}^\star, ..., \Delta u_{k+m-1|k}^\star\} \tag{3.32}$$

and the MPC control law can thus be written as:

$${}^{MPC}u_k = \widehat{u}_{k-1} + \Delta u_{k|k}^\star \tag{3.33}$$

Equation 3.26 is a general linear inequality (polyhedral) constraint that can be used to represent many different types of constraints on y and u, including upper/lower bounds and box constraints. Constraints on Δu (Equation 3.27) are used to avoid excessive variations on u, i.e. to smooth the control action profile, reducing for example the process noise due to control.

When designing a MPC controller one should take into account the following: a more aggressive controller with higher fuel consumption and better tracking performances is obtained with smaller prediction horizon p, tighter constraints on control variables and higher ratio between state error weight and control action weight; a larger control horizon m leads to an optimization problem with more degrees of freedom, thus a "more" optimal (better performance) but more complex (computationally demanding) control.

The values of m and p are therefore a trade-off solution between: (a) optimality of the MPC solution and performances of the controller, e.g. reference trajectory tracking accuracy or fuel consumption; and (b) on board resources consumption in terms of computing power / time requirements (related also to electric power consumption) and memory requirements for data storage. In the scenarios presented in the next chapters, the length of the control horizon and the prediction horizon are the results of such a trade-off solution. For experimental tests, a preliminary tuning of the MPC parameters was conduced using a software simulator. The resulting MPC control law was then tested in the real hardware to validate the proposed solution and understand the effects of the non-modeled disturbances.

3.4.4 From a MPC problem to a Quadratic Programming problem

As you can see from the previous subsection, a MPC optimization problem is a Quadratic Programming (QP) problem with linear inequality constraints. To be solved, it must be re-written into the following standard QP problem form, for which many solvers are available:

$$\begin{aligned} \min_{X} \ & \tfrac{1}{2} X^T \mathcal{H} X + \mathcal{G} X^T \\ s.t. \ & \Omega X \leq \omega \end{aligned} \tag{3.34}$$

The following definition are given:

- $\mathcal{Z}_k = [\, y_{k|k}^T \,, \ldots, \, y_{k+p|k}^T \,]^T$ is the sequence of the controlled variables within the prediction horizon;
- $\mathcal{T}_k = [\, y_{r,k|k}^T \,, \ldots, \, y_{r,k+p|k}^T \,]^T$ is the sequence of the reference (or target) for the controlled variables within the prediction horizon;
- $\mathcal{U}_k = [\, u_{k|k}^T \,, \ldots, \, u_{k+p-1|k}^T \,]^T$ is the sequence of the control variables within the prediction horizon;
- $\Delta\mathcal{U}_k = [\, \Delta u_{k|k}^T \,, \ldots, \, \Delta u_{k+m-1}^T \,]^T$ is the sequence of the control variables increments within the control horizon;
- $\mathcal{U}_{r,k} = [\, u_{r,k|k}^T \,, \ldots, \, u_{r,k+p-1}^T \,]^T$ is the sequence of the target control variables within the prediction horizon.

Using Equations 3.20 and 3.21, $\mathcal{Z}_k$ can be written as:

$$\mathcal{Z}_k = \Psi x_{k|k} + \Upsilon u_{k-1} + \Theta \Delta\mathcal{U}_k \tag{3.35}$$

where:

$$\Psi = \begin{bmatrix} CA \\ \vdots \\ CA^{m-1} \\ CA^{m} \\ \vdots \\ CA^{p} \end{bmatrix}; \Upsilon = \begin{bmatrix} CB \\ \vdots \\ \sum_{i=0}^{m-1} CA^{i}B \\ \sum_{i=0}^{m} CA^{i}B \\ \vdots \\ \sum_{i=0}^{p-1} CA^{i}B \end{bmatrix}; \Theta = \begin{bmatrix} CB & \cdots & 0 \\ CAB + CB & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{m-1} CA^{i}B & \cdots & CB \\ \sum_{i=0}^{m} CA^{i}B & \cdots & CAB + CB \\ \vdots & \vdots & \vdots \\ \sum_{i=0}^{p-1} CA^{i}B & \cdots & \sum_{i=0}^{p-m} CA^{i}B \end{bmatrix}.$$

Given an estimation of the current state and the control variable at the previous control step, using the previous equation it is possible to compute the sequence of the controlled variable within the prediction horizon for any given sequence of the control variable. We introduce also the following "free tracking error" vector:

$$\varepsilon_k = \mathcal{T}_k - \Psi x_{k|k} - \Upsilon u_{k-1} \tag{3.36}$$

which is the difference between the future target trajectory and the free response of the system, namely the response that would occur over the prediction horizon if no input changes were made, that is if $\Delta\mathcal{U}_k = 0$.

Cost function in standard form

The cost function given in Equation 3.22 can be written in matrix form as:

$$V_k = \| \mathcal{Z}_k - \mathcal{T}_k \|^2_{\mathcal{W}_y} + \| \Delta\mathcal{U}_k \|^2_{\mathcal{W}_{\Delta u}} + \| \mathcal{U}_k - \mathcal{U}_{r,k} \|^2_{\mathcal{W}_u} + \rho_\epsilon \epsilon_k^2 \tag{3.37}$$

where $\mathcal{W}_y$, $\mathcal{W}_{\Delta u}$ and $\mathcal{W}_u$ are the output variable sequence, the controlled variable increment sequence and the control variable sequence weighting matrices computed from matrices Q_j, $R_{u,j}$ and $R_{\Delta u,j}$ respectively. $\epsilon \geq 0$ is a slack variable used to relax the constraints on u, Δu and y.

This cost function can to be rewritten as a quadratic fiction of $\Delta\mathcal{U}_k$ and ϵ. First, we can write $\mathcal{U}_k$ as a function of $\Delta\mathcal{U}_k$ as:

$$\mathcal{U}_k = \Lambda \Delta\mathcal{U}_k + \mathbf{1}\, u_{k-1} \tag{3.38}$$

where:

$$\Lambda = \begin{bmatrix} I_{n_u \times n_u} & 0_{n_u \times n_u} & \cdots & \cdots & 0_{n_u \times n_u} \\ I_{n_u \times n_u} & I_{n_u \times n_u} & \ddots & \cdots & 0_{n_u \times n_u} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ I_{n_u \times n_u} & I_{n_u \times n_u} & \cdots & I_{n_u \times n_u} & 0_{n_u \times n_u} \\ I_{n_u \times n_u} & I_{n_u \times n_u} & \cdots & I_{n_u \times n_u} & I_{n_u \times n_u} \end{bmatrix} \in \mathbb{R}^{n_u m \times n_u m} \tag{3.39}$$

is the map from $\mathcal{U}_k$ to $\Delta\mathcal{U}_k$, and:

$$\mathbf{1} = \left.\begin{bmatrix} I_{n_u \times n_u} \\ \vdots \\ I_{n_u \times n_u} \end{bmatrix}\right\} m \text{ times} \tag{3.40}$$

Then, we can write:

$$\mathcal{U}(k) - \mathcal{U}_t(k) = \Lambda \Delta\mathcal{U}(k) + \mathbf{1}\, u(k-1) - \mathcal{U}_t(k) = \Lambda\, \Delta\mathcal{U} + \mu(k) \tag{3.41}$$

where we have defined:

$$\mu_k = \mathbf{1}\, u_{k-1} - \mathcal{U}_{r,k} \tag{3.42}$$

Taking into account Equation 3.36 and after some cumbersome calculation it is possible to write V_k as:

$$V_k = \Delta\mathcal{U}_k^T\, \tilde{\mathcal{H}} \Delta\mathcal{U}_k + \Delta\mathcal{U}_k^T\, \tilde{\mathcal{G}}_k + \rho_\epsilon\, \epsilon_k^2 + c_k \tag{3.43}$$

where we have defined:

$$\tilde{\mathcal{H}} = \Theta^T \mathcal{W}_y \Theta + \mathcal{W}_{\Delta u} + \Lambda^T \mathcal{W}_u \Lambda \tag{3.44}$$

$$\tilde{\mathcal{G}}_k = -2\Theta^T \mathcal{W}_y \varepsilon_k + 2\Lambda^T \mathcal{W}_u \mu_k \tag{3.45}$$

$$c_k = {\varepsilon_k}^T \mathcal{W}_y \varepsilon_k + {\mu_k}^T \mathcal{W}_u \mu_k \tag{3.46}$$

At this point we introduce the independent variable of the optimization problem:

$$X_k = [\, \Delta \mathcal{U}_k^T,\, \epsilon_k \,]^T \tag{3.47}$$

The cost function in a standard form for the QP solver results:

$$V_k = X_k^T \, \mathcal{H} \, X_k + X_k^T \, \mathcal{G}_k \tag{3.48}$$

where:

$$\mathcal{H} = \begin{bmatrix} \tilde{\mathcal{H}} & 0 \\ 0 & \rho_\epsilon \end{bmatrix} \tag{3.49}$$

$$\mathcal{G}_k = \begin{bmatrix} \tilde{\mathcal{G}}_k \\ 0 \end{bmatrix} \tag{3.50}$$

Note that c_k has been dropped off since constant[5] and it is not necessary to calculate its value.

Constraints in standard form

Let us assume a constant polyhedral constraint set, i.e. representable as a time-invariant linear inequality system[6]. At each time step $k+j$, the constraint set can therefore be written as:

$$\Gamma \begin{bmatrix} y_{k+j|k} \\ u_{k+j|k} \\ \Delta u_{k+j|k} \end{bmatrix} \leq \gamma + \epsilon \, \mathcal{V} \tag{3.51}$$

where $\mathcal{V}$ is the vector of the Equal Concern for the Relaxation (ECR) parameters that allow to soft the corresponding constraint: the larger $\mathcal{V}_\mathrm{i}$, the softer the corresponding constraint.

Using Equations 3.35 and 3.38, the inequality constraints for all time steps within the prediction horizon can be written as:

$$\Omega \, X \leq \omega_k \tag{3.52}$$

where $\omega_k = \omega_x \, x_{k|k} + \omega_{u_{k-1}} \, u_{k-1} + \omega_{cnst}$.

[5] The value of c_k is different for each time step.

[6] This assumption is quite general and allows to represent a wide variety of constraint sets.

3.4.5 On-line MPC Engine

This subsection describes the MPC engine that was used to execute the rendez vous and capture maneuver for the SPHERES MOSR scenario (see Chapter 7).

In order to efficiently implement the MPC algorithm and consequently reduce the cumputing time, which is very important for real-time applications, we need to understand which QP solver input elements must be updated at each control step and which ones are constant, since not dependent from the current dynamic state of the system.

The QP problem in standard form is solved using the Dantzig-Wolfe's active set method [70]. Dantzig-Wolfe's inputs and outputs are listed in Table 3.1.

Table 3.1: Dantzig-Wolfe's active set method inputs and outputs.

Inputs	
tabi	initial tableau
basi	initial basis
ibi	initial setting of the variables index vector
ili	initial setting of the Lagrange multipliers index vector
maxiter	maximum number of iterations

Outputs	
tab	final tableau
bas	final basis vector
ib	index vector for the variables
il	index vector for the Lagrange multipliers
iter	iteration counter

We need first to rewrite both the cost function and the linear inequality constraint as explicit functions of the time-dependent parameters of the MPC problem, i.e. reference trajectories and dynamic state of the system. The only time-dependent term of the cost function is $\tilde{\mathcal{G}}_k$ (Equation 3.45), which can be rewritten as:

$$\tilde{\mathcal{G}}_k = \tilde{\mathcal{G}}_{\mathcal{T}}\,\mathcal{T}_k + \tilde{\mathcal{G}}_x\,x_{k|k} + \tilde{\mathcal{G}}_{u_{k-1}}\,u_{k-1} + \tilde{\mathcal{G}}_{u_r}\,\mathcal{U}_{r,k} \tag{3.53}$$

For what concern the constraints system, let v be a variable on which a constraint is imposed, i.e. either y or Δu or u. If we assume that a polyhedral constraint is imposed on each of these

variables individually, then each constraint can be written as:

$$\Gamma_v \, v_{k+j|k} \leq \gamma_v \, + \epsilon \, \mathcal{V}_v \tag{3.54}$$

After some cumbersome calculations, it is possible to write this constraint as a linear inequality system in the optimization variable X for all time steps of the prediction horizon as:

$$\left[\begin{array}{c|c} \mathcal{F}_v & f_{v,\epsilon} \end{array} \right] X \leq \mathcal{F}_{v,x_k} \, x_k + \mathcal{F}_{u,u_{k-1}} \, u_{k-1} + f_v \tag{3.55}$$

Ω and ω_k matrices in Equation 3.52 result:

$$\Omega = \left[\begin{array}{cc} \mathcal{F}_y & f_{y,\epsilon} \\ \mathcal{F}_u & f_{u,\epsilon} \\ \mathcal{F}_{\Delta u} & f_{\Delta u,\epsilon} \\ 0 & -1 \end{array} \right] \tag{3.56}$$

$$\omega_k = \left[\begin{array}{c} \mathcal{F}_{y,x_k} \, x_k + \mathcal{F}_{y,u_{k-1}} \, u_{k-1} + f_y \\ \mathcal{F}_{u,u_{k-1}} \, u_{k-1} + f_u \\ f_{\Delta u} \\ 0 \end{array} \right] = \left[\begin{array}{c} \omega_{y,k} \\ \omega_{u,k} \\ \omega_{\Delta u} \\ 0 \end{array} \right] \tag{3.57}$$

The following parameters are defined:

$$mnu = n_u \, m + 1 \tag{3.58}$$

$$nc = 4 \, n_u \, m + n_{cy} p + 1 \tag{3.59}$$

where mnu is the dimension of the optimization vector, n_{cy} is the total number of constraints on y and nc is the total number of constraints of the QP problem.

The QP solver input matrices *tabi* and *basi* are computed from $x_{k|k}$,u_{k-1} , $\mathcal{T}$ and the MPC problem parameters as follows:

$$tabi = \begin{bmatrix} -\mathcal{H}^{-1} & \mathcal{H}^{-1}\,\Omega^T \\ \Omega\,\mathcal{H}^{-1} & -\Omega\,\mathcal{H}^{-1}\,\Omega^T \end{bmatrix} = \text{const} \in \mathbb{R}^{(mnu+nc)\,\mathrm{x}\,(mnu+nc)} \tag{3.60}$$

$$basi_k = bas_{\mathcal{T}}\,\mathcal{T} + bas_x\,x_{k|k} + bas_{u_{k-1}}\,u_{k-1} + bas_{cnst} \in \mathbb{R}^{(mnu+nc)\,\mathrm{x}\,1} \tag{3.61}$$

where bas terms are matrices that can be computed using $\mathcal{F}$'s and f's matrices of Equation 3.52 and $\tilde{\mathcal{G}}$'s matrices of Equation 3.45. A more detailed description of these mathematical passages is provided in [71].

The data flow of the MPC engine is presented in Figure 3.4. As you can see, only the $basi_k$ matrix has to be updated at each control step before calling the QP solver, while $tabi$ and bas's matrices remain constant and can be computed off-line as functions of the MPC parameters and the dynamic system.

Once the QP solver has provided the solution matrices bas and il, the sequence of the optimal control variable increments $\Delta\mathcal{U}_k^\star$ can be computed as:

$$\Delta\mathcal{U}_k^\star(j) = bas(\,il(j)\,) + xmin(j) \tag{3.62}$$

where (j) means the j^{th} component of the corresponding vector. The current optimal control action is then computed as:

$$^{MPC}u_k^\star = \widehat{u}_{k-1} + \Delta u_{k|k}^\star \tag{3.63}$$

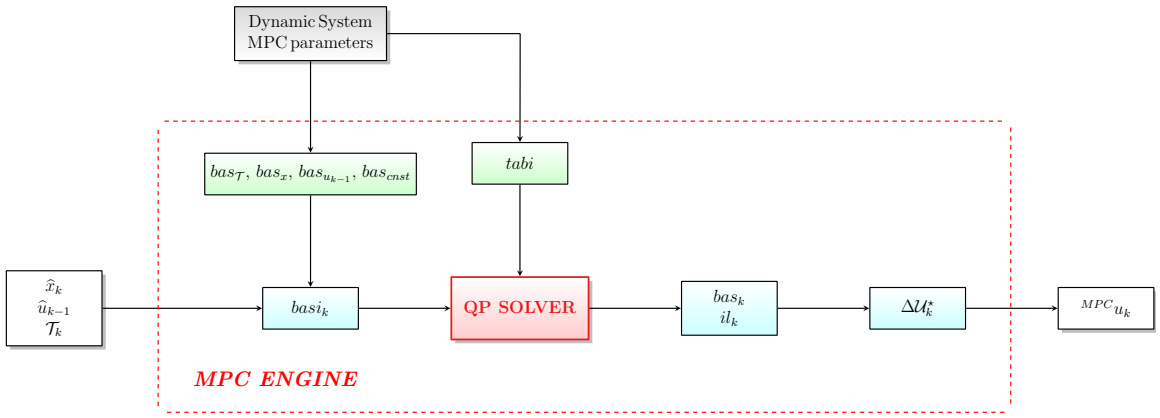


Figure 3.4: Block diagram of the MPC engine data flow. Off-line computations in green, on-line computations in cyan and red.

3.4.6 Explicit Model Predictive Control

In [72] Bemporad et al. show that an explicit solution to constrained LTI MPC problems can be computed off-line by solving a multi-parametric Quadratic Program (mpQP), thus allowing to run predictive controllers in systems with limited power and CPU resources.

Let us consider the following MPC regulation problem:

$$\min_{\mathcal{U}_k} \quad J(\mathcal{U}_k, x_k) \tag{3.64}$$

$$s.t. \quad y_{min} \leq y_{k+i|k} \leq y_{max} \quad i = 1, ..., p \tag{3.65}$$

$$u_{min} \leq u_{k+i} \leq u_{max} \quad i = 1, ..., m \tag{3.66}$$

$$x_{k|k} = x_k \doteq \widehat{x}(t_k) \tag{3.67}$$

$$x_{k+i+1|k} = A\, x_{k+i|k} + B\, u_{k+i} \quad i \geq 0 \tag{3.68}$$

$$y_{k+i|k} = C\, x_{k+i|k} \tag{3.69}$$

$$u_{k+i} = K\, x_{k+i|k} \quad m \leq i \leq p-1 \tag{3.70}$$

where

$$J(\mathcal{U}_k, x_k) = x_{k+p|k}^T\, P\, x_{k+p|k} + \sum_{i=0}^{p-1} \left\{ x_{k+i|k}^T\, Q\, x_{k+i|k} + u_{k+i}^T\, R\, u_{k+i} \right\} \tag{3.71}$$

$\mathcal{U}_k = [u_k^T, ..., u_{k+m-1}^T]^T \in \mathbb{R}^{mn_u}$ and (A, B) is a stabilizable pair. When the final cost matrix P and gain matrix K are computed from the algebraic Riccati equation, assuming that the constraints are not active for $i \geq m$ and $i \geq p$, the MPC problem exactly solves the constrained infinite horizon LQR problem for the system on hand, with weigh matrices Q and R.

By treating x_k as a vector of parameters ϑ_k, Bemporad et al. showed that the MPC problem can be rewritten as the following mpQP[7]:

$$\min_{\mathcal{U}_k} J(\mathcal{U}_k, \vartheta_k) = \min_{U_k} \frac{1}{2}\, \mathcal{U}_k^T\, H\, \mathcal{U}_k + \vartheta_k^T\, F\, \mathcal{U}_k \tag{3.72}$$

$$s.t. \quad G\, \mathcal{U}_k \leq W + S\, \vartheta_k \tag{3.73}$$

where matrices H, F, G, W and S are obtained from the MPC problem as described in Section 3.4.4. It is important to note that this mpQP problem is strictly convex and the Karush-Kuhn-

[7] Problems depending on a vector of parameters are called multi-parametric programs in the jargon of operations research.

Tucker (KKT) conditions are necessary and sufficient conditions for optimality, guaranteeing the uniqueness of the solution.

Also, the resulting explicit MPC control law ${}^{MPC}u = f(\vartheta_k)$ is continuous and Piece Wise Affine (PWA):

$$f(\vartheta_k) = F^i\vartheta_k + g^i, \quad \text{if } H^i\vartheta_k \leq k^i, \;\; i = 1, ..., N_{mpc} \tag{3.74}$$

where N_{mpc} is the number of neighboring convex polytopes (or regions) into which the ϑ-space is divided. The on-line MPC computation effort thus reduces to a table-lookup: for a given vector of parameters ϑ_k, which is equals to the state of the system x_k in this case, a first linear search through the polyhedral regions $H^i\vartheta \leq k^i$ is performed to locate the one which contains ϑ_k, and then the corresponding feedback gains F^i and g^i are lookup to compute the MPC control action ${}^{MPC}u$.

In [72] the authors propose some extensions to this basic formulation, including reference tracking, measured disturbances compensation, soft constraints and variable constraints. In real applications with constraints on the output variable, the introduction of slack variables to soft constraints is a solution to avoid infeasibility. In fact, hard constraints on the output variable may cause infeasibility when, for instance, the initial conditions are infeasible, or noise causes the output to go outside the feasible solution space in some future time steps or in the presence of serious model uncertainties.

An explicit solution extends the applicability of MPC to situations in which anti-windup schemes are inadequate or where on-line computing requirements are prohibitive for either technical or cost reasons. The decision between an on-line MPC and an explicit MPC must be a trade-off solution between CPU computing demanding to solve in real-time a QP problem and memory for storing the explicit MPC solution.

Chapter 4

FORMATION FLIGHT SOFTWARE SIMULATOR

4.1 Introduction

The potential for formation flying satellites to serve as a feasible alternative to single satellite missions is strongly dependent on the development of suitable Guidance, Navigation and Control (GNC) systems. These systems must deliver robust and reliable performance to accuracies and fuel consumption levels defined by the mission objectives, while accommodating orbital perturbations, sensor noises and actuation errors.

We developed a simulator suitable for formation flying application that enable the user to design and validate GNC strategies and actuator or sensor architectures. The simulator also allows to analyze the performances achieved by adopting different approaches for the problem of propagating the relative motion of a spacecraft formation. In order to test the robustness of a GNC system, as well as to validate an on-board relative dynamics predictor, it is indeed important to have a reliable and accurate dynamics propagator. Furthermore, the simulator present a modular structure that enables the user first to design particular elements of the simulator, such as control or guidance strategies, and then to evaluate its efficiency and robustness taking into account reliable representations both of the environment in which the satellite formation will operate and of the behavior of the sensor and/or actuator systems.

This chapter describes the main general features of the Matlab Simulator, including architecture, data flow and implementation notes. The Matlab simulator was used to simulate the formation flying maneuvers described in the next chapters. Simulator modules and algorithms developed for those particular scenarios will be described in detail from time to time in the respective sections.

4.2 Simulator Architecture and Data flow

The Matlab Simulator is divided into eight modules or macro-functionalities:

1. a DYNAMIC PROPAGATOR module that performs the dynamic evolution of the system;
2. a SENSORS module that models the behavior of the sensor system while producing measures from the dynamic state of the system;
3. a NAVIGATION module that performs an (optimal) estimation of the current state of the system using models of its dynamic behavior and measures from the sensor module;
4. a GUIDANCE module that computes the desired trajectory that each satellite of the formation has to follow;
5. a CONTROL module that computes (optimal) control actions that should be performed by the actuator system in order to track the desired trajectory;
6. an ACTUATORS module that models the behavior of the actuator system while performing the control actions provided by the control module;
7. an ENVIRONMENT module that computes the perturbative effects (forces and torques) on the system dynamics;
8. a MODE MANAGER module that menages the transition between different formation flying modes.

Figure 4.1 shows the top level architecture and data flow of the simulator. The modules are grouped in two main sections: the *PLANT system*, which represents the real world, and the *GNC system*, which is the brain of the loop and that interacts with the real world through sensors and actuators. A more detailed description of each module is presented below, following an input-output scheme.

DYNAMIC PROPAGATOR

This module performs the dynamic evolution of the whole physical system through the integration of the differential equations system used to model the dynamic behavior of the whole PLANT:

$$\dot{\mathbf{X}}(t) = \mathbf{F}(\mathbf{X}(t),\, t,\, \mathbf{U}_{comm}(t),\, \mathbf{U}_d(t)) \tag{4.1}$$

where $\mathbf{X}(t)$ is the dynamic state of the whole PLANT, which contains environment state and sensors, actuators, attitude and orbital absolute and/or relative states for each satellite of the formation; $\mathbf{U}_{comm}(t)$ is the control vector that is formed by both linear and angular accelerations

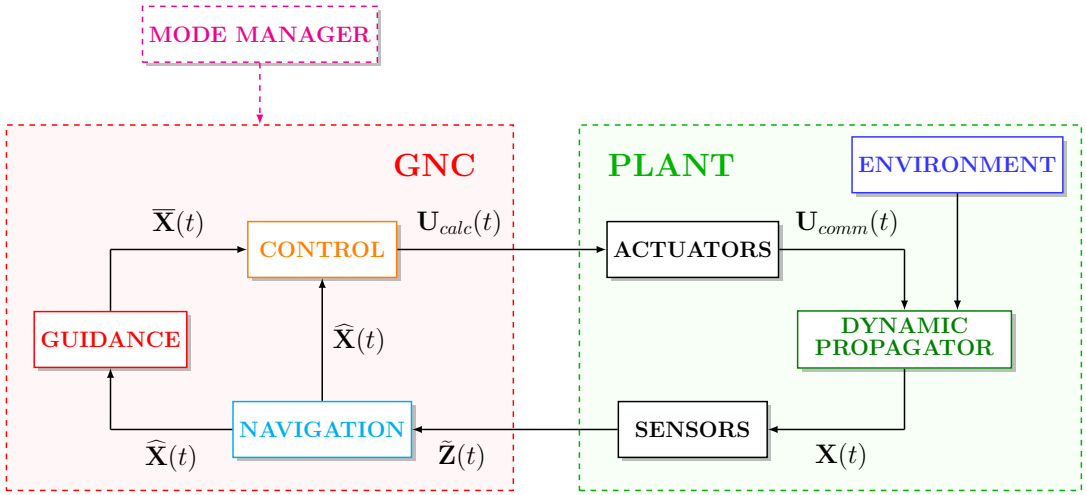


Figure 4.1: Top level Architecture and Data flow.

and that is performed[1] by the actuator system; $\mathbf{U}_d(t)$ is the perturbative action vector, which contains both linear and angular perturbative actions.

INPUT: dynamic state of the whole PLANT at time t_k, $\mathbf{X}(t_k)$; control actions performed by the actuators system $\mathbf{U}_{comm}(t)$; forces and torques due to the interaction of the system with the environment, $\mathbf{U}_d(t)$.

OUTPUT: dynamic state of the whole PLANT at time $t_{k+1} = t_k + \Delta T_{sim}$, $\mathbf{X}(t_{k+1})$,where ΔT_{sim} is the simulation step.

SENSORS

The aim of this module is to simulate the sensors system behavior while producing desired measures from the dynamic state of the system. The behavior of the sensors system is described by models whose accuracy and consequent complexity depend on the final purposes of the simulation. Measures $\widetilde{\mathbf{Z}}(t)$ can be obtained adding a bias and a (white) noise to a component that is function of the dynamic state of the PLANT:

$$\widetilde{\mathbf{Z}}(t) = f(\mathbf{X}(t)) + \text{bias} + \text{noise} \tag{4.2}$$

INPUT: dynamic state of the whole PLANT at time t_k, $\mathbf{X}(t_k)$.

OUTPUT: desired measures at time t_k, $\widetilde{\mathbf{Z}}(t_k)$.

[1] *comm* stands for "commanded".

NAVIGATION

This module performs an (optimal) estimation of the current state of the system $\widehat{\mathbf{X}}(t)$ using a set of measures coming from the sensors module $\widetilde{\mathbf{Z}}(t)$ and a set of models that describe the dynamics of the system (state equations) and the sensor system behavior (measure equations). Different estimation techniques can be used for this purposes, as for instance Kalman filtering or Bayessian estimation, whose selection have to be evaluated accordingly to the sensor system type and the scenario on hand.

INPUT: measures from the SENSORS module at time t_k, $\widetilde{\mathbf{Z}}(t_k)$.

OUTPUT: best estimation of the current state of the system at time t_k, $\widehat{\mathbf{X}}(t_k)$.

Depending on the estimation algorithm used, this module may require additional information as input, e.g. the control vector actuated at the previous time step when a Kalman Filter is used.

GUIDANCE

Taking into account the environmental conditions, the current estimated state of the system $\widehat{\mathbf{X}}(t)$, the Actuator system characteristics, and the activated mission mode, this module performs the calculation of the desired trajectory $\overline{\mathbf{X}}(t)$ (translational and/or attitude absolute or relative motion) that each satellite of the formation has to follow.

INPUT: Mode that is activated from the MODE SWITCHER; constraints related to actuator and sensor systems; environmental conditions; Navigation data $\widehat{\mathbf{X}}(t_k)$.

OUTPUT: Reference trajectory that each spacecraft of the formation has to track at time t_k, $\overline{\mathbf{X}}(t_k)$.

CONTROL

This module computes the control actions $\mathbf{U}_{calc}(t)$ that the actuator system should perform to make the current estimated state of the system $\widehat{\mathbf{X}}(t)$, from the NAVIGATION module, matching the desired one $\overline{\mathbf{X}}(t)$, from the GUIDANCE module. The tracking error depends not only on the control type in use but also on the characteristics of SENSORS, NAVIGATION and ACTUATORS modules.

INPUT: estimation of the system state at time t_k, $\widehat{\mathbf{X}}(t_k)$; reference trajectory to be tracked at time t_k, $\overline{\mathbf{X}}(t_k)$; environmental conditions; actuator system constraints.

OUTPUT: computed control actions $\mathbf{U}_{calc}(t)$.

ACTUATORS

This module simulates the actuator system behavior while performing the desired control actions. The behavior of the actuators system is described by models whose accuracy and consequent complexity depend on the final purposes of the simulation. The control actions that each element of the actuators system has to perform, $\mathbf{U}_{comm}$, can be obtained first using a function of the calculated control actions coming from the CONTROL module, $\mathbf{U}_{calc}$, and then adding to it a bias and a (white) noise:

$$\mathbf{U}_{comm}(t) = f\,(\,\mathbf{U}_{calc},\,\mathbf{X}(t),\,\text{bias}(t),\,\text{noise}(t)) \tag{4.3}$$

INPUT: control actions computed in the CONTROL module at time t_k, $\mathbf{U}_{calc}(t_k)$; dynamic state of the whole PLANT at time t_k, $\mathbf{X}(t_k)$.

OUTPUT: control actions performed by the actuators system at time t_k, $\mathbf{U}_{comm}(t_k)$.

ENVIRONMENT

The Environment module menages all computations needed to determine perturbative effects (forces and torques) that come from the interaction of the spacecraft(s) with the environment, as for example the central body current position and/or attitude with respect to the Earth Centered Inertial (ECI) reference frame, which is needed to compute spacecraft acceleration due to higher harmonics of the central body, solar radiation pressure, or physical properties of the central body atmosphere for the drag force computation.

INPUT: current Julian Date.

OUTPUT: parameters needed to compute perturbative effects on the spacecraft(s).

In particular, the perturbative accelerations due to a non spherical shape of the Earth are computed as the gradient of the gravity potential expanded in spherical harmonics (see for instance [12]). Also, the perturbative acceleration due to atmospheric drag acting on a single spacecraft in the ECI reference frame is modeled as:

$$\mathbf{f}_{drag} = -\frac{1}{2}\rho\frac{C_D S}{M}V^2\,\widehat{\mathbf{V}} \tag{4.4}$$

where ρ is the air density, computed as function of the satellite altitude following a classical Jacchia's 1977 model for average solar activity, M and S are the mass and the cross section, respectively, of the spacecraft, C_D is the drag coefficient (equal to 2.2), $\mathbf{V}$ is the satellite absolute velocity vector in the ECI reference frame.

MODE MANAGER

Taking into account the current estimated position of the spacecrafts in the formation and the mission requirements, this module menages the transition between different formation flying modes. The transition between one mode to another one occurs if some conditions, which depend on the formation state, are verified, e.g. if the distance between the satellites is less then a reference value, the collision-avoidance mode is activated.

INPUT: current position of the spacecraft of the formation at time t_k, $\mathbf{X}(t_k)$.

OUTPUT: the formation flying mode that is currently activated.

4.3 Dynamic Propagators

Two dynamic propagators are available in the formation flight simulator. The first one is based on a classical Cowell's method [12], which integrates a system of six differential equations in the Cartesian Earth Centered Inertial (ECI) reference frame, with time as independent variable.

The second dynamic propagator is a fast and accurate nonlinear dynamics propagator for formation flying orbital motion, called DROMO-FF [73], based on a a very efficient regularized method developed by Pelàez et al. in 2006 [74]. This new formulation for the two body-problem equations of motion borrows elements of rigid-body dynamics. The basic idea of this method is to track the evolution of an orbital frame moving with the point mass and link to it a new set of orbital elements by applying the variation of parameter technique. The independent variable is changed from time to a pseudo-anomaly, which coincides with the true anomaly in the pure Kepler motion. The variation of parameters technique is applied to a particular set of 7 elements, which are chosen as the integrals of the unperturbed motion. The first three elements define the shape of the orbit and its orientation on the orbital plane, and the remaining four elements, which correspond to the components of a unit quaternion, are devoted to fixing the attitude of the orbital plane with respect to an inertial reference frame.

Pelàez's method is capable of propagating the orbital dynamics of a single point mass with better accuracy and faster computational run time, when compared to classical methods such as Cowell's or Encke's methods, as well as newer methods like the Kustaanheimo–Stiefel or the Sperling–Burdet methods. A fast dynamic propagator is obtained by using Pelàez's method with variable step numerical integration routines with effective step control, as Runge–Kutta–Fehlberg or Dormand–Prince types. Routines with fixed step size can however be used without reduction in performances and the new independent variable allows analytical step regulation even in this case.

The absolute dynamics of each space vehicle flying in formation is first obtained using the dynamic propagator, while the the relative dynamics of each spacecraft with respect to a reference

satellite is then properly computed from the absolute dynamics [75].

4.4 Simulator implementation

The formation flight software simulator was developed in Matlab, which allows the user to perform numerical calculations and visualize the results without the need for complicated and time consuming programming, and many specific toolboxes, such as the Control Toolbox or the MPC toolbox, are available. On the other hand, Matlab is an interpreted language and poor programming practices and very computationally burdensome problems can make it unacceptably slow. For this reason, some functions, such as those for the Earth gravity model implementation with up to 360x360 harmonics, were implemented in C and mex-interfaced with Matlab. The use of Matlab is however very useful for preliminary design and validation of GNC strategies, software simulator architecture and modules structure.

The formation flight Matlab simulator was developed using the Object Oriented Programming (OOP). One of the most important characteristics of OOP is the data encapsulation concept, which means that there is a very close attachment between data items and procedures or methods. Normally, objects of a given type are instances of a class, whose definition specifies the private (internal) working of these objects as well as their public interface. The public interface functions completely defines how to use this object, so that programs that want to manipulate an object do not have to worry about how these tasks are achieved nor the internal structure of the object. If compared to the procedural programming, the OOP presents the following advantages:

- simplicity: software objects model real world objects, so the complexity is reduced and the program structure is very clear;
- modularity: each object forms a separate entity whose internal workings are decoupled from other parts of the system;
- modifiability: it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods;
- extensibility: adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones;
- maintainability: objects can be maintained separately, making locating and fixing problems easier;
- re-usability: objects can be reused in different programs.

The Matlab program is organized into three main sections (see Figure 4.2):

1. pre-processing, in which the scenario is defined and all objects, structure and variables are defined and initialized;

2. main loop with a main for loop with the updating sequence of all modules at each simulation time step;

3. post-processing for final calculations, results saving and visualization.

The interfacing fuctions of each class can be divided into three main groups: (1) setup functions used in the pre-processing to initialize the objects; (2) update and data printing functions used in the main loop to update the object at the current simulation time step, to visualize data in the Matlab command window and/or save it in the corresponding output files; and (3) post-processing and plotting functions, used in post-processing to save simulation results and generate diagrams.

A *Set-Up* m-file allows the user to define all those parameters that characterize the scenario under studying, such as: absolute and/or relative dynamic state of each space vehicle, type of dynamic propagator, type of sensors/actuators, GNC strategies, etc. The user can also enable or disable each single module through a corresponding enabling flag in the Set-Up m-file. This for example allows the user to first design a particular control algorithm without sensors/actuators noises or other perturbative effects, and then to evaluate the same control strategy in a more representative environment.

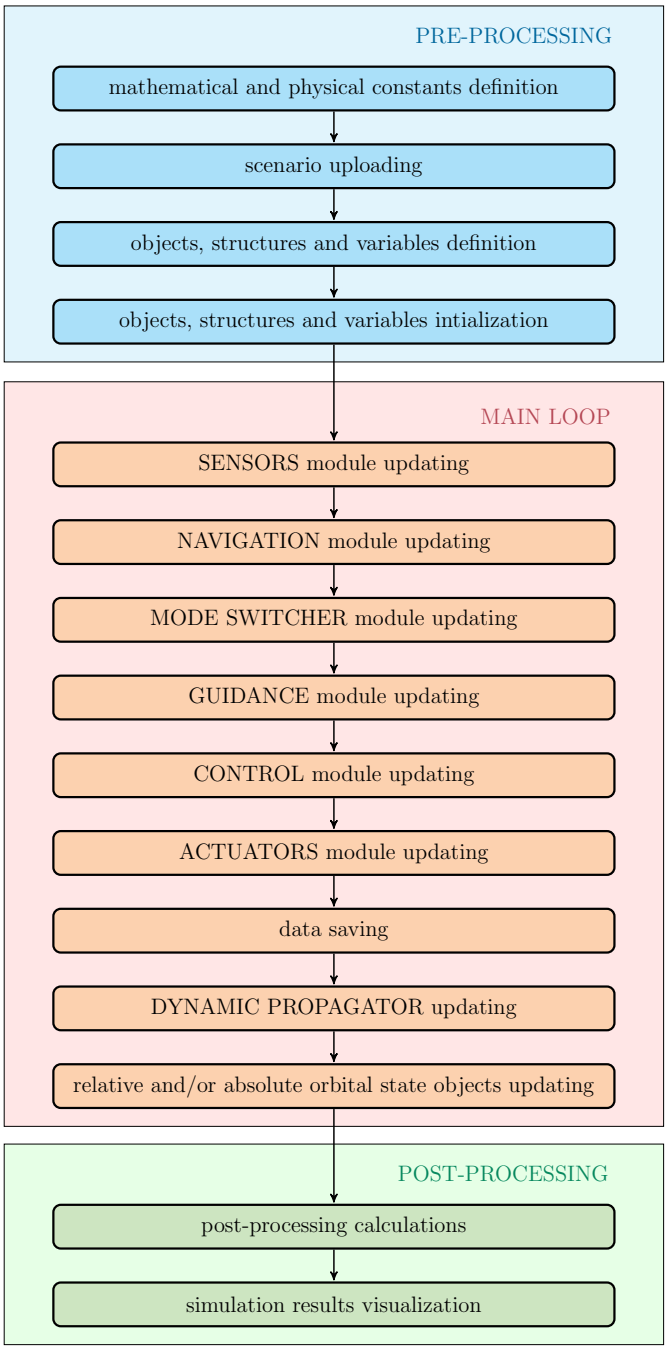


Figure 4.2: Block diagram of the Formation Flight Matlab Simulator.

Chapter 5

MPC APPLICATIONS TO SPACECRAFT FORMATION ACQUISITION

In this chapter we present a study on the MPC application to a formation acquisition maneuver for two space vehicles, taking into account two scenarios: a Leader-Follower formation and a Projected Circular Orbit formation. The performances of an MPC controller are compared with those of a LQR controller in carrying out the same maneuver, evaluating also the effects of the gravitational harmonics J_2 , J_3 and air drag perturbations on the proposed maneuvers.

5.1 Leader-Follower formation

Let us consider the acquisition maneuver of a Leader-Follower formation (see Figure 5.1a) with two space vehicles $500\,m$ apart that have active constraints on the maximum control acceleration that their propulsion subsystems can actuate.

The initial dynamic state of the Chief is given in Table 5.1 in terms of orbital elements of the osculating keplerian orbit, while the initial dynamic state of the Deputy is given in Table 5.2 as relative position $\mathbf{r}_i$ and velocity $\mathbf{v}_i$ vectors with respect to the Hill's frame attached to the chief spacecraft. The previous relative conditions are computed using the Hill- Clohessy-Wiltshire (HCW) solution with the 1^{st}-order bounded relative motion condition (Equation 2.12) with amplitudes of the in-plane and out-of-plane bounded motion $\rho_x = 8\,m$ and $\rho_z = 8\,m$, respectively, along-track offset $\Delta y = 500\,m$, and in-plane and out-of-plane phase angles $\varphi_{xy} = 45\,deg$ and $\varphi_z = 90\,deg$, respectively. Therefore, we assume that before the formation acquisition maneuver the Deputy follows a 1^{st}-order bounded parking trajectory near the reference relative state $\overline{\mathbf{x}} = [0,\, 500\,m,\, 0,\, 0,\, 0,\, 0]^T$.

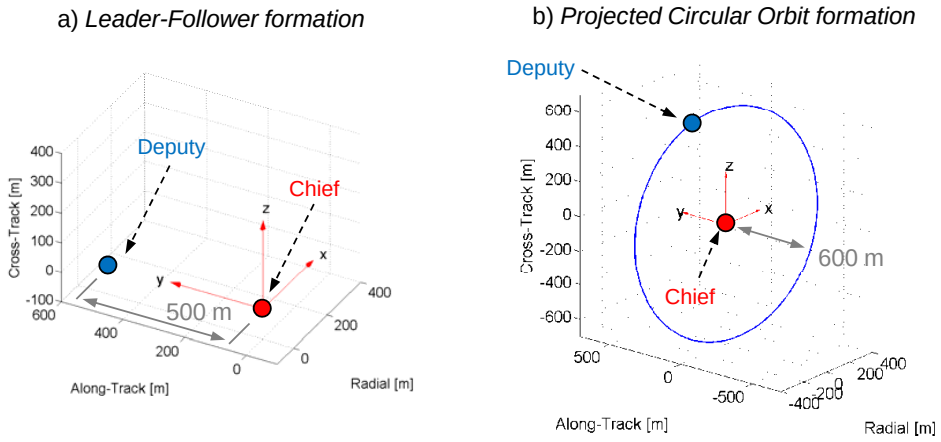


Figure 5.1: (a) Leader-Follower formation, and (b) Projected Circular Orbit Formation.

Table 5.1: Chief keplerian orbital elements – Leader-Follower formation.

semi-major axis a [km]	eccentricity e [ ]	RAAN Ω [deg]	inclination i [deg]	argument of perigee ω [deg]	true anomaly ϑ [deg]
6700	0	0	97.87	0	0

Table 5.2: Deputy relative initial position and velocity vectors – Leader-Follower formation.

relative initial position $\mathbf{r}_i$ [m]	relative initial velocity $\mathbf{v}_i$ [10^{-3} m/s]
$[5.657\,,\,51.314\,,\,-8.000]^T$	$[-6.512\,,\,-1.025\,,\,0.000]^T$

The following simulation results refer to a 10 minutes acquisition maneuver, with a control interval $T_s = 5\,s$ and a maximum control acceleration $u_{max} = 5 \cdot 10^{-3}\,m/s^2$, so that:

$$|u_j| \leq u_{max}\,,\; j = x, y, z \tag{5.1}$$

The parameters of the LQR and MPC controllers are listed in Table 5.3, where n is the Chief mean motion. The parameters of both controllers are tuned in order that the components of the position error vector present the same settling time $t_{set} \approx 2\,min$, i.e. the two controllers presents similar performances in reaching the same reference relative state.

Table 5.3: LQR and MPC parameters – Leader-Follower formation.

${}^{LQR}\mathbf{Q}$	$\begin{bmatrix} 1.5\,\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & 0.1\,\mathbf{I}_{3\times3} \end{bmatrix}$
${}^{LQR}\mathbf{R}$	$1.1 \cdot 10^{11}\, n^2\, \mathbf{I}_{3\times3}$

${}^{MPC}\mathbf{Q}$	${}^{LQR}\mathbf{Q}$
${}^{MPC}\mathbf{R}$	$1.25 \cdot 10^{9}\, n^2\, \mathbf{I}_{3\times3}$
control horizon m	4
prediction horizon p	20

The position error trajectory in Hill's LVLH reference frame is shown in Figure 5.2, while the position error components are plotted in Figure 5.3. In particular, for both controller types under investigation, the magnitude of the position error vector is less that $0.382\,m$ after $2.00\,min$ from the beginning of the maneuver.

Figure 5.5 shows the control acceleration components actually performed by the propulsion system $U_{comm}(t)$ (see Figure 4.1), in blue for the LQR controller and in red for the MPC controller. As can be seen from this figure, the control acceleration components of both controllers sometimes reach the maximum values $\pm 5 \cdot 10^{-3}\, m/s^2$. However, the MPC controller takes the control acceleration constraints into account while performing the optimal control computation, so that the output of the MPC CONTROL module $U_{calc}(t)$ always satisfies the saturation constraint given by Equation 5.1. On the other hand, the LQR optimal control computation is performed without taking into account the saturation constraint, and for this reason $U_{calc}(t)$ from the LQR CONTROL module can be greater than the maximum control acceleration that can be carried out by the propulsion system.

The required Δv is taken as index of the control system performance, defined as:

$$\Delta v(t) = \int_0^t \|\mathbf{u}(\tau)\|_2 \, d\tau \tag{5.2}$$

This index is used to compare the performances of the two controllers LQR and MPC. The following LQR and MPC Δv requirements over 10 minutes of simulation were obtained: $\Delta v_{LQR} = 0.68187\, m/s$ and $\Delta v_{MPC} = 0.39995\, m/s$. The percentage difference of these Δv requirements is:

$$\delta_{\Delta v} = \frac{\Delta v_{MPC} - \Delta v_{LQR}}{\Delta v_{LQR}}\, 100 = -41.34\,\% \tag{5.3}$$

The profiles of the Δv requirements and the correspondent percentage differences are depicted in Figures 5.6 and 5.7, respectively.

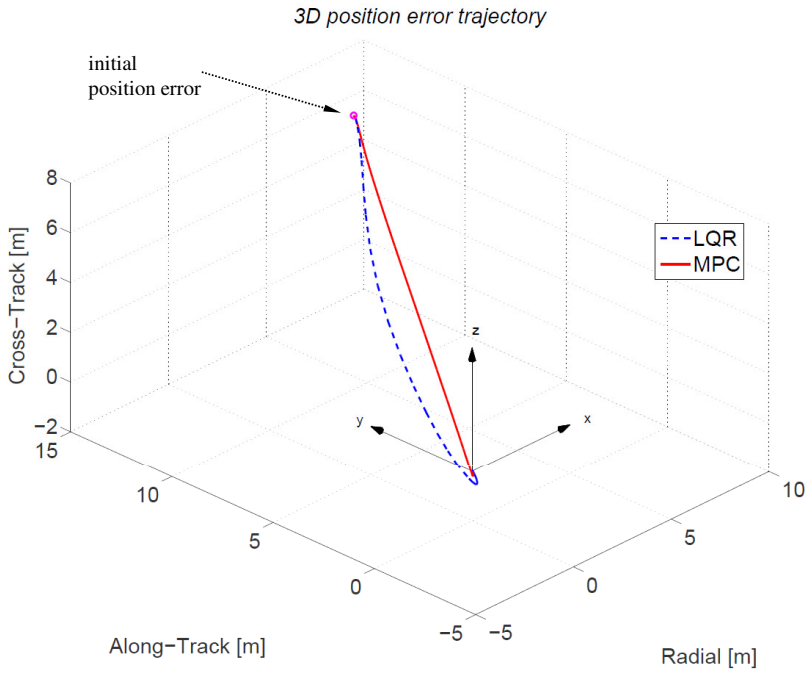


Figure 5.2: 3D position error trajectory. Initial position error in magenta – Leader-Follower formation.

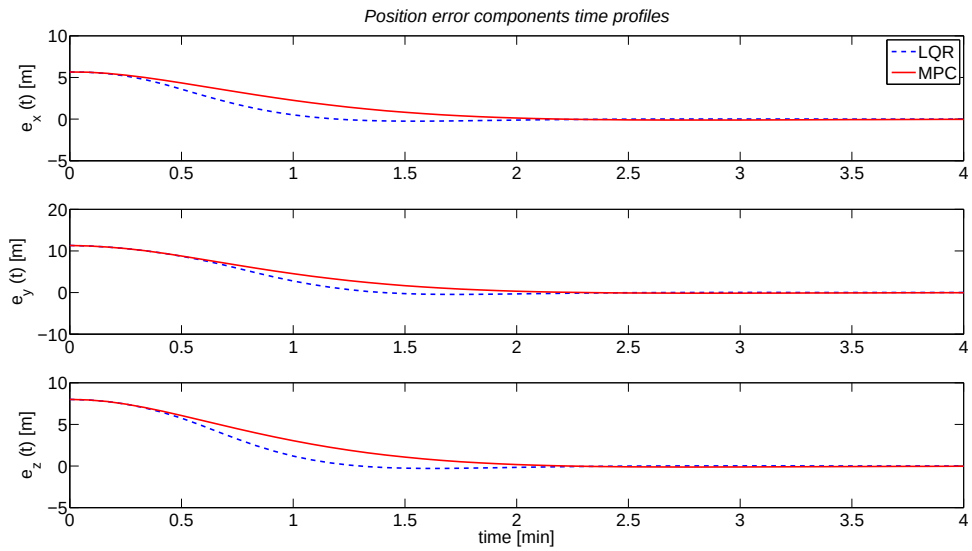


Figure 5.3: Position error components time profiles – Leader-Follower formation.

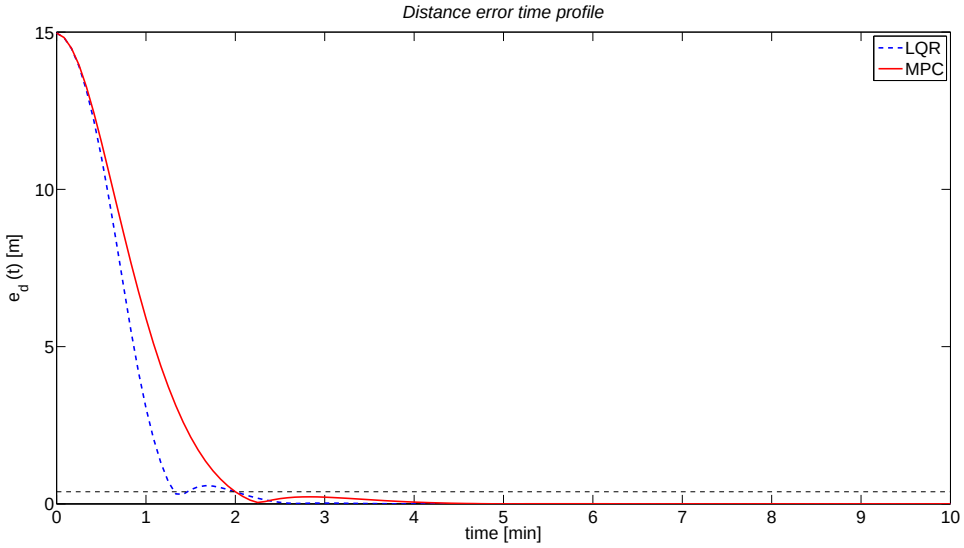


Figure 5.4: Distance error time profile – Leader-Follower formation.

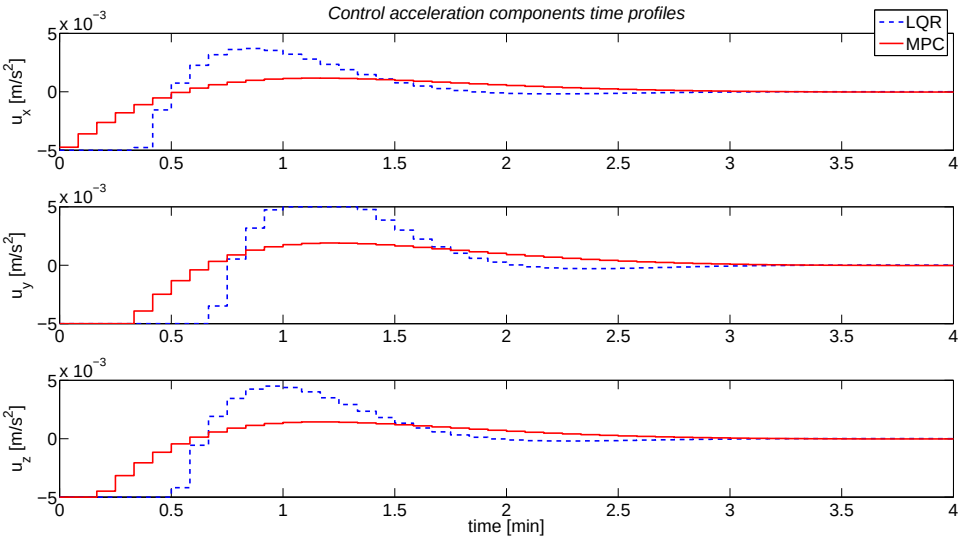


Figure 5.5: Control acceleration components of LQR and MPC controllers – Leader-Follower formation.

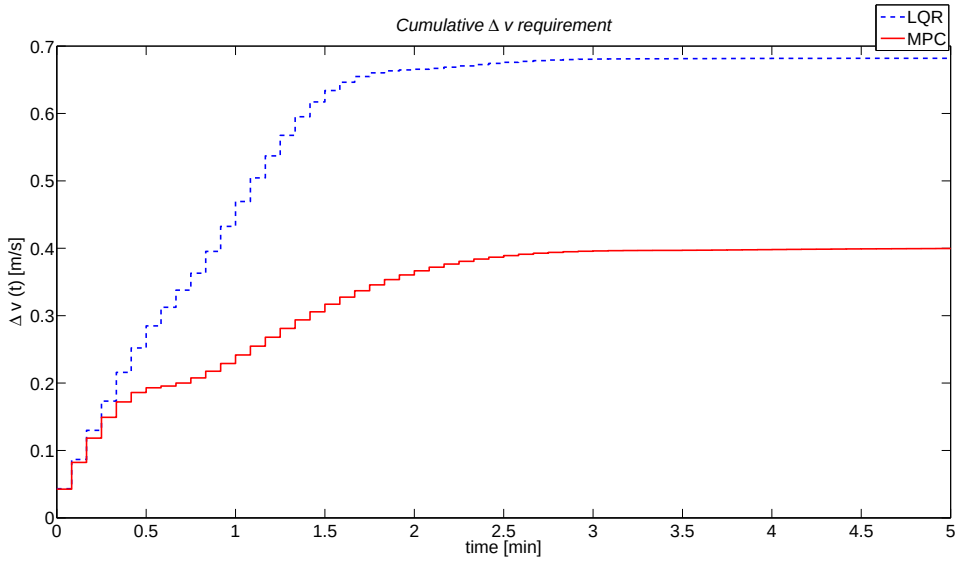


Figure 5.6: Δv requirements of LQR and MPC controllers – Leader-Follower formation.

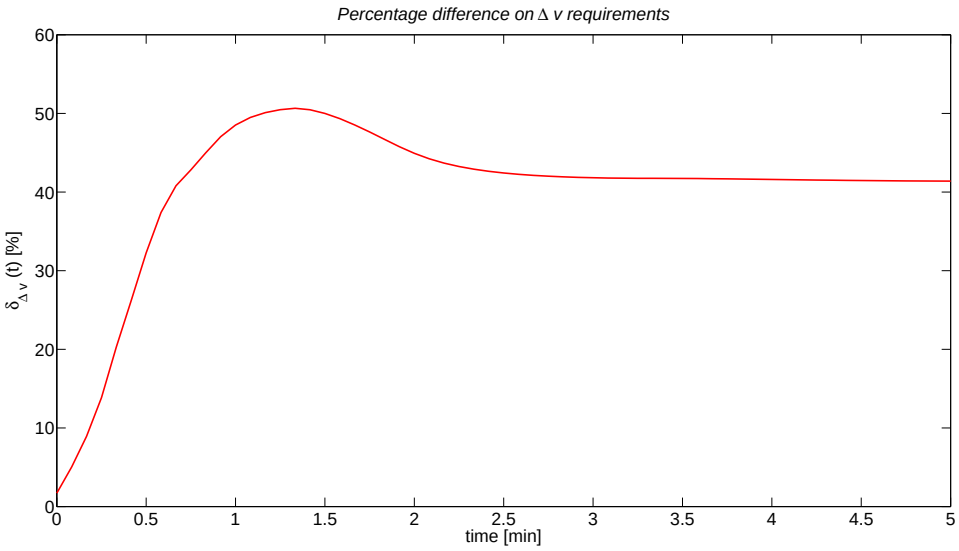


Figure 5.7: Percentage difference on Δv requirement – Leader-Follower formation.

The previous results are obtained assuming a dynamic evolution of the formation with no perturbations. In order to evaluate the effect of perturbations on the previous formation acquisition maneuver, other two simulations were carried out taking into account J_2 , J_3 and atmospheric drag[1] perturbations (see Chapter 4).

Table 5.4 lists the Δv requirements of both LQR and MPC controllers for unperturbed and perturbed cases. The position error components profile and the control acceleration components

[1] Air density $\rho \approx 1.9 \cdot 10^{-11}\, kg/m^3$ at the initial formation altitude.

profile are similar to those obtained in the unperturbed case. Also, in the perturbed scenarios the use of an MPC controller can reduce significantly the total Δv requirement with respect to an LQR controller. However, the percentage difference $\delta_{\Delta v}$ between LQR and MPC Δv requirements for the case with J_2, J_3 and drag perturbations is reduced by 0.49 % with respect to the unperturbed case.

Table 5.4: Δv requirements of LQR and MPC controllers for perturbed and unperturbed cases.

perturbations	$\Delta v_{LQR}\,[m/s]$	$\Delta v_{MPC}\,[m/s]$	$\delta_{\Delta v}\,[\%]$
none	0.68187	0.39995	−41.34
$J_2 + J_3$	0.68321	0.40118	−41.28
$J_2 + J_3 + drag$	0.69126	0.40886	−40.85

5.2 Projected Circular formation

In this section we compare and discuss the performances of a MPC controller and a LQR controller in carrying out the same formation acquisition maneuver for a two spacecraft Projected Circular Orbit (PCO) formation (see Figure 5.1b).

The initial dynamic state of the Chief is the same used for the Leader-Follower scenario and reported in Table 5.1. The Deputy presents an initial relative dynamic state, listed in Table 5.5 and computed using Equations 2.13 - 2.15 with $\rho_x = 295\,m$, $\rho_z = 590\,m$, $\Delta y = 0\,m$, and $\varphi_{xy} = \varphi_z = 45\,deg$, while the reference relative trajectory is given by the same equations with $\rho_x = 300\,m$, $\rho_z = 600\,m$, $\Delta y = 0\,m$, and $\varphi_{xy} = \varphi_z = 45\,deg$, , i.e. a PCO with a radius of $600\,m$ in the along-track – cross-track plane.

As in the Leader-Follower formation, we consider a 10 minutes maneuver, with a control interval $T_s = 5\,s$ and a maximum control acceleration $u_{max} = 5{\cdot}10^{-3}\,m/s^2$. The parameters of the LQR and the MPC controllers, listed in Table 5.6, are designed to obtain the same settling time $t_{set} \approx 2\,min$ for the components of the position error vector.

Table 5.5: Deputy relative initial position and velocity vectors – Projected Circular formation.

relative initial position $\mathbf{r}_i$ [m]	relative initial velocity $\mathbf{v}_i$ [10^{-1} m/s]
$[\,208.597\,,\,417.193\,,\,417.193\,]^T$	$[\,2.401\,,\,-4.803\,,\,4.802\,]^T$

Table 5.6: LQR and MPC parameters – Projected Circular formation.

${}^{LQR}\mathbf{Q}$	$\begin{bmatrix} 1.5\,\mathbf{I}_{3\text{x}3} & \mathbf{0}_{3\text{x}3} \\ \mathbf{0}_{3\text{x}3} & 0.1\,\mathbf{I}_{3\text{x}3} \end{bmatrix}$
${}^{LQR}\mathbf{R}$	$1.1 \cdot 10^{11}\, n^2\, \mathbf{I}_{3\text{x}3}$

${}^{MPC}\mathbf{Q}$	${}^{LQR}\mathbf{Q}$
${}^{MPC}\mathbf{R}$	$6.50 \cdot 10^{8}\, n^2\, \mathbf{I}_{3\text{x}3}$
control horizon m	4
prediction horizon p	30

Table 5.7: Δv requirements of LQR and MPC controllers for perturbed and unperturbed cases – Projected Circular formation.

perturbations	$\Delta v_{LQR}\,[m/s]$	$\Delta v_{MPC}\,[m/s]$	$\delta_{\Delta v}\,[\%]$
none	0.52151	0.36522	-29.97
$J_2 + J_3$	0.52239	0.36607	-29.92
$J_2 + J_3 + drag$	0.52906	0.37130	-29.81

The position error trajectory in the Hill's LVLH reference frame is shown in Figure 5.8, while the position error components are depicted in Figure 5.9. LQR and MPC Δv requirements over 10 minutes of simulation and their percentage differences are: $\Delta v_{LQR} = 0.52151\,m/s$ and $\Delta v_{MPC} = 0.36522\,m/s$ and $\delta_{\Delta v} = -29.97\,\%$. Figure 5.11 shows the control acceleration components produced by the propulsion system, while the profiles of the Δv requirement and the corresponding percentage differences are depicted in Figures 5.12 and 5.13, respectively. Also in this scenario, the use of a MPC controller can reduce significantly the total Δv requirement with respect to an LQR controller.

The effect of J_2 , J_3 and drag perturbations on the PCO formation acquisition maneuver for both controllers are reported in Table 5.7. Simulation results confirm that also in the perturbed scenarios the MPC controller is more efficient (requires less Δv) than the LQR controller, with a small reduction of $\delta_{\Delta v}$ (by 0.16%) between the unperturbed case and the one affected by $J_2 + J_3 + drag$ perturbations.

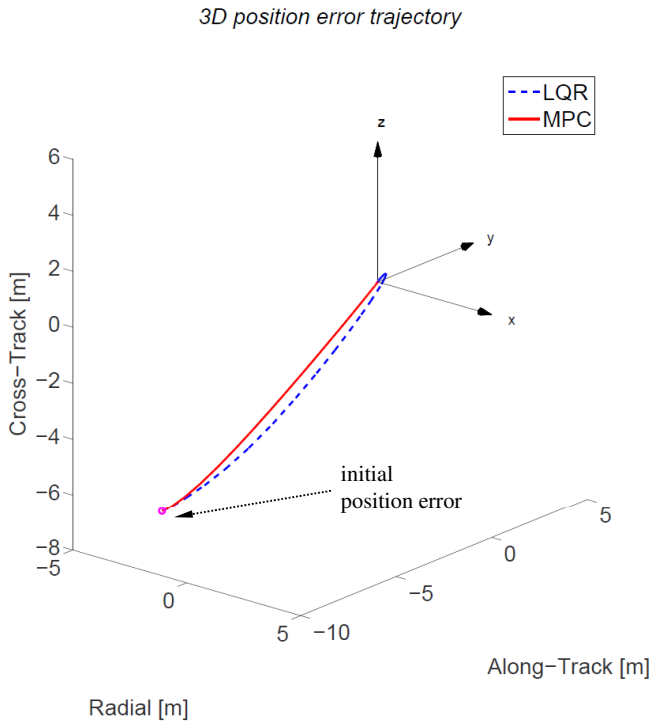


Figure 5.8: 3D position error trajectory – Projected Circular formation.

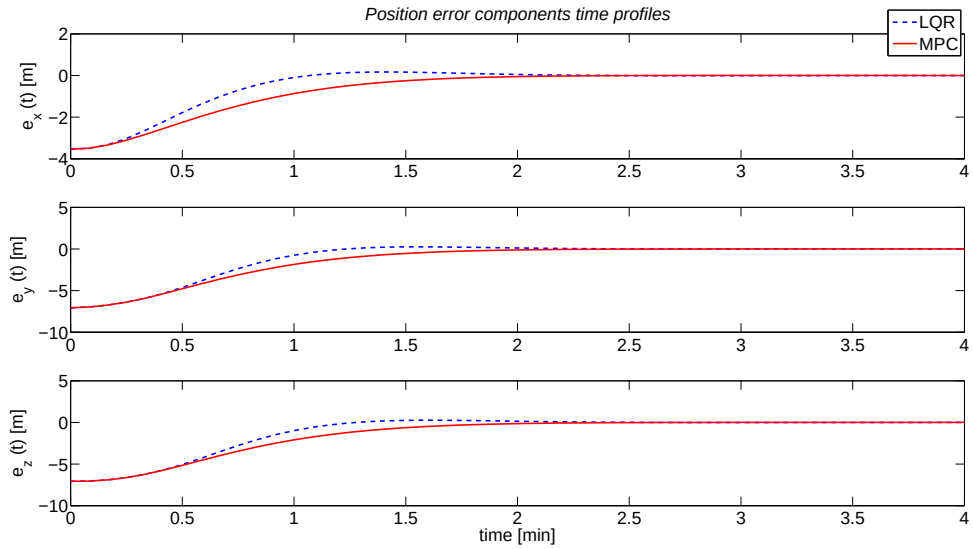


Figure 5.9: Position error components time profiles – Projected Circular formation.

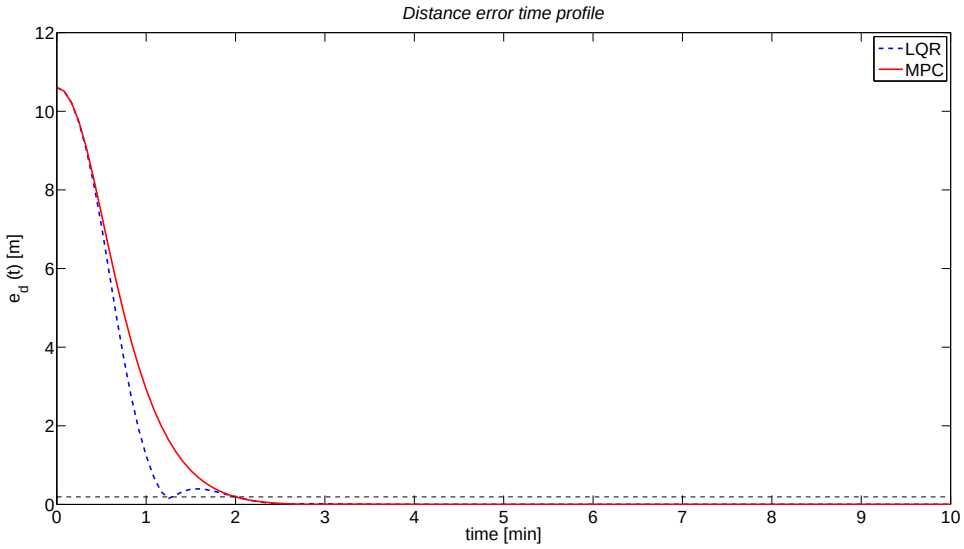


Figure 5.10: Distance error time profile – Projected Circular formation.

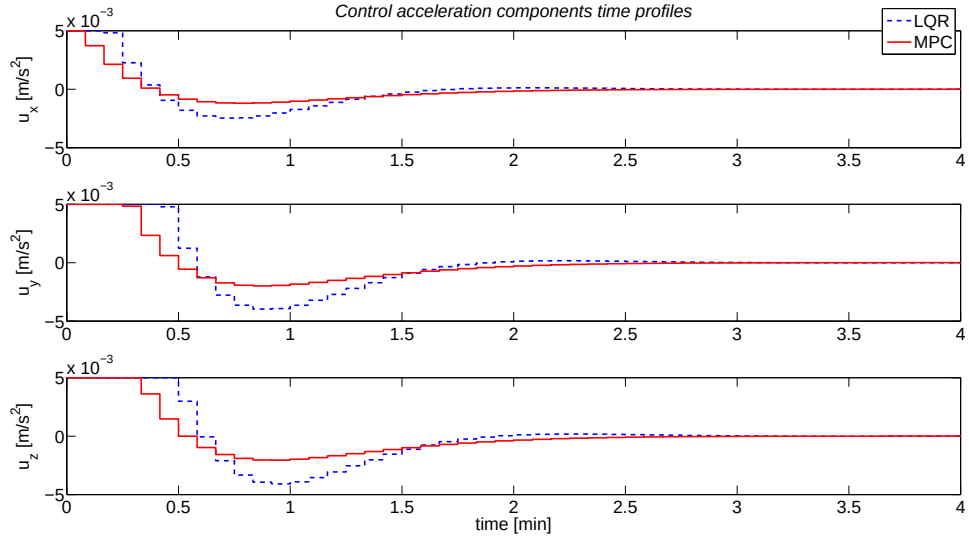


Figure 5.11: Control acceleration components of LQR and MPC controllers – Projected Circular formation.

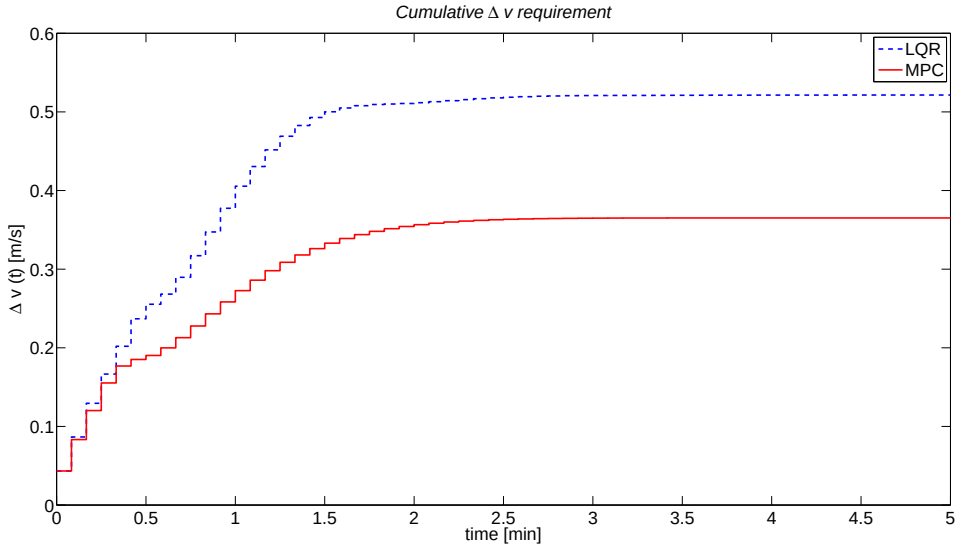


Figure 5.12: Δv requirements of LQR and MPC controllers – Projected Circular formation.

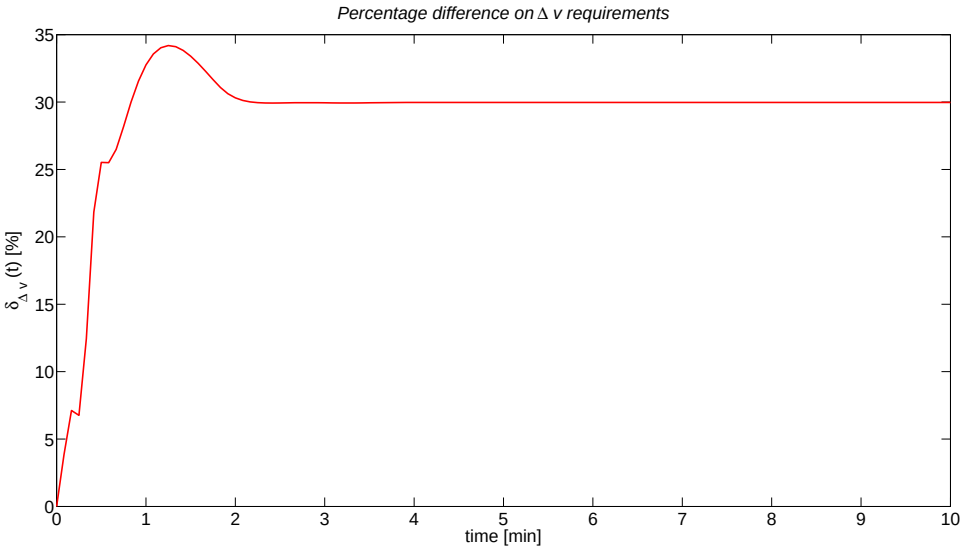


Figure 5.13: Percentage difference on Δv requirement – Projected Circular formation.

Chapter 6

COLLISION AVOIDANCE GUIDANCE AND CONTROL STRATEGIES

6.1 Introduction

The aim of this chapter is to study guidance and control strategies for collision-avoidance formation flying maneuvers. As mentioned in Section 2.3, Guidance has in charge the determination of reference trajectories that the satellites of a formation have to follow in order to accomplish a desired task, as for instance collision-avoidance, formation-reconfiguration or station-keeping. On the other hand, Control is responsible for forces and/or torques exploitation to track guidance commands, aiming to have the current state "equal" to the desired one.

The collision-avoidance strategy described in this chapter is based on the use of Genetic Algorithms (GAs), a global optimization method based on the genetic evolution process of biological organisms: an initial population evolve over many generation according to the principles of natural selection. Genetic algorithms are especially powerful techniques for the research of global solutions in highly non-linear and multi-task optimization problems.

Genetic Algorithms has been already used in the spacecraft formation flying field. Kim et al. in [76] and by Tian and al. in [77] have studied the use of Genetic Algorithms for optimal reconfiguration of satellites flying in formation. Kim et al. developed a hybrid optimization method based first on a Genetic Algorithm to perform a global search and find two-impulse trajectories, and then on a primer vector analysis to find multiple-impulsive local optimal trajectories with the two-impulse trajectories as initial guesses. As pointed out by the authors, the necessity of use a Genetic Algorithm to perform a global search is due to the sensitivity of calculus of variations methods to

the choice of the initial guess. In fact, if the initial guess is arbitrarily chosen, the global optimality of the solution is not guaranteed, especially with highly non-convex objective functions. Tian et al. developed optimal controllers based on Genetic Algorithms for spacecraft formation reconfiguration including the constraints of minimum fuel, avoiding collision and final configuration. The authors emphasize that both the problem formulation and the genetic parameter selection have a big effect on the solution obtained with these methods.

6.2 The Collision-Avoidance mode

The prime objective of the Collision-Avoidance mode (one of the GNC modes) is to ensure the immediate and long-term safety of the formation, in terms of avoiding collision and preventing formation evaporation. Fuel conservation is a secondary objective.

We define as Chief the free-flying spacecraft following temporary off-nominal conditions and as Deputy the spacecraft performing collision avoidance maneuvers. We also define two ellipsoidal regions centered at the Chief: the Avoidance Region (AR), with semi-axes a_{AR}, $b_{AR} = 2\,a_{AR}$ and $c_{AR} = a_{AR}$ (a 1x2x1 ellipsoid) in x, y and z directions of the Hill's reference frame, respectively, and the Nominal Boundary (NB) with semi-axes $a_{NB} = k\,a_{AR}$, $b_{NB} = k\,b_{AR}$ and $c_{NB} = k\,c_{AR}$, where k is a safety margin. A picture of the Avoidance Region and Nominal Boundary in Hill's frame is shown in Figure 6.1.

The Collision-Avoidance strategy is based on a Separation Guidance and on a Nominal Guidance [78, 79], summarized in Figure 6.1. If a collision is predicted and the estimated relative position between the Deputy and the Chief is within the Avoidance Region, then a separation maneuver is performed (Separation Guidance): the distance between the two spacecrafts has to increase monotonically and the Deputy has to exit the Nominal Boundary within a prescribed time. When the Deputy reaches a relative position outside the Nominal Boundary, an evasive maneuver is planned (Nominal Guidance), allowing the Deputy to reach a safe or "parking" trajectory: the evasive trajectory, from the actual position to the safe trajectory, must not intersect the Nominal Boundary, and the safe trajectory must remain inside a prescribed Safe Region that does not intersect the Nominal Boundary and is far enough from the Chief. A Nominal Guidance maneuver can also be planned for cases in which the sensed relative position is already outside the Nominal Boundary while the Collision-Avoidance mode is activated only when a collision is predicted.

Two possible scenarios may occur:

1. in the first one, the Deputy is already outside the Nominal Boundary and the Collision-Avoidance mode is switched on, e.g. a collision is predicted in a certain amount of time. Nominal Guidance is performed and the Deputy achieves a Safe Trajectory.

2. in the second one, the Deputy is inside the Avoidance Region and the Collision-Avoidance mode is switched on. Separation Guidance is first performed, and the Deputy is placed outside the Nominal Boundary within a prescribed time. Once Deputy is outside the Nominal Boundary, the Nominal Guidance algorithm takes over and the Deputy achieves a Safe Trajectory.

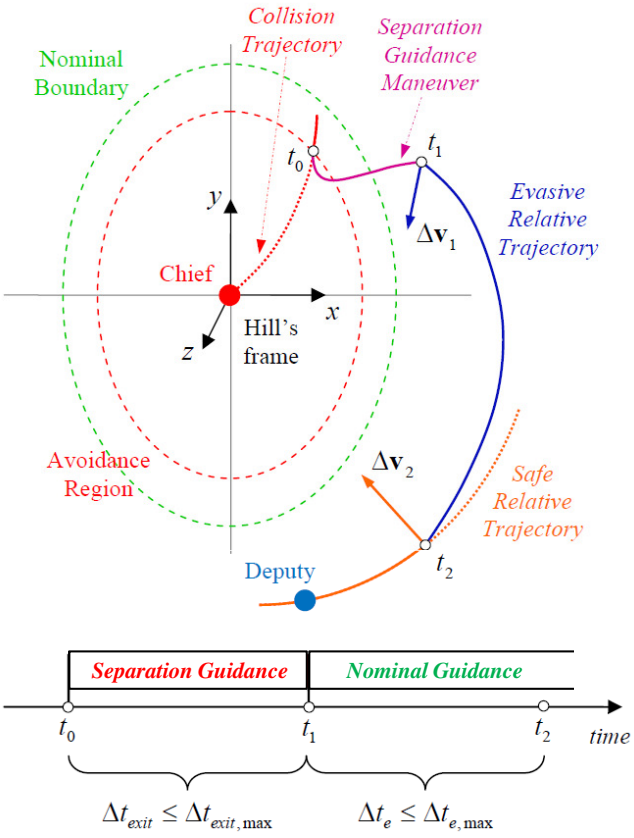


Figure 6.1: Collision Avoidance strategy and time sequence.

6.3 Separation Guidance

The Separation Guidance has to meet the following two main requirements: (1) the Deputy has to exit the Nominal Boundary within a specified time period $\Delta t_{exit,max}$; and (2) the distance between the two space vehicles has to nominally increase while inside the region. The Separation Guidance is more critical for the formation safety than the Nominal Guidance, and a reliable solution that meets the SG requirements must be obtained with minimal computing time. For this reason, the Separation Guidance algorithm is a computationally simple, deterministic and closed-form algorithm, so that a valid solution is always available without delay. The algorithm must also be

robust to practical levels of uncertainty in the initial estimated relative position and velocity used to plan the maneuver. The actual control action computation is indeed performed taking into account only the distance of the Deputy from the Nominal Boundary and the maximum time needed for the Deputy to exit the Nominal Boundary, $\Delta t_{exit,max}$. $\Delta t_{exit,max}$ is a tunable parameter: smaller exit times lead to higher delta-v and therefore higher fuel consumption, but a safe condition is reached in a shorter time. The algorithm presents some differences with respect to the one proposed by Larsson et al. in [78].

Let $\mathbf{r}_k = [\, x_k,\, y_k,\, z_k \,]^T$ and $\mathbf{v}_k$ be the relative position and velocity vectors at the current time t_k. Let t_1 be the initial time, i.e. when the separation maneuver starts, $t_f = t_i + \Delta t_{exit,max}$ be the final separation time, which is the time the Deputy has to be outside the Nominal Boundary, and $\Delta t_k = t_f - t_k$ be the current separation time. We define the Current Ellipsoid (CE) as a 1x2x1 ellipsoid centered at the Chief, with x semi-axis a_k and passing through the current relative position of the Deputy, $\mathbf{r}_k$ (see Figure 6.2).

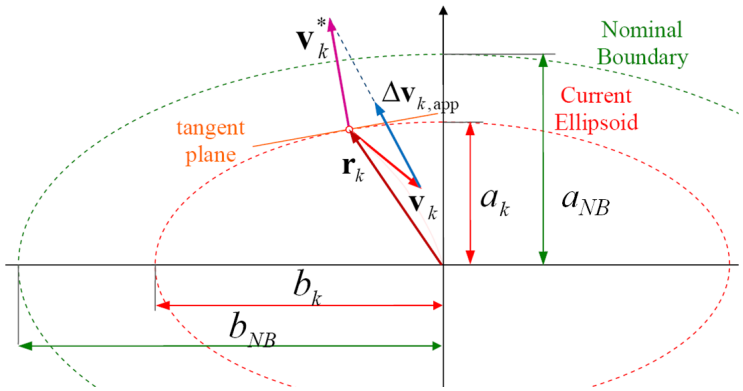


Figure 6.2: Separation Guidance strategy.

A desired current separation velocity $\mathbf{v}_k^*$ is first computed based only on the current position vector $\mathbf{r}_k$. $\mathbf{v}_k^*$ is proportional to the difference between the semiaxes in y direction of the Nominal Boundary, b_{NB}, and the Current Ellipsoid, b_k , and inversely proportional to the current separation time Δt_k:

$$v_k^* = \frac{b_{NB} - b_k}{\Delta t_k}\,. \tag{6.1}$$

$\mathbf{v}_k^*$ is aligned with the external unit vector $\hat{\mathbf{n}}_k$ orthogonal to the plane that is tangent in $\mathbf{r}_k$ to the Current Ellipsoid:

$$\hat{\mathbf{n}} = \frac{1}{\sqrt{(4x_k)^2 + (y_k)^2 + (4z_k)^2}} \begin{bmatrix} 4x_k, & y_k, & 4z_k \end{bmatrix}^T . \tag{6.2}$$

The current delta-v $\Delta\mathbf{v}_k$ is:

$$\Delta\mathbf{v}_k = \mathbf{v}_k^* - \mathbf{v}_k \,. \tag{6.3}$$

The thruster system may not be able to perform the current delta-v vector, and the current control acceleration is therefore computed as:

$$u_k = \frac{1}{T_s} min\,(\,\mathbf{u}_{max}\,T_s,\,\Delta\mathbf{v}_k\,) \tag{6.4}$$

where T_s is the control interval of the separation control and $\mathbf{u}_{max}$ is the maximum acceleration that the thruster system can apply in the $\Delta\mathbf{v}_k$ direction. If the Deputy is already flying away from the Chief, meeting the Separation Guidance requirements, the current control acceleration is not applied.

6.4 Nominal Guidance

The Nominal Guidance starts if the sensed relative position between the Deputy and the Chief is outside the Nominal Boundary and the Collision-Avoidance mode is activated. The Nominal Guidance algorithm computes two basic relative trajectories (see Figure 6.1): an Evasive Relative Trajectory (ERT) and a Safe Relative Trajectory (SRT). A first delta-v Δv_1 is planned at time t_1 so that the Deputy follows the ERT outside the Nominal Boundary; then, a second delta-v Δv_2 is planned at time t_2 and the Deputy finally achieves the SRT.

The couple of trajectories SRT-ERT has to satisfy the following safe requirements: (a) the SRT has to evolve within a Safe Region (Section 6.4.1); (b) the minimum distance of the SRT projected in the cross-track plane has to be greater than a safe minimum distance (Section 6.4.2); and (c) the ERT has to be outside the Nominal Boundary (Section 6.4.3). Furthermore, the whole maneuver has to be performed within a maximum time interval $\Delta t_{e,max}$, requirement (d), and the total delta-v Δv_{tot} used to plan the Nominal Guidance maneuver has to be minimized.

As discussed in the following sections, each of the previous safe requirements is converted in a correspondent constraint condition for those parameters that define the couple of trajectories SRT-ERT.

6.4.1 Safe Relative Trajectory definition

The Safe Region is designed with the double intent to assure the formation safety, keeping the two space vehicles sufficiently far apart, and to prevent formation evaporation, keeping the spacecraft sufficiently close for the communication and relative navigation systems to operate.

Two types of Safe Regions are designed taking into account both in-plane and out-of-plane safe constraints: a Fly-Around type, which defines SRTs that evolve around the Chief, and a No-Fly-Around type, that does not fly around the Chief and whose geometric center is properly placed along track. In order to reduce both the computational complexity of the Nominal Guidance algorithm and the propellant consumption, the previous two SRT types are designed using the Hill-Clohessy-Wiltshire (HCW) solution with the 1^{st} order bounded relative motion condition $\dot{y}_0 = -2\,n\,x_0$ (see Section 2.1.1). A SRT is therefore defined by 5 parameters: $\rho_y = 2\,\rho_x$ and ρ_z, which are the amplitudes of the in-plane and out-of-plane bounded motion, Δy, which is the along-track offset, and φ_{xy} and φ_z, which are the phase angles of the in-plane and out-of-plane motion (Equations 2.13 - 2.15). The value of these five independent parameters have to be properly selected in order for each SRT to evolve inside the corresponding Safe Region.

6.4.1.1 Fly-Around Safe Relative Trajectories

The Fly-Around Safe Region is a 1x2x1 elliptical section cylinder with a central empty region (Figure 6.3).

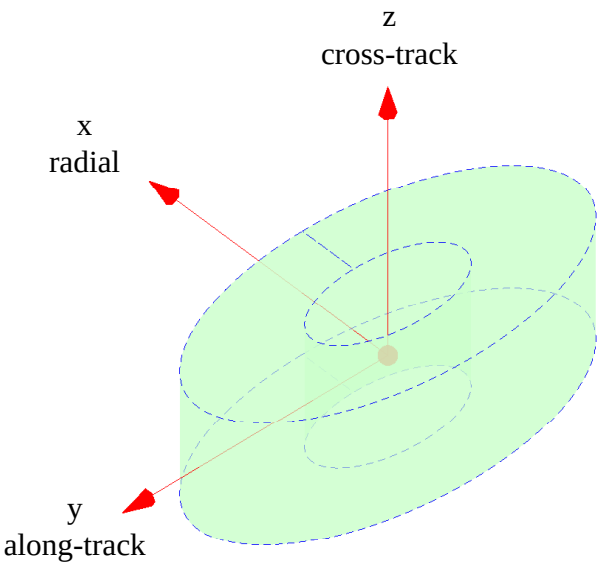


Figure 6.3: 3D visualization of the Safe Region for the Fly-Around Safe Relative Trajectory case.

The in-plane evolution of the SRT can be designed independently of the out-of-plane evolution, since they are uncoupled in the HCW model. The in-plane projection of the Safe Region and the constraint conditions (c/c) for Fly-Around SRTs are presented in Figure 6.4. A Fly-Around SRT evolves within the Fly-Around Safe Region if the values of ρ_y and Δy satisfy the following two constraint conditions c/c 1 and c/c 2:

$$\text{c/c 1:} \begin{cases} y_{min} \geq y^*_{min} \\ y_{min} = \rho_y - |\Delta y| \end{cases} \Rightarrow \rho_y - |\Delta y| \geq y^*_{min} \tag{6.5}$$

$$\text{c/c 2:} \begin{cases} y_{max} \leq y^*_{max} \\ y_{max} = |\Delta y| + \rho_y \end{cases} \Rightarrow \rho_y + |\Delta y| \leq y^*_{max} \tag{6.6}$$

Two limit cases can be considered:

- if $\Delta y = 0$, then $\rho_y \in [\, y^*_{min},\, y^*_{max}\,]$;
- if $y^*_{min} + y^*_{max} = 2\,\rho^*_y$, i.e. $\rho^*_y \doteq \frac{y^*_{min}+y^*_{max}}{2}$, then $\Delta y^* \doteq \rho^*_y - y^*_{min} = \frac{y^*_{max}-y^*_{min}}{2}$.

This means that if a value of Δy is selected between 0 and Δy^*, then the value of ρ_y has to be properly chosen in order to satisfy constraint conditions c/c 1 and c/c 2 (Equations 6.5 and 6.6).

For what concerns the out-of-plane evolution of the SRT, a constraint on the maximum amplitude in this direction is considered that translates into:

$$\rho_z \in [\, 0,\, z^*_{max}\,] \tag{6.7}$$

where z^*_{max} is the maximum amplitude of the out-of-plane motion.

The in-plane and out-of-plane phase angles φ_{xy} and φ_z can assume a value between 0 and $2\,\pi$.

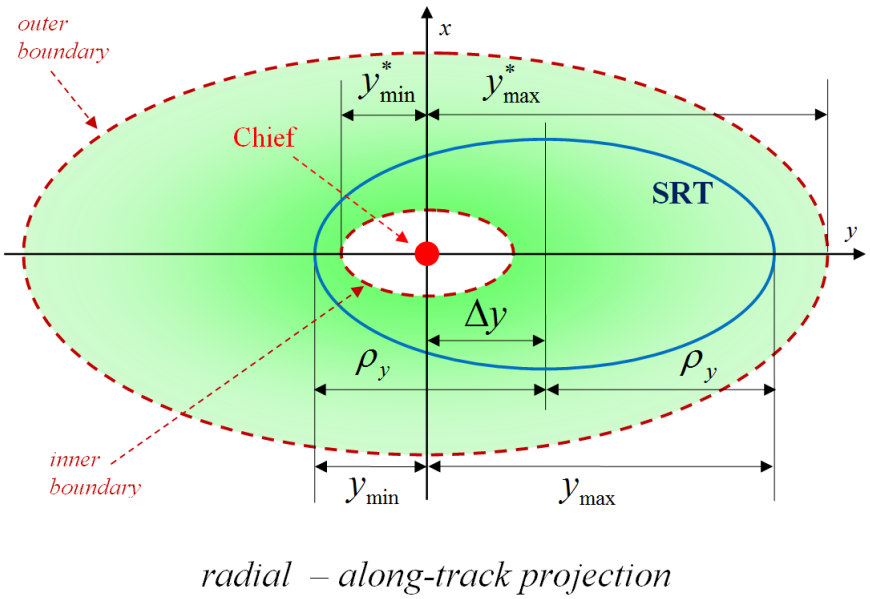


Figure 6.4: Constraints for the In-plane motion of the Fly-Around Safe Relative Trajectory.

6.4.1.2 Not-Fly-Around Safe Relative Trajectories

The Not-Fly-Around Safe Region consists of two parallelograms with zero offset in radial and cross-track directions and with equal and opposite offset in the along-track direction (Figure 6.5).

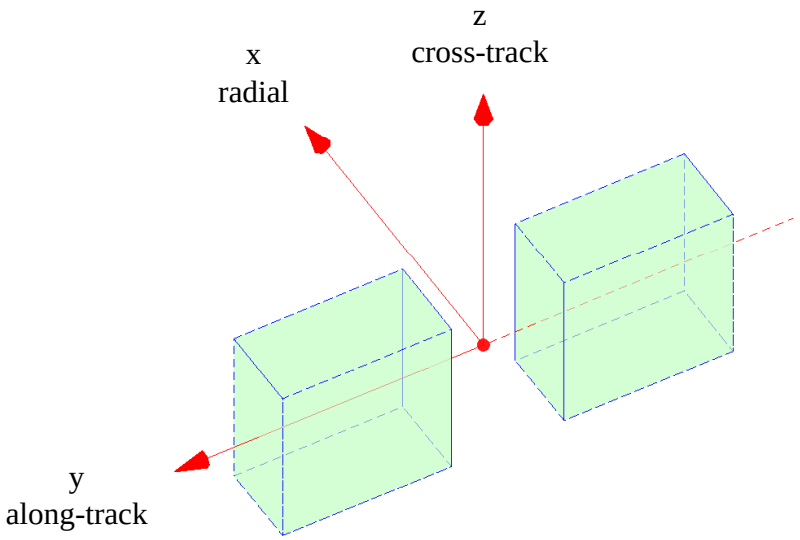


Figure 6.5: 3D visualization of the Safe Region for the Not-Fly-Around Safe Relative Trajectory case.

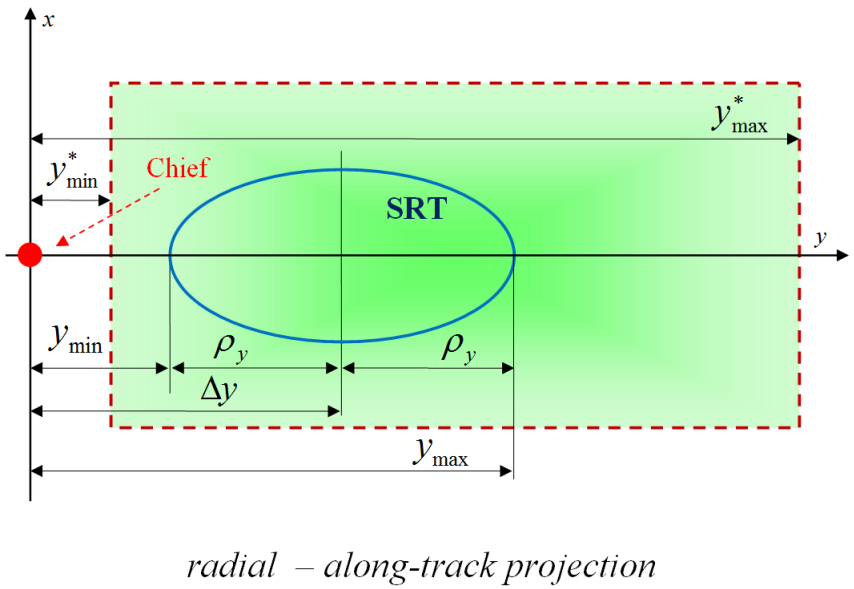


Figure 6.6: Constraints for the In-plane motion of the Not-Fly-Around Safe Relative Trajectory (positive y-direction case).

Figure 6.6 represents the in-plane projection of the Safe Region and the constraint conditions for No-Fly-Around SRTs. A No-Fly-Around SRT evolves within the No-Fly-Around Safe Region if

the following two constraint conditions, c/c 3 and c/c 4, are satisfied by ρ_y and Δy:

$$\text{c/c 3:} \begin{cases} y_{min} \geq y^*_{min} \\ y_{min} = |\Delta y| - \rho_y \end{cases} \Rightarrow |\Delta y| - \rho_y \geq y^*_{min} \tag{6.8}$$

$$\text{c/c 4:} \begin{cases} y_{max} \leq y^*_{max} \\ y_{max} = |\Delta y| + \rho_y \end{cases} \Rightarrow |\Delta y| + \rho_y \leq y^*_{max} \tag{6.9}$$

Two limit cases can be considered:

- if $\Delta y = y^*_{min}$ or $\Delta y = y^*_{max}$, then $\rho_y = 0$;
- if $\Delta y^* = \frac{y^*_{min}+y^*_{max}}{2}$, then $\rho^*_y \doteq \frac{y^*_{max}-y^*_{min}}{2}$, maximum value for ρ_y.

It a value of Δy is therefore selected between y^*_{min} and y^*_{max}, then the value of ρ_y has to be chosen in order to satisfy constraint conditions c/c 3 and c/c 4 (Equations 6.8 and 6.9).

The amplitude of the out-of-plane motion can get any value between 0 and the maximum value z^*_{max}. The in-plane and out-of-plane phase angles φ_{xy} and φ_z can assume a value between 0 and $2\,\pi$.

6.4.2 Cross-track minimum distance

In some scenarios it is essential to ensure a minimum distance in the cross-track plane to guarantee the safety of the formation flying satellites, even in the presence of along-track drift. Some small amount of along-track drift is always present after a maneuver due to unavoidable thruster inaccuracies, navigation errors and perturbations. Therefore, it is important to find safe relative trajectories that are independent of the along-track position.

We define *cross-track avoidance region* as a safe circle in the cross-track plane, centered at the Chief and with radius r_{CT}. The minimum distance of the SRT projected in the cross-track plane is greater than the safe minimum distance r_{CT}, requirement (b), if the semi-minor axis b of the ellipse obtained by projecting the SRT in the cross-track plane is greater than r_{CT} , that is (Figure 6.7):

$$\text{c/c 5:}\, b = \sqrt{\frac{\rho_x^2 + \rho_z^2 - \sqrt{\rho_x^4 + \rho_z^4 + 2\,\rho_x^2\,\rho_z^2\,cos\,2\alpha}}{2}} \geq r_{CT} \tag{6.10}$$

where $\alpha = \varphi_{xy} - \varphi_z$

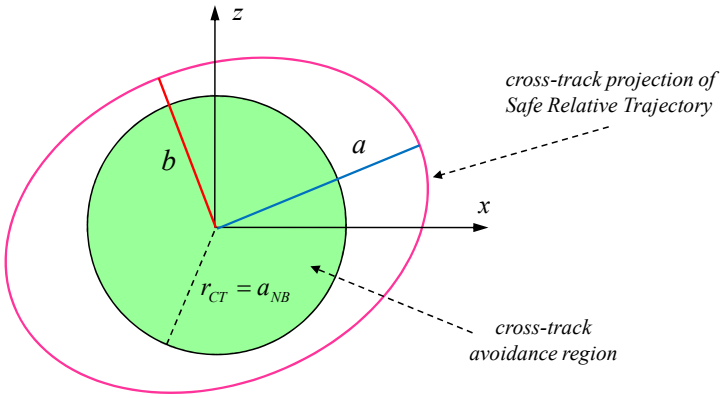


Figure 6.7: Constraint on SRT cross-track minimum distance.

6.4.3 Evasive trajectories constraints

To check if the ERT is outside the Nominal Boundary, the ERT is divided into m points $P_{ERT,j}$, with $j = 1, ..., m$. Let $d_{ERT,j}$ be the distance of the point $P_{ERT,j}$ from the origin of the Hill's frame, and let $d_{NB,j}$ be the distance from the frame origin of the point $P_{NB,j}$ obtained intersecting the ray from the origin to $P_{ERT,j}$ with the Nominal Boundary (see Figure 6.8).

The ERT evolve outside the Nominal Boundary if the following constraint condition c/c 6 is satisfied:

$$\text{c/c 6: } \min_{j=1,...,m} (d_{ERT,j} - d_{NB,j}) \geq 0 \tag{6.11}$$

Let $\mathbf{r}_P = \begin{bmatrix} x_P, & y_P, & z_P \end{bmatrix}^T$ be the position of a point $P = P_{ERT}$ in the ERT. The equation of the line connecting the frame origin O and P is:

$$\begin{cases} x = \frac{x_P}{z_P} z \\ y = \frac{y_P}{z_P} z \end{cases} . \tag{6.12}$$

The z coordinates of the two points obtained by intersecting this line with the Nominal Boundary ellipsoid are:

$$\overline{z} = \sqrt{\frac{z_P^2}{(x_P/a_{NB})^2 + (y_P/b_{NB})^2 + (z_P/c_{NB})^2}} \tag{6.13}$$

The coordinates $r_Q = \begin{bmatrix} x_Q, & y_Q, & z_Q \end{bmatrix}^T$ of the intersection point $Q = P_{NB}$ between the ellipsoid and the OP ray are as follow:

$$
\begin{aligned}
z_Q &= \begin{cases} \overline{z} & \text{if } z_P \geq 0 \\ -\overline{z} & \text{otherwise} \end{cases} \\
x_Q &= z_Q \frac{x_P}{z_P} \\
y_Q &= z_Q \frac{y_P}{z_P}
\end{aligned}
\tag{6.14}
$$

Constraint condition c/c 6 can be rewritten as:

$$
\text{c/c 6: } \min_{j=1,\ldots,m} \left(\left\| \mathbf{r}_{P_{ERT,j}} \right\|_2 - \left\| \mathbf{r}_{P_{NB,j}} \right\|_2 \right) \geq 0 \tag{6.15}
$$

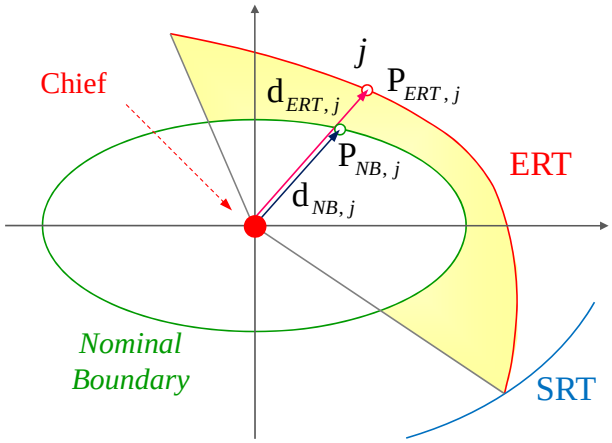


Figure 6.8: Evasive Relative Trajectory constraints.

6.5 Nominal Guidance Algorithm

The Nominal Guidance algorithm uses Genetic Algorithms[1] (GAs) to find a ERT-SRT couple that satisfies (a) (b) (c) and (d) requirements for both Fly-Around and Not-Fly-Around SRTs and minimizes the total delta-v consumption Δv_{tot}.

6.5.1 Genetic Algorithm setup

The optimization variables used in GAs are ρ_y, ρ_z, Δy, φ_{xy}, φ_z and Δt_e. We develop two objective functions (one for the Fly-around case and the other for the Not-Fly-Around case) that receive in input a vector of six elements (optimization variables) whose values are chosen by the Genetic Algorithm tool in the following ranges:

[1] Genetic Algorithm and Direct Search Toolbox™ 2. The MathWorks, Inc., 2009.

- Fly-Around case
  - $\rho_y \in [\, y^*_{min},\, y^*_{max}\,]$;
  - $\rho_z \in [\, 0,\, z^*_{max}\,]$;
  - $\Delta y \in [\, -\Delta y^*,\, \Delta y^*\,]$;
  - $\varphi_{xy} \in [\, 0,\, 2\,\pi\,]$;
  - $\varphi_z \in [\, 0,\, 2\,\pi\,]$;
  - $\Delta t_e \leq \Delta t_{e,max}$.

- Not-Fly-Around case
  - $\rho_y \in \left[\, 0,\, \rho^*_y\,\right]$;
  - $\rho_z \in [\, 0,\, z^*_{max}\,]$;
  - $\Delta y \in [\, y^*_{min},\, y^*_{max}\,]$ and $\Delta y \in [\, -y^*_{max},\, -y^*_{min}\,]$;
  - $\varphi_{xy} \in [\, 0,\, 2\,\pi\,]$;
  - $\varphi_z \in [\, 0,\, 2\,\pi\,]$;
  - $\Delta t_e \leq \Delta t_{e,max}$.

Parameters y^*_{min}, y^*_{max}, z^*_{max} and $\Delta t_{e,max}$, Fly-Around or Not-Fly-Around SRT type and the constraint on the minimum distance of the SRT in the cross-track plane are elements properly chosen in accordance with the specific mission scenario.

Choosing appropriate parameters and methods in GA is very important, since a good configuration might cause the algorithm to converge to best results in a short time while a worse setting might cause the algorithm to run for a long time before finding a good solution or even it might never be able to find a good solution.

Many simulations were conduced to find an acceptable configuration of GA parameters in order to obtain a solution sufficiently near the best one in a reasonable time. Finding a solution that is as near the best one as possible means less fuel consumption, but needs higher computational time and CPU resources. Since in the Collision Avoidance mode formation safety is more important than fuel consumption, the algorithm has to give an acceptable solution in a relatively short time.

Results show that smaller size of population looses the diversity very soon, before even finding a good solution; higher population size provides a higher diversity and therefore contains more sample solutions. As a result, converging to better solutions happens sooner than smaller population sizes. Bigger populations needs however more time for the algorithm to run, specially the time taken for

sorting and evaluate the fitness of individuals is very CPU intensive. Population size is therefore a trade off solution.

Once a solution is obtained with GA , a Pattern Search (PS) algorithm is run for 10 s to refine GA solution, giving about a 3% reduction of the cost function. PS algorithm use the same objective functions of GA and use GA solution as starting point.

6.5.2 Total delta-v computation

The total delta-v computation Δv_{tot} is computed using the HCW model. Let $\mathbf{v}_1^-$ and $\mathbf{v}_1^+$ be the relative velocity vectors of Deputy before and after the application of $\Delta\mathbf{v}_1$, respectively. Similarly for $\mathbf{v}_2^-$, $\mathbf{v}_2^+$ and $\Delta\mathbf{v}_2$. Let $\mathbf{r}_2$ and $\mathbf{v}_2^+$, which define the SRT, be given position and velocity vectors that the Deputy should has at time t_2. $\mathbf{r}_2$ and $\mathbf{v}_2^-$ can be written as function of $\mathbf{r}_1$, $\mathbf{v}_1^-$ and $\Delta\mathbf{v}_1$ using the HWC state transition matrix $\mathbf{\Phi}$ as follow:

$$\begin{bmatrix} \mathbf{r}_2 \\ \\ \mathbf{v}_2^- \end{bmatrix} = \mathbf{\Phi} \begin{bmatrix} \mathbf{r}_1 \\ \\ \mathbf{v}_1^+ \end{bmatrix} = \mathbf{\Phi} \begin{bmatrix} \mathbf{r}_1 \\ \\ \mathbf{v}_1^- + \Delta\mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}_{rr} & \mathbf{\Phi}_{rv} \\ \\ \mathbf{\Phi}_{vr} & \mathbf{\Phi}_{vv} \end{bmatrix} \left(\begin{bmatrix} \mathbf{r}_1 \\ \\ \mathbf{v}_1^- \end{bmatrix} + \begin{bmatrix} 0 \\ \\ \Delta\mathbf{v}_1 \end{bmatrix} \right). \quad (6.16)$$

From the last equation, it is possible to compute $\Delta\mathbf{v}_1$ so that the Deputy reaches position $\mathbf{r}_2$ at time t_2:

$$\mathbf{r}_2 = \mathbf{\Phi}_{rr}\mathbf{r}_1 + \mathbf{\Phi}_{rv}\mathbf{v}_1^- + \mathbf{\Phi}_{rv}\Delta\mathbf{v}_1; \quad (6.17)$$

$$\mathbf{\Phi}_{rv}\Delta\mathbf{v}_1 = \mathbf{r}_2 - \mathbf{\Phi}_{rr}\mathbf{r}_1 + -\mathbf{\Phi}_{rv}\mathbf{v}_1^-. \quad (6.18)$$

$\Delta\mathbf{v}_1$ is obtained solving the last linear system. $\Delta\mathbf{v}_2$ is computed as follow:

$$\mathbf{v}_1^+ = \mathbf{v}_1^- + \Delta\mathbf{v}_1; \quad (6.19)$$

$$\mathbf{v}_2^- = \mathbf{\Phi}_{vr}\mathbf{r}_1 + \mathbf{\Phi}_{vv}\mathbf{v}_1^+; \quad (6.20)$$

$$\Delta\mathbf{v}_2 = \mathbf{v}_2^+ - \mathbf{v}_2^-. \quad (6.21)$$

The total delta-v is then:

$$\Delta v_{tot} = \|\Delta \mathbf{v}_1\| + \|\Delta \mathbf{v}_2\| \,. \tag{6.22}$$

6.5.3 Constraint conditions

For each given optimization vector, a ERT-SRT couple is defined. (a)-(b)-(c) requirements have to be satisfied by this ERT-SRT couple. This is true if the corresponding constraint conditions are verified, according to the following Table 6.1.

To ensure that the six conditions of Table 6.1 are met, six penalization terms are properly added to Δv_{tot}. These six penalization terms are equal to zero only if the corresponding constraint conditions are satisfied; on the other hand, they present an exponential rise as explained below.

A couple of *checking parameters* (μ , μ^*) is introduced for each constraint condition, which can indeed be rewritten in the form $\mu_i \geq \mu_i^*$, with $i = 1, 2, \ldots, 6$. The definition of μ and μ^* for each constraint condition is reported in Table 6.2. If the i^{th} constraint condition is not verified, then $\mu_i < \mu_i^*$ and the corresponding penalization term is:

$$p_i = e^{k_i \left(1 - \frac{\mu_i}{\mu_i^*}\right)} - 1, \tag{6.23}$$

where

$$k_i = \frac{ln\,(\gamma_i - 1)}{1 - \alpha_i}. \tag{6.24}$$

This means that if $\mu_i = \alpha_i \, \mu_i^*$, with $\alpha \in [\,0,\, 1\,)$, then $p_i = \gamma_i$. k_i is introduced to control the rising rate of each penalization term in relation to the others and it can also be used to weight one constraint violation more than the other ones.

Table 6.1: Requirements – Constraint conditions correspondence.

Requirements	Constraint Conditions
(a)	(1)-(2) for a Fly-Around SRT (3)-(4) for a Not-Fly-Around SRT
(b)	(5)
(c)	(6)

Table 6.2: Definition of μ_i and μ_i^* parameters for each constraint condition.

c/c	c/c not satisfied	μ_i	μ_i^*
(1)	$\rho_y - \lvert\Delta y\rvert < y^*_{min}$	$\rho_y - \lvert\Delta y\rvert$	y^*_{min}
(2)	$\rho_y + \lvert\Delta y\rvert > y^*_{max}$	$-\rho_y - \lvert\Delta y\rvert$	$-y^*_{max}$
(3)	$\lvert\Delta y\rvert - \rho_y < y^*_{min}$	$\lvert\Delta y\rvert - \rho_y$	y^*_{min}
(4)	$\lvert\Delta y\rvert + \rho_y > y^*_{max}$	$-\lvert\Delta y\rvert - \rho_y$	$-y^*_{max}$
(5)	$b < r_{CT}$	b	r_{CT}
(6)	$d_{ERT,min} < d_e$	$d_{ERT,min}$	d_e

6.6 Control Strategies

The Collision-Avoidance algorithms have been implemented in Matlab to develop the following two modules for the Formation Flight Matlab Simulator: the Collision Avoidance Separation Guidance (CASG) module, the Collision Avoidance Nominal Guidance (CANG) module and the section of the Mode Manager module that is in charge of the Collision-Avoidance. The Mode Manager has to manage in general all collision scenarios that can occur in each formation flying phase of a mission. It also uses a Collision Detector which, starting from the actual relative dynamic state of the formation, foresees possible collisions within a given time window. The following elements of the proposed Collision Avoidance strategy have to be appointed for each formation flying phase:

- parameters and conditions for the Collision Avoidance mode activation. e.g. relative position and/or velocity and collision detection;
- values of those parameters used by separation and nominal guidance algorithms, e.g. a_{NB}, a_{AR}, SRT type, $\Delta t_{e,max}$.

Referring to Figure 6.9, if the Collision- Avoidance mode is activated, the Mode Manager handles activation/deactivation of the CASG and CANG modules through S^{MM}_{CASG} and S^{MM}_{CANG} commands. When the CASG module is activated, it computes the control actions that the thrusters system has to execute, based on the estimation of the actual relative position $\widehat{\mathbf{x}}$. When the CANG is activated, it carries out the computation of the reference state trajectory $(\overline{\mathbf{x}}, \overline{\mathbf{v}})$ starting from the current estimated relative dynamic state $(\widehat{\mathbf{x}}, \widehat{\mathbf{v}})$. The reference trajectory is utilized in the control module to compute the control acceleration for trajectory tracking.

Two types of control strategies were used to perform the tracking of the reference trajectories: the first one is based on Linear Quadratic Regulator (LQR), and the second one is based on Model Predictive Control (MPC). Please see Chapter 3 for a detailed description of these two controllers.

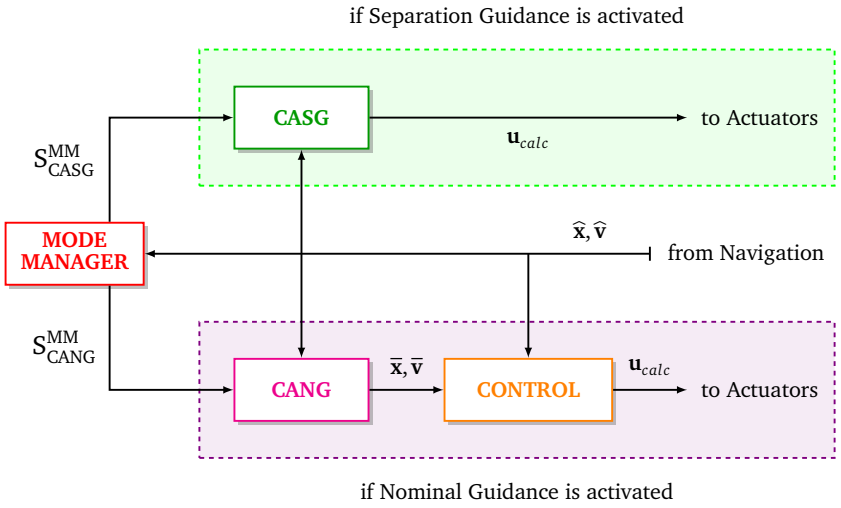


Figure 6.9: Guidance and Control block diagram for the Collision-Avoidance mode.

6.7 Simulation Results

In this section we present results of a Collision-Avoidance maneuver in which the Deputy spacecraft is inside the Avoidance Region at the initial time t_i and the Collision-Avoidance mode is activated. Therefore, Separation Guidance is performed at first with the aim of place the Deputy outside the Nominal Boundary within a prescribed time; once the Deputy is outside the Nominal Boundary, the Nominal Guidance takes over to park it in a safe trajectory inside a prescribed safe region.

The initial dynamic state of the Chief is given in Table 6.3 in terms of orbital elements of the osculating keplerian orbit. The initial dynamic state of the Deputy is given in Table 6.4 as relative position $\mathbf{r}_i$ and velocity $\mathbf{v}_i$ vectors with respect to the Hill's frame. In order to simulate a collision, the value of $\mathbf{v}_i$ is obtained forcing the Deputy to reach the origin of the Hill's frame in 5 minutes. Separation Guidance parameters and Nominal Guidance parameters are listed in Table 6.5. A $\Delta t_{e,max}$ of 69.56 minutes is equal to 0.75 times the initial orbital period of the Chief.

Table 6.3: Chief keplerian orbital elements.

semi-major axis a [km]	true anomaly ϑ [deg]	eccentricity e [ ]	RAAN Ω [deg]	inclination i [deg]	argument of perigee ω [deg]
6778	60	0	30	45	20

Table 6.4: Deputy relative initial position and velocity vectors.

relative initial position $\mathbf{r}_0$ [m]	relative initial velocity $\mathbf{v}_0$ [10^{-3} m/s]
$[-120\,,\,50\,,\,-60]^T$	$[-374.761\,,\,-300.006\,,\,192.260]^T$

Table 6.5: Separation Guidance parameters (left) and Nominal Guidance parameters (right).

Separation Guidance parameters					Nominal Guidance parameters				
a_{AR} [m]	a_{NB} [m]	$\Delta t_{exit,max}$ [min]	T_s [s]	u_{max} [m/s^2]	r_{CT} [m]	$\Delta t_{e,max}$ [min]	y^*_{min} [m]	y^*_{max} [m]	z^*_{max} [m]
100	200	7	5	$3 \cdot 10^{-2}$	200	69.56	450	1000	700

Table 6.6: Genetic Algorithms and Pattern Search results – Nominal Guidance maneuver.

	ρ_y [m]	ρ_z [m]	Δy [m]	φ_{xy} [rad]	φ_z [rad]	Δt_e [s]	Δv_{tot} [m/s]
GA	711.300	201.946	−260.883	5.4253	3.7412	35.94.86	$3.5593 \cdot 10^{-1}$
PS	711.300	201.946	261.300	5.3896	3.7040	3620.87	$3.5495 \cdot 10^{-1}$
difference [%]	0.00	0.00	0.16	−0.66	−0.99	0.72	−0.27

Table 6.7: Position, velocity and delta-v vectors – Nominal Guidance maneuver.

t	$\mathbf{r}(t)$ [m]	$\mathbf{v}(t^-)$ $\cdot 10^{-3}$ [m/s]	$\mathbf{v}(t^+)$ $\cdot 10^{-3}$ [m/s]	$\Delta\mathbf{V}(t)$ $\cdot 10^{-3}$ [m/s]	$\|\Delta\mathbf{V}(t)\|$ $\cdot 10^{-3}$ [m/s]
t_1	183.553 −25.470 −78.728	−299.887 −255.258 −83.595	270.444 −434.602 −49.100	−29.443 −179.343 132.696	225.031
t_2	−354.900 −215.125 10.052	44.747 783.773 −101.089	26.120 803.043 −228.191	−18.627 19.270 −127.122	129.917

Figure 6.10 shows the time profile of the control acceleration components during the Separation Guidance maneuver. Thrusters are switched on for the first 24 seconds of the maneuver to ensure that the Deputy is outside the Nominal Boundary within 7 minutes. Figure 6.11 shows the Deputy/Chief distance profile: the Deputy is outside the Avoidance Region and the Nominal Boundary after 34 seconds and 6.37 minutes, respectively.

Results of Genetic Algorithm and Pattern Search for the Nominal Guidance maneuver are presented in Table 6.6. These results were obtained using Matlab Genetic Algorithm and Direct Search Toolbox™, with a population size of 4000 individuals and with a maximum computational time for GA and PS algorithms of 1.5 minutes. The corresponding position, velocity and delta-v vectors are listed in Table 6.7. Figures 6.12 and 6.13 show in-plane and out-of-plane projection, respectively, of the Fly- Around Safe Region and the resulting couple of trajectories SRT-ERT. As can be noted from these figures, all constraint conditions are satisfied.

A comparison of LQR and MPC control acceleration components near times t_1 and t_2 is shown in Figures 6.15 and 6.17, respectively. The corresponding tracking error time profiles are reported in Figures 6.14 and 6.16. These results are obtained with a control horizon $m = 5$, a prediction horizon $p = 25$, a control interval $T_s = 5\,s$ for both controllers, while the state error and the control weighting matrices of both controllers are set to obtain similar performances in tracking the same reference trajectories. For the Nominal Guidance maneuver, the total delta-v obtained with LQR and MPC are $\Delta v_{LQR} = 1.4757\,m/s$ and $\Delta v_{MPC} = 1.3914\,m/s$, respectively. In this scenario, the use of an MPC controller reduces the total delta-v of 5.71 % with respect to an LQR controller. This is due also to the anticipative action of the MPC, as can be noted in Figure 6.17.

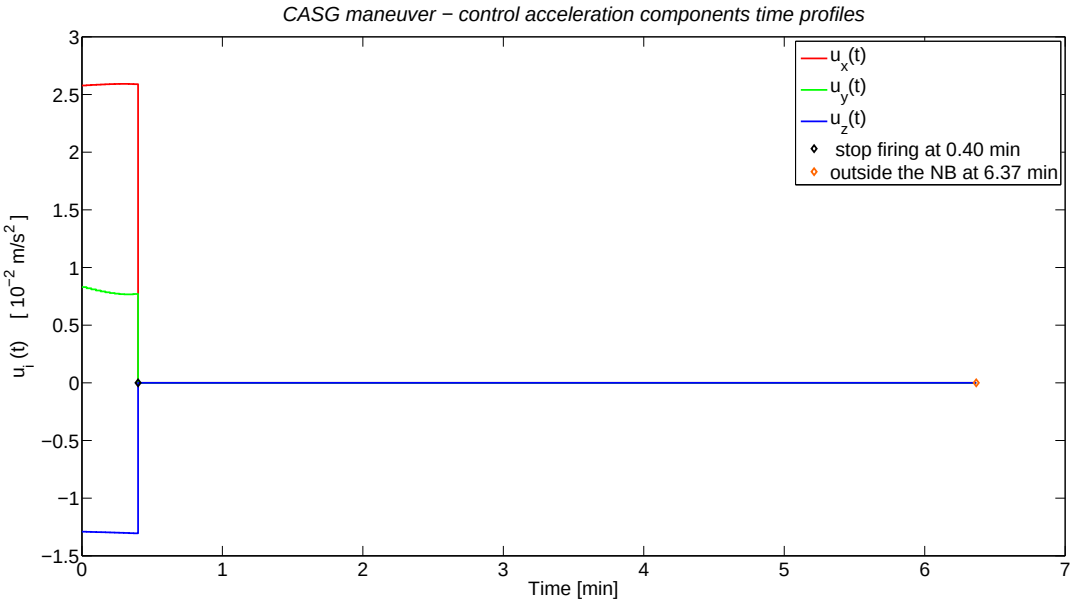


Figure 6.10: Control acceleration components time profiles – Separation Guidance maneuver. $\Delta v_{tot,CASG} = 1.80\,m/s$.

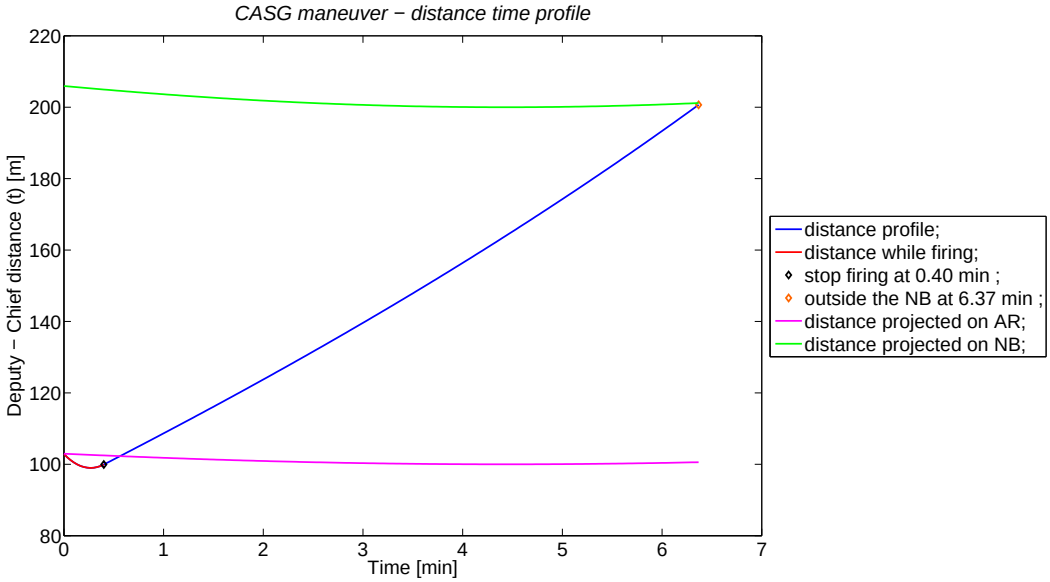


Figure 6.11: Distance profiles – Separation Guidance maneuver.

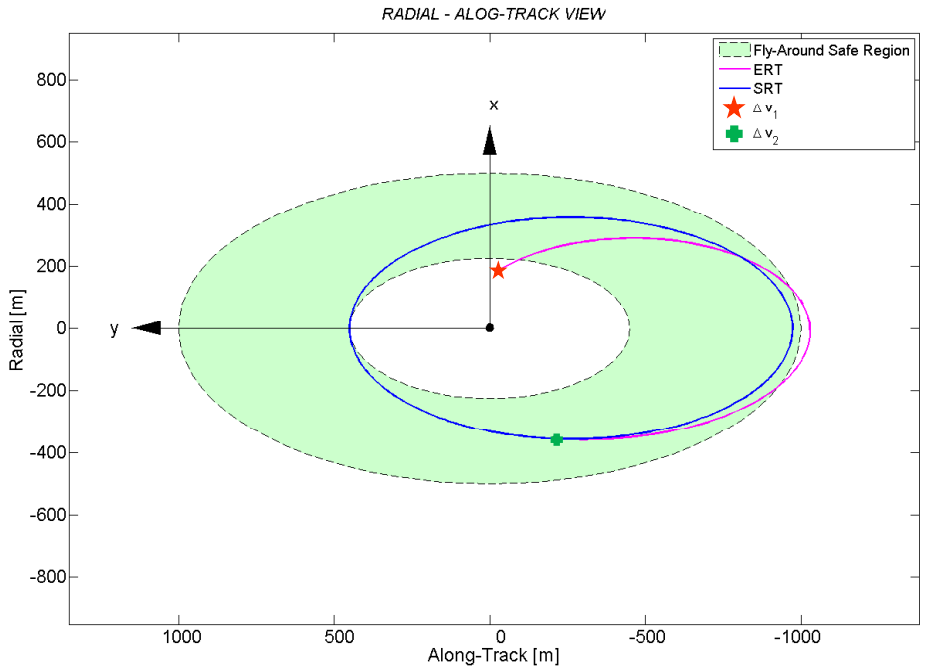


Figure 6.12: Out-of-plane projection of the Nominal Guidance maneuver. Δv_1 application point marked with an orange star, Δv_2 marked with a green cross.

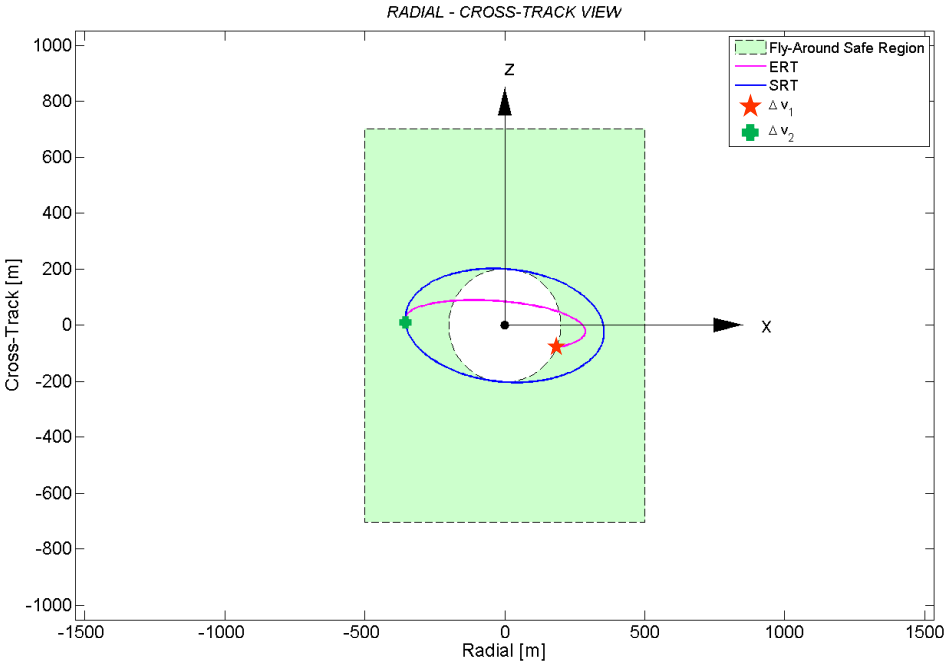


Figure 6.13: Out-of-plane projection of the Nominal Guidance maneuver. Δv_1 application point marked with an orange star, Δv_2 marked with a green cross.

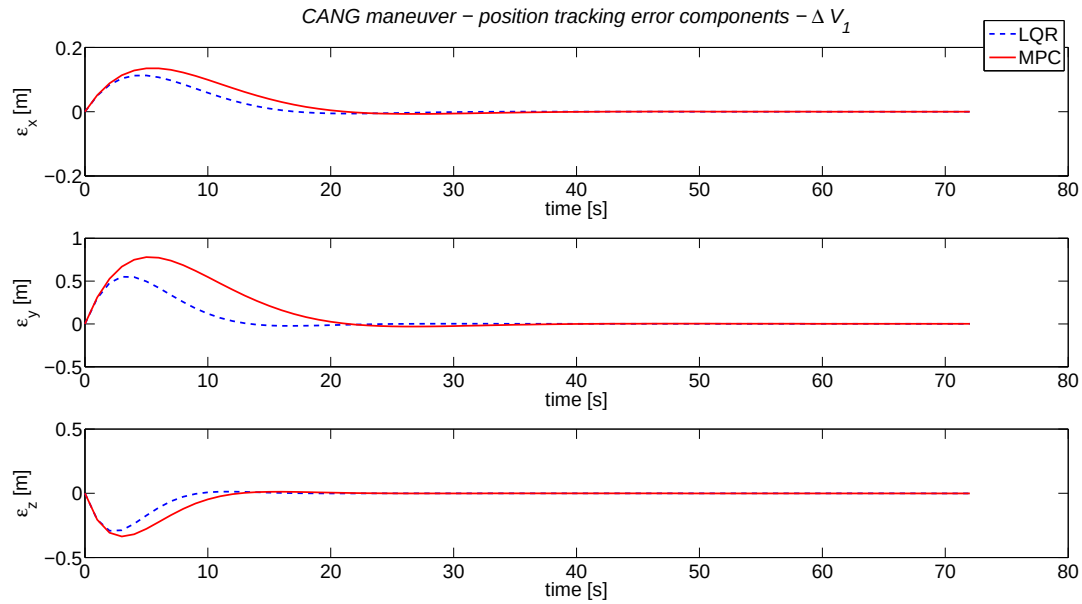


Figure 6.14: LQR and MPC tracking error comparison near time t_1.

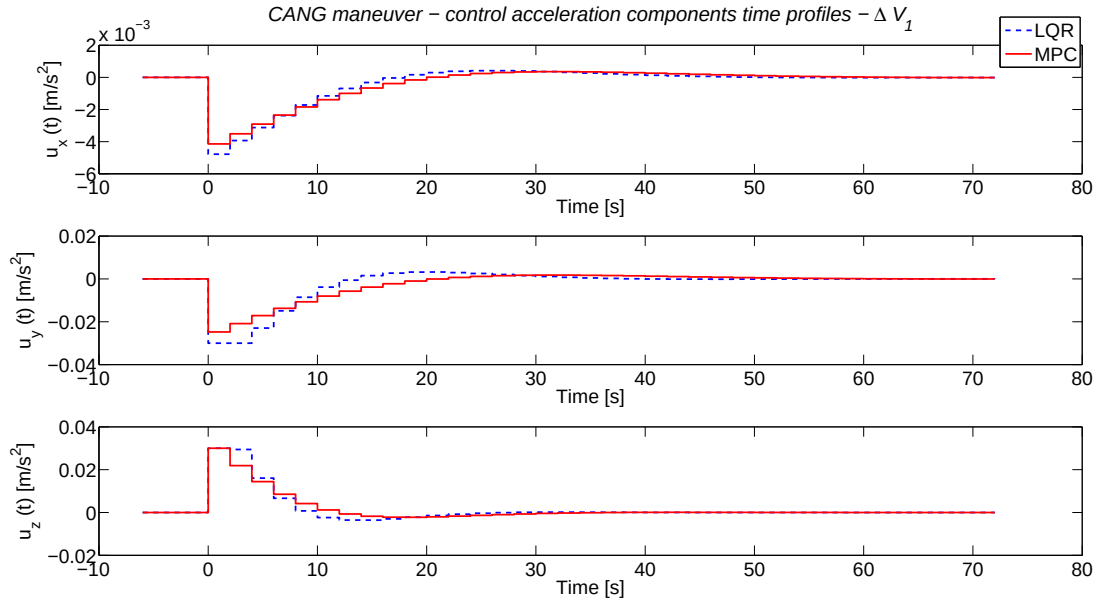


Figure 6.15: LQR and MPC control acceleration components time profiles near time t_1.

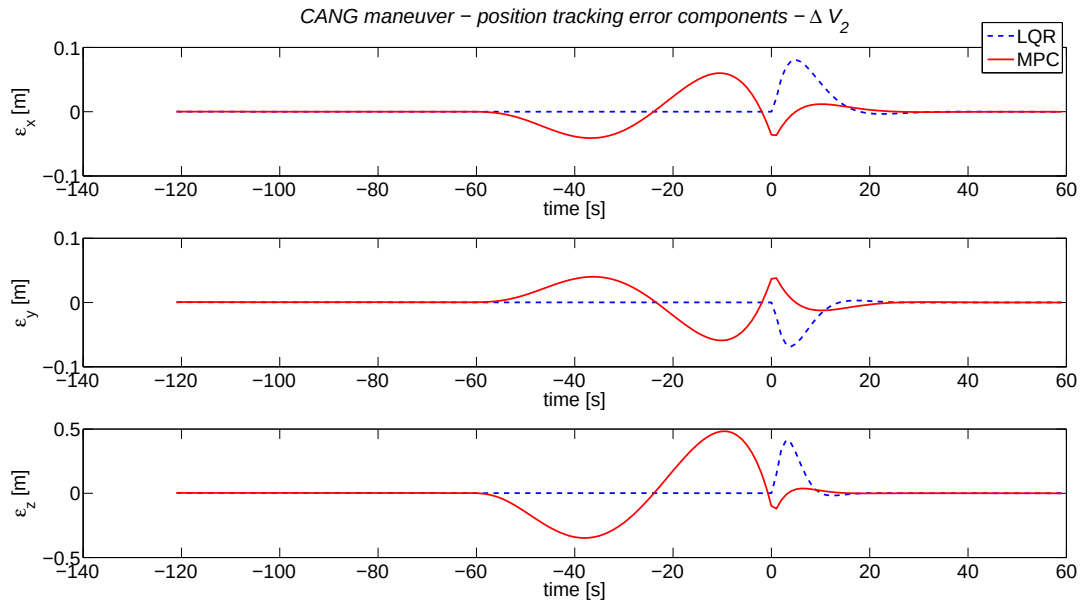


Figure 6.16: LQR and MPC tracking error comparison near time t_2.

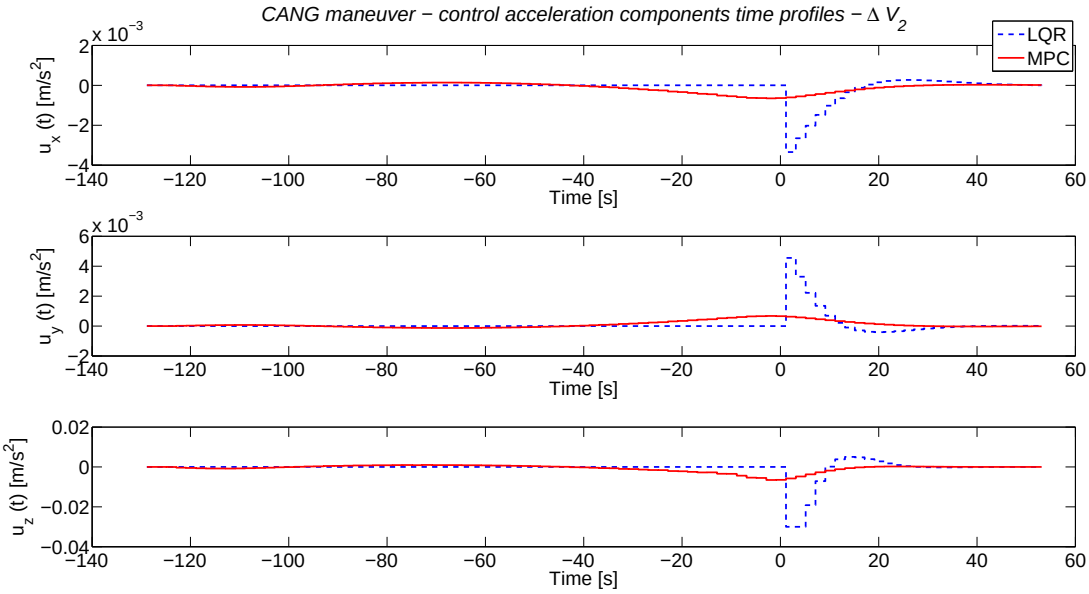


Figure 6.17: LQR and MPC control acceleration components time profiles near time t_2.

Chapter 7

MPC FOR THE SPHERES MOSR SCENARIO

7.1 MOSR Scenario Overview

The current mission scenario of the NASA's Mars Sample Return (MSR) mission [80] utilizes a small Orbiting Sample (OS) satellite, launched from the surface of Mars and with some geological samples on board recovered by a catching rover, which will rendezvous with an Earth Return Vehicle (ERV) in Martian orbit (Figure 7.1). The guidance of the OS into the capture mechanism on the ERV is considered to be one of the highest-risk operations. Since the OS will most likely be passive during this maneuver - possibly outfitted with a radio beacon for long-distance detection, but with no means of active propulsion or attitude control - the ERV must determine the OS' location in Martian orbit using a single visual-band camera, and maneuver itself to capture it.

Figure 7.1: Artist's rendition of Mars sample launching from MSR lander (left); MSR Orbiter performing OS target search and acquisition in Mars orbit (right).

The Massachusetts Institute of Technology (MIT), in a partnership with Aurora Flight Sciences, developed the Mars Orbital Sample Return (MOSR) system with the final aim of testing the visual tracking and relative motion control algorithms [81] (Figure 7.2). This system uses the Synchronized Position Hold Engage & Reorient Experimental Satellite (SPHERES) test bed to emulate the combined motion of the capture satellite and the OS. The key elements of the system are: (1) a moving SPHERES satellite modified with a white and black shell to match the on orbit operative lighting conditions (Figure 7.3); (2) a stationary capture mechanism; and (3) an optical tracking system with cameras mounted on the capture mechanism. Software on the capture mechanism implements a vision tracking algorithm (based on the Hough transform, astronomical photometry and Kalman Filtering), computes the likely maneuver commands for a capture satellite, which are then translated into relative motions to be performed by a SPHERES satellite, acting as the OS (Figure 7.4).

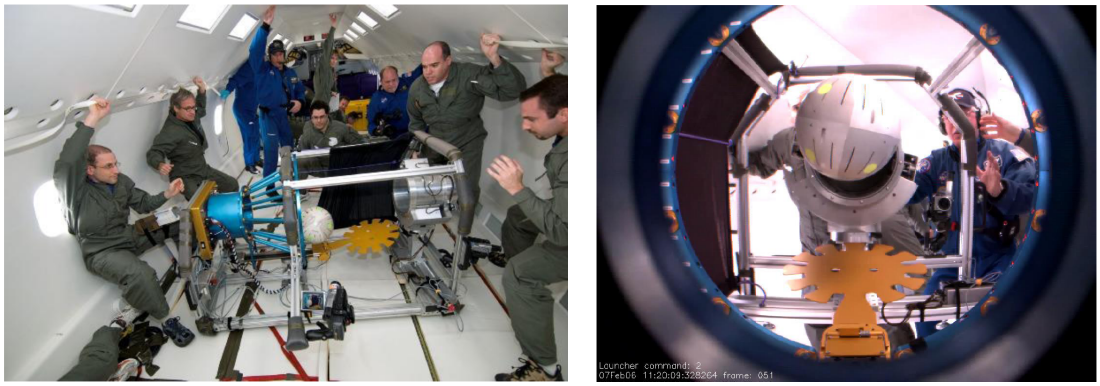


Figure 7.2: SPHERES-MOSR test bed performing OS contact dynamics experiments on reduced gravity flight (left); boresight view of SPHERES-MOSR testbed (right) (courtesy of [82]).

Figure 7.3: OS Shell with a SPHERE satellite for comparison (left); OS on an air bearing support structure (right) (courtesy of [82]).

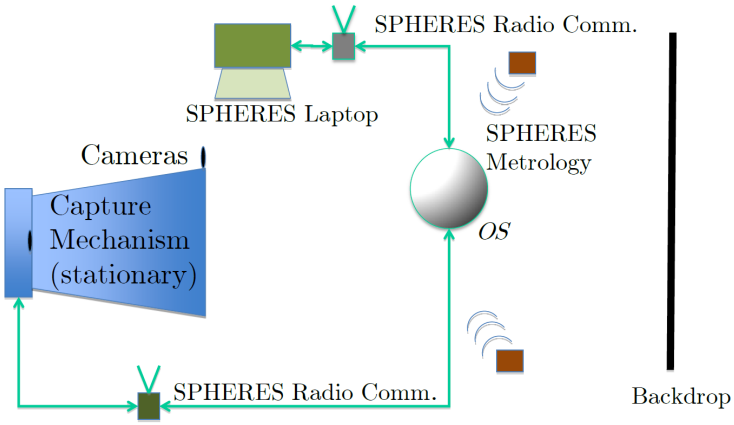


Figure 7.4: Communications architecture for SPHERES MOSR system (courtesy of [82]).

7.2 Controller Requirements and Constraints

There are two possible experimental configurations that can be considered.

1. In the first configuration, a planning algorithm is used to generate a (safe) reference state trajectory[1], starting from an initial estimated state of the chaser s/c and ending to a target state. A controller, such as a MPC-based one, could be used to track the reference trajectory.

2. In the second configuration, a MPC-based controller is used to both compute a reference trajectory and track it at the same time.

We adopted the second control strategy with the aim of testing the MPC capability to compute and track a reference trajectory for the close-proximity phase of the rendezvous and capture maneuver.

Some requirements and/or constraints have to be taken into account (see Figure 7.5) with the aim of:

- guarantee the safety for both the chaser s/c and the target s/c;
- guarantee proper operational conditions for all the elements used to execute the rendezvous and capture maneuver;
- improve the control system performance taking into account the actual system.

Requirements and constraints for the SPHERES MOSR scenario are described in the following.

7.2.1 Field of view constraint

It is required that the chaser remains within the FOV cone of the vision-based sensing system during the rendezvous maneuver.

[1]All trajectories/states are considered as relative trajectories/states of the chaser s/c w.r.t. the target s/c.

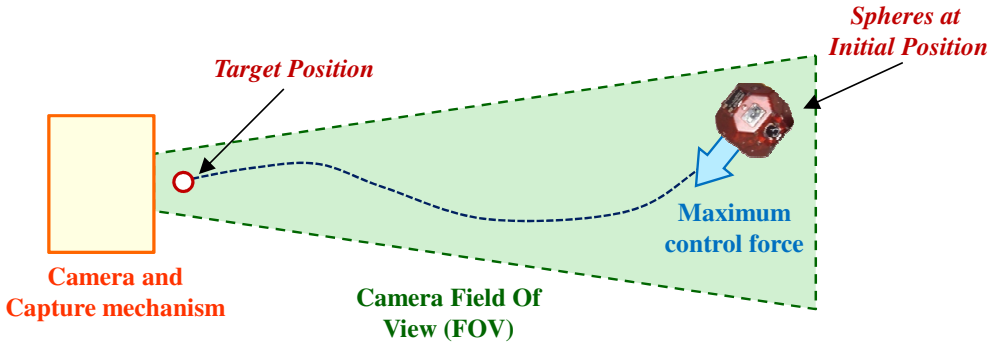


Figure 7.5: Controller requirements and constraints.

This FOV requirement can be represented by the following linear inequality constraint on the relative position vector **p**:

$$\mathbf{H}_{FOV}\,\mathbf{p}(t_k) \leq \mathbf{k}_{FOV} \tag{7.1}$$

for each time step t_k of the rendezvous maneuver. $\mathbf{H}_{FOV}$ and $\mathbf{k}_{FOV}$ are a matrix and a vector, respectively, that define the camera FOV as a 3D convex polytope (see Figures 7.5 and 7.6). The camera FOV has an amplitude in the horizontal plane $fov_h = 60\,deg$, and in the vertical plane $fov_v = 30\,deg$.

7.2.2 Limited control authority

This constraint is due to the maximum control force that can be actuated by the CO_2 propulsion system on board the chaser satellite.

Each thruster on board SPHERES can perform a force $F_{thr} = 0.112\,N$. Taking into account the SPHERES thruster system configuration, the maximum force that can be applied in any direction is $F_{max} = 2\,F_{thr} = 0.224\,N$ (worst case when the force direction is parallel to the thruster firing direction). The mass of the SPHERES satellite is considered constant and equal to $m_{sph} = 4.3\,kg$. To conduct tests at the MIT SSL, each SPHERES satellite is placed on an air carriage with a mass $m_{air\,carriage} = 6.7\,kg$ that allows a low friction 2D translational motion. The resulting maximum control acceleration that the thruster system can actuate at the MIT SSL is then $u_{max} = F_{max}/(m_{sph} + m_{air\,carriage}) = 0.02\,m/s^2$.

The MPC algorithm assumes a Piece Wise Constant (PWC) control acceleration profile, with a control period $\Delta t_{ctrl} = 1s$. A control acceleration is applied using a Pulse Width Modulation (PWM) strategy with a maximum pulse width $pw_{max} = 0.2\,s$, i.e. the acceleration is equal to u_{max} and the width, or time duration, of each pulse is computed in order to preserve the impulse of the acceleration. The maximum equivalent PWC acceleration, which is used in the MPC algorithm,

can be computed as follows:

$$^{MPC}u_{max} = \frac{pw_{max}}{\Delta t_{ctrl}}\, u_{max} = DC_{ctrl}\, u_{max} \tag{7.2}$$

which is $^{MPC}u_{max} = 0.004\, m/s^2$ at the MIT SSL. DC_{ctrl} is the Duty Cycle of the PWM.

Tanking into account these constraints within the control algorithm, MPC is expected to have better performance with respect to other classical control strategies.

7.2.3 Terminal constraints

Terminal constraints are applied to the relative dynamic state of the chaser satellite in proximity of the target position both for safety purposes and to guarantee good conditions for the capture mechanism to work.

We imposed the following constraints on the relative velocity when the SPHERES satellite is close to the target position:

- the relative velocity has to be less than a safety value along the camera focal axis direction;
- the relative velocity has to be almost zero in the other two directions.

This means that the SPHERES satellite approaches the target position following a straight path along the camera focal axis and with a reduced velocity. Constraints on the relative velocity are met by properly tuning the MPC weights (PD gains).

7.2.4 Attitude control

The attitude of the chaser s/c is maintained equal to a target orientation during close-proximity operations. A quaternion-based PD controller is used to regulate the attitude of the SPHERES satellite during the whole maneuver.

7.2.5 Reference Frames definition

We define the following two Reference Frames (RFs) (see Figure 7.6).

- Camera RF (C). It has x and y axes in the camera focal plane and z axis along the camera focal axis. The relative position estimate provided by the vision tracking algorithm is expressed in the Camera RF at first.
- Flat Floor Global RF (G). It is the RF used by the SPHERES Ultra-Sound Global Metrology System. All position and velocity vectors reported in the following are expressed in the Flat Floor Global RF.

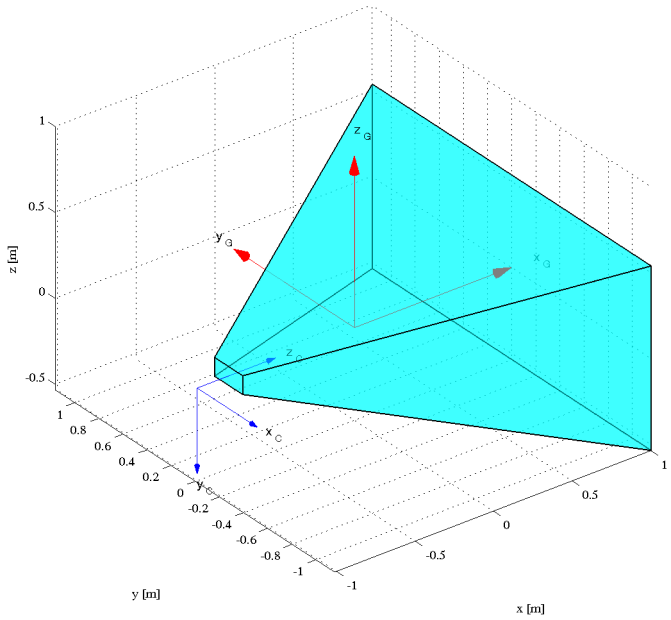


Figure 7.6: Flat Floor Global Reference Frame (G) in red and Camera Reference Frame (C) in blue. The camera FOV is represented by the cyan convex polyhedron.

7.3 MPC Engine for SPHERES

Due to its high computational time when run on board SPHERES, the MPC Engine (see Section 3.4.5) can not be used directly on board SPHERES when a control frequency $f_{ctrl} < 0.1\,\text{Hz}$ is required. In these situations, the MPC problem is solved using the flight laptop and the computed control forces are transmitted back to the SPHERES satellite for the actuation. To manage the time delay due to MPC computation and MPC data exchange between SPHERES and the flight laptop, given an estimation of the dynamic state x_k of the plant at time t_k, a model of the plant is first used to estimate its dynamic state f control steps forward in time, x_{k+f}. The new estimate dynamic state at time t_{k+f} is then used as initial condition for the MPC Engine in order to obtain a sequence of f control accelerations that should be actuated starting at time t_{k+f}, i.e. f control steps forward in time with respect to the initial time t_k.

Figure 7.7 shows the block diagram and the data flow when the position and velocity vectors of the SPHERES satellite are provided by the Vision-Based Relative Navigation System on board the Capture Mechanism. We obtain a similar configuration when the SPHERES satellite state is given by the SPHERES satellite itself using the Ultra-Sound Global Metrology System.

The time schedule of the MPC operations are presented in Figure 7.8, where:

- Δt_{est} is the time interval needed by the Vision-Based Relative Navigation System to estimate position and velocity of the SPHERES satellite w.r.t. the Capture Mechanism;

- $\Delta t_{tx,CPT->MAT}$ is the transmission delay from the Capture Mechanism to Matlab;
- Δt_{MPC} is the time interval needed to solve the MPC problem; it includes forward propagation, QP data updating, QP solver and control sequence updating;
- $\Delta t_{tx,MAT->SPH}$ is the transmission delay from Matlab to the SPHERES satellite.

MPC data is exchanged between SPHERES and Matlab using two types of data packets:

- MPC computation data packet. This data packet is transmitted from SPHERES to Matlab to execute the MPC computation using position and velocity vectors transmitted with this packet as initial dynamic state.
- MPC control acceleration data packet. This data packet, transmitted from Matlab to SPHERES, includes the control acceleration sequence that have to be actuated on SPHERES.

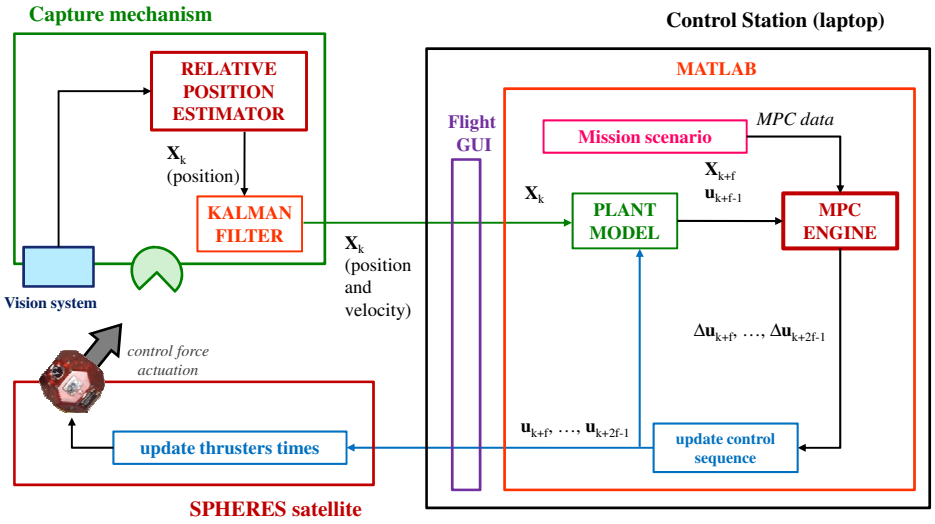


Figure 7.7: Block diagram and data flow of the MPC on-line operations.

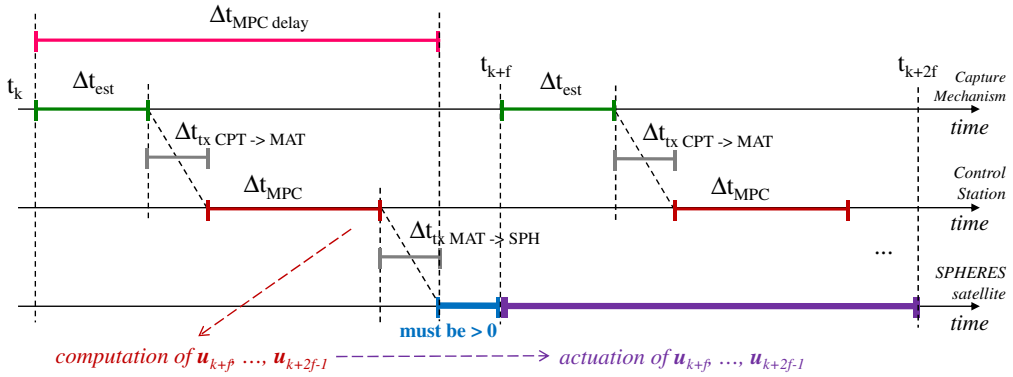


Figure 7.8: Time schedule of the MPC on-line operations.

7.4 Preliminary Simulation Results

In this section we evaluate the behavior of a MPC-based controller in executing the rendezvous and capture maneuver, comparing its performance with a standard Proportional-Derivative (PD) controller.

Two simulation environments were considered:

- the Matlab simulator, which is an ideal environment with no representative SPHERES properties, no sensor/actuator noises and without the attitude dynamics;
- the SPHERES simulator, which is a representative environment of the SPHERES test bed with 6 degrees of freedom.

7.4.1 Matlab Simulator MPC vs Matlab Simulator PD

The initial dynamic state x_i and the final or target dynamic state x_f for this simulation are as follows:

$$x_i = [0.9\,m,\ 1\,m,\ 0.48\,m,\ 0\,m/s,\ 0\,m/s,\ 0\,m/s]^T \tag{7.3}$$

$$x_f = [\,-0.75\,m,\ 0\,m,\ 0\,m,\ 0\,m/s,\ 0\,m/s,\ 0\,m/s]^T\,m \tag{7.4}$$

PD and MPC parameters are tuned in order to obtain comparable settling time for the position error components profiles. Both MPC and PD parameters are listed in Table 7.1. Other parameters used for this simulation are:

- control step $\Delta t_{ctrl} = 1\,s$;
- control acceleration constraints in all directions: $u_{min} = -10^{-2} m/s^2$, $u_{max} = 10^{-2} m/s^2$;
- no forward computation in MPC (see Subsection 7.3).

Simulation results for both MPC and PD controller are compared in the following figures:

- Figure 7.9: 3D position trajectory;
- Figure 7.10: 2D projection of the position trajectory on the yx and zx planes;
- Figure 7.11: position error components (current - reference) vs. time;
- Figure 7.12: velocity error components (current - reference) vs. time;
- Figure 7.13: control acceleration components vs. time;

- Figure 7.14: cumulative Δv requirement vs. time.

From these results you can see that the MPC total Δv requirement is about 11% less than the PD result, with similar settling times for all the position error components of both controllers. This is due to MPC's prediction capability and actuator saturation handling ability (see Figure 7.13).

Table 7.1: MPC and PD parameters - MPC vs PD in the Matlab simulator.

PD		MPC	
${}^{G}k_P$	$10^{-2}[5,\ 8,\ 15]$	${}^{G}w_y$	$10^{-5}[1,\ 15,\ 15]^T$
${}^{G}k_D$	$10^{-1}[5.00,\ 4.65,\ 6.40]$	${}^{G}w_u$	$10^{-1}[1,\ 1,\ 1]^T$
		m	6
		p	$8\,m = 48$

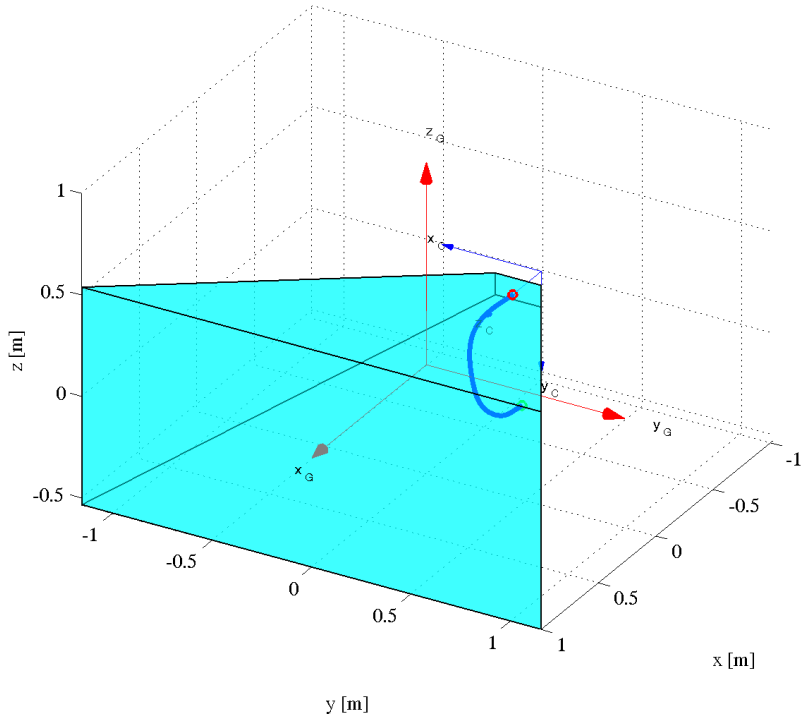


Figure 7.9: 3D Position Trajectory - Matlab Sim. MPC vs Matlab Sim. PD.

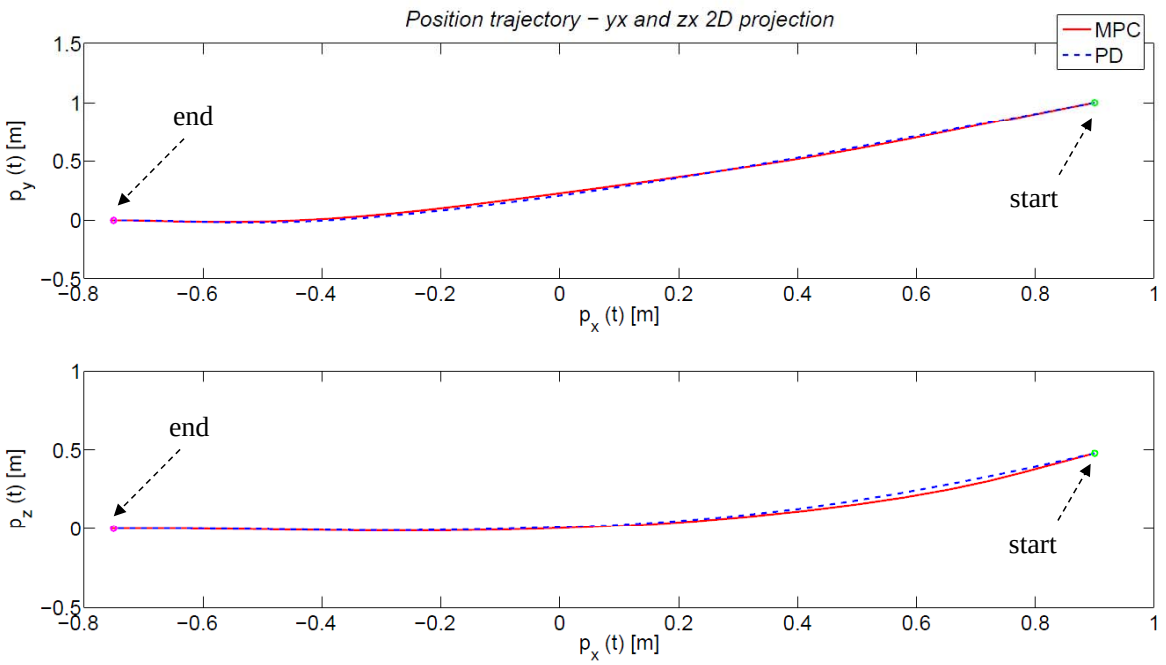


Figure 7.10: 3D Position Trajectory - Matlab Sim. MPC vs Matlab Sim. PD. Initial and final positions are marked with a green circle and a magenta circle respectively.

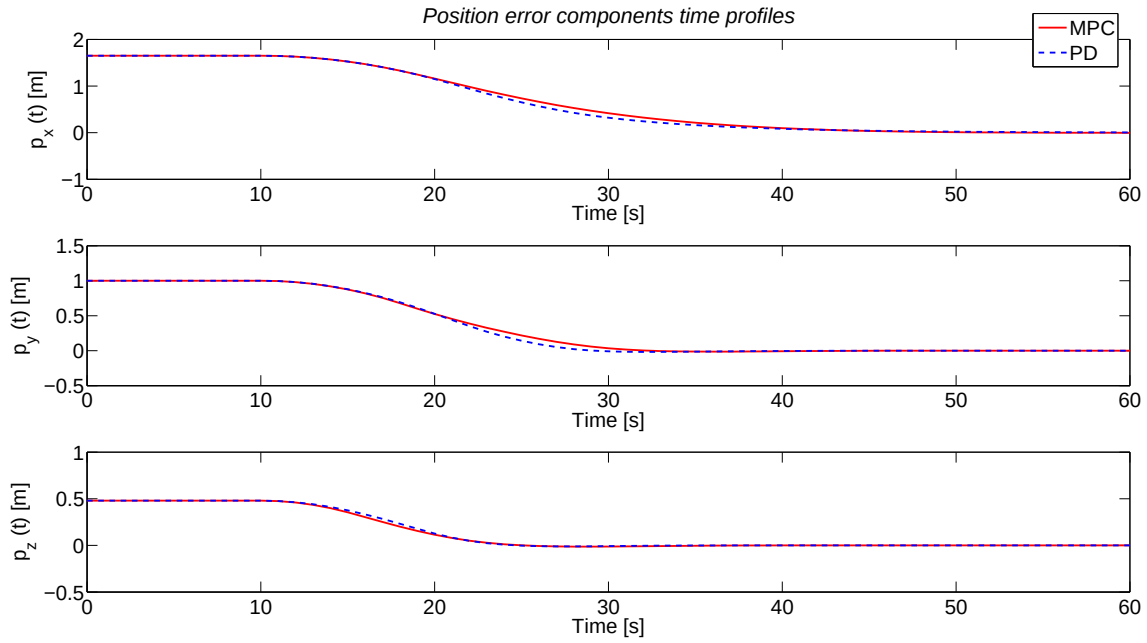


Figure 7.11: Position components error vs. time - Matlab Sim. MPC vs Matlab Sim. PD.

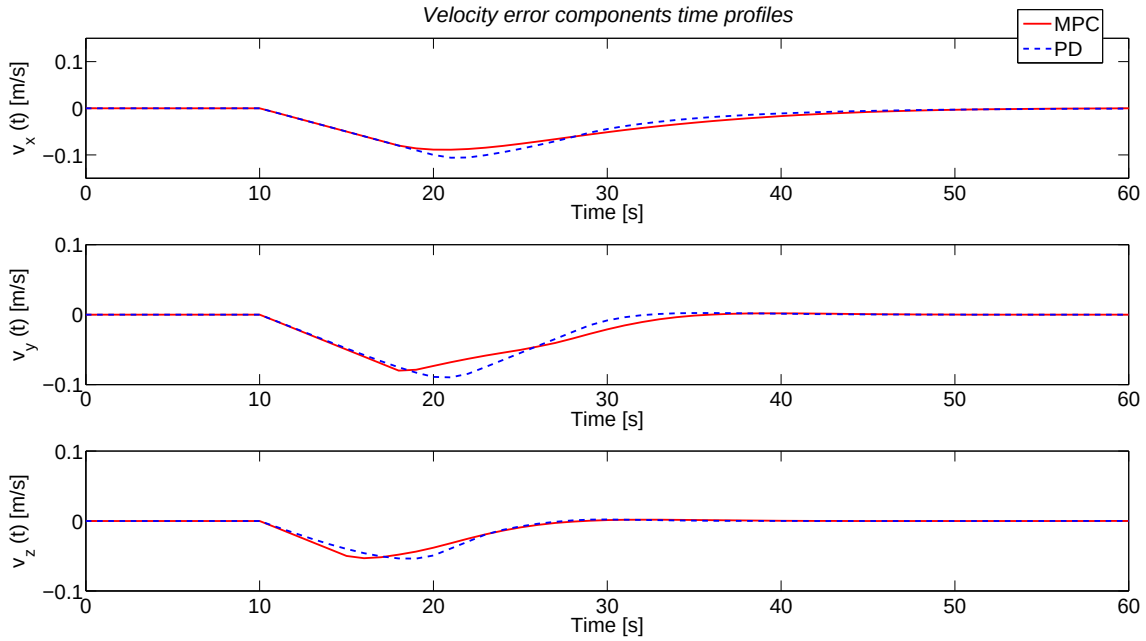


Figure 7.12: Velocity components error vs. time - Matlab Sim. MPC vs Matlab Sim. PD.

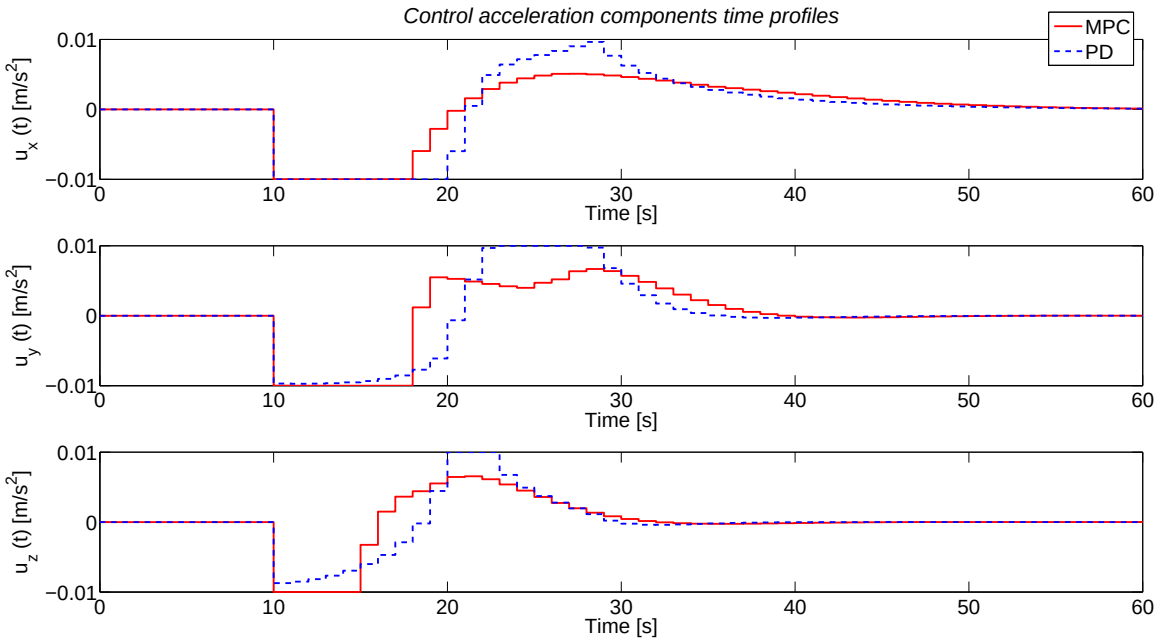


Figure 7.13: Control acceleration components vs. time - Matlab Sim. MPC vs Matlab Sim. PD.

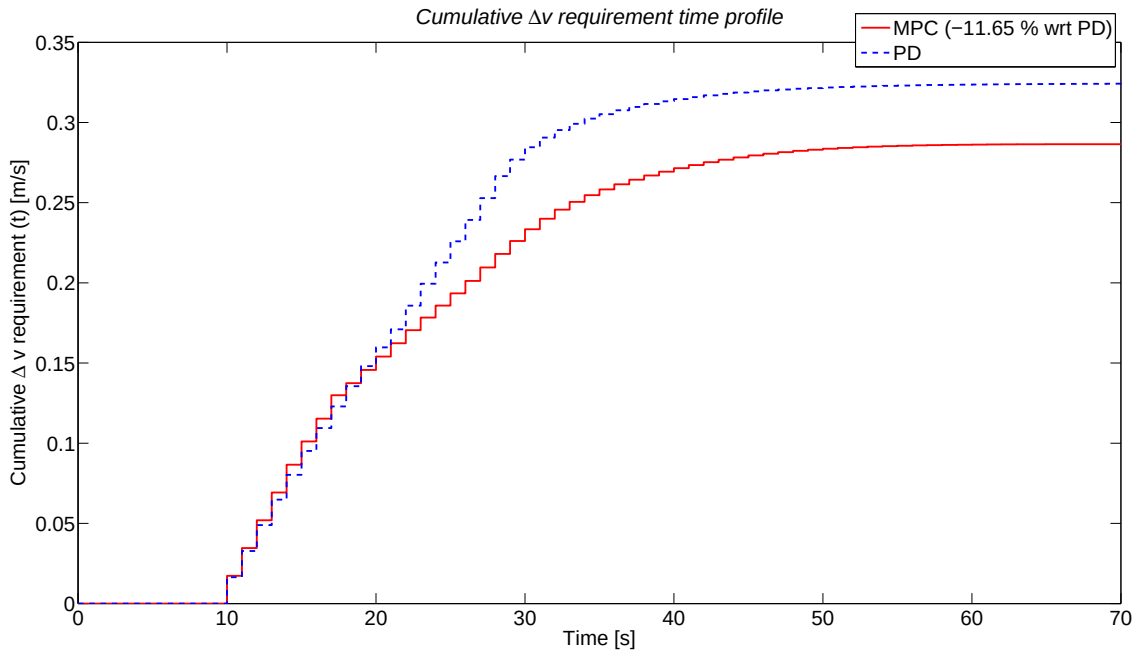


Figure 7.14: Cumulative Δv requirement vs. time - Matlab Sim. MPC vs Matlab Sim. PD.

7.4.2 SPHERES Simulator MPC vs SPHERES Simulator PD

The initial dynamic state x_i and the final or target dynamic state x_f for this simulation are as follows:

$$x_i = [0.9\,m,\ 1\,m,\ 0.48\,m,\ 0\,m/s,\ 0\,m/s,\ 0\,m/s]^T \tag{7.5}$$

$$x_f = [\,-0.75\,m,\ 0\,m,\ 0\,m,\ 0\,m/s,\ 0\,m/s,\ 0\,m/s]^T\,m \tag{7.6}$$

PD and MPC parameters are tuned in order to obtain comparable settling time for the position error components profiles. Both MPC and PD parameters are listed in Table 7.2. Other parameters used for this simulation are:

- Control step $\Delta t_{ctrl} = 1\,s$;
- Control acceleration constraints in all directions: $u_{min} = -10^{-2} m/s^2$, $u_{max} = 10^{-2} m/s^2$;
- No forward computation in MPC (see Subsection 7.3).

Results for this preliminary test are shown in Figures 7.15 - 7.17. As you can see from these results, the MPC total Δv requirement is about 13% less than the PD one, with similar settling times for all the position error components of both controllers.

Note also that SPHERES simulation results were obtained with same initial and final system states and very similar PD gains and MPC weights as Matlab simulator results. A delayed behavior both in position and velocity between Matlab and SPHERES simulators results for both PD and MPC is observed (as expected).

We can conclude that also in a more representative simulation environment, i.e the SPHERES simulator, a MPC-based controller has higher performances than a classical PD controller.

Table 7.2: MPC and PD parameters - MPC vs PD in the Matlab simulator.

PD		MPC	
${}^{G}k_P$	$10^{-2}[5,\ 8.5,\ 19.1]$	${}^{G}w_y$	$10^{-5}[1,\ 15,\ 15]^T$
${}^{G}k_D$	$10^{-1}[5.7,\ 4.65,\ 6.6]$	${}^{G}w_u$	$10^{-1}[1,\ 1,\ 1]^T$
		m	6
		p	$8\,m = 48$

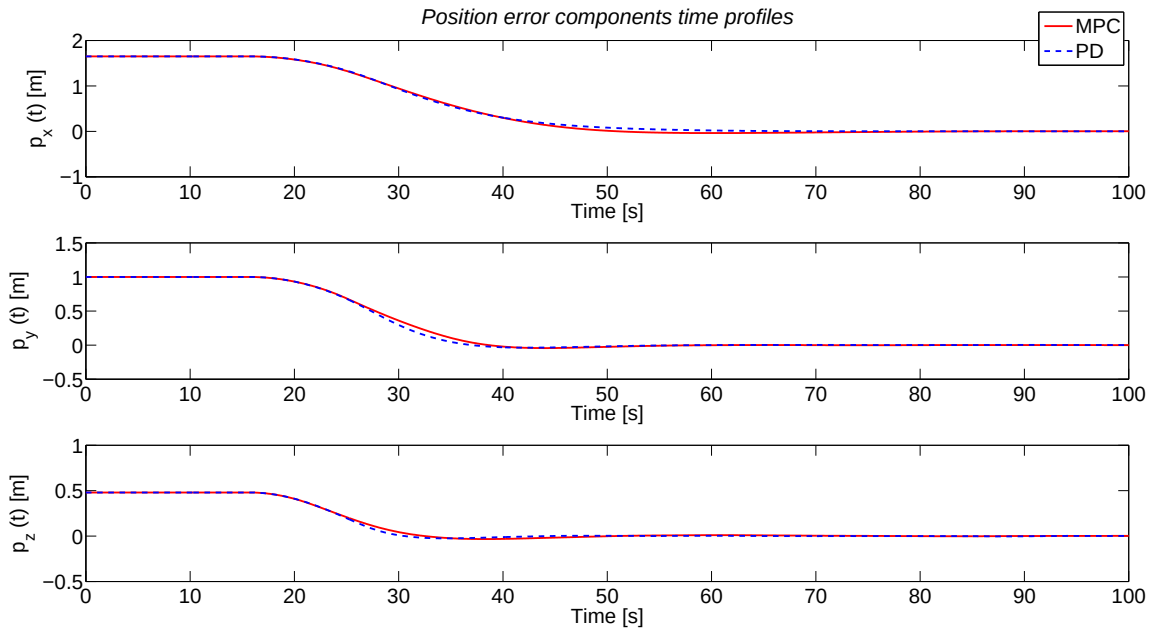


Figure 7.15: Position components error vs. time - SPHERES Sim. MPC vs SPHERES Sim. PD.

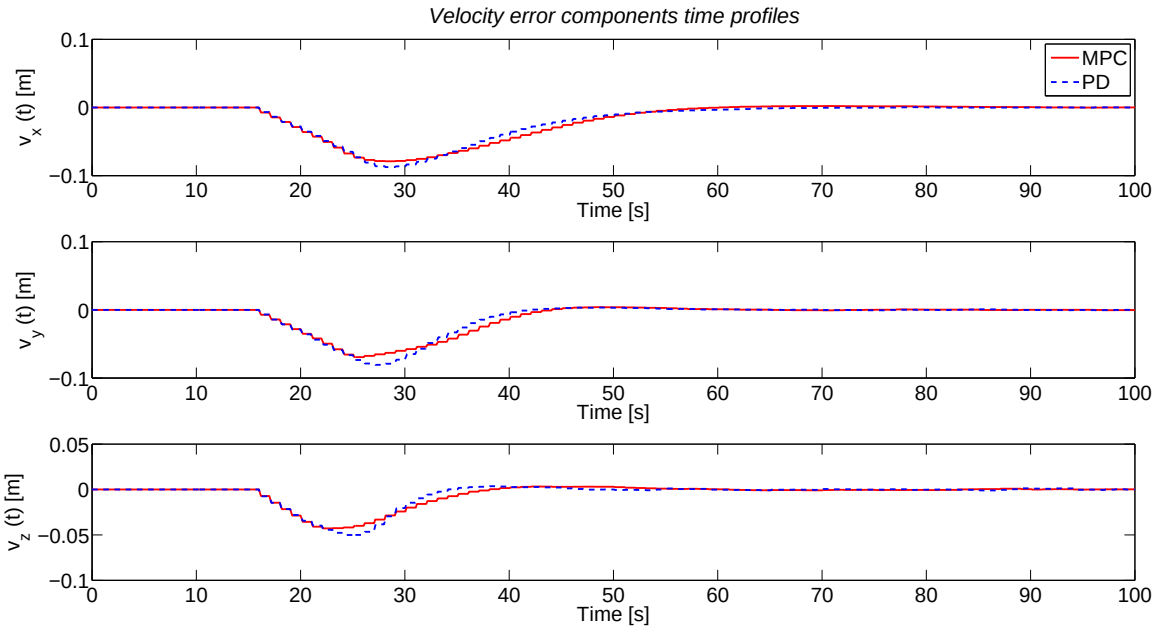


Figure 7.16: Velocity components error vs. time - SPHERES Sim. MPC vs SPHERES Sim. PD.

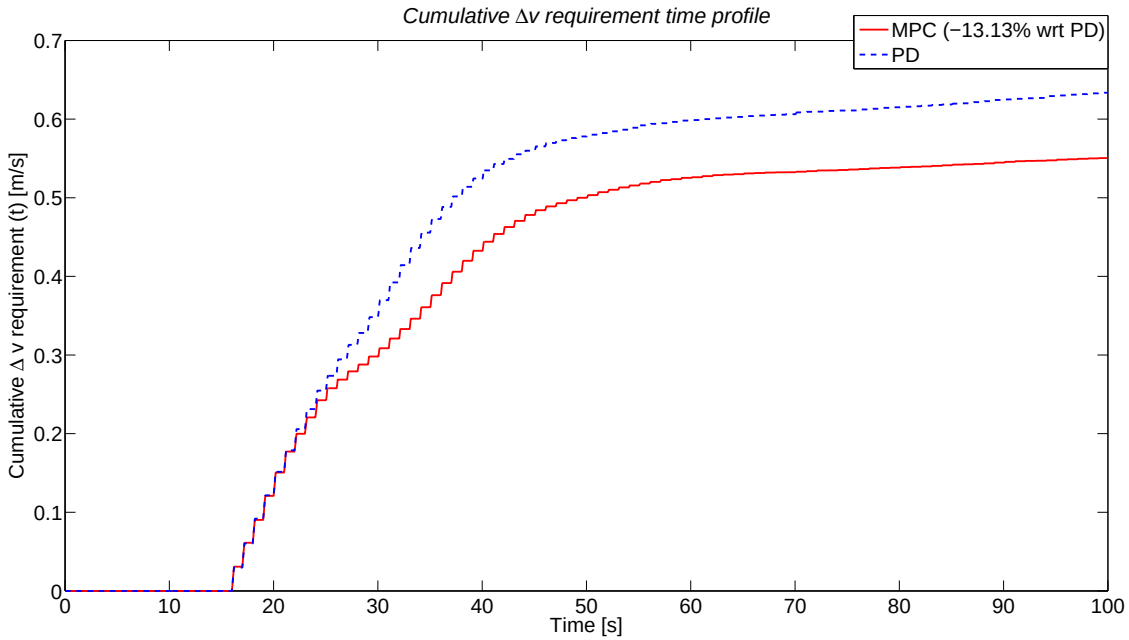


Figure 7.17: Cumulative Δv requirement vs. time - SPHERES Sim. MPC vs SPHERES Sim. PD.

7.5 SPHERES Test Results

7.5.1 SPHERES Test Plan

The SPHERES test plan is presented in Table 7.3. As you can see, Test 1 is used for both debugging C and Matlab codes and evaluating the MPC time delay, i.e. MPC computing time and data transmission between SPHERES and Matlab (see Figure 7.8 also). The performance of MPC and PD controllers in executing the MOSR rendezvous and capture maneuver are evaluated running Test 2 and Test 3, respectively.

Initial position and velocity, final position and other parameters for Test 2 and Test 3 are listed in Table 7.4.

Table 7.3: SPHERES Test Plan.

TEST NUMBER	MANUVER NUMBER	RUN TIME [S]	DESCRIPTION
1	1	70	MPC time delay evaluation. Position and velocity vectors are taken from the global metrology system
2			Rendez-vous and capture maneuver using an MPC controller. Position and velocity vectors are taken from the global metrology system.
	1	40	state estimator convergence
	2	80	initial position acquisition
	3	1	initial state acquisition
	4	100	rendez-vous and capture maneuver
	5	30	station keeping
3			Rendez-vous and capture maneuver using an PD controller. Position and velocity vectors are taken from the global metrology system.
	1	40	state estimator convergence
	2	80	initial position acquisition
	3	1	initial state acquisition
	4	100	rendez-vous and capture maneuver
	5	30	station keeping

Table 7.4: Test 2 and Test 3 parameters.

Test 2 parameters	
${}^{FF}p_i$	$[-0.4737,\ -0.1611]^T\, m$
${}^{FF}v_i$	$[-3.15,\ 0.13]^T\, 10^{-3}\, m/s$
${}^{FF}p_f$	$[0.7,\ 0.4]^T\, m$
Δt_{ctrl}	$1\, s$
${}^{FF}w_y$	$[\,3 \cdot 10^{-5},\ 1.35 \cdot 10^{-1}]^T$
${}^{FF}w_u$	$[0.1,\ 0.1]^T$
m	6
p	$8\, H_u = 48$
u_{max}	$4 \cdot 10^{-3}\, m/s^2$
u_{min}	$-4 \cdot 10^{-3}\, m/s^2$

Test 3 parameters	
${}^{FF}p_i$	$[-0.5035,\ -0.1213]^T\, m$
${}^{FF}v_i$	$[-0.0008,\ 0.0011]^T\, 10^{-3}\, m/s$
${}^{FF}p_f$	$[0.7,\ 0.4]^T\, m$
Δt_{ctrl}	$1\, s$
${}^{FF}k_p$	$[0.08,\ 0.22]^T$
${}^{FF}k_d$	$[1,\ 1.674]^T$

7.5.2 Test 1 - MPC computing time evaluation

The MPC computing time is between 300 ms ad 600 ms with active constraints on both the output variable and the control variable.

7.5.3 Test 2 - MPC rendezvous maneuver

Test results are shown in the following figures:

- Figure 7.18: computing time to solve the MPC problem in Matlab; notice that the MPC problem for this test is solved in less than $60\, ms$ with less than 20 iterations;
- Figure 7.19: 2D position trajectory, with initial position in blue and final position in green;
- Figure 7.20: position profiles;
- Figure 7.21: velocity profiles;
- Figure 7.22: control acceleration profiles;
- Figure 7.23: cumulative Δv requirement profile.

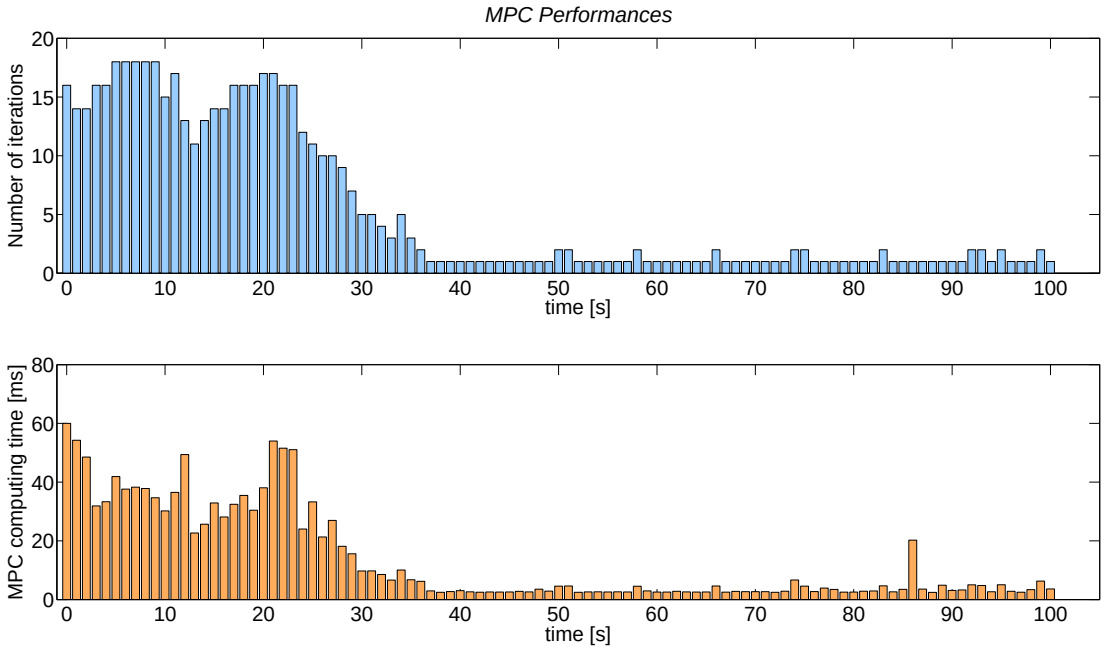


Figure 7.18: MPC computing time in Matlab - Test 2.

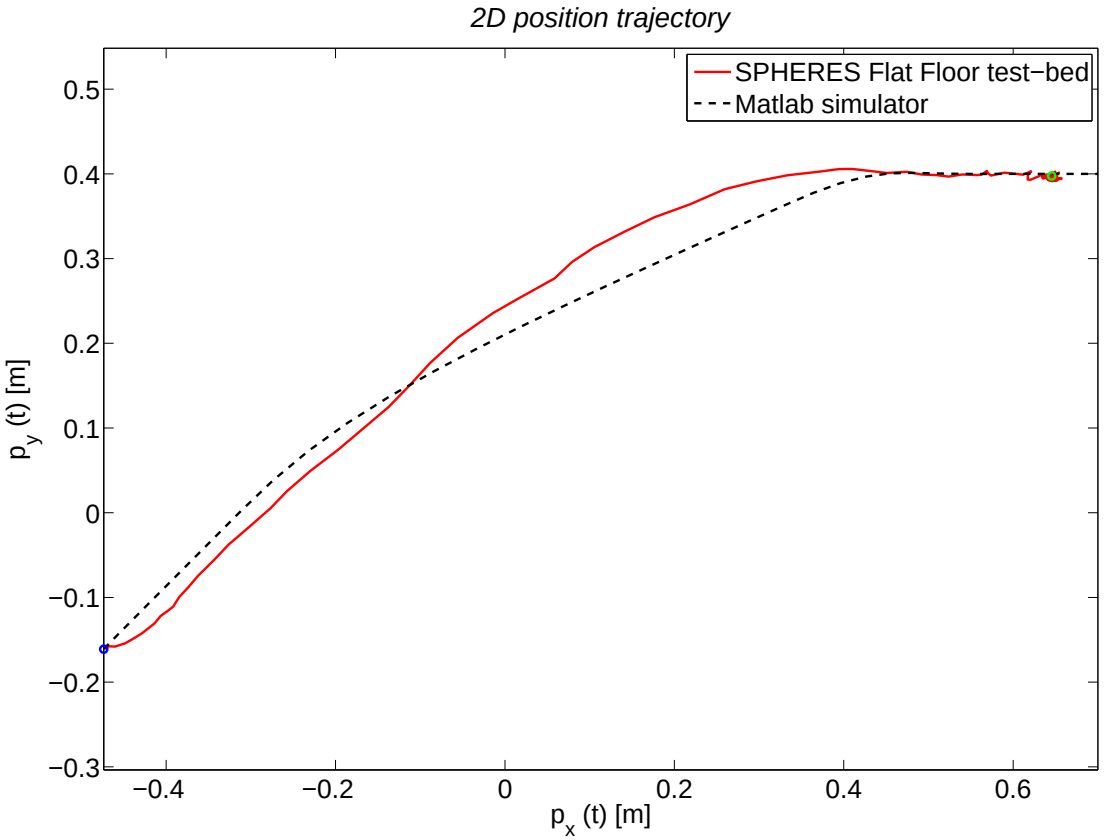


Figure 7.19: MPC 2D position trajectory - Test 2.

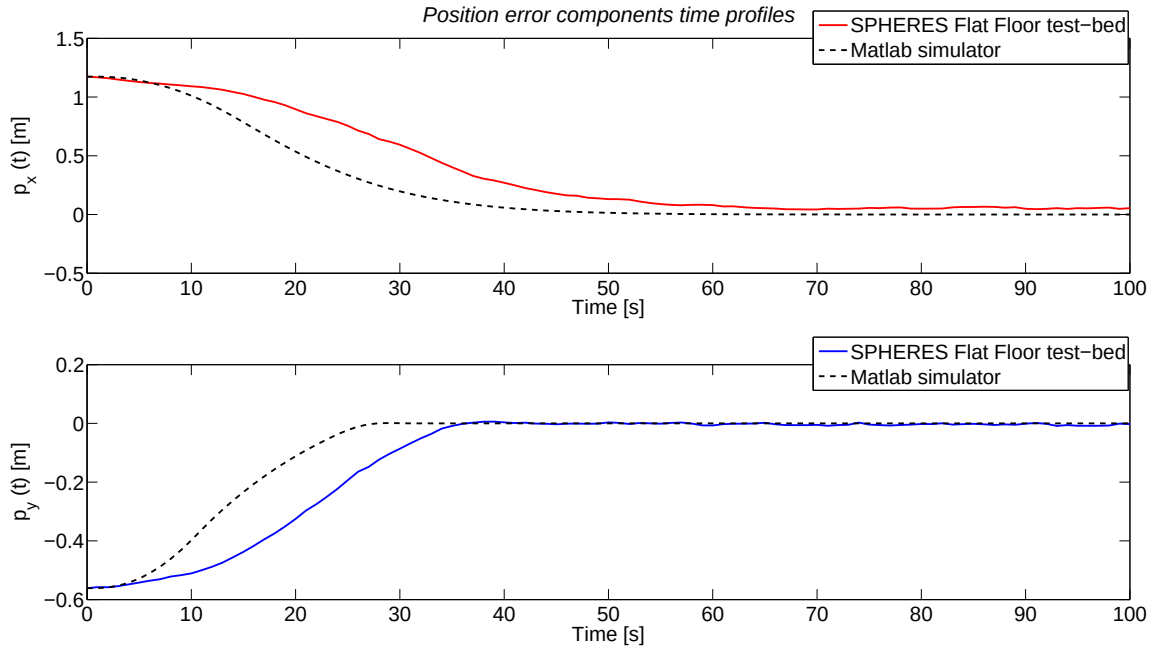


Figure 7.20: MPC position components error vs. time - Test 2.

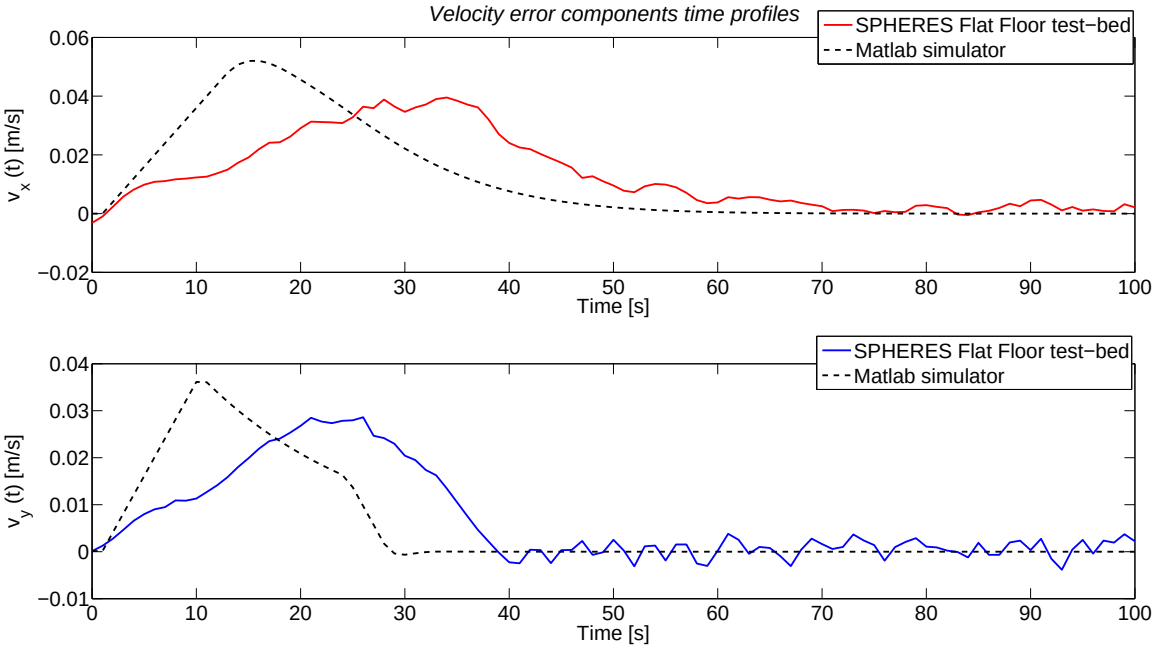


Figure 7.21: MPC velocity components error vs. time - Test 2.

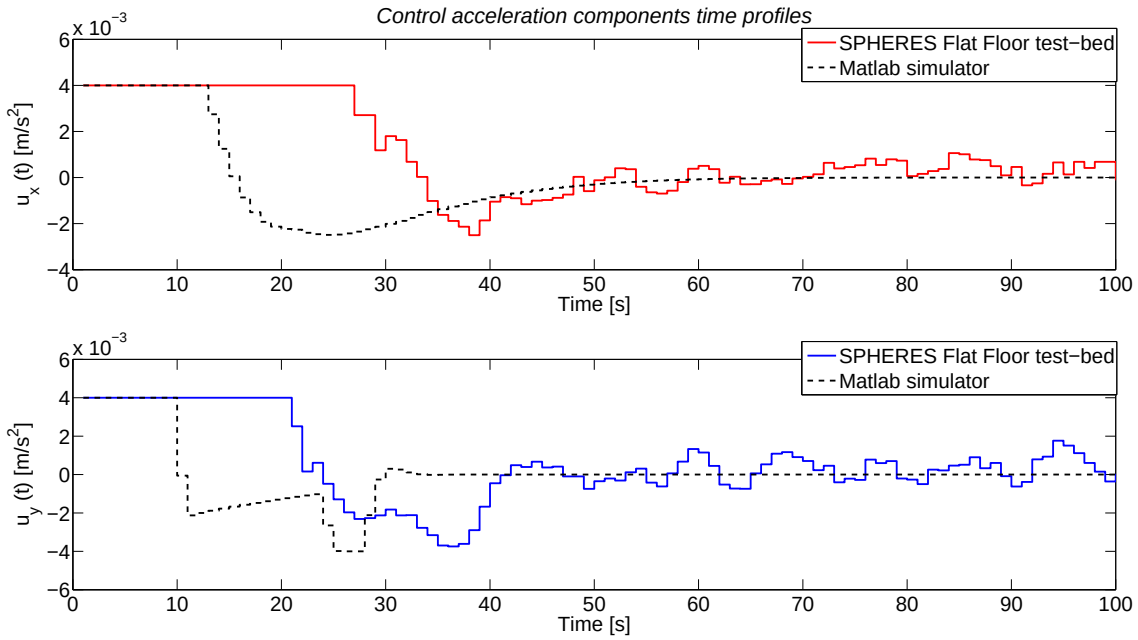


Figure 7.22: MPC control acceleration components vs. time - Test 2.

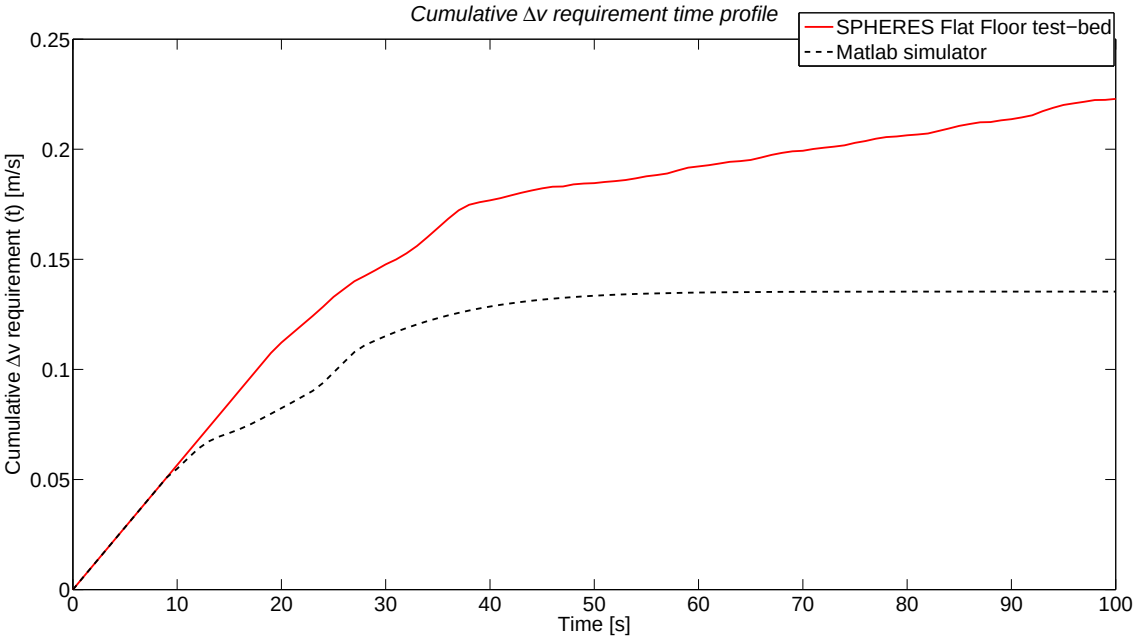


Figure 7.23: MPC Cumulative Δv requirement vs. time - Test 2.

7.5.4 Test 3 - PD rendezvous maneuver

Results for Test 3 are shown in Figures 7.24 - 7.28.

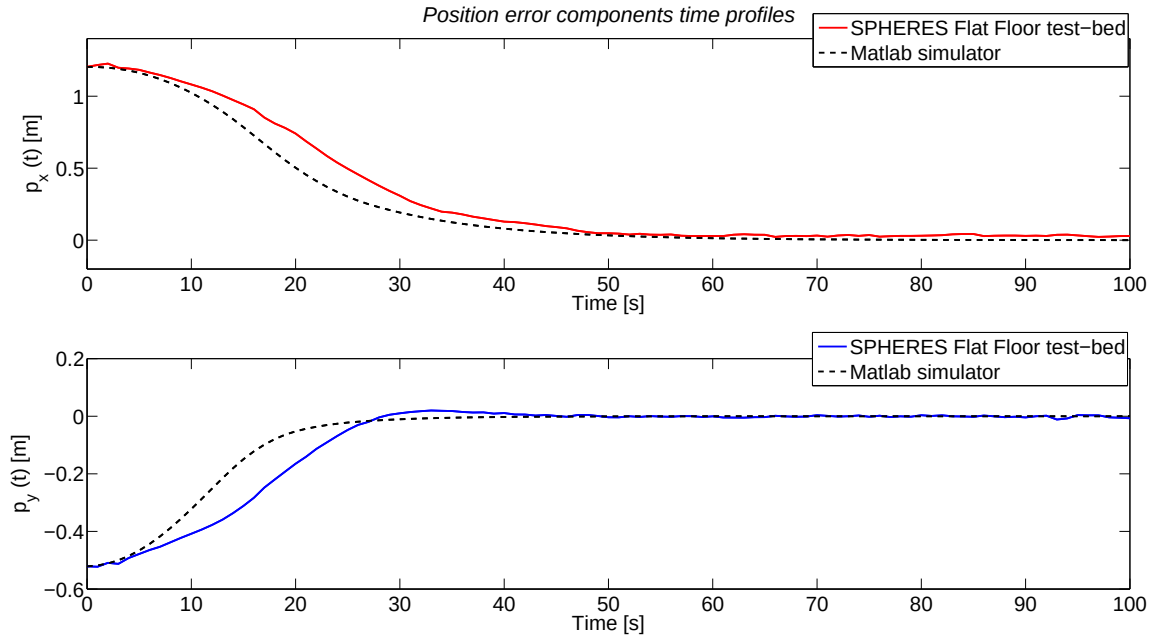


Figure 7.24: PD position components error vs. time - Test 3.

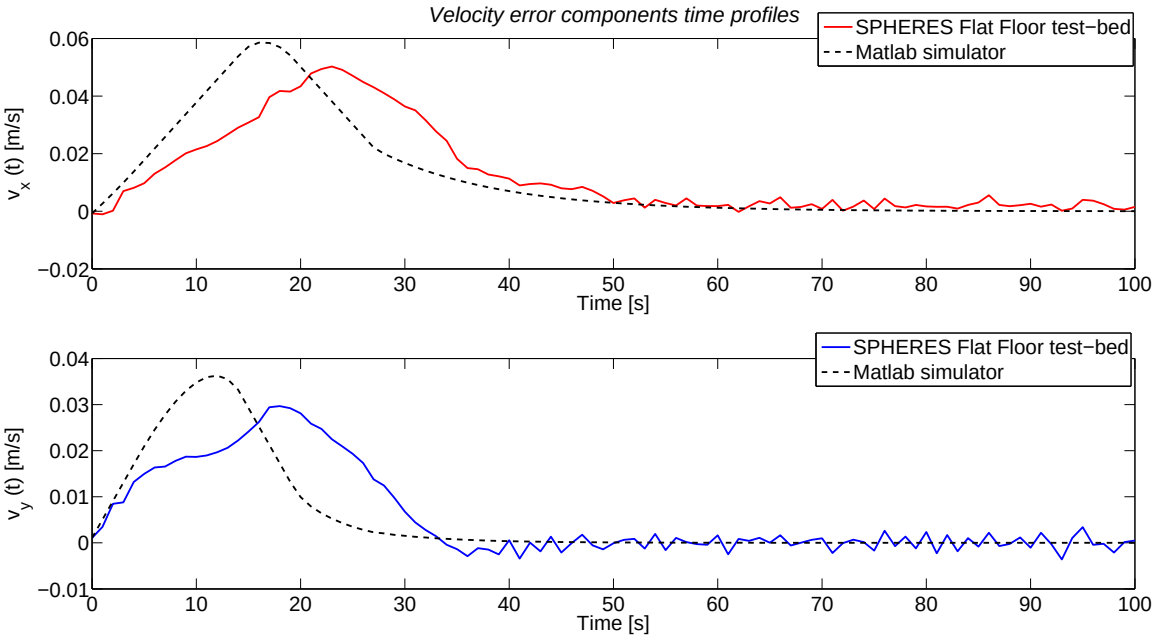


Figure 7.25: PD velocity components error vs. time - Test 3.

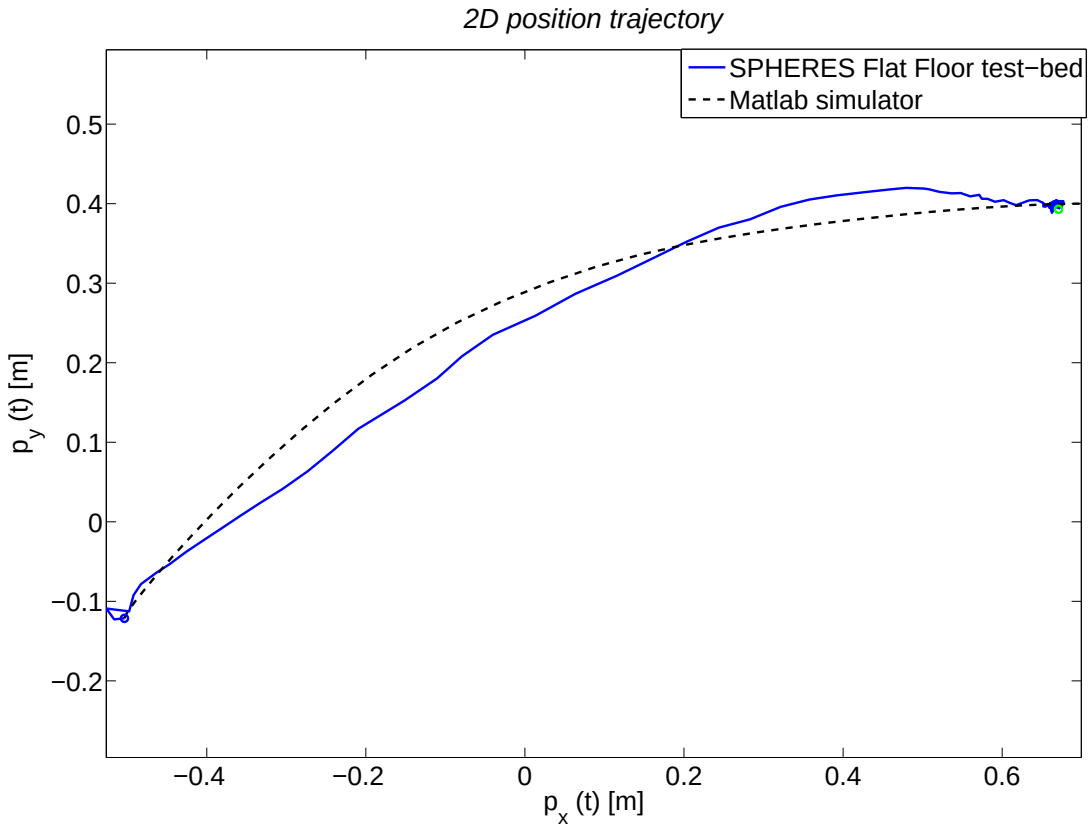


Figure 7.26: PD 2D position trajectory vs. time - Test 3.

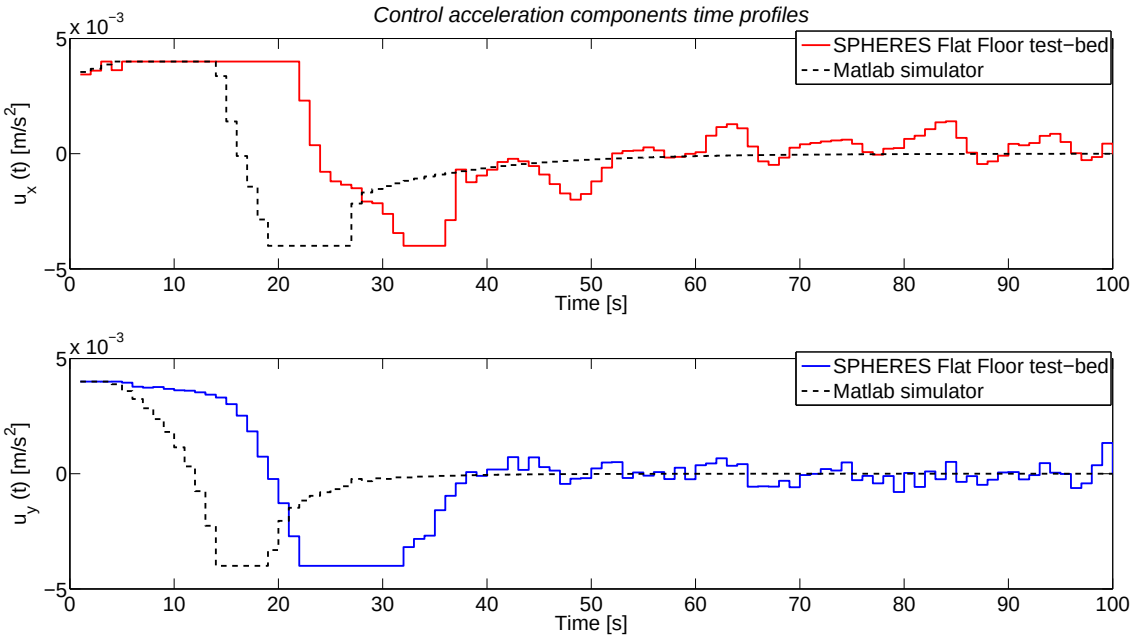


Figure 7.27: PD control acceleration components vs. time - Test 3.

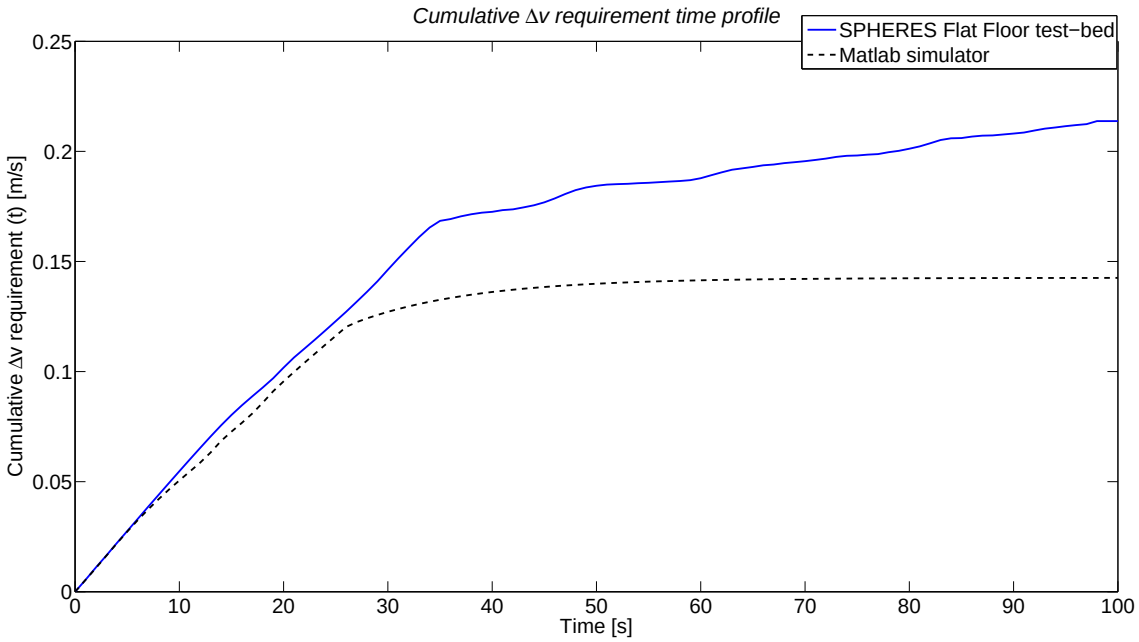


Figure 7.28: PD cumulative Δv requirement vs. time - Test 3.

7.5.5 Explanation of results

Both MPC and PD controllers are effective, since the final target position is reached with a small final error. This steady-state error is mainly due to the friction between the air carriage and the flat floor and the control force minimum pulse that the propulsion subsystem on board the SPHERES satellite can actually apply. An Integral component in the control policy should fix that steady-state error.

A noisy and delayed behavior between Matlab Simulator and SPHERES results for both PD and MPC is observed. This may be due to the following causes:

- friction between the air carriage and the flat floor that impedes the SPHERES motion;
- noise of the SPHERES estimated state;
- because of a minimum pulse width, the control acceleration that is actually performed is different from the computed one;
- two different control actuation schemes were used, PWC in Matlab and PWM in SPHERES;
- noise of the actuator system;
- time delay in actuating the control actions.

Chapter 8

FORMATION FLIGHT HARDWARE SIMULATOR

8.1 Introduction

The final aim of this project is to design, realize and validate a representative facility, called SFF Hardware Simulator, to carry out research activities in the fields of Satellite Formation Flying (SFF) and Rendez-vous and Docking. This separated vehicles testbed will be a representative dynamic environment on the ground for the development and verification of coupled position and attitude relative Guidance Navigation and Control algorithms. The main requirements that led the facility design are: fatefully representation of a satellite formation dynamics, easy configuration changes, low cost, reliability and safety.

The SFF testbed is made up by two or more Spacecraft Simulators representing the units of a satellite formation and a Control Station (laptop) (see Figure 8.1). Each Simulator has an Attitude Module (AM) with three rotational degrees of freedom and a Translational Module (TM) with two position degrees of freedom that translate on a glass plane using a low friction air cushion system. A laptop is part of the facility too, and its functions are to boot-load the Simulators, to transmit commands to the Simulators and to receive and store telemetry data.

At the moment of writing this thesis, only the AM is almost completed, while the TM is planned to be developed in the future. The main features of the Attitude Module are described in Section 8.3 after a brief review of spacecraft simulators.

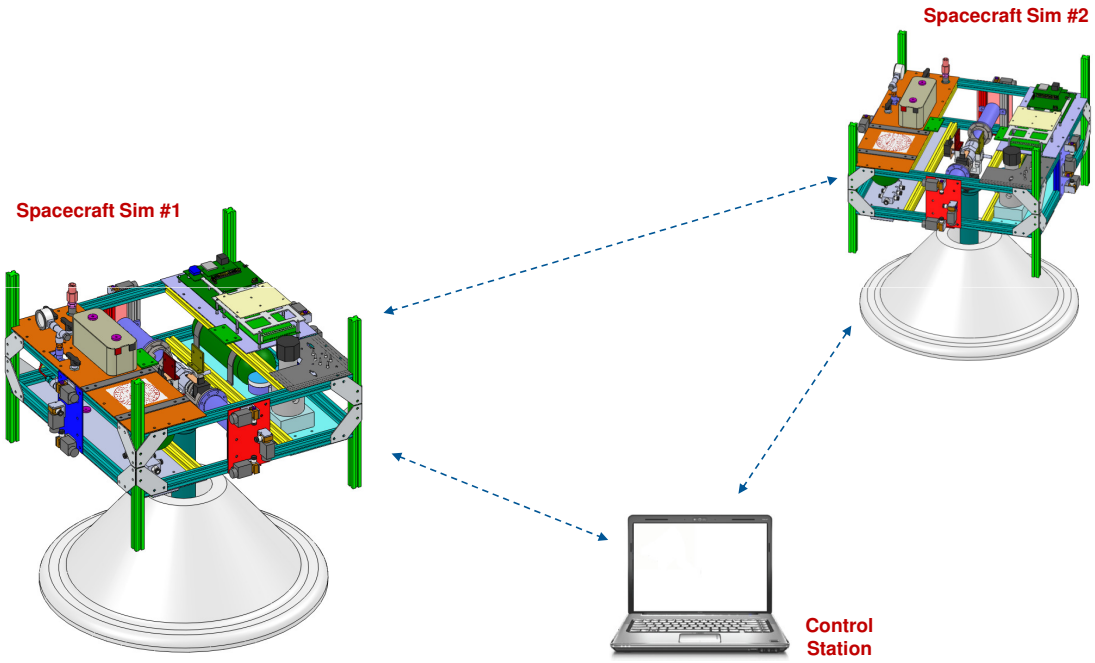


Figure 8.1: Formation Flight Hardware Simulator testbed overview.

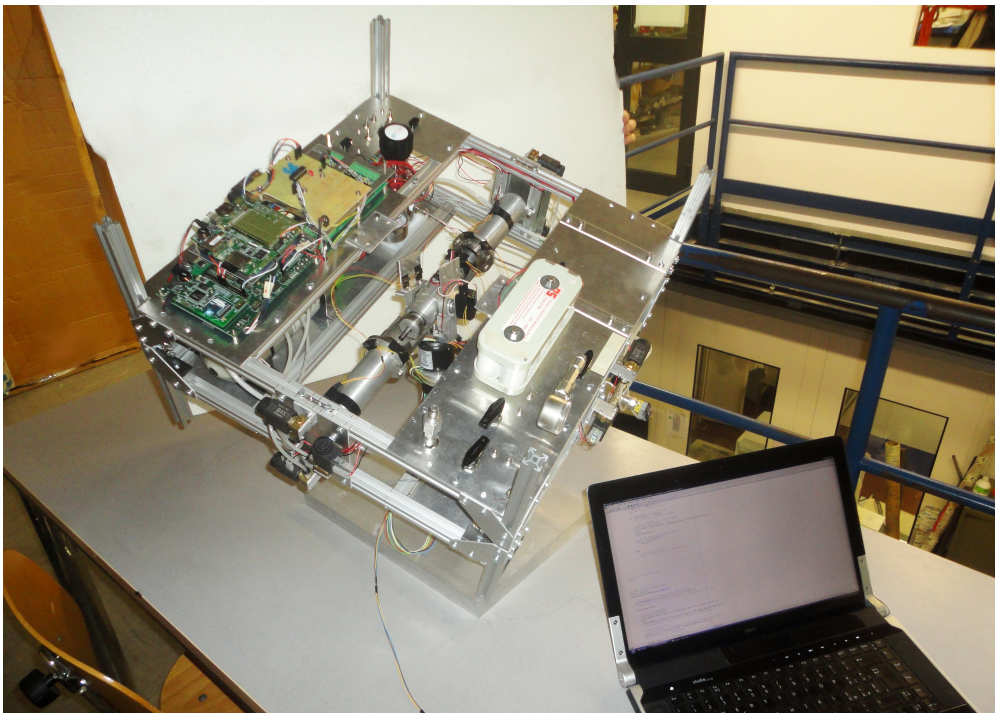


Figure 8.2: The Attitude Module (AM) at the current sate of the system.

8.2 Review of Spacecraft Simulators

Since the beginning of space exploration, ground-based testbeds have been used for both hardware and software development and verification. There are many solutions to the problem of simulating the space dynamics on the ground and some particular techniques are more applicable and representative in one situation than another. A classification of spacecraft testbeds is shown in Table 8.1, and some advantages and disadvantages of the proposed solutions are listed in Table 8.2.

Table 8.1: Classification of spacecraft testbeds.

<table>
<tr><td rowspan="9">Ground-based Testbeds</td><td rowspan="6">Spacecraft Simulators</td><td rowspan="4">Rotational Systems</td><td rowspan="3">Air-bearing</td><td>Tabletop</td></tr>
<tr><td>Umbrella</td></tr>
<tr><td>Dumbbell</td></tr>
<tr><td>Ball-and-socketd joints</td></tr>
<tr><td>Multiple gimbals</td></tr>
<tr><td colspan="2">Planar Systems</td></tr>
<tr><td colspan="2">Combined Systems</td></tr>
<tr><td rowspan="2">Water Tanks</td></tr>
<tr><td>Drop Towers</td></tr>
<tr><td colspan="4">Parabolic Flights</td></tr>
<tr><td colspan="4">On-orbit Testbeds</td></tr>
</table>

Table 8.2: Advantages and disadvantages of some proposed spacecraft testbeds.

TESTBEDS	PROS	CONS
On-orbit Testbeds	fully representative of operative conditions	very expensive
Parabolic Flights	excellent microgravity siulation quite easy to interact with the payload	expensive microgravity last about 20 s not possible to test payload in vacuum or to study thermal cycles effects
Water Tanks	good microgravity simulation long lasting microgravity conditions	hight viscous forces / torques need for a water resistant payload
Spacecraft Simulators	good microgravity simulation very easy to interact with the payload	disturb forces / torques need to be mitigated partial motion of freedom

In [83] Schwartz et al. provide an historical review of air-bearing spacecraft simulators. In that paper, spacecraft simulators are grouped into three main categories: planar systems, rotational systems and combination systems. Both in planar and rotational air bearings, pressurized air passes through small holes creating a thin film or cushion between coupled moving sections. That air film supports the weight of moving sections and, acting as a lubricant, reduces the friction between the two sections of the bearing, allowing to obtain virtually torque-free rotational and force-free translational motions.

Planar systems have one rotational and two translational Degrees Of Freedom (DOF) and are generally used for validating both formation flight and rendezvous and docking scenarios control strategies. Examples of planar air bearing systems are listed in the following. The Stanford University's Aerospace Robotics Laboratory have several air-bearing test facilities used both to study formation flying issues and for testing on-orbit construction, servicing, assembly and repair maneuvers with the use of a robotic arm. Also the University of Victoria has a planar air bearing system hosting a single robotic arm that is used to investigate how optimal joint trajectories can be found with the aim of minimizing vibration excitation within the arm elements during a designated maneuver.

The MIT Space Systems Laboratory (SSL) developed the SPHERES (Synchronized Position Hold, Engage, and Reorient Experimental Satellites) project that can be used in two main testbed configurations. The Flat Floor and the Glass Table test facilities at the SSL are planar systems with three degrees of freedom, while the six degrees of freedom system flies on board the International Space Station in a fully representative microgravity environment.

Rotational air bearing systems provide attitude freedom about three axes. A completely free 3 DOF attitude motion is difficult and expensive to achieve. Full freedom of spin in the yaw axis can be obtained adopting a Tabletop- and Umbrella-style configuration (panels A and B in Figure 8.3), but pitch and roll motion are typically constrained to angles of less than 90 deg. Tabletop and Umbrella systems have been developed by many agencies and research centers including the NASA Goddard Space Flight Center, the NASA Jet Propulsion Laboratory, the Utah State University and the School of Aerospace Engineering at the Georgia Tech.

Another possible configuration that greatly reduces structural interference to the rotational motion is the Dumbbell configuration (panel C in Figure 8.3). Using two opposing arms, the mounting area is placed away from the center of rotation, thereby achieving unconstrained motion in both the roll and yaw axes. Examples of dumbbell configuration systems are the Virginia Tech's Whorl-II (panel A in Figure 8.4), the University of Michigan's Triaxial Air Bearing Testbed (panel B in Figure 8.4) and the Virginia Tech's Distributed Spacecraft Attitude Control System Simulator.

In Combination systems both planar and rotational air bearings are used to obtain up to

six completely unconstrained degrees of freedom simulators. Combination spacecraft simulators have been developed by the NASA Marshall Space Flight Center's Flight Robotics Laboratory and the Lawrence Livermore National Laboratory with the ability to perform precision maneuvers autonomously, including rendezvous, inspection, proximity operations, formation flying, docking, and servicing. Figure 8.5 shows the Formation Control Testbed developed by JPL-CalTech.

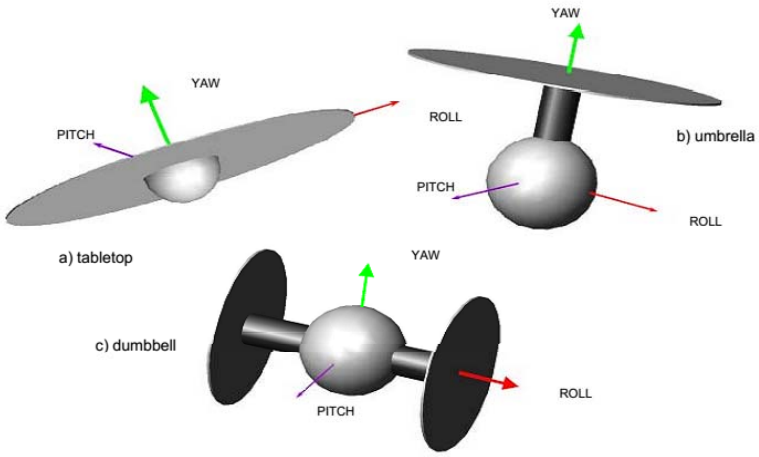


Figure 8.3: Tabletop configuration (panel A), Umbrella configuration (panel B) and Dumbbell configuration (panel C). Tabletop and Umbrella systems provide full freedom about the yaw axis, while the Dumbbell systems provide full freedom about yaw and roll axes.

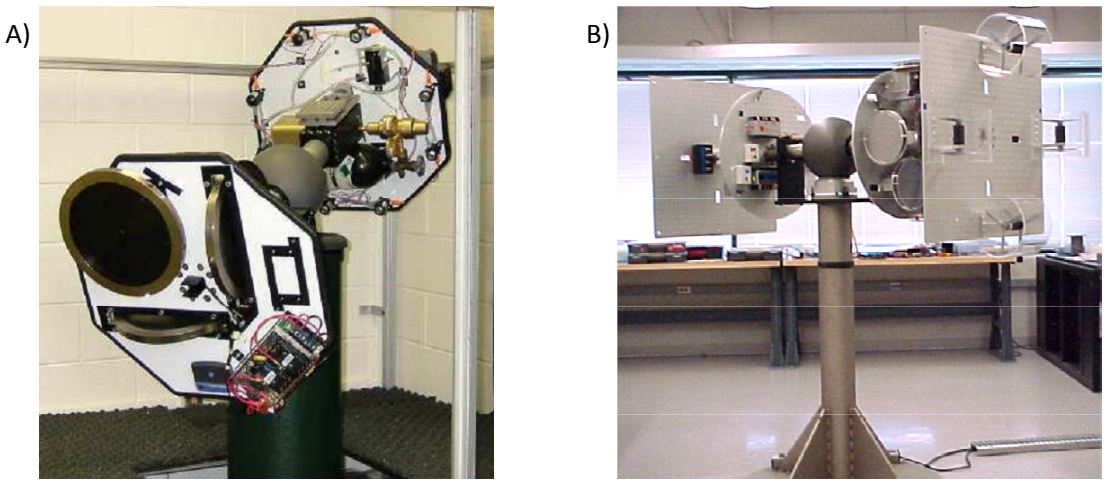


Figure 8.4: Virginia Tech's Whorl-II (panel A) and University of Michigan's Triaxial Air Bearing Testbed (panel B).

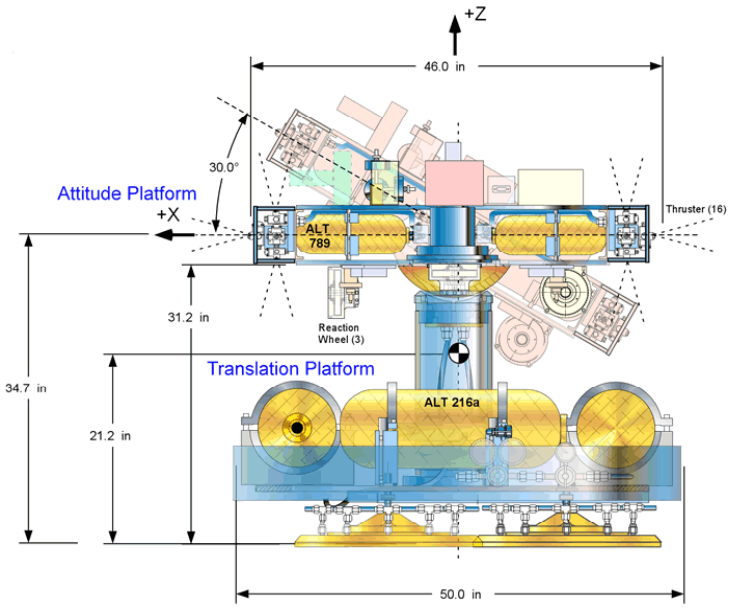


Figure 8.5: JPL - CalTech Formation Control Testbed.

8.3 Attitude Module On Board Subsystems

The Attitude Module (AM) is equipped with 5 main subsystems that allow it to execute an autonomous attitude maneuver:

1. the Structural Subsystem;
2. the Propulsion Subsystem;
3. the Attitude Determination and Control Subsystem (ADCS);
4. the Electric and Power Subsystem;
5. the Communication and Data Handling Subsystem.

These on board subsystems are briefly described in Subsections 8.3.1 - 8.3.4.

The brain of the AM is a micro-controller that manages information coming from sensors and computes the control torque that the actuators (thrusters) have to perform in order to autonomously execute a given attitude maneuver. Furthermore, the AM can interact with the Control Station (laptop) that transmits commands and receives house keeping and telemetry data about the attitude maneuver under execution. Both the on board and the control station software architecture are presented in short in Subsection 8.3.6.

8.3.1 Structural Subsystem

The AM's structure is made up by the following main components (see Figure 8.6):

- a Three-Joints System on which the AM is mounted and that allows the AM to rotate around any axis;
- an aluminium framework made of 22 Bosh Rexroth 20 mm x 20 mm aluminium profile beams with 4 lateral T-grooves, joined together with stainless steel quick connectors;
- 4 upper and lower main rectangular plates to support the other components of the system;
- 16 reinforcing lateral corner plates;
- 4 reinforcing lateral rectangular plates;
- 12 right-angle brackets for the thrusters support.

All plates are connected to the frame with a fastener / T-nut system.

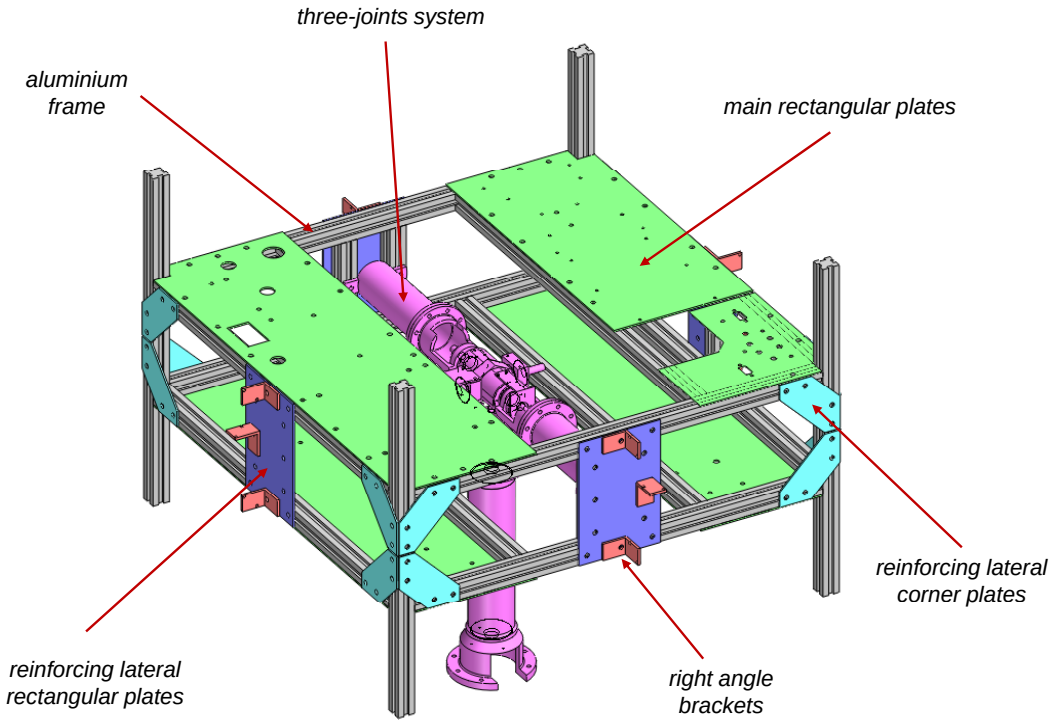


Figure 8.6: Main components of the Attitude Module's Structure.

More precisely, the Three-Joints System (see Figure 8.7) is made of three consecutive rotational cylindrical joints[1] whose axes are orthogonal and meeting in a rotational center. This systems of joints is designed so that the first joint allows a rotation about the yaw axis coinciding with the vertical axis, the second one allows a rotation with respect to the pitch axis and the third one allows a rotation about the roll axis. Thank to this joint system, the AM can freely rotate around the yaw direction, while both roll and pitch angles are limited in the range $[-40°, +40°]$.

In nominal conditions, i.e. with no friction and in a perfect balancing condition, the AM's center of mass coincides with the Three-Joints System rotational center allowing to have a platform whose attitude dynamics does not depend on torques due to gravity but only on control torques.

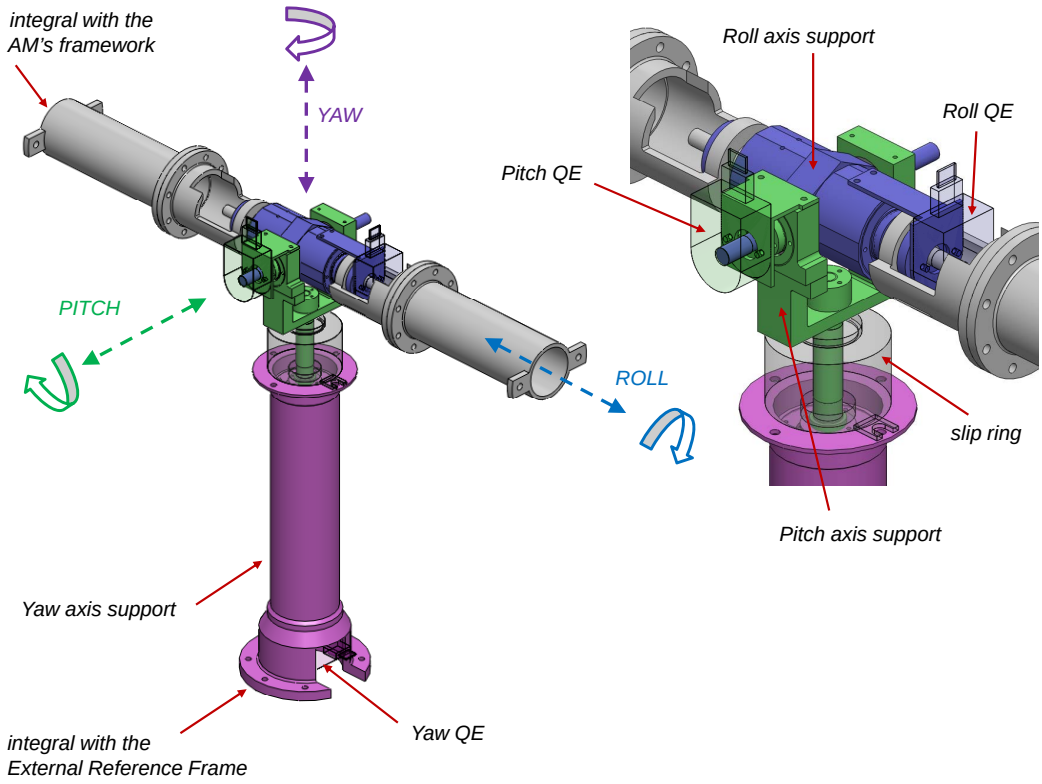


Figure 8.7: The Three-Joints System with roll axis in blue, pitch axis in green and yaw axis in violet.

8.3.2 Propulsion Subsystem

The architecture of the Propulsion Subsystem is shown in Figure 8.8. As you can see from this figure, the on board Propulsion Subsystem consists on a high-pressure side at 200 bar maximum and a low-pressure side at 10 bar. On the high-pressure side of the system, the main components

[1] Realized using Al7075 and Ti-8Mn components to achieve low mass and stiffness at the same time and preloaded, low friction, spherical SKF bearings.

include (see Figure 8.9 also):

- two Luxfer composite (aluminium, carbon fiber and glass fiber) tanks with a total capacity of 2 Lit. for the air storage at 200 bar maximum;
- a fill & vent system for the pneumatic circuit loading, consisting of a single-end shutoff (SESO) quick connector and a first ball valve plus a second ball valves for manual emptying;
- an Air Liquide pressure-reducing regulator that reduces the air pressure from the storage level, i.e. high pressure at 200 bar maximum (red circuit in Figure 8.8) to the nozzle operative level, i.e. low pressure at 10 bar (blue circuit in Figure 8.8);
- some Swagelok stainless steel pipe fittings to connect the high-pressure components.

The low-pressure side of the subsystem includes:

- two 6-ways manifolds that divide the air flow coming from the pressure regulator outlet supplying the 12 thrusters;
- 12 thrusters to actuate the control torques;
- some variable size plastic pipes to connect the elements of the low-pressure pneumatic circuit.

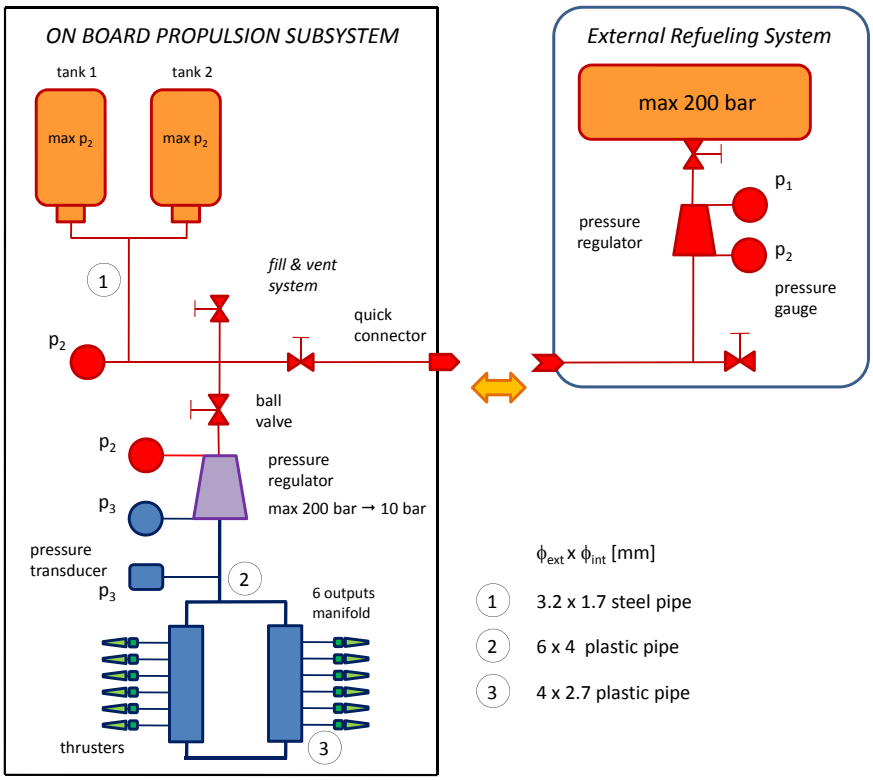


Figure 8.8: Architecture of the on board Propulsion Subsystem and the External Refueling System.

In more detail, each thruster is composed of an UNIVER U1 electro-valve and a converging nozzle. The electro-valve is a solenoid valve that opens when powered at 12-24 VDC, letting the air flow pass through the nozzle. The nozzle is made of a M5 screw with a central 0.75 mm diameter hole.

The 12 thrusters firing axes are placed in order that using four thrusters it is possible to actuate a positive or negative torque only around each of the x_B, y_B or z_B AM's Body axes (nominally). Combining the actuation of the 6 thruster couples is then possible to actuate a control torque around any axis in the AM's Body Reference Frame.

A PPE-P10A pressure transducer is used for monitoring the pressure level at the low pressure side of the pneumatic system.

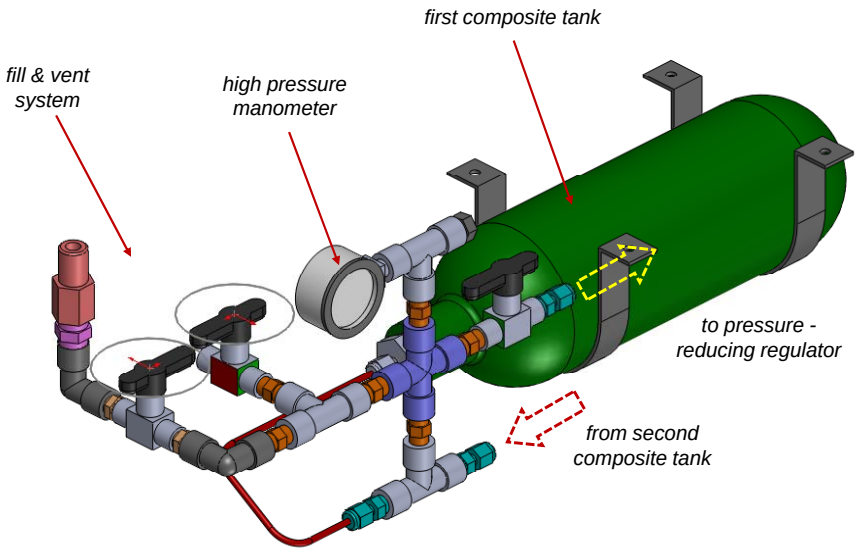


Figure 8.9: Part of the High Pressure Pneumatic Circuit.

8.3.3 ADCS

The ADCS consists on attitude sensors that provide an estimation of the AM's orientation with respect to a Local Vertical - Local Horizontal (LVLH) External Reference Frame and electronic boards that perform the control action computation and properly command the Propulsion Subsystem for their actuation.

Each simulator is equipped with two kind of attitude sensors:

- 3 Avago Technologies HEDM 5500- B13 rotational optical incremental Quadrature Encoders (QE), that are used to measure the rotation of the joints with a resolution of 0.09°, thus directly providing Roll, Pitch and Yaw angles;

- a Microstrain 3DM-GX1 Inertial Measurement Unit (IMU), that provides an estimation of the AM attitude and attitude rate with respect to the External Reference Frame.

Since the attitude accuracy achievable with QE's is by far better than the one achievable with the IMU, the QE angular measurements are used as a reference to evaluate drift and bias on the IMU measurements.

The Guidance, Navigation and Control (GNC) functions are performed by a Digi International Rabbit micro-controller. At the moment the AM is equipped with a Rabbit RIO Prototyping Board with a Rabbit RIO chip for the QE's reading and a Rabbit Core Module 4510W with a Rabbit 4000 29.49 MHz micro-controller (see Figure 8.10).

Custom electronic was developed to interface the avionics with both the propulsion subsystem (Thruster Command Board) and the laptop (RS-232 transceiver).

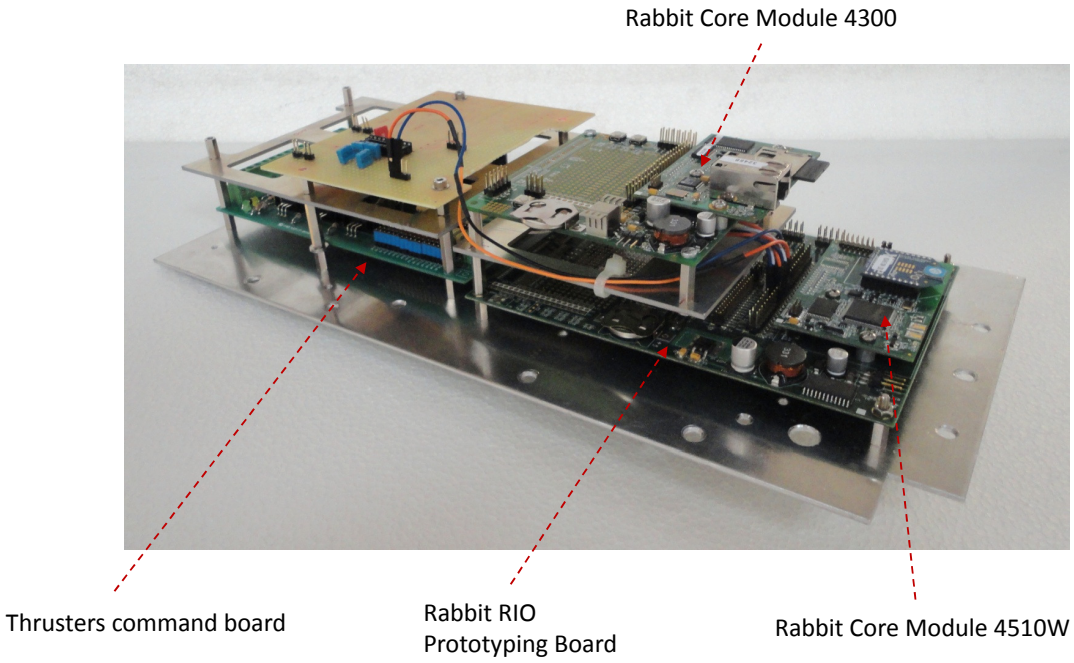


Figure 8.10: Electronic boards assembly.

8.3.4 Electric and Power Subsystem

The electrical power is provided by 2 rechargeable 12 VDC Ni-Cd battery connected in series. A custom board with DC/DC converters (see Figure 8.11) allows to regulate the electrical power from the battery voltage level to the operative voltage level required by the all on board subsystems, including sensors, the Micro-Controller Board and the Thruster Command Board. The architecture of the current on board Electric and Power Subsystem is showed in Figure 8.12.

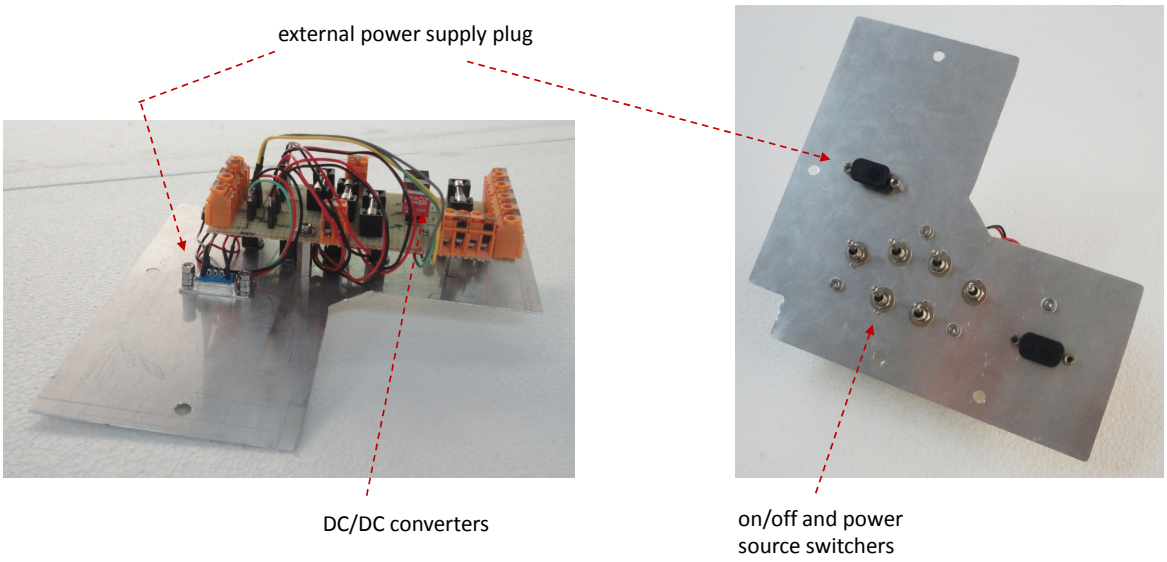


Figure 8.11: Custom Electric board assembly.

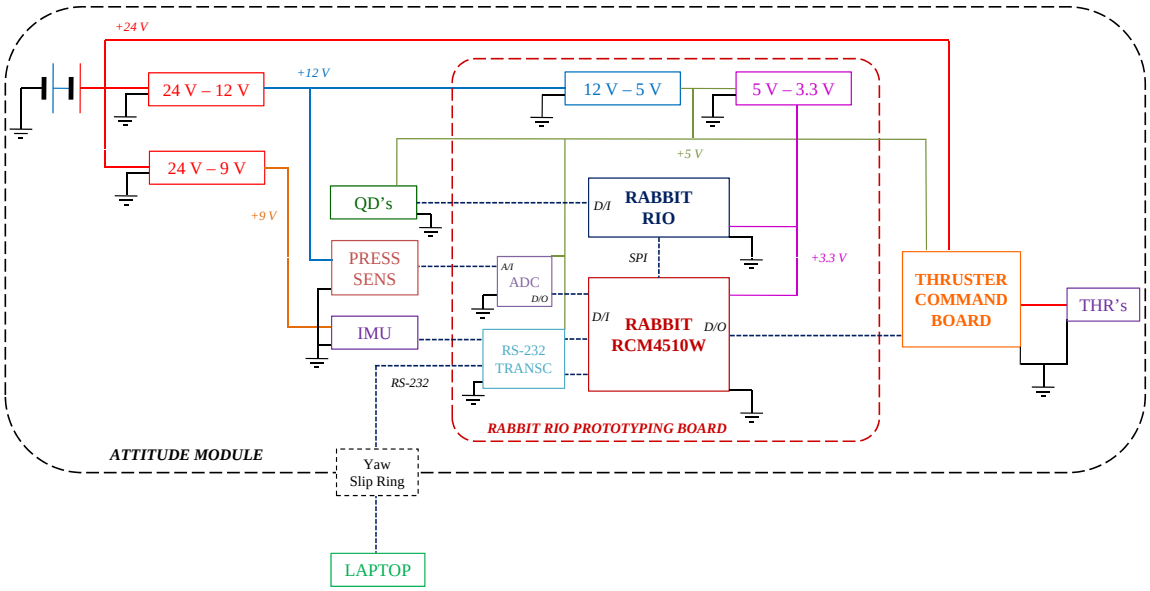


Figure 8.12: Electric and Power Subsystem and on board avionic elements communication.

8.3.5 Communication & Data Handling Subsystem

In the final configuration of the test bed, both the Spacecraft Simulators and the laptop use a Zeeg-Bee module to communicate each other in a peer-to-peer network. Two separate channel are planned to be used: (1) a Simulator-to-Simulator channel will be used to exchange information between the units flying in formation, allowing to simulate different formation flying architecture

(see Chapter 2); (2) a Simulator-to-Laptop channel will be used for the transmission of commands from and telemetry data to the laptop control station. In the current state of the project, the wireless communication is not yet available, and only the Simulator-to-Laptop channel has been realized with the AM's micro-controller communicating by wire with the laptop using a RS-232 transceiver and the yaw slip ring. Figure 8.12 shows the communication links between the on board avionic elements and the laptop

The communication protocol is based on variable size packets whose structure consists of a 5-byte header, a n-byte data payload or message and a final 2-byte checksum (see Figure 8.13). The header carries information about the transmitted data payload: (1) SOH is the Start Of Header byte equal to 1; (2) FROM_ID is the transmitting unit ID; (3) TO_ID is the receiver unit ID; (4) MSG_ID is the data message ID; and (5) COMPL_ID is the 255 complement of the 8-bit checksum of FROM_ID, TO_ID and MSG_ID[2]. Each unit exchanging data, i.e. each Simulator and the Laptop control station, as long as each message, has a unique ID, and each message ID is associated to a data payload with a pre-defined number of bytes and content. SOH and COMPL_ID are used by the decoding algorithm to identify a new incoming data packet. The final 2-byte checksum is the 16-bit checksum of the n bytes message, with the Least Significant 8 Bits as first element (CHK_LSB). The final checksum is used for error detection in the transmitted data.

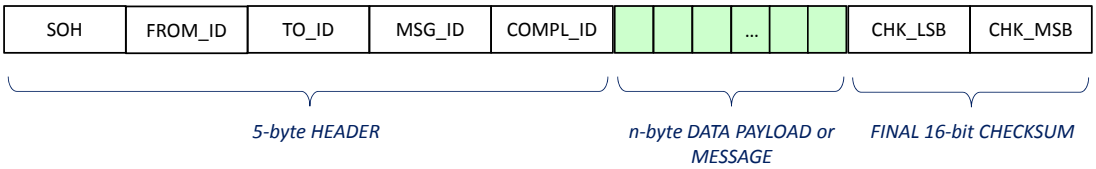


Figure 8.13: Data Packet Structure with an Header, a Data Payload or Message and a Final Checksum.

Messages exchanged between units can be divided into 5 types: (1) an Initiate Link Message is used to initialize the communication between units; (2) Acknowledge Messages are used to ensure that commands from the laptop (or a chief Spacecraft) are received and executed synchronously between multiple units[3]; (3) Data Request Messages are used to request the transmission, either multiple or single, of some telemetry or housekeeping data; (4) Data Messages are used to transmit telemetry, housekeeping and GNC data; (5) Command Messages are used to command the execution of a specific operation (except for Data Messages transmission).

All data are transmitted as signed or unsigned 8-, 16- or 32- bit integers, depending on the

[2]A k-bit checksum is a simple unsigned sum of the unsigned bytes, truncated to k bits.
[3]Acknowledge messages are also very useful for debugging.

variability range of each data. For example, the current on board time is transmitted as a unsigned 32-bit integer, while almost all float values are first rescaled and then transmitted as a truncated signed 16-bit integer, with a truncation error that is less than the float value accuracy. This allow to reduce data packets size without significantly degrade data accuracy.

8.3.6 On Board and Control Station Software

The on board software is written in Dynamic C, which is a C-based language developed by Digi International that allows to easily and efficiently implement multitasking operations, i.e. a sequence of operations that can *appear* to execute in parallel. Multiple tasks[4] can therefore be executed almost in parallel at a different frequency or priority. For example, GNC tasks, which have in general the highest priority, are executed at an higher frequency that housekeeping tasks.

The on board Dynamic C code is organized into 6 main libraries, one for each main on board process: (1) Sensors (SENS); (2) Position and Attitude Determination System (PADS); (3) Controller (CTRL); (4) Propulsion (PROP); (5) Communication (COMM); and (6) Housekeeping (HKP). The process Controller includes Guidance and Control. Also, each library includes two types of functions: (1) Initialization Functions, for the library data structures and variables (re)initialization; and (2) Update Functions, either periodic or event-driven executed, which are executed to update the library variables, such as current dynamic state updating or control action computation.

Figure 8.14 shows the main processes managed by the on board micro-controller. Two types of processes are used. Periodic Interrupt Processes perform repetitive, time dependent operations such as GNC operations or thrusters firing times setting. Event-driven background and GNC tasks are used to implement not-time-dependent operations such as software system initialization when the micro-controller is powered on, or formation flying mode changing when a collision is detected. Some features, as for example data transmission, are both periodic and event-driven, since telemetry data are transmitted at a given frequency while warning messages are transmitted only when some kinds of events occur. Periodic GNC operations are synchronized, since it is very important that control actions are computed using updated data sensors, as depicted in Figure 8.15. Also, SENS and PADS processes can be run at a frequency, $1/\Delta t_{SENS}$ and $1/\Delta t_{PADS}$ respectively, which is (at least) equal or higher than the CTRL and Propulsion processes frequency $1/\Delta t_{ctrl}$.

The on board operations are divided into three hierarchical levels: programs, tests, and maneuvers [84]. Each program is associated with a particular executable file (with an assigned ID) consisting of one or more stand alone experiments or tests. Each test may in turn consists of a linear or non-linear sequence of maneuvers, which are convenient tools to separate a complex motion in a sequence of simpler movements.

[4] A task is defined as a pre-defined sequence of operations.

In order to simplify the implementation of the algorithms being tested, a set of primary interface functions to the existing on board software were developed. These functions are associated both to periodic and to event-driven processes defined by the guest scientist, and have pre-defined inputs and outputs that the scientist can use to implement his own experiment.

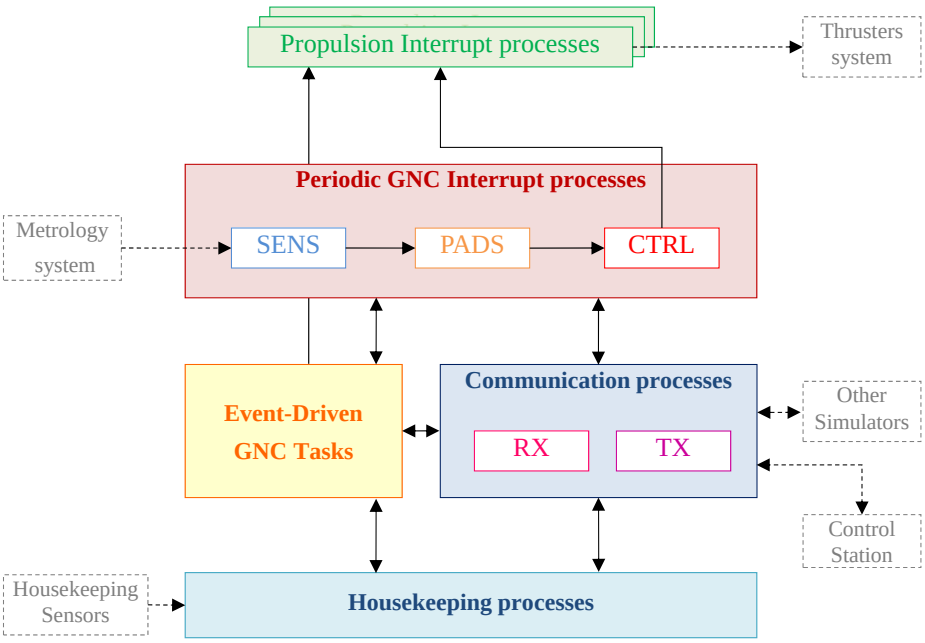


Figure 8.14: Micro-controller main processes and data flow.

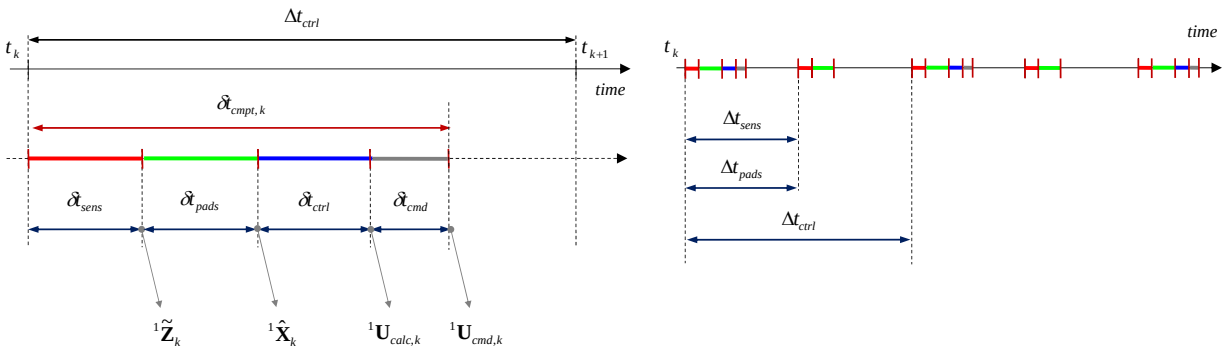


Figure 8.15: Periodic GNC processes time schedule.

8.4 Matlab Software Simulator

We developed a Matlab Software Simulator with the aim of preliminarily simulate both translational and attitude dynamics of the Spacecraft Hardware Simulators flying in formation and test some formation control strategies. As done in the Formation Flight Matlab Simulator described in Chapter 4, we adopted an Object Oriented Programming (OOP) mainly for its characteristics of modularity, extensibility and re-usability. The same classes can indeed be used for each Hardware Simulator flying in formation. The software architecture of each unit flying in formation is shown in the block diagram of Figure 8.16, which refers only to the attitude dynamics. In this diagram, q is the quaternion vector, ω is the angular velocity vector, τ represents a torque vector, ϕ, θ and ψ are the roll, pitch and yaw angles, respectively, and t is the current time.

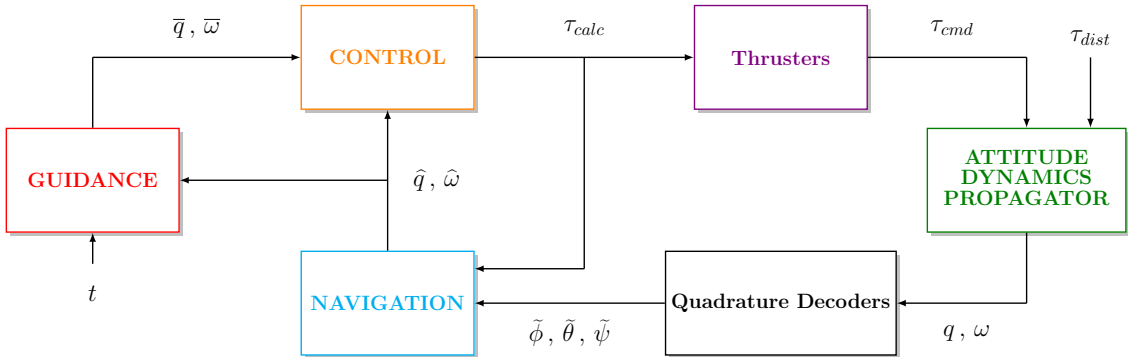


Figure 8.16: Software architecture of the attitude dynamics of the Matlab Software Simulator for the SFF hardware simulator.

At the moment of writing this thesis, the Matlab Software Simulator allows to simulate the AM's attitude dynamics with 3 DOF and control its orientation using two types of controllers. The first controller is a 1 DOF MPC both in its on-line and explicit formulations, as described in details in Chapter 3. The second controller is a 1DOF PID controller modified with a proportional and a derivative kick avoidance scheme and and anti-reset windup strategy, as mentioned in Chapter 3. In a discrete-time system, the control action is computed at time instants t_k's with a frequency $f_{ctrl} = 1/\Delta t_{ctrl}$. We therefore implemented a discrete time version of the previous PID control law using a forward finite differences method for the integral term discretization and a backward differences method for the derivative term discretization. The discrete-time version of the PID controller is then:

$$u(t_k) = \left[K_p\,\beta + \left(\frac{N\,K_p\,T_d}{N\,\Delta t_{ctrl} + T_d}\right)\,\gamma\right]\,r(t_k) + \left[-K_p - \left(\frac{N\,K_p\,T_d}{N\,\Delta t_{ctrl} + T_d}\right)\right]\,y(t_k) + \\ + \left[\frac{K_p\,\Delta t_{ctrl}}{T_i} - \left(\frac{N\,K_p\,T_d}{N\,\Delta t_{ctrl} + T_d}\right)\,\gamma\right]\,r(t_{k-1}) + \left[-\frac{K_p\,\Delta t_{ctrl}}{T_i} + \left(\frac{N\,K_p\,T_d}{N\,\Delta t_{ctrl} + T_d}\right)\right]\,y(t_{k-1}) + \\ + I(t_{k-1}) + \frac{\Delta t_{ctrl}}{T_t}\left[u'(t_{k-1}) - u(t_{k-1})\right] + \left[\frac{T_d}{N\,\Delta t_{ctrl} + T_d}\right] + D(t_{k-1}) \tag{8.1}$$

The SFF team is moving toward quaternion-based PID and MPC control schemes to control the AM in 3 DOF.

The Guidance module implements pseudospectral optimal control methods used to find attitude reference trajectories. These methods can be used in general to solve optimal control problems by approximating the time-dependent state and control variables as a weighted sum of Legendre or Chebyshev polynomial basis. This allows to discretize the differential equations and continuous constraints of the optimal control problem over a grid of orthogonal collocation nodes, thus obtaining a set of algebraic equations. The integration in the cost functional is approximated by well known Gauss quadrature rules, consisting of a weighted sum of the function values at the discretization nodes. The original optimal control problem is therefore directly discretized to formulate a nonlinear programming problem, which is then solved numerically using a sparse nonlinear programming solver.

Pseudospectral methods main features include their exponential (or spectral) rate of convergence, which is faster than any polynomial rate, and a good accuracy achievable with relatively coarse grids [85]. Pseudospectral techniques have emerged over the last 15 years as important computational methods for solving optimal control problems in many fields, including space. NASA used such methods for generating real time trajectories for a spacecraft maneuver [86].

8.5 Experimental Activities

In order to test some attitude control strategies, we needed to carry out the following preparatory experimental activities using a torque system with a steel wire as sensitive element (see Table 8.3).

TEST 1 Estimation of the rotational elastic stiffness k of the wire; using a Torque Pendulum system, the wire rotational stiffness k is estimated from measurements of the pendulum free oscillation period T.

TEST 2 Estimation of the force that an AM's thruster can perform; using a Torque Balance system, the thrust force S is estimated from the equilibrium angle θ_{eq} at which the

torque due to a couple of thrusters is balanced by the elastic torque due to the wire.

TEST 3 Characterization of the AM's inertia properties; using a Torque Pendulum system, the AM's moment of inertia around the wire axis is estimated from measurements of the pendulum free oscillation period T. Repeating this test for at least 6 independent AM's axes, it is possible to obtain an estimation of the AM's inertia matrix.

Each test with main experimental results will be briefly described in the following subsections. For a more detailed description see [87].

Table 8.3: Experimental activities conducted.

Test	Estimated quantity	Used system	Measured quantity
1	wire rotational stiffness, k	Torque Pendulum	free oscillation period, T
2	thrust force, S	Torque Balance	equilibrium angle, θ_{eq}
3	AM's inertia properties	Torque Pendulum	free oscillation period, T

8.5.1 Torque Wire Characterization

The final aim of this test is to estimate the rotational stiffness of the wire that is the sensitive element of the systems we used for the next two tests. For this purpose, we built a Torque Pendulum system with well known moment of inertia about the wire axis, I, and with the center of mass along the wire axis, as shown in Figure 8.17, using components with well known mass and geometry. Making the Pendulum oscillate around the equilibrium position and taking several measurements of its free oscillation period T, the wire rotational elastic stiffness k can be computed according to the following formula:

$$k = 4\,\pi^2\,\frac{I}{T^2} \tag{8.2}$$

To measure the oscillation period of the pendulum, we used two kind of systems. In the first one, called photo-resistance system, a laser is mounted on the Torque Pendulum and a photo-resistance is placed along the trajectory that the laser follows during the pendulum oscillation. The photo-resistance is connected in series with a $1\,k\Omega$ measure resistance R_m and integrated into the electronic circuit shown in Figure 8.17. An Arduino UNO board connected to a laptop is used to measure the voltage V_m between the measure resistance ends over time. When the laser ray passes over the photo-resistance, the electric resistance of the last one decreases, resulting in a peak on the measured voltage V_m (see Figure 8.18). A Matlab program was then used to evaluate the

oscillation period T as the time period between two corresponding peaks in the acquired voltage time profile.

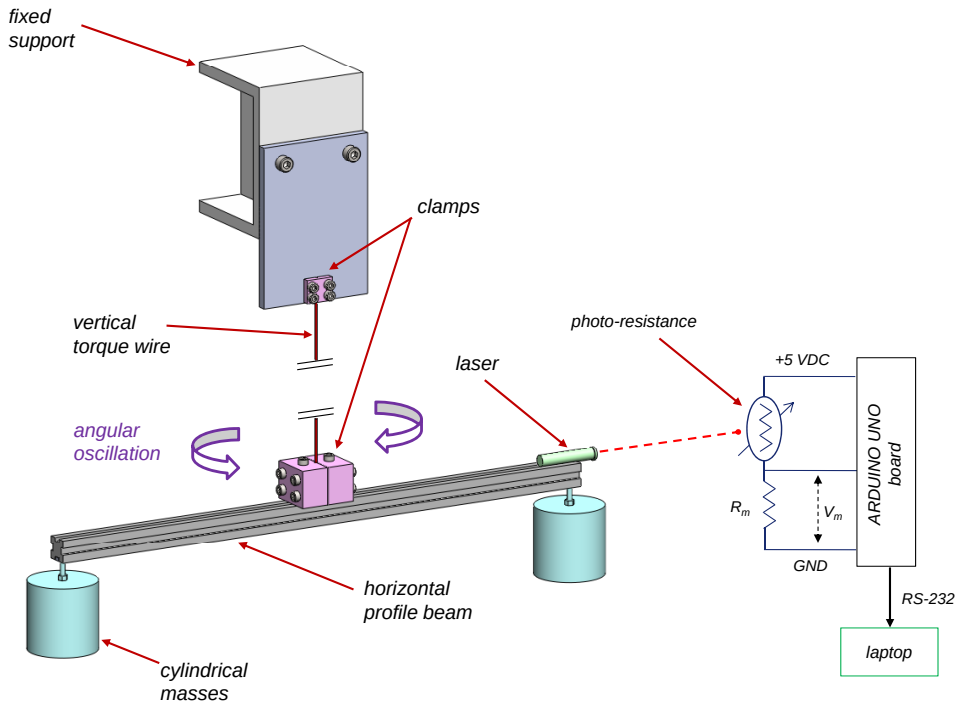


Figure 8.17: Torque wire characterization experimental setup with the photo-resistance based acquisition system.

In order to validate the photo-resistance method, we used a second method, called the web cam method. In this method, we used a web cam to record the laser ray passage over a reference point P_r marked on a white screen. We then manually evaluated the time instants t_k when the laser passed over P_r on the screen, calculating the pendulum oscillation period T as difference of corresponding couples of t_k's.

Several tests were conduced using both the photo-resistance and the web cam methods with a maximum angular amplitude of oscillation between 30° and 140° and a wire length of $l_{w,1} = 956\,mm$. Tests results are listed in Table 8.4 and summarized in Figure 8.19.

The expected value of k for a steel wire with radius $r = 1\,mm \pm 0.02\,mm\,(2\%)$, length $l_{w,1} = 956\,mm \pm 1\,mm\,(0.1\%)$ and modulus of rigidity $G = 75\,GPa \pm 6\,GPa\,(8\%)$ can be computed as:

$$k = \frac{\pi\, r^4\, G}{2\, l_{w,1}} = 0.123\,\frac{N\,m}{rad} \pm 0.014\,\frac{N\,m}{rad}\,(14\%) \tag{8.3}$$

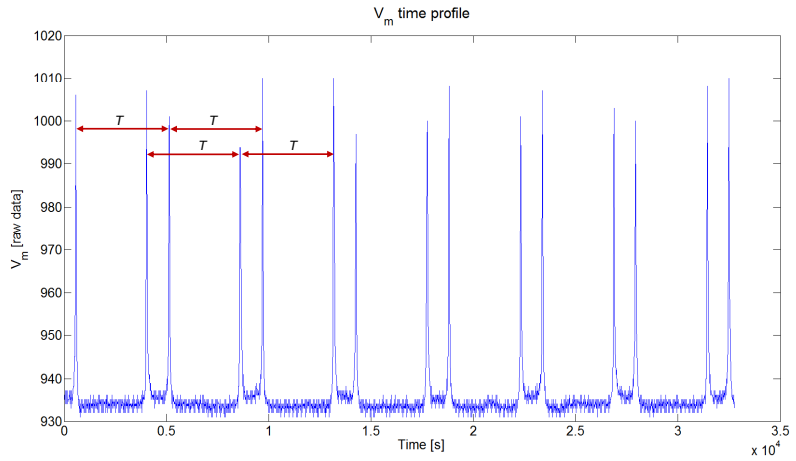


Figure 8.18: Typical V_m time profile.

Table 8.4: Test results of the photo-resistance method vs. the web cam method.

Test		photo-resistance		web cam	
#	θ_{max} [°]	$\bar{T}$ $[s]$	σ_T $[s]$	$\bar{T}$ $[s]$	σ_T $[s]$
1	30	4.69	0.02	4.68	0.02
2	40	4.69	0.02	4.68	0.02
3	90	4.70	0.02	4.69	0.02
4	100	4.70	0.02	4.69	0.02
5	130	4.71	0.02	4.70	0.02
6	140	4.71	0.02	4.71	0.02

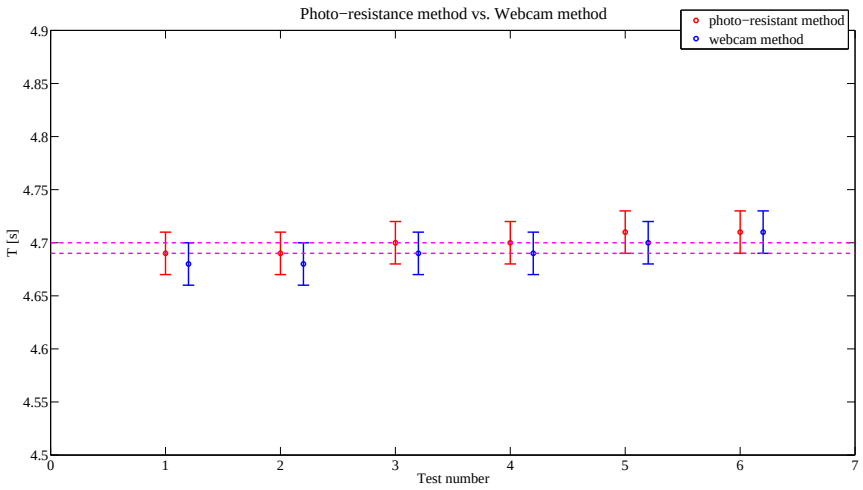


Figure 8.19: Test results of photo-resistance vs. web cam methods.

We estimated a Torque Pendulum moment of inertia about the wire axis of $I = 6.07 \cdot 10^{-2} kg\, m^2 \pm 4 \cdot 10^{-4}\, kg\, m^2$ and a free oscillation period $T = 4.70\, s \pm 0.02\, s$. Using Equation 8.2, the rotational elastic stiffness of the wire k results:

$$k_1 = 0.109\, \frac{N\, m}{rad} \pm 0.001\, \frac{N\, m}{rad}\, (1\sigma) \tag{8.4}$$

where the uncertainty of k_1 is computed with the following Kline-McClintock formula:

$$i_k = \frac{4\,\pi^2\, I}{T^2} \sqrt{\left(\frac{i_I}{I}\right)^2 + 4\left(\frac{i_T}{T}\right)^2} \tag{8.5}$$

$i_I$and i_T are the uncertainties of I and T respectively. The relative uncertainty of k is $i_{k,rel} = 1\,\%\,(1\sigma)$.

Analyzing the experimental results we can state that measurements of T acquired with different θ_{max} and with the two different methods previously described are consistent, since there exist a band between $4.69\, s$ and $4.70\, s$ which is common to all the measurements error bands. We can therefore exclude systematic error intrinsic to both methods. The photo-resistant method has therefore been tested and it was used in the next two tests since simpler and more automatable than the web cam method. We can also conclude that the wire torque stiffness does not significantly depend on the maximum angular amplitude of oscillation if $\theta_{max} < 140°$, since the mean value of T stays within the root mean square of each measurement.

The experimental value of k_1 is also consistent with the expected theoretical value computed in Equation 8.3. We conduced further tests setting the length of the wire to $l_{w,2} = 935\, mm$ by changing the wire length tightened between the two clamps. Experimental results gave us $k_2 = 0.111\, Nm/rad$, which is consistent with the previous experimental result k_1 since $k_1 l_{w,1} = k_2 l_{w,2} = \pi\, r^4\, G/2 = const = 0.104\, Nm^2/rad$.

8.5.2 Thrust Force Estimation

The objective of this second test is to estimate the thrust force performed by the AM's thrusters. We therefore built a Torque Balance system (see Figure 8.20) with a vertical torque wire (the same one we characterized in Test 1), a couple of AM's thrusters placed on an horizontal profile beam at a distance $b/2$ from the wire, two lasers at the ends of the profile beam, a pneumatic circuit attached under the beam (similar to the one mounted on board the AM) to supply thrusters, a balancing system to place the center of mass of the system along the wire axis, and a goniometer with an angular resolution of 0.5° to detect the angular position θ of the Torque Balance.

When the couple of thrusters are powered on, the Torque Balance system is in an equilibrium condition with $\theta = \theta_{eq}$ when the torque generated by the couple of thrusters τ_{thr} is balanced by

the elastic torque due to the wire τ_w:

$$\tau_{thr} = S\,b = \tau_w = k\,\theta_{eq} \tag{8.6}$$

Taking measurements of the equilibrium angle θ_{eq} with known k and b, it is possible to estimate the thrust force as:

$$S = \frac{k\,\theta_{eq}}{b} \tag{8.7}$$

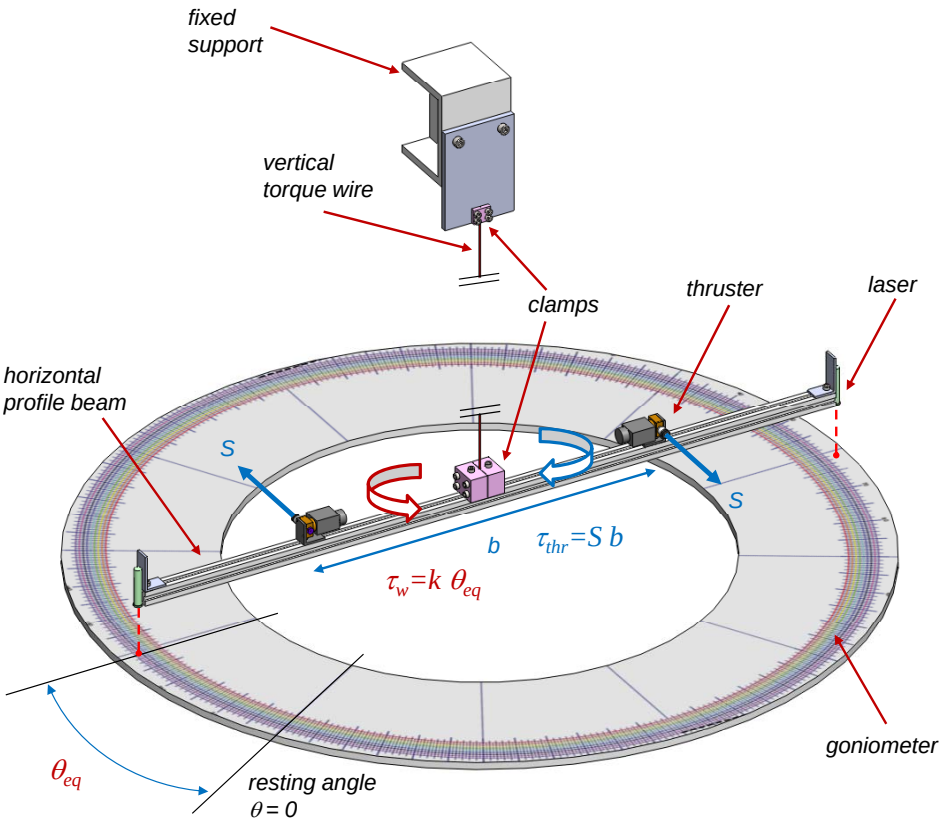


Figure 8.20: Thrust force estimation experimental setup.

To characterize the AM's thrusters in terms of thrust force as a function of the total pressure incoming the nozzle, we conduced several tests measuring the Torque Balance equilibrium angle θ_{eq} for given values of the total pressure in the range $1\,bar$ - $10\,bar$. The total pressure incoming the nozzle was set by adjusting the pressure-reducing regulator and measured using a Rabbit RCM4300 with an on board A/D converter. We also measured the overshoot angle θ_{os} of the Torque Balance dynamic response to a step torque applied by the thrusters couple at the initial resting condition. Indeed, for a very low damping system, θ_{os} is almost twice θ_{eq}.

Table 8.5: Experimental results of the AM's thrusters characterization.

Test #	$p_0\,[bar]$	$\theta_{eq}\,[^\circ]$	$\theta_{os}\,[deg]$	$S\,[N]\,(1\sigma)$
1	1.7 ± 0.1	6.0 ± 0.5	12 ± 1	0.028 ± 0.002
2	2.6 ± 0.1	15.0 ± 0.5	29 ± 1	0.070 ± 0.002
3	4.5 ± 0.1	32.5 ± 0.5	63 ± 1	0.151 ± 0.003
4	6.4 ± 0.1	49.5 ± 0.5	97 ± 1	0.230 ± 0.003
5	8.3 ± 0.1	67 ± 0.5	132 ± 1	0.311 ± 0.004
6	9.2 ± 0.1	76 ± 0.5	148 ± 1	0.353 ± 0.004
7	9.7 ± 0.1	82 ± 0.5	158.5 ± 1	0.380 ± 0.004
8	9.7 ± 0.1	81.5 ± 0.5	158 ± 1	0.378 ± 0.004

The expected S - p_0 profile is linear, as stated by the following formula that is valid for an isoentropic nozzle with sonic flow conditions at the exit area A_e:

$$S = \left\{ \left[\left(\frac{2}{k+1} \right)^{\frac{k}{k-1}} + k \left(\frac{2}{k+1} \right)^{\frac{k+1}{2(k-1)}} \right] p_0 - p_a \right\} A_e \tag{8.8}$$

where p_a is the ambient pressure outside the nozzle and $k = 1.4$ is the air specific heat ratio.

Experimental results are listed in Table 8.5 and shown in Figures 8.21 and 8.22. The θ_{os} uncertainty is greater than the θ_{eq} one since θ_{os} is measured in a dynamic condition and consequently more sensitive to lateral oscillations that were easily reduced by hand in static conditions as for the θ_{eq} acquisitions.

The experimental profile of S as a function of p_0 is linear with good accuracy for $2\,bar \leq p_0 \leq 10\,bar$, as confirmed by the following results of a least squares linear regression:

$$S(p_0) = 0.0435 \frac{N}{bar}\, p_0 - 0.0457\,(r > 99\%) \tag{8.9}$$

where r is the linear correlation coefficient of the two variables and p_0 is in bar.

We can therefore conclude that, when supplied with air at the operative total pressure $p_{0,thr} = 10\,bar$, the AM's thrusters can perform a thrust force of:

$$S = 0.389 \pm 0.007\,N\,(2\sigma) \tag{8.10}$$

The force performed by an AM's thruster is estimated with a relative uncertainty $i_{S,rel} = 1.80\,\%\,(2\sigma)$.

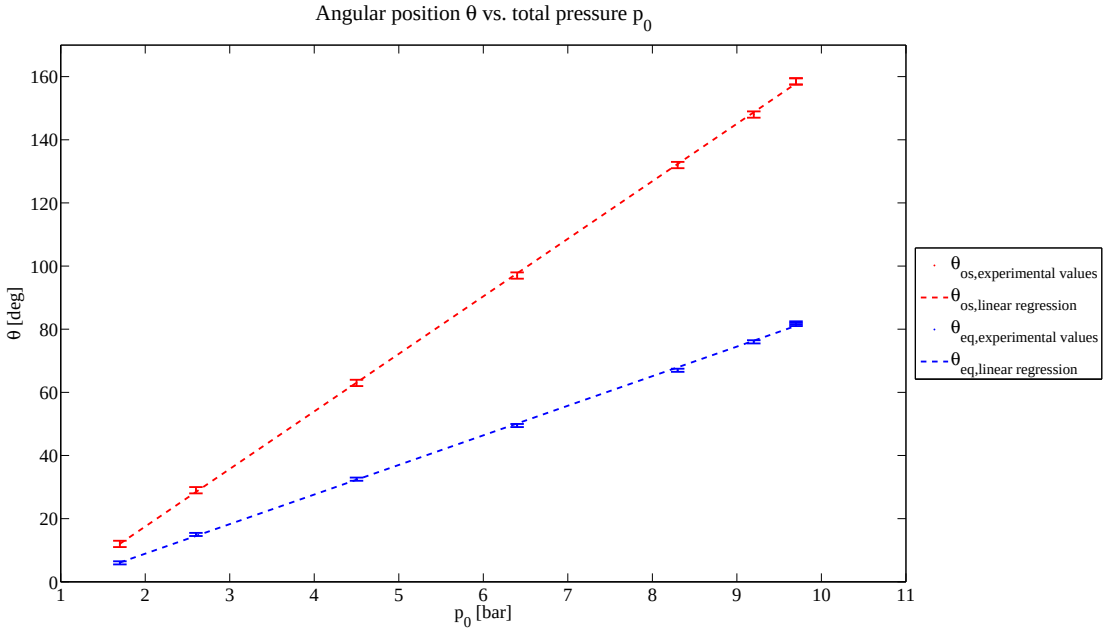


Figure 8.21: Experimental results of: (a) θ_{eq} vs. p_0 in blue, and (b) θ_{os} vs. p_0 in red.

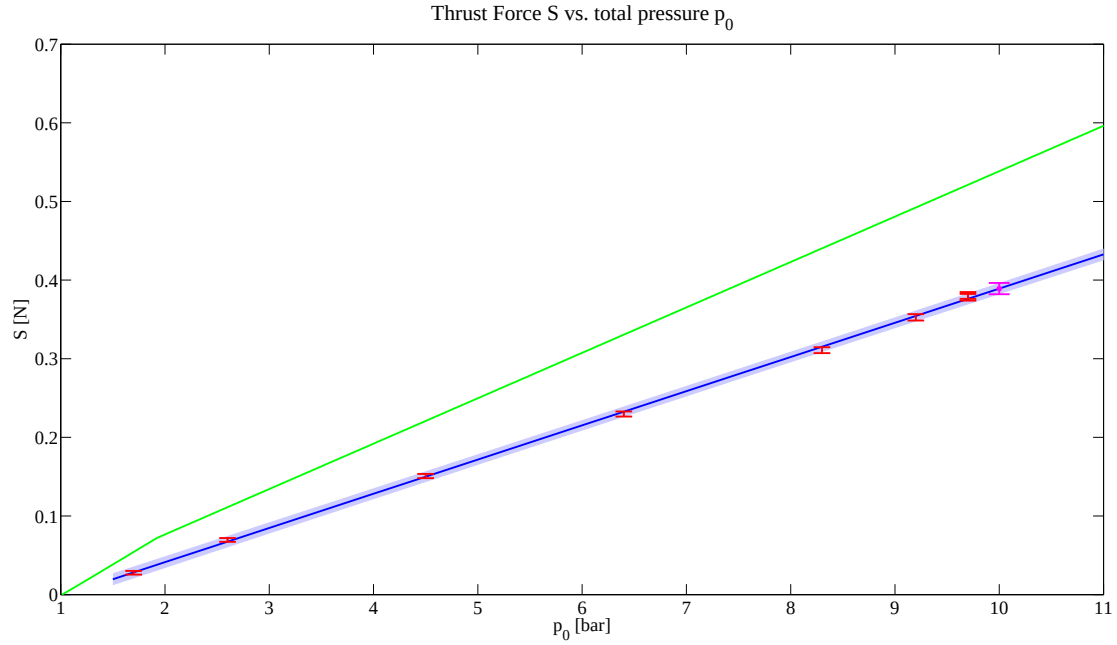


Figure 8.22: Thrust force S vs. total pressure p_0 with: (a) experimental results with error band in red, (b) least square linear regression results with 2σ error band in blue, and (c) theoretical thrust profile of an isoentropic nozzle in green.

8.5.3 Inertia Properties Determination

The aim of this last experimental activity is to estimate the AM's inertia tensor in the Body Reference Frame. As described in [88], different methods for the inertia tensor identification are available. Torsional or torque pendulum methods can be used to estimate the moment of inertia about a specified rotation axis passing through the system center of mass. The system under test oscillates about the rotation axis by means of a restoring torque generated by springs (or an equivalent system). The moment of inertia with respect to the rotation axis can be calculated from the measured oscillation period (or frequency) if the rotational spring stiffness is known. When the last quantity is not known or is not easy to estimate, a relative method can be adopted. This method consists on adding an additional known mass to the test system and the moment of inertia of the last one, I_1, is calculated using the measured oscillation frequencies of two tests performed with the unmodified an modified test system, f_1 and f_2 respectively:

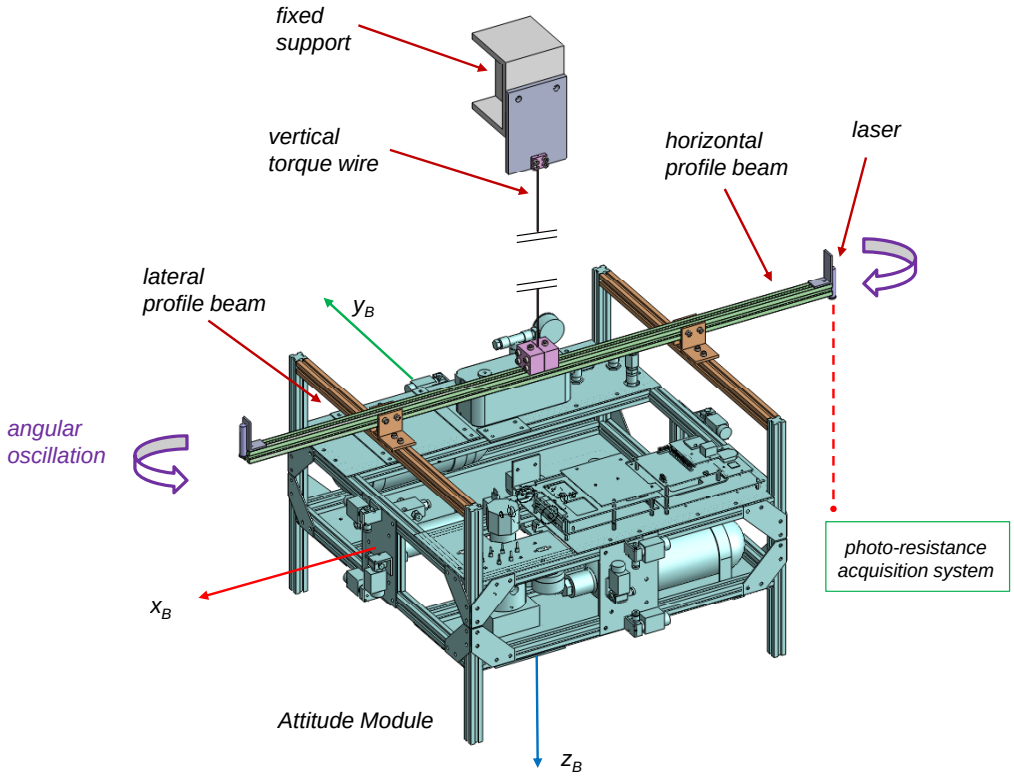
$$I_1 = \frac{I_0 f_2^2}{f_1^2 - f_2^2} \tag{8.11}$$

where I_0 is the moment of inertia of the additional mass about the given rotation axis.

At least six tests with different specified rotation axes are in general needed in order to identify the complete inertia tensor, since in a non principal reference frame the inertia tensor is symmetric with 6 independent components. The torsional pendulum method is safe, simple experimental setup and software are needed and the procedure is well approved in industry. The skill requirements of the testing personal is basic (once the procedures and/or software needed have been identified and verified). However, time requirements are high, since no multiple moment of inertia identification is possible simultaneously. As stated in [88] a high accuracy of $\pm(0.5 - 1.5)\%$ can be achieved.

Other methods that can be used to this purpose involve multi-filar pendulum systems, e.g. bi-filar or tri-filar [89], where the test system is suspended by wires and the restoring torque is generated by the gravity.

We therefore built a Torque Pendulum in order to measure the AM's moment of inertia about at least 6 independent directions. The Torque Pendulum used for this third test is similar to the one we used for test 1. We have indeed the wire in vertical position, an horizontal profile beam, two lasers at the ends of the profile beam and some lateral profile beams to support the AM with the required orientation. The AM is supposed to be balanced and with its center of mass along the wire axis. Figure 8.23 depicts the experimental setup used to estimate the the AM's moment of inertia about its z_B axis.



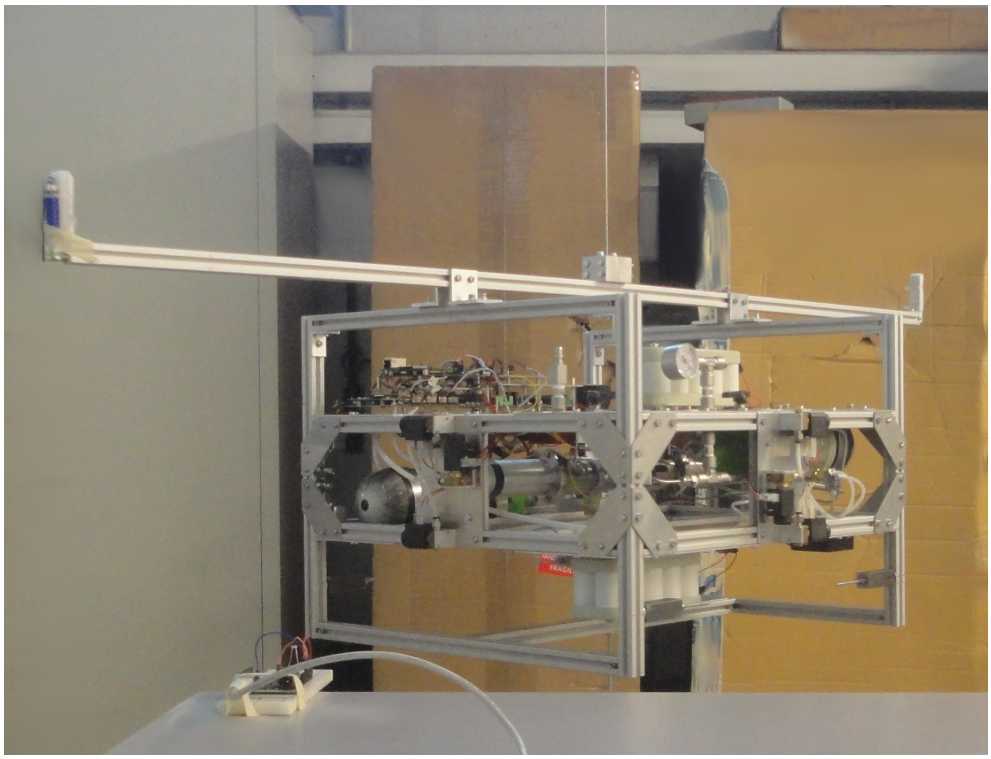


Figure 8.23: Experimental setup used to estimate the AM's moment of inertia about its z_B axis.

Table 8.6: Experimental results of the Torque Pendulum free oscillation period T used to estimate the AM's moment of inertia about its z_B axis.

Test Number	# samples	mean value [s]	standard deviation [s]
Empty Tanks			
1	16	23.40	0.09
2	14	23.38	0.10
3	24	23.38	0.08
Filled Tanks			
4	22	23.42	0.07
5	20	23.42	0.10
6	18	23.42	0.07

In this last test the wire rotational stiffness k is known. Taking therefore measurements of the Torque Pendulum free oscillation period T with the photo-resistance setup, we can estimate the moment of inertia of the rotating system I_{tot}, i.e. the AM's moment of inertia about its z_B axis $I_{AM,zz}$ plus the moment of inertia of the support system moving part I_{supp}:

$$I_{tot} = I_{AM,zz} + I_{supp} = \frac{k\,T^2}{4\,\pi^2} \tag{8.12}$$

I_{supp} can be either computed, if the support system is simple with well known geometrical and mass properties, or experimentally evaluated using the same setup (without the AM) otherwise.

To evaluate the contribution of the air stored in the tanks to the AM's inertia, we conducted some tests with empty tanks and some others with tanks filled at $40\,bar$[5]. Experimental results are listed in Table 8.6. As it can be seen, the contribution of the air mass within tanks to the oscillation period T is less than the estimation uncertainty, and therefore negligible. Combining results of Table 8.6 we estimated an oscillation period $T = 23.41\,s \pm 0.09\,s$. With a wire length of $l_{w,2}$ and therefore a wire rotational stiffness $k = 0.111\,N\,m/rad \pm 0.001\,N\,m/rad$, we obtained the following result:

$$I_{tot} = 1.54\,kg\,m^2 \pm 0.03\,kg\,m^2\,(2\sigma) \tag{8.13}$$

Since the support system used to estimate $I_{AM,zz}$ was very simple, I_{supp} was computed, obtaining the following estimation:

[5]The maximum allowable pressure at the moment of test execution.

$$I_{supp} = 0.171\,kg\,m^2 \pm 0.008\,kg\,m^2\,(2\sigma) \tag{8.14}$$

The the AM's moment of inertia about its z_B axis results:

$$I_{AM,zz} = 1.37\,kg\,m^2 \pm 0.03\,kg\,m^2\,(2\sigma) \tag{8.15}$$

with a relative uncertainty $i_{I_{AM,zz},rel} = 2.19\,\%\ (2\sigma)$.

8.6 Test of Attitude Control Maneuvers

At the moment of writing this thesis, the AM was not fully operative for 3 DOF attitude control tests. Therefore, this section presents some significant test results obtained with the AM in the 1 DOF configuration. A simple and light mechanical system was used to block the roll and pitch motions allowing rotations only about the yaw axis that coincides with the AM z_B axis.

8.6.1 Preliminary Tests and Kalman Filter Tuning

The aims of these preliminary tests are:

- to design a linear Kalman Filter (KF) with the use of the Matlab Simulator in order to obtain a better estimation of the AM angular velocity about its z_B axis, ω, with respect to a simpler Incremental Ratio (IR);

- to test the KF performances in estimating ω in the real system, i.e. the Attitude Module testbed.

8.6.1.1 The Kalman Filter

In linear systems with Gaussian process and measurement noises, the Kalman Filter, which was first introduced by Rudolph E. Kalman in his paper [90], is the optimal closed form solution to the recursive Bayesian estimation[6]. The purpose of the discrete-time Kalman filter is to provide a state optimal estimate of discrete-time linear systems, which can be represented by the following equations:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\,\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\,\mathbf{u}_{k-1} + \mathbf{W}_{k-1}\,\mathbf{w}_{k-1} \tag{8.16}$$

$$\mathbf{z}_k = \mathbf{H}_k\,\mathbf{x}_k + \mathbf{V}_k\,\mathbf{v}_k \tag{8.17}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{z} \in \mathbb{R}^m$ is the measurement vector, $\mathbf{u} \in \mathbb{R}^p$ is the input or control vector, $\mathbf{A} \in \mathbb{R}^{nxn}$ is the state transition matrix of the system, $\mathbf{B} \in \mathbb{R}^{nxp}$ is the control matrix, $\mathbf{H} \in \mathbb{R}^{nxm}$ is the measurement or observation matrix and $\mathbf{W} \in \mathbb{R}^{nxn}$ and $\mathbf{V} \in \mathbb{R}^{mxm}$ are the state and measurement noise matrices, respectively. The state noise vector $\mathbf{w} \in \mathbb{R}^n$ and the measurement noise vector $\mathbf{v} \in \mathbb{R}^m$ are assumed uncorrelated, zero-mean Gaussian and white noise.

[6]The most general form of optimal nonlinear state estimation.

In mathematical notation we have:

$$\begin{aligned}
\mathbf{w}_k &\backsim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}} = \mathbf{0},\, \mathbf{Q}_k) \\
\mathbf{v}_k &\backsim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{v}} = \mathbf{0}, \mathbf{R}_k) \\
cov(\mathbf{w}_k, \mathbf{w}_l) &= E\left\langle \mathbf{w}_k\, \mathbf{w}_l^T \right\rangle = \mathbf{Q}_k\, \delta(k-l) \\
cov(\mathbf{v}_k, \mathbf{v}_l) &= E\left\langle \mathbf{v}_k\, \mathbf{v}_l^T \right\rangle = \mathbf{R}_k\, \delta(k-l) \\
cov(\mathbf{w}_k, \mathbf{v}_k) &= E\left\langle \mathbf{w}_k\, \mathbf{v}_k^T \right\rangle = \mathbf{0}
\end{aligned} \tag{8.18}$$

where $\mathbf{Q} \in \mathbb{R}^{nxn}$ and $\mathbf{R} \in \mathbb{R}^{mxm}$ are the process noise covariance matrix and the measurement noise covariance matrix, respectively, and $\delta(k-l)$ represents the Kronecker delta. In addition, for all $k \geq 0$, $\mathbf{x}_k$ is uncorrelated with $\mathbf{v}_l$ for all l and $\mathbf{x}_k$ is uncorrelated with $\mathbf{w}_l$ for all $l \geq k$.

The discrete-time Kalman Filter algorithm is shown in Figure 8.24, where $\widehat{\mathbf{x}}_k^-$ is the *a priori* state estimate at step k given knowledge of the process prior to step k, $\widehat{\mathbf{x}}_k^+$ is the *a posteriori* state estimate at step k given measurement $\mathbf{z}_k$, and $\mathbf{P}_k^-$ and $\mathbf{P}_k^+$ are the corresponding error covariance matrices. The KF equations fall into two groups (as for the Bayesian estimator) resulting in a *predictor-corrector algorithm*. The *time update equations* are responsible for projecting forward in time the current state and the error covariance estimates to obtain the a priori estimates for the next time step. The *measurement update equations* are responsible for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

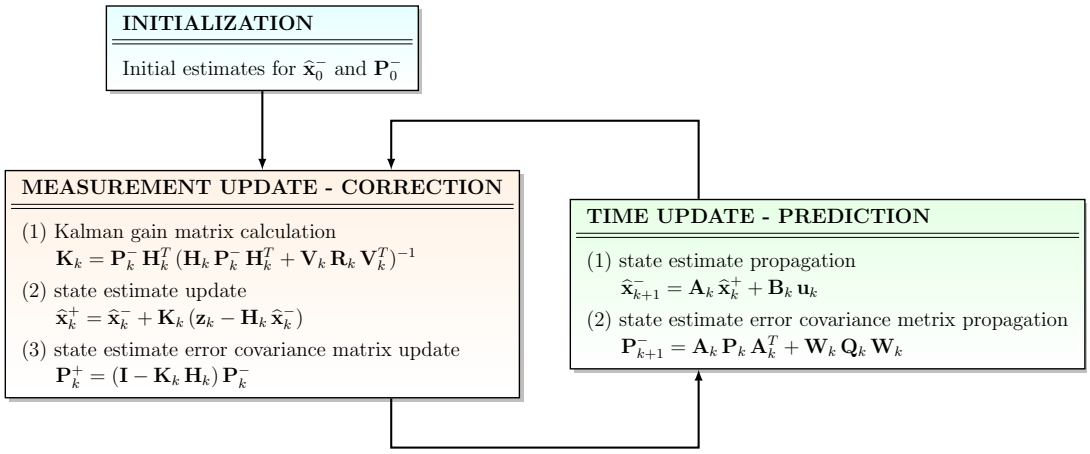


Figure 8.24: Discrete-time Kalman Filter algorithm.

The matrix $\mathbf{K}_k$ is chosen to be the *gain* or *blending factor* that minimizes the a posteriori error covariance matrix:

$$\mathbf{K}_k = \mathbf{P}_k^- \, \mathbf{H}_k^T \, (\mathbf{H}_k \, \mathbf{P}_k^- \, \mathbf{H}_k^T + \mathbf{R}_k)^{-1} = \frac{\mathbf{P}_k^- \, \mathbf{H}_k^T}{(\mathbf{H}_k \, \mathbf{P}_k^- \, \mathbf{H}_k^T + \mathbf{R}_k)} \tag{8.19}$$

Looking at Equation (8.19) we see that as the measurement error covariance $\mathbf{R}_k$ approaches zero, the gain $\mathbf{K}_k$ weights the residual $(\mathbf{z} - \mathbf{H}_k \, \widehat{\mathbf{x}}_k^-)$ more heavily, that is:

$$\lim_{\mathbf{R}_k \to \mathbf{0}} \mathbf{K}_k = \mathbf{H}_k^{-1} \Longrightarrow \widehat{\mathbf{x}}_k^+ \to \mathbf{H}_k^{-1} \, \mathbf{z}_k \tag{8.20}$$

This means that the actual measurement $\mathbf{z}_k$ is trusted more and more, while the predicted measurement $\mathbf{H}_k \, \widehat{\mathbf{x}}_k^-$ is trusted less and less. On the other hand, as the a priori estimate error covariance $\mathbf{P}_k^-$ approaches zero, the gain $\mathbf{K}_k$ weights the residual less heavily, that is:

$$\lim_{\mathbf{P}_k^- \to \mathbf{0}} \mathbf{K}_k = \mathbf{0} \Longrightarrow \widehat{\mathbf{x}}_k^+ \to \widehat{\mathbf{x}}_k^- \tag{8.21}$$

and the actual measurement $\mathbf{z}_k$ is trusted less and less.

After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. This recursive nature is one of the very appealing features of the Kalman Filter - it makes practical implementations much more feasible than, for example, an implementation of a Wiener filter which is designed to operate on all of the data directly for each estimate. The Kalman Filter instead recursively conditions the current estimate on all of the past measurements.

8.6.1.2 Kalman Filter tuning

For the problem on hand, $\widehat{\mathbf{x}} = [\,\widehat{\theta}\,, \widehat{\omega}\,]^T \in \mathbb{R}^2$ is the state vector estimate with angular position θ in rad and velocity in rad/s, $\mathbf{z} = \widetilde{\theta} \in \mathbb{R}$ is the yaw angle measurement in rad provided by the yaw Quadrature Decoder (QD), $\mathbf{u} = \tau \in \mathbb{R}$ is the control torque in $N\,m$, $n = 2$, $m = 1$ and $p = 1$. Preliminary experimental tests revealed that the GNC loop can be run at a maximum frequency of $f_{GNC,max} = 5\,Hz$, i.e. a GNC minimum sampling time interval of $\Delta t_{GNC,min} = 0.2\,s$. The state matrix $\mathbf{A}$, the control matrix $\mathbf{B}$ and the measurement matrix $\mathbf{H}$ for the discrete-time system result:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.2 \\ \\ 0 & 1 \end{bmatrix} \tag{8.22}$$

$$\mathbf{B} = \begin{bmatrix} 0.01429 \\ \\ 0.1429 \end{bmatrix} \tag{8.23}$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{8.24}$$

After the KF tuning we obtained the following process noise covariance matrix and measurement noise covariance:

$$\mathbf{Q} = \begin{bmatrix} 5 \cdot 10^{-4} & 0 \\ \\ 0 & 1.5 \cdot 10^{-3} \end{bmatrix} \tag{8.25}$$

$$\mathbf{R} = 1.767145 \cdot 10^{-4} \tag{8.26}$$

with an initial estimate error covariance matrix of:

$$\mathbf{P}_0 = \begin{bmatrix} 3 \cdot 10^{-3} & 0 \\ \\ 0 & 3 \cdot 10^{-2} \end{bmatrix} \tag{8.27}$$

To test the KF performance also when a control torque is performed by the on board thruster system, the control torque time profile shown in Figure 8.25 was applied to the system. Matlab simulation results are represented in Figures 8.26 - 8.28. Estimation errors plotted in Figure 8.28 were computed as difference between the estimated value and the true value. As you can see, the QD and the KF angular position estimate errors are almost the same, while the KF angular velocity estimate error is at least one order of magnitude less than the IR one (Figure 8.28, panel B). From Figure 8.27 we can also notice that:

- when a control torque is applied, the IR ω time profile has a delay w.r.t. the true value, leading to a bias of about $\pm 1\, deg/s$ in ω;
- when any control torque is applied, the IR ω is more noisy than the KF one.

Using a simple linear KF it is possible both to cancel the IR ω time delay and to reduce its noise.

Further Matlab simulations showed that the IR ω bias can be reduced by decreasing the sampling time interval Δt, but this leads to a more noisy angular velocity estimate, since the IR ω error component due to the QD angular resolution is $\epsilon_\theta / \Delta t$, where $\epsilon_\theta = 0.09\, deg$ is the QD angular resolution.

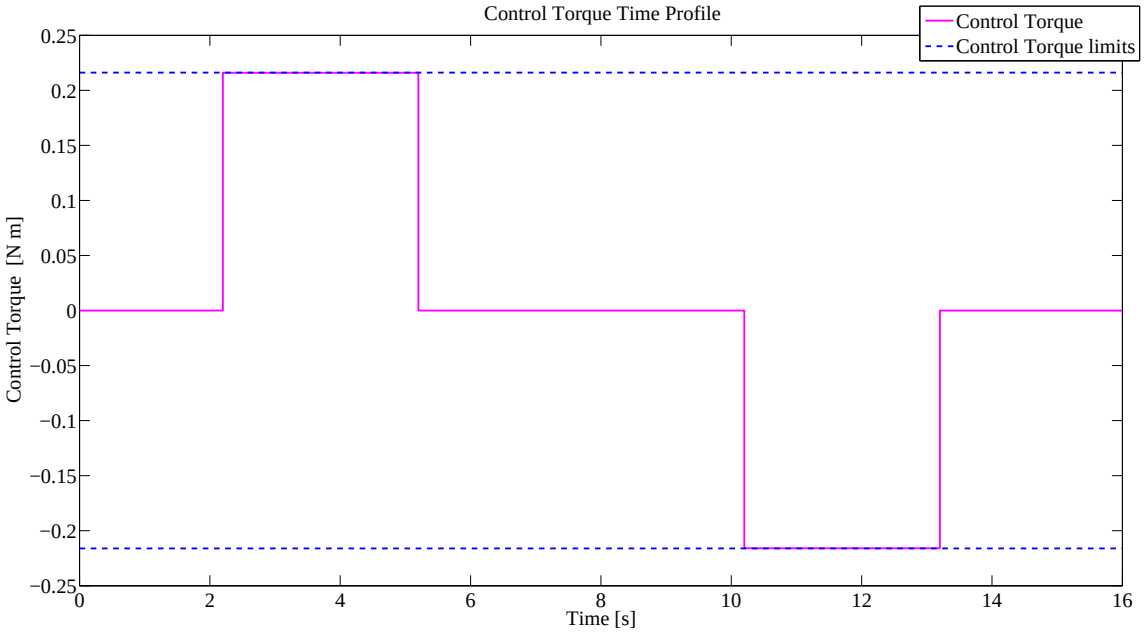


Figure 8.25: Control torque time profile – Test 1.

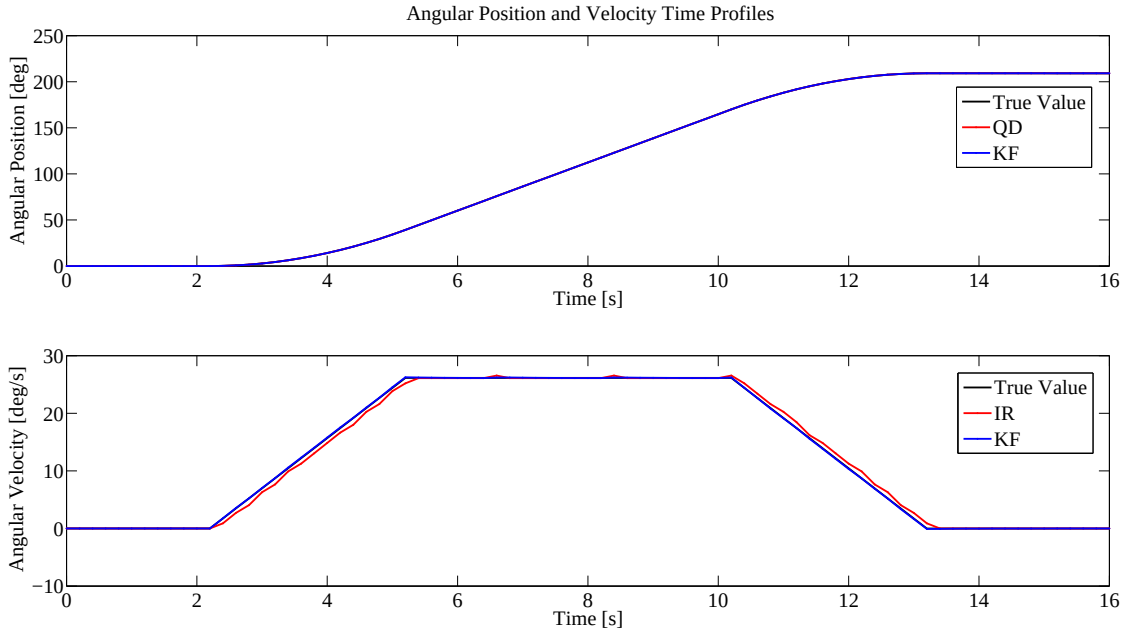


Figure 8.26: Angular position (top) and velocity (bottom) time profiles – Test 1 Matlab simulation results.

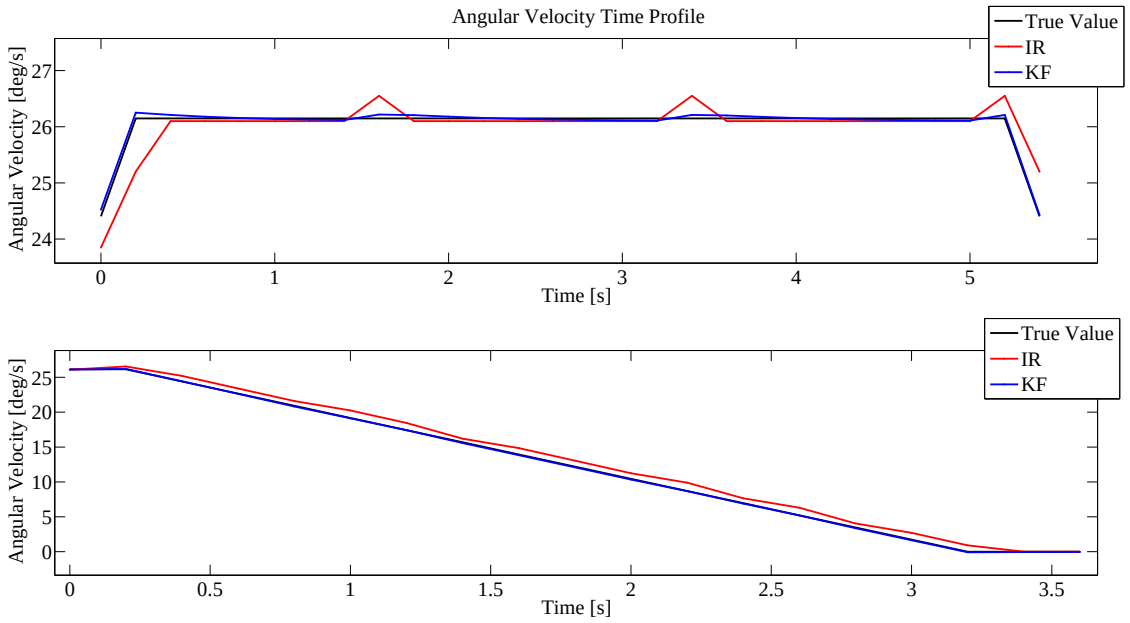


Figure 8.27: Particulars of experimental angular velocity time profile – Test 1 Matlab simulation results.

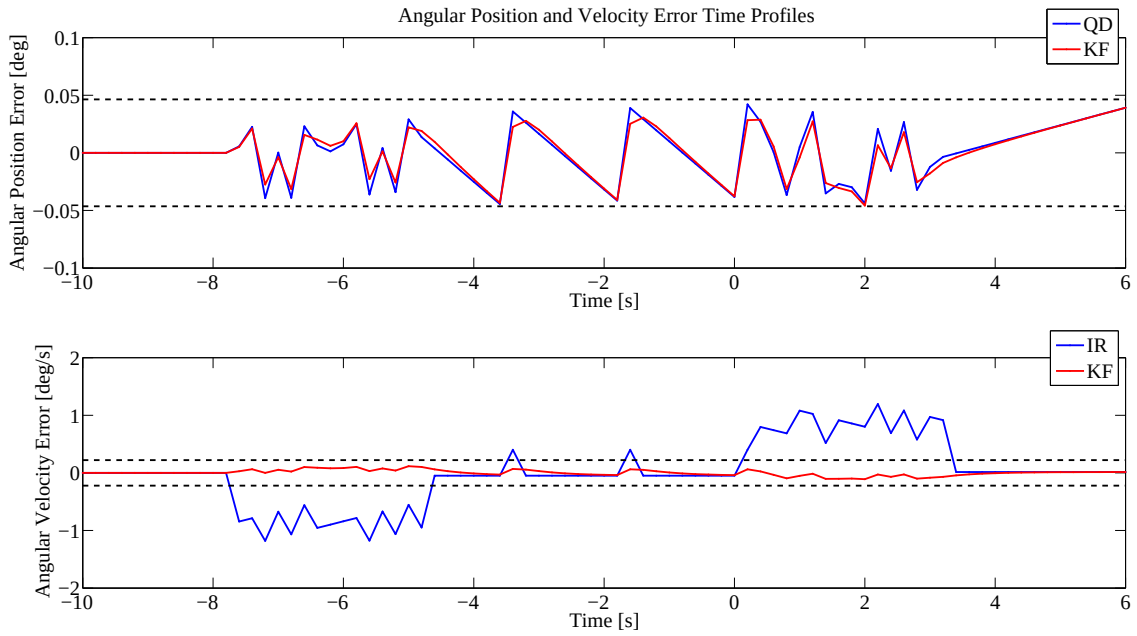


Figure 8.28: Angular position (top) and velocity (bottom) error time profiles with 3σ error band in dashed black lines – Test 1 Matlab simulation results.

8.6.1.3 Experimental results

KF equations were first rewritten and simplified for the problem on hand, and then implemented in the AM on board micro-controller as follows.

```
1  // Get the current angular position measurement
2  yKF_meas = MY_DEG2RAD*sens_qd_data[YAW].ang_pos;
3
4  // STATE ESTIMATE UPDATE
5  yKF_aux_1 = (yKF_x_min[0] - yKF_meas) / (yKF_P_min[0][0] + yKF_rm);
6  yKF_x_plus[0] = yKF_x_min[0] - yKF_P_min[0][0]*yKF_aux_1;
7  yKF_x_plus[1] = yKF_x_min[1] - yKF_P_min[1][0]*yKF_aux_1;
8
9  // STATE ESTIMATE ERROR COVARIANCE MATRIX UPDATE
10 yKF_aux_1 = 1.0/(gsp_yKF_P_min[0][0] + gsp_yKF_rm);
11 yKF_P_plus[0][0] = - yKF_P_min[0][0]*( yKF_P_min[0][0]*yKF_aux_1 - 1.0);
12 yKF_P_plus[0][1] = - yKF_P_min[0][1]*( yKF_P_min[0][0]*yKF_aux_1 - 1.0);
13 yKF_P_plus[1][0] = yKF_P_min[1][0] - yKF_P_min[0][0]*yKF_P_min[1][0]*yKF_aux_1;
14 yKF_P_plus[1][1] = yKF_P_min[1][1] - yKF_P_min[0][1]*yKF_P_min[1][0]*yKF_aux_1;
15
16 ...
17
18 // STATE ESTIMATE PROPAGATION
19 yKF_x_min[0] = yKF_u_k_1*1.428571428571429E-2 + yKF_x_plus[0] + yKF_x_plus[1]*2.0E
       -1;
20 yKF_x_min[1] = yKF_u_k_1*1.428571428571429E-1 + yKF_x_plus[1];
21
22 // STATE ESTIMATE ERROR COVARIANCE MATRIX PROPAGATION
23 yKF_P_min[0][0] = yKF_P_plus[0][0] + yKF_P_plus[0][1]*2.0E-1 + yKF_P_plus[1][0]*2.0E
       -1 + yKF_P_plus[1][1]*4.0E-2 + yKF_qt;
24 yKF_P_min[0][1] = yKF_P_plus[0][1] + yKF_P_plus[1][1]*2.0E-1;
25 yKF_P_min[1][0] = yKF_P_plus[1][0] + yKF_P_plus[1][1]*2.0E-1;
26 yKF_P_min[1][1] = yKF_P_plus[1][1] + yKF_qw;
```

Time profiles of experimental angular position and velocity are shown in Figures 8.29 - 8.31. Estimation errors depicted in Figure 8.31 were computed taking the KF estimated value as reference. Comparing these figures with the Matlab simulation ones, it can be noticed a little difference between the simulated and the real angular position and velocity time profiles shape. This difference is mainly due to a small friction torque at the yaw joint. However, KF experimental results are consistent with the ones obtained using the Matlab simulator, confirming the KF better performances also in the real system.

Taking into account the results of these preliminary tests we can conclude that:

- the AM angular position is estimated with an error less than 0.1 deg using mainly Quadrature Decoder measurements;

- the AM angular velocity is estimated with an error less than 0.1 deg/s using a linear Kalman Filter.

The use of a simple linear Kalman Filter slightly increase the Navigation computing cost, but reduce the angular velocity estimation error by one order of magnitude with respect to an Incremental Ratio approach.

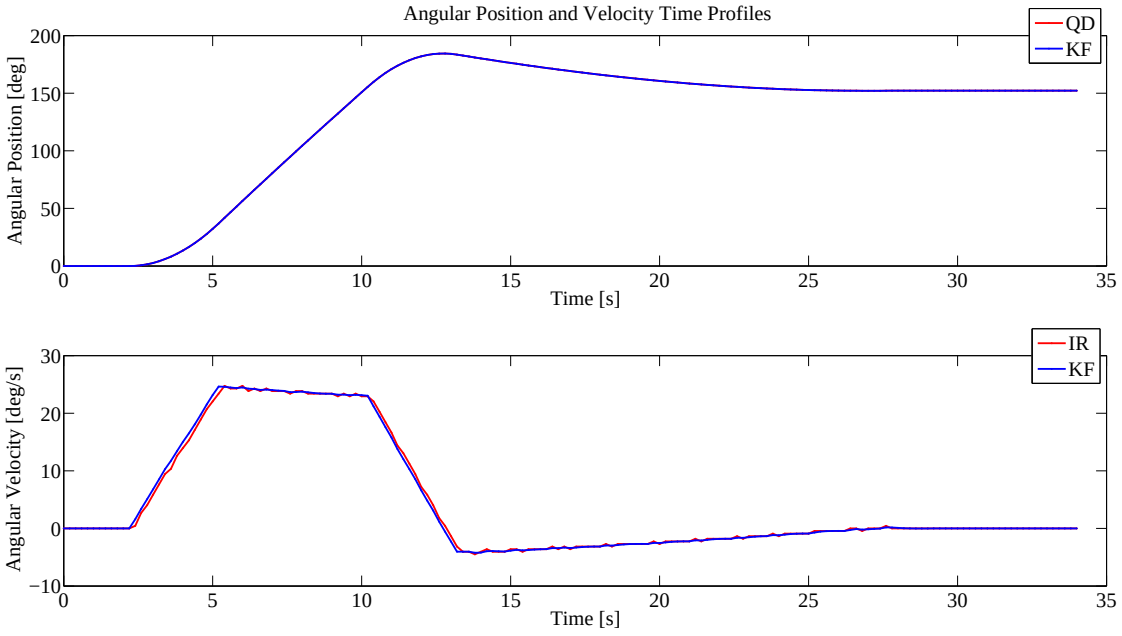


Figure 8.29: Angular position (top) and velocity (bottom) time profiles – Test 1 experimental results.

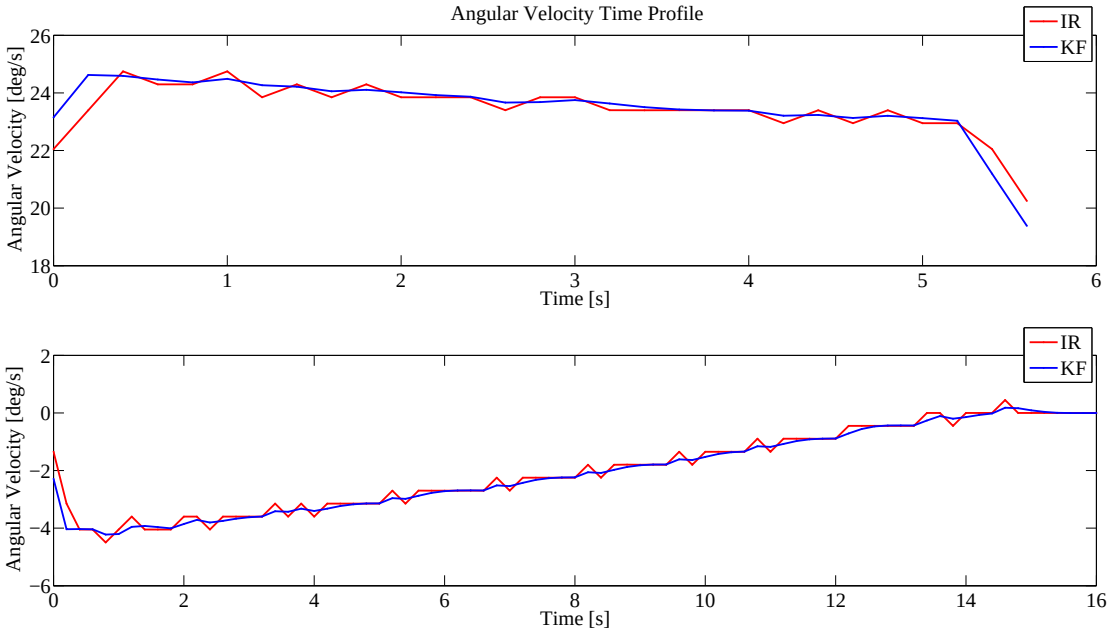


Figure 8.30: Particulars of experimental angular velocity time profile – Test 1 experimental results.

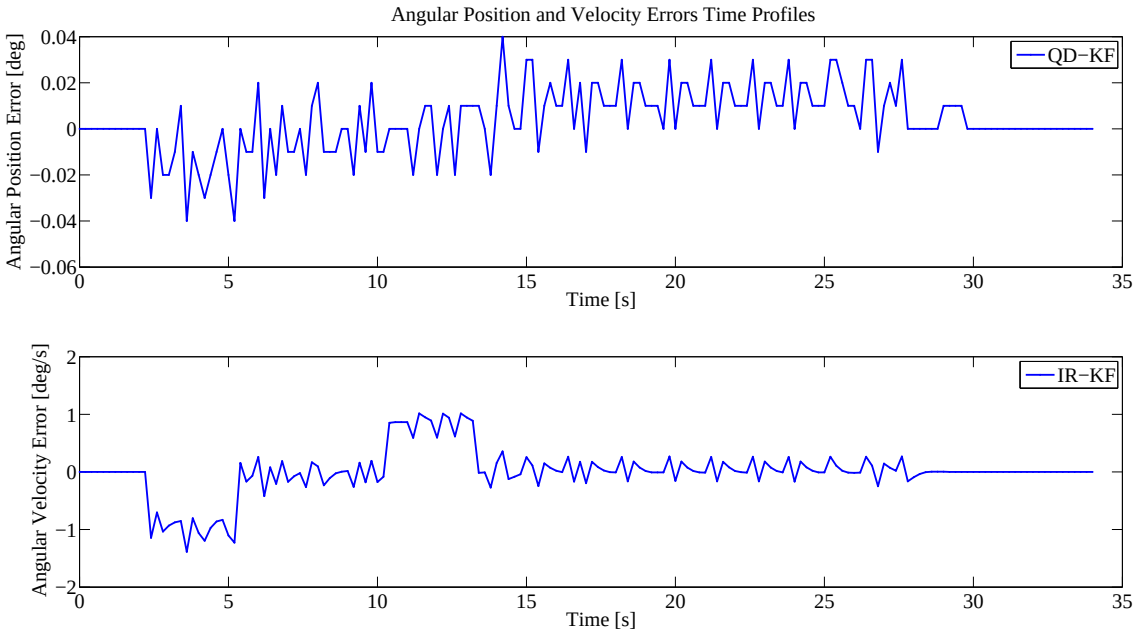


Figure 8.31: Angular position (top) and velocity (bottom) error time profiles – Test 1 experimental results.

8.6.2 Fixed Set Point Reaching

The main purpose of this second test is to evaluate MPC capabilities in making the AM to execute a slew maneuver about its z_B axis. In particular, starting from an initial angle $\theta_i = 0°$, the AM has to reach a given final target angular position $\theta_f = -170°$.

To overcome the main drawback of MPC, i.e. its on-line computation effort, and being able to implement a MPC scheme in the micro-controller on board the AM, the computational cost was moved off-line following the strategy proposed by Bemporad et al. in [72] and described in Section 3.4.6.

The attitude dynamics of the AM about its z_B axis was modeled with the following discrete-time LTI state space model:

$$x_{k+1} = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.01429 \\ 0.1429 \end{bmatrix} u_k \tag{8.28}$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k \tag{8.29}$$

where $x = [\theta - \theta_f, \omega]^T$ is the state vector, $u = \tau$ is the control torque and $y = \theta - \theta_f$ is the system output. The discrete-time model was obtained from a continuous-time double integrator model using a zero-order hold approach[7] with a control sampling time interval $\Delta t_{ctrl} = 0.2\,s$. In this problem, the vector of parameters ϑ coincides with the state vector x, i.e. $\vartheta = x = [\theta - \theta_f, \omega]^T$. As described in Section 8.6.1, a linear Kalman Filter was used to estimate the current angular velocity from yaw angle QD measurements.

An explicit MPC controller was obtained using the Hybrid Toolbox for Matlab by Bemporad [91][8] by defining the problem as a constrained regulation problem to the origin with the following parameters:

$$m = 3 \quad \text{(control horizon)} \tag{8.30}$$

$$p = 5 \quad \text{(prediction horizon)} \tag{8.31}$$

[7] Each control component is kept constant within each control time interval, resulting in Piece Wise Constant (PWC) control profiles.

[8] The Multi-Parametric Toolbox [92] can be used for the same purpose.

$$R = 1 \tag{8.32}$$

$$Q = \begin{bmatrix} 0.2 & 0 \\ 0 & 0 \end{bmatrix} \tag{8.33}$$

The weighting matrix P on the terminal state error was obtained as the solution of the associated Riccati equation. The following constraint on the control action was imposed: $-0.2161\,N\,m \leq \tau \leq 0.2161\,N\,m$. Also, the explicit control law was computed for $\theta \in [-\pi, \pi]\,rad$ and $\omega \in [-\pi/3, \pi/6]\,rad/s$.

With the previous parameters, the state space is divided into 13 polyhedral regions as depicted in Figure 8.32. Each state space region is represented by the matrix pair (H^i, k^i), with $H^i \in \mathbb{R}^{36x2}$, $k^i \in \mathbb{R}^{36x1}$, $i = 1, ..., 13$, through the matrix inequality $H^i\,\vartheta \leq k^i$, and the corresponding control sequence $U = [u^T(0),\, u^T(1),\, u^T(2)]^T$ is computed as $U(\vartheta) = F^i\vartheta + g^i$ with $F^i \in \mathbb{R}^{13x2}$ and $g^i \in \mathbb{R}^{13x1}$. Figure 8.33 shows the control torque for each state space point in terms of both a surface (panel on the left) and a contour plot (panel on the right). The on-line computation consists on a simple table-lookup: for a given state of the system, the region to which that state belongs is first determined, and the corresponding control sequence is then computed.

Matlab simulation and real system results are compared in Figures from 8.34 to 8.36. In particular, Figure 8.34 shows angular position and velocity time profiles in dashed red line for the Matlab simulation and in continuous blue line for the real system. The final target angular position is reached with an error less than 0.5° in $20\,s$. Also, as can be seen from this figure, in Matlab simulation we decided to set the MPC parameters to obtain an angular position time profile with a little overshoot in order to compensate the friction effect of the real system. Figure 8.35 represents the control torque time profile (top) and the corresponding MPC region number (bottom). The Matlab simulation and the real system control torque time profiles have the same shape and the real system results are more damped than the Matlab simulation results because of the friction effects. To conclude, Figure 8.36 plots the state space polyhedral partition and the state space trajectories in dashed red line for the Matlab simulation and in continuous blue line for the real system.

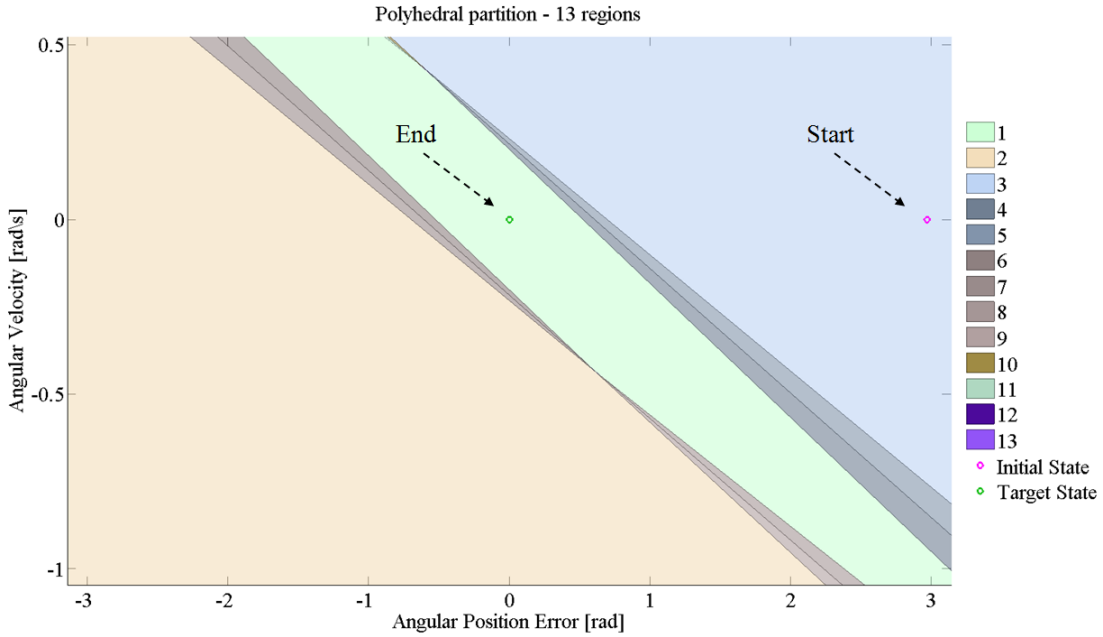


Figure 8.32: State space polyhedral partition into 13 regions with initial state in magenta circle and final target state in green circle – Test 2.

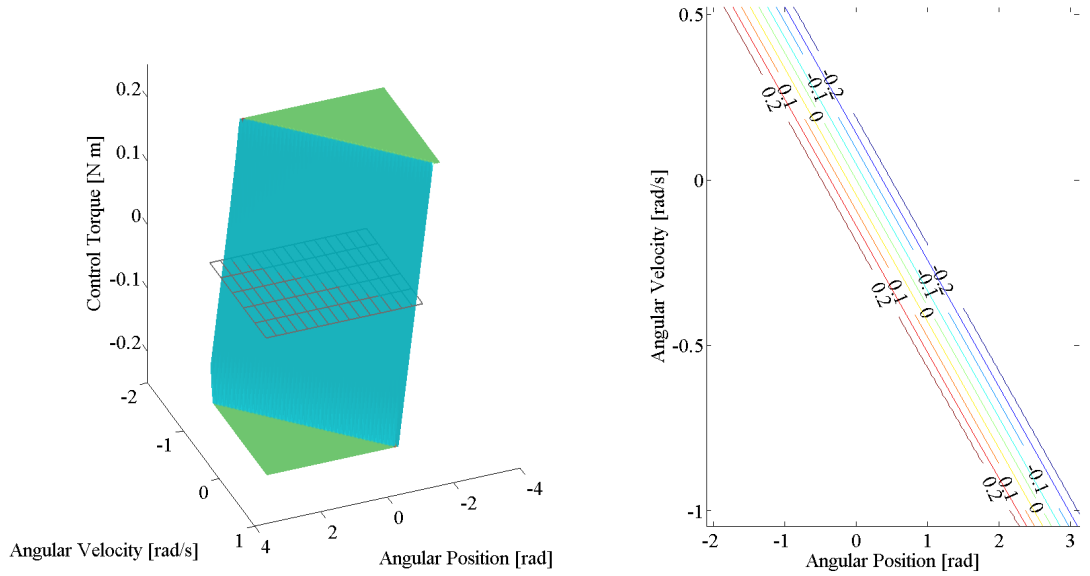


Figure 8.33: Explicit control torque for each state space point in terms of both a surface (panel on the left with $\tau = 0$ plane) and a contour plot (panel on the right) – Test 2.

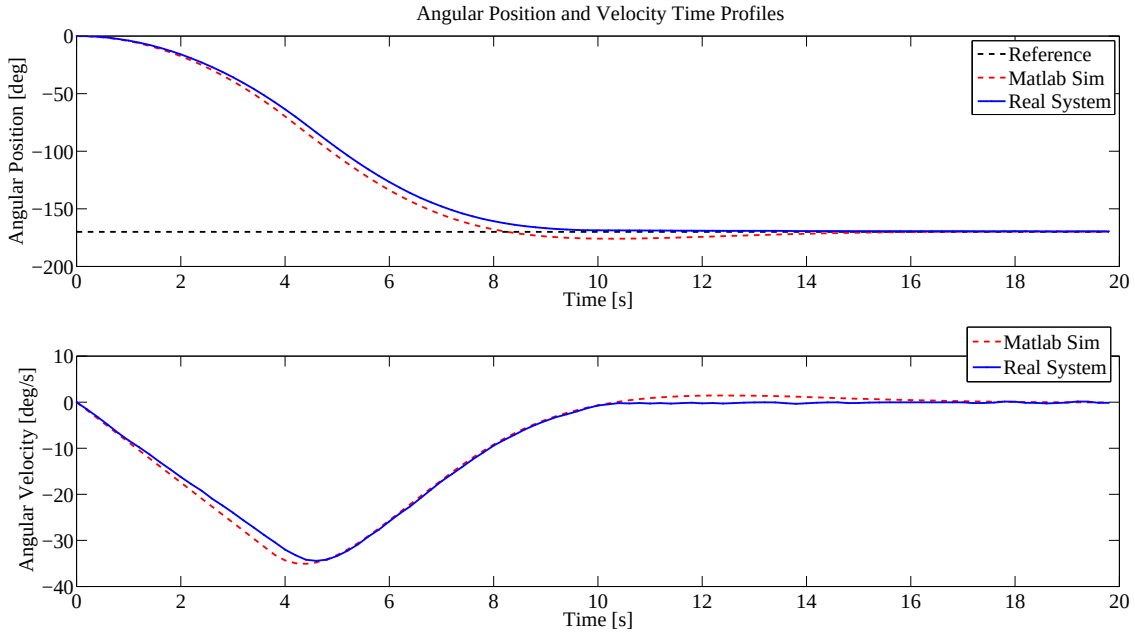


Figure 8.34: Angular position (top) and velocity (bottom) time profiles – Matlab Simulator vs. Real System comparison – Test 2.

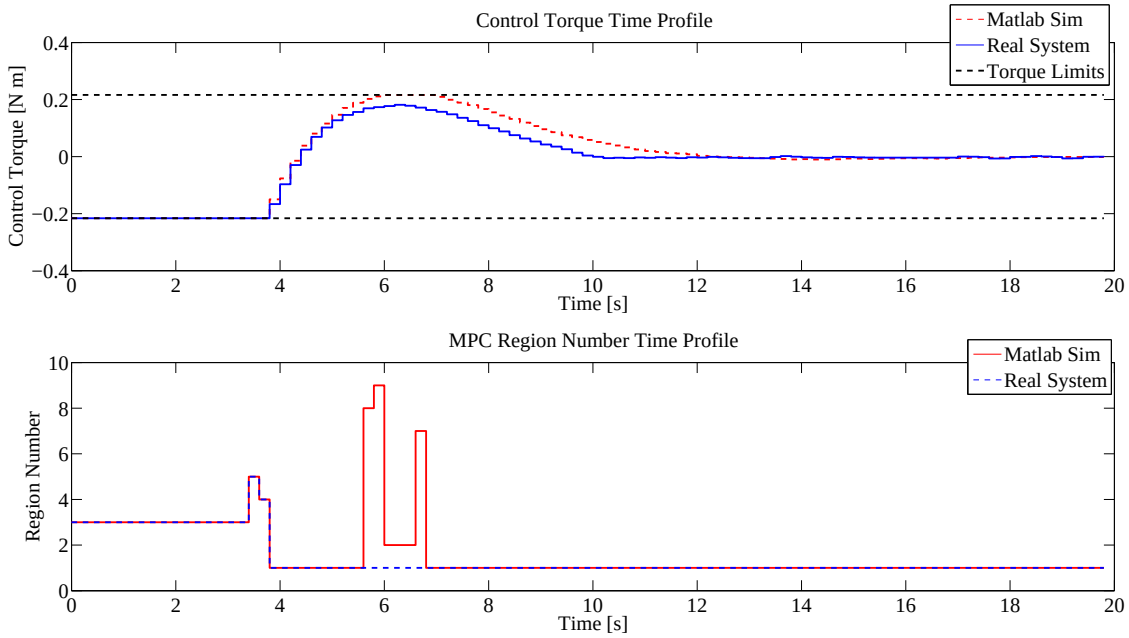


Figure 8.35: Control torque (top) and MPC region number (bottom) time profiles – Matlab Simulator vs. Real System comparison – Test 2.

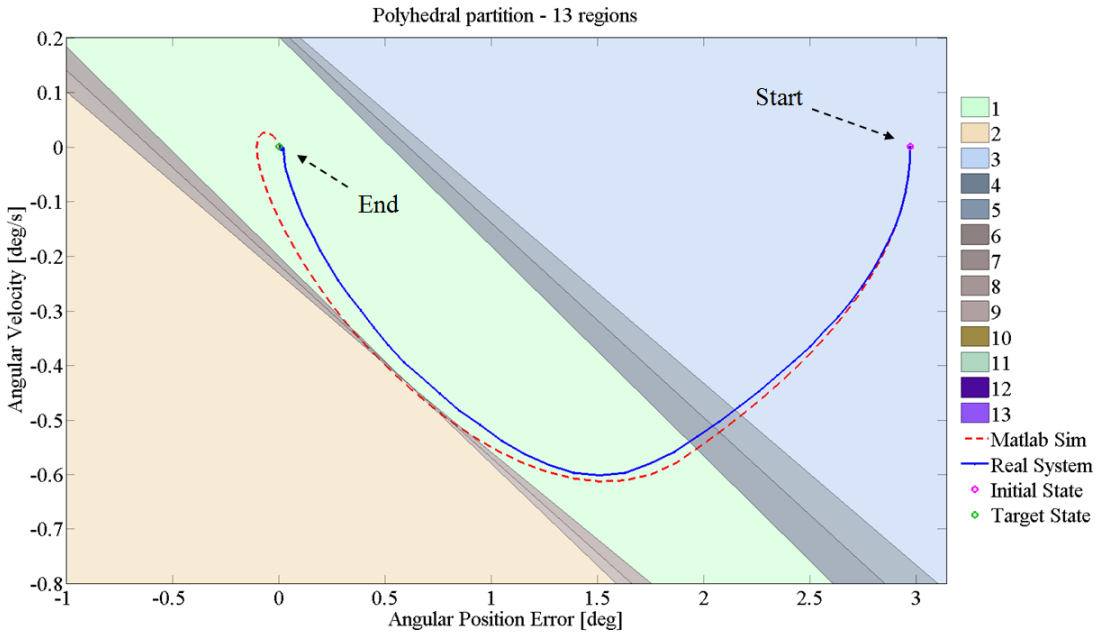


Figure 8.36: State space trajectory with MPC polyhedral partition – Matlab Simulator vs. Real System comparison – Test 2.

8.6.3 Time-Varying Trajectory Tracking

The aim of this third test is to simulate the following coarse repointing maneuver: a spacecraft with its $\mathbf{x}_B$ axis pointing the initial direction $\hat{\mathbf{u}}_i$ at time $t_i = 0$ has to be maneuvered to get its $\mathbf{x}_B$ axis aligned along the final direction $\hat{\mathbf{u}}_f$ at time $t_f = 30\,s$, minimizing the propellant consumption. The $\mathbf{x}_B$ axis could represent the optical axis of an on board instrument and during the attitude maneuver it has to avoid a set of five pre-defined keep-out-cones, which are represented by an axial direction $\mathbf{c}_j$ and a vertex angle $2\beta_j$ (see Figure 8.37 and Table 8.7).

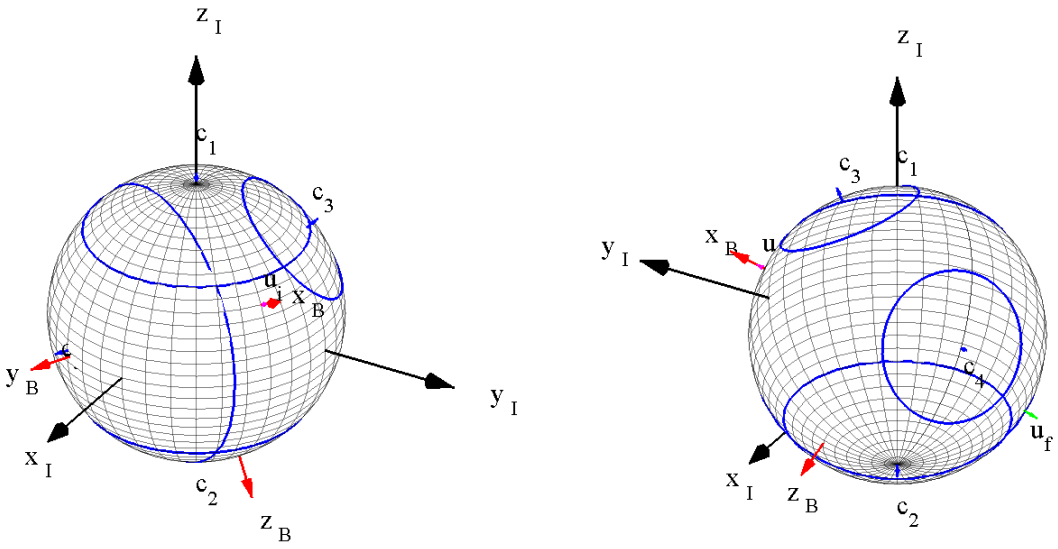


Figure 8.37: 3D view of the coarse repointing maneuver with Inertial reference frame in black, Body reference frame in red, keep-out-cones in blue, initial direction $\mathbf{u}_i$ in magenta and final direction $\mathbf{u}_f$ in green – Test 3.

Table 8.7: Repointing maneuver parameters – Test 3.

	$\phi\,[deg]$	$\lambda\,[deg]$
$\hat{\mathbf{u}}_i$	30	60
$\hat{\mathbf{u}}_f$	330	250

	$\phi\,[deg]$	$\lambda\,[deg]$	$\beta\,[deg]$
$\hat{\mathbf{c}}_1$	180	90	50
$\hat{\mathbf{c}}_2$	180	-90	50
$\hat{\mathbf{c}}_3$	120	42	30
$\hat{\mathbf{c}}_4$	180	328	31
$\hat{\mathbf{c}}_5$	330	0	75

This attitude repointing maneuver can be formulated as the following optimal control problem. Find the control trajectory $\mathbf{u}(t) = [\tau_x, \tau_y, \tau_z]^T(t)$ and the state trajectory $\mathbf{x}(t) = [\mathbf{q}^T, \boldsymbol{\omega}^T]^T(t)$, with $t = [t_i, t_f]$, to minimize the following cost function:

$$J = \int_{t_i}^{t_f} \|\mathbf{u}(\tau)\|_2^2 \, d\tau \tag{8.34}$$

subject to the differential constraints:

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\,\mathbf{T}(\mathbf{q}(t))\,\mathbf{q}(t) \tag{8.35}$$

$$\dot{\boldsymbol{\omega}}(t) = \mathbf{J}^{-1}\left[\mathbf{u}(t) - \boldsymbol{\omega} \times \mathbf{J}\,\boldsymbol{\omega}\right] \tag{8.36}$$

the path constraints:

$$-1 \leq \mathbf{x}_B \cdot \hat{\mathbf{c}}_j \leq cos(\beta_j),\ j = 1, ..., 5 \tag{8.37}$$

and the boundary conditions:

$$\mathbf{q}(t_i) = \mathbf{q}_i \tag{8.38}$$

$$\boldsymbol{\omega}(t_i) = \mathbf{0} \tag{8.39}$$

$$\mathbf{x}_B(t_f) = \mathbf{x}_{B,f} \tag{8.40}$$

$$\boldsymbol{\omega}(t_f) = \mathbf{0} \tag{8.41}$$

where $\mathbf{q} = [q_1, q_2, q_3, q_4]^T(t)$ is the quaternion vector, $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T(t)$ is the spacecraft angular velocity relative to the inertial reference frame and expressed in the body frame, $\mathbf{T}(\mathbf{q}(t))$ is given by:

$$\mathbf{T}(\mathbf{q}(t)) = \begin{bmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & -q_4 & q_3 \\ q_4 & q_1 & -q_2 \\ -q_3 & q_2 & q_1 \end{bmatrix}(t) \tag{8.42}$$

and $\mathbf{x}_B(q(t))$ is the x Body axis of the spacecraft. We assumed $\mathbf{J} = 1.37\,kg\,m^2\,\mathbf{I}_{3\text{x}3}$ as the inertia matrix of the spacecraft (see below). Repointing maneuver parameters are listed in Table 8.7. The initial attitude is given by $\mathbf{q}_i = [0.1477, -0.8335, -0.4877, -0.2135]^T$.

The optimal control problem associated to the attitude repointing maneuver was solved using

the C++ software package PSOPT[9] [93, 94] with a Legendre collocation method, a nonlinear problem tolerance of 10^{-5} and an ordinary differential equations tolerance of 10^{-4}. The optimal control problem solution, with a 65 collocation points final mesh, is showed in Figures from 8.38 to 8.40 in terms of quaternion, angular velocity and control torque components time profiles. Figure 8.41 shows a polar diagram, in terms of azimuth and elevation angles with respect to the inertial reference frame, with keep-out-cones in cyan areas and spacecraft $\mathbf{x}_B$, $\mathbf{y}_B$ and $\mathbf{z}_B$ axes trajectories in continuous red, green and blue lines, respectively. A 3D visualization of the Body axes reference trajectories is provided in Figure 8.42. As you can see from these figures, the $\mathbf{x}_B$ trajectory (the red line) keeps outside the keep-out zones reaching its final position in 30 s.

In order to simulate part of this coarse repointing maneuver using the 1 DOF hardware attitude simulator, the reference attitude position, obtained from PSOPT in terms of quaternion components time profiles, was converted into Euler angles time profiles with a roll-pitch-yaw sequence. The yaw angle time profile and the z component of the angular velocity time profile were then used as angular position and velocity reference trajectories, respectively, and uploaded into the on board micro-controller. Figure 8.43 presents these reference trajectories with the z component of the control torque in blue. This figure shows also in magenta the electro-valve firing time associated to the reference control profile with the nominal minimum firing time in black (16 ms). The duty-cycle of the corresponding PWM action ranges from 0% to a maximum value of 35%.

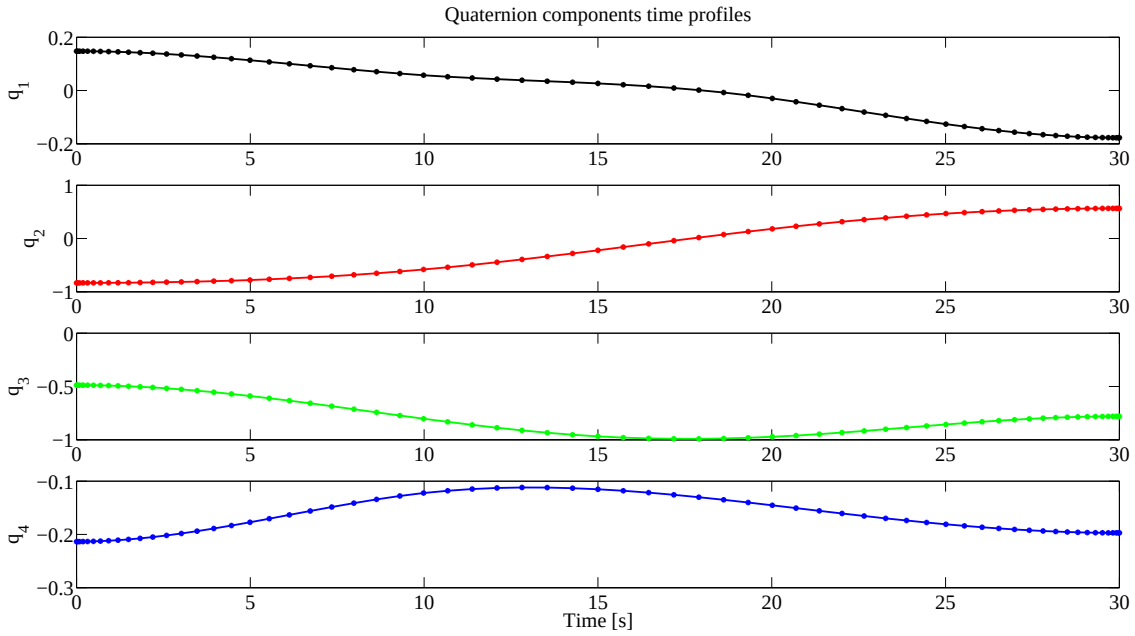


Figure 8.38: Quaternion components time profiles – Test 3.

[9]Available at: http://code.google.com/p/psopt/ downloads/list.

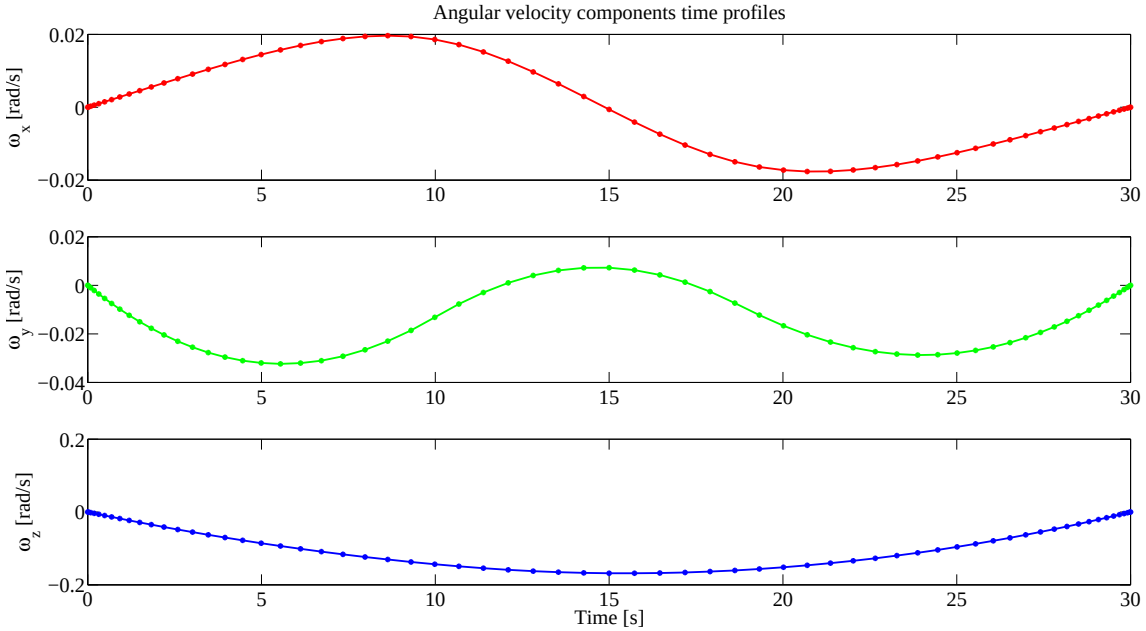


Figure 8.39: Angular velocity components time profile – Test 3.

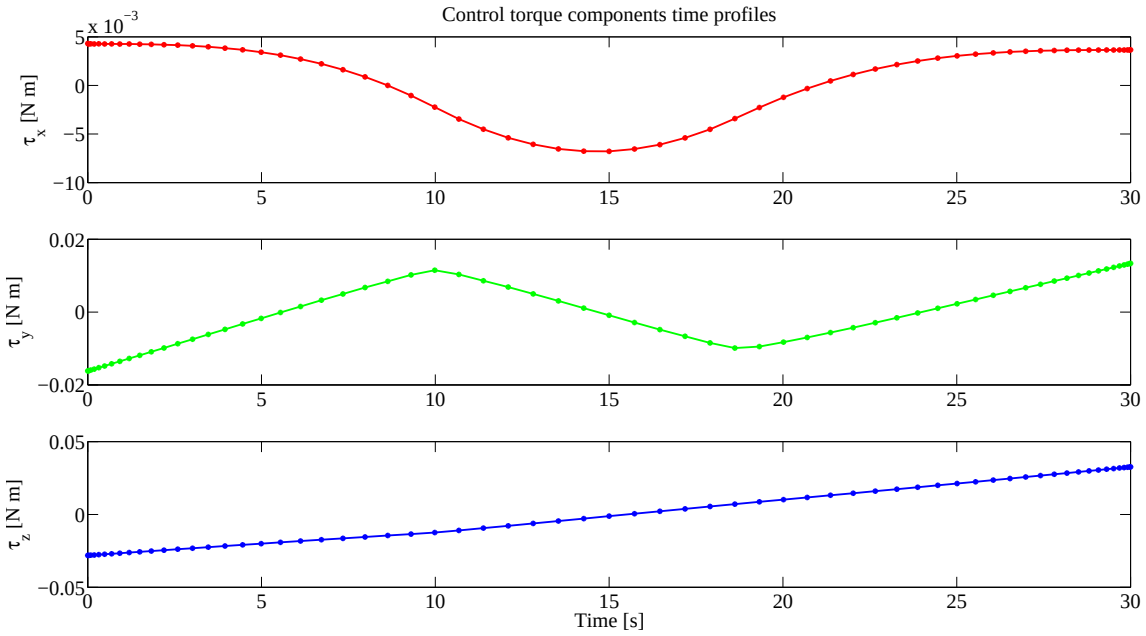


Figure 8.40: Control torque components time profiles. All control torque components are equal to zero for $t > 30\,s$ – Test 3.

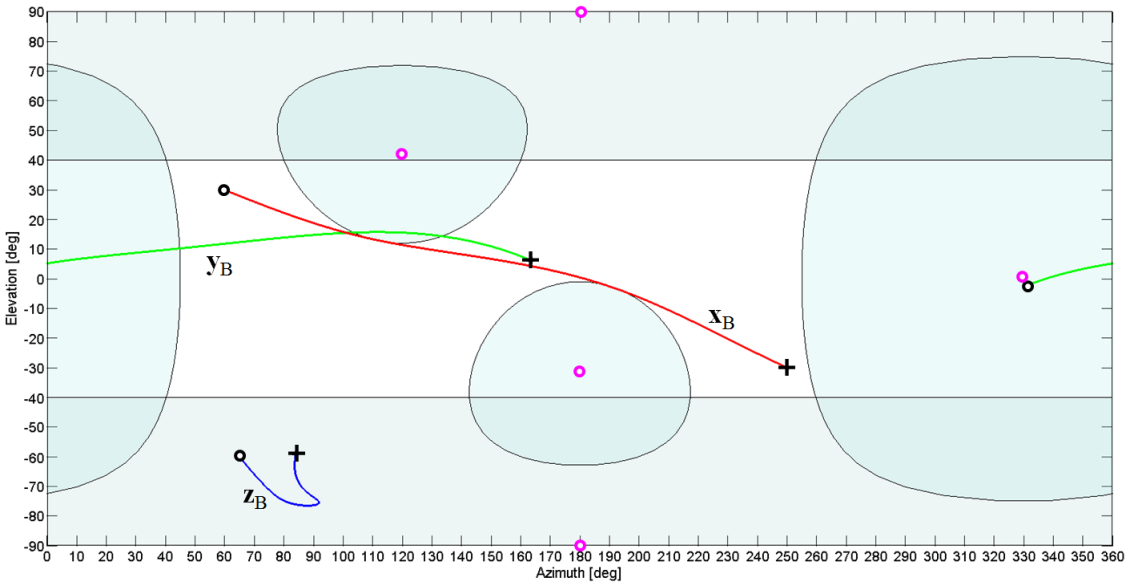


Figure 8.41: Polar diagram showing keep-out-cones in cyan, $\mathbf{x}_B$ trajectory in red line, $\mathbf{y}_B$ trajectory in green line and $\mathbf{z}_B$ trajectory in blue line. Initial and final position is represented with black circles and crosses, respectively. Keep-out-cones central axes, i.e. $\hat{\mathbf{c}}_j$ directions, are marked with magenta circles – Test 3.

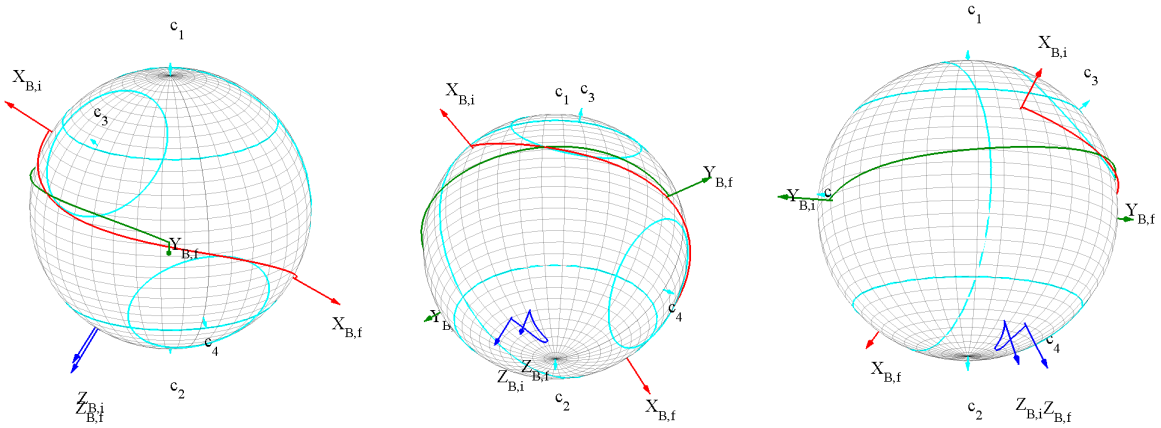


Figure 8.42: 3D reference trajectories for $\mathbf{x}_B$ axis in red line, $\mathbf{y}_B$ axis in green line and $\mathbf{z}_B$ axis in blue line. Keep-out-cones are plotted in cyan – Test 3.

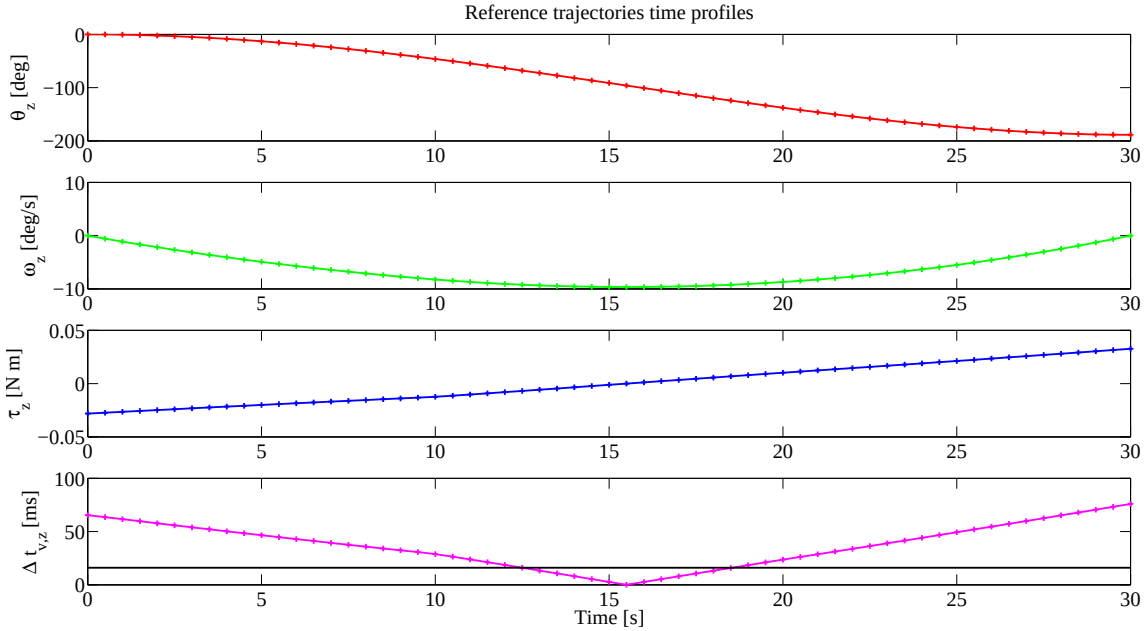


Figure 8.43: Reference trajectories time profiles for angular position θ_z, angular velocity ω_z, control torque τ_z and electro-valve firing time $\Delta t_{v,z}$. $\tau_z = 0$ for $t > 30\,s$ – Test 3.

8.6.3.1 PID control

To track the reference trajectories computed with pseudospectral methods, we first used the discrete-time PID control law described in Section 8.4 with the following parameters: $\Delta t_{ctrl} = 0.5\,s$, $K_p = 1.4$, $K_i = 0.45$, $K_d = 1.6$, $K_t = 0.11$, $N = 10$ and $\beta = \gamma = 1$. The yaw angle was not estimated with a Kalman Filter, but it was taken directly from the QD.

Test results are presented in Figures 8.44 and 8.45 in terms of yaw angular position and velocity and trajectory tracking errors. In dynamic conditions, the reference angular position is tracked with an error $|\theta_{estim} - \theta_{ref}| < 3°$, while the reference angular velocity with an error $|\omega_{estim} - \omega_{ref}| < 1.5°/s$. The final target angular position is reached with an error less than the QD resolution in $35\,s$. The trajectory tracking errors, both in position and velocity, have an higher oscillation in the first part of the maneuver, while in the second part ($t > 17\,s$) they converge faster to zero. These initial oscillations are mainly due to the static friction at the yaw joint.

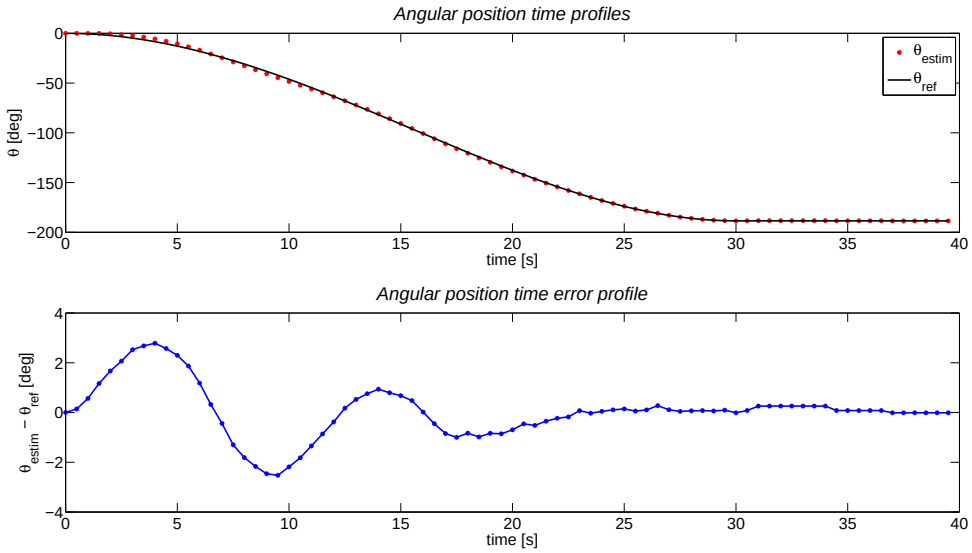


Figure 8.44: Time profiles of: (top) estimated angular position (red) and reference angular position (black); (bottom) angular position error (blue) – Test 3.

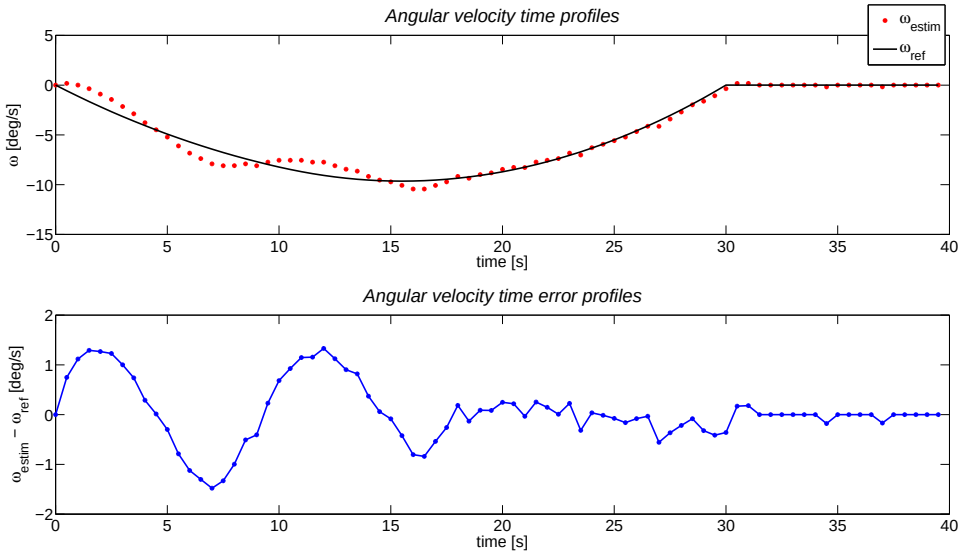


Figure 8.45: Time profiles of: (top) estimated angular velocity (red) and reference angular velocity (black); (bottom) angular velocity error (blue) – Test 3.

8.6.3.2 MPC control

An explicit MPC control law was obtained using the Hybrid Toolbox for Matlab by defining the problem as a constrained state trajectory tracking problem. In this final test, we considered the following discrete-time LTI state space model with a control sampling time interval $\Delta t_{ctrl} = 0.2\,s$:

$$x_{k+1} = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.01429 \\ 0.1429 \end{bmatrix} u_k \tag{8.43}$$

$$y_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k \tag{8.44}$$

where $x = [\theta, \omega]^T$ is the state vector, $u = \tau$ is the control torque and $y = x$ is the output. In this problem, the vector of MPC parameters ϑ at time t_k is given by $\vartheta_k = [\theta_k, \omega_k, \tau_{k-1}, \theta_{ref,k}, \omega_{ref,k}]^T$ and the vector to be optimized is the sequence of control action increments within the control period, i.e. $[\Delta u^T(0), ..., \Delta u^T(m-1)]^T$. As described in Section 8.6.1, a linear Kalman Filter was used to estimate the current angular velocity from yaw angle QD measurements. We imposed a constraint on the maximum control torque $-0.2161\,N\,m \leq \tau \leq 0.2161\,N\,m$ and we adopted the following MPC parameters:

$$m = 3 \quad \text{(control horizon)} \tag{8.45}$$

$$p = 5 \quad \text{(prediction horizon)} \tag{8.46}$$

$$R = 5 \cdot 10^{-2} \tag{8.47}$$

$$Q = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.45 \end{bmatrix} \tag{8.48}$$

With these parameters, the MPC parameters space $\Theta \subseteq \mathbb{R}^5$ is divided into 15 polyhedral regions as depicted in Figures 8.46 - 8.47. In particular, in Figure 8.46 the polyhedral partition is projected on the 2D plane $\theta - \omega$ assuming the other 3 components of the MPC parameters vector equal to zero, i.e. $u_{k-1} = \theta_{ref} = \omega_{ref} = 0$. Figure 8.47 represents a 3D section of the polyhedral partition obtained imposing $\theta_{ref} = \omega_{ref} = 0$.

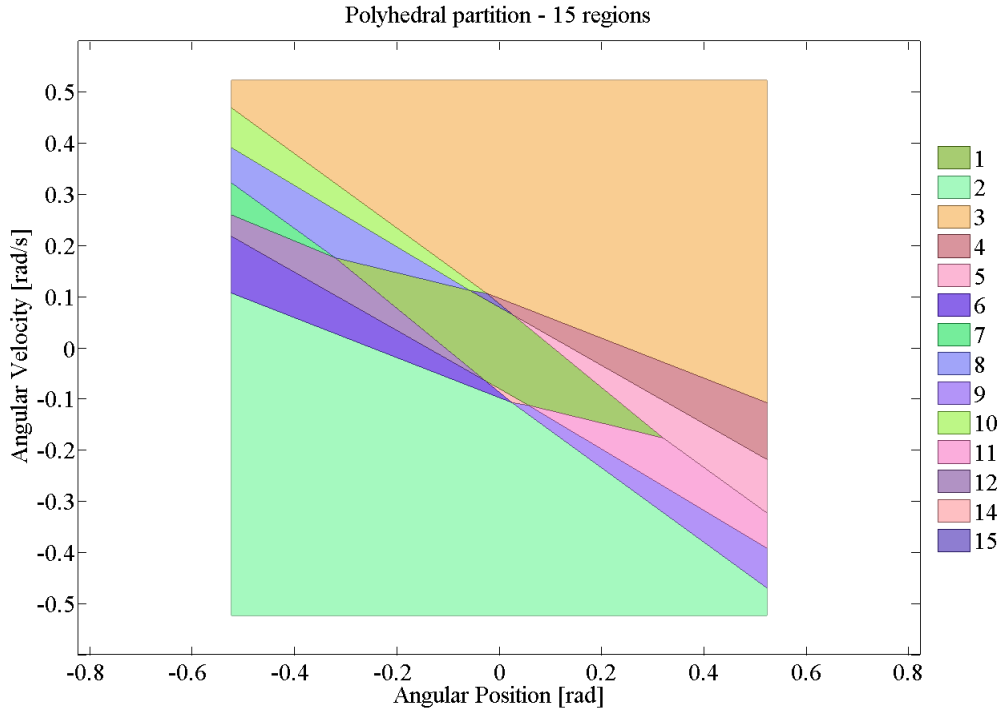


Figure 8.46: 2D MPC control law polyhedral partition with 15 regions projected on the state space $\theta - \omega$ with $u_{k-1} = \theta_{ref} = \omega_{ref} = 0$ – Test 3.

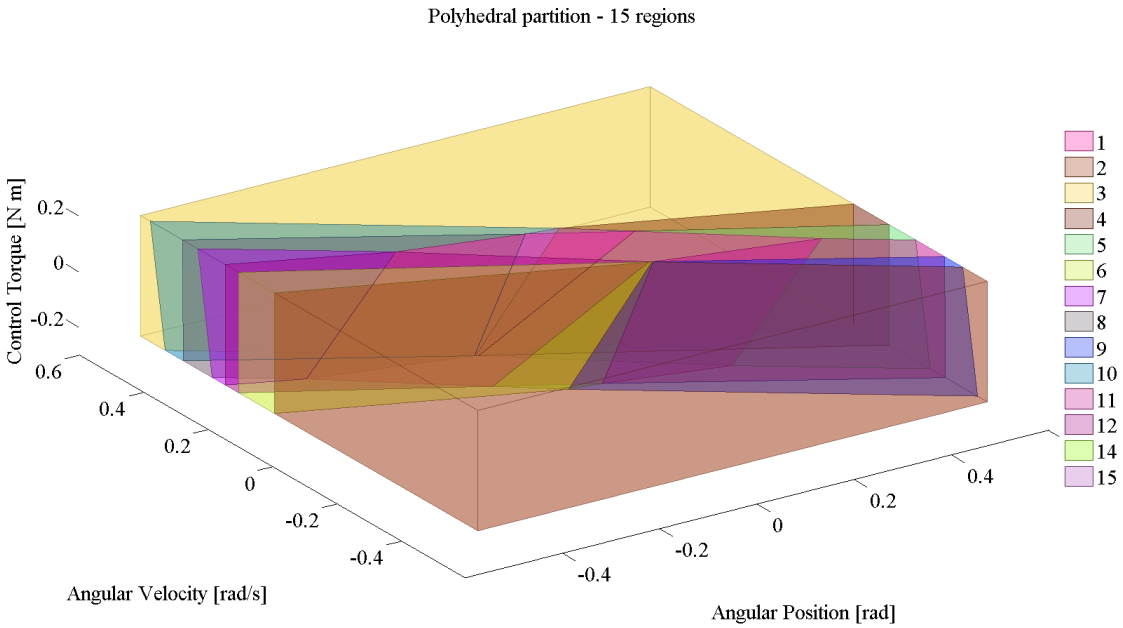


Figure 8.47: 3D MPC control law polyhedral partition with 15 regions projected on the space $\theta - \omega - u_{k-1}$ with $\theta_{ref} = \omega_{ref} = 0$ – Test 3.

Matlab simulation and real system results are compared in Figures 8.48 - 8.52. Figure 8.48 shows the time profiles of the angular position (top) and velocity (bottom) in dashed red line for the Matlab simulation and in continuous blue line for the real system, and the corresponding reference time profile in continuous black line. Figure 8.49 plots the same results in terms of state space trajectories. In Figures 8.50 and 8.51 it can be noted that:

- the reference angular position profile is tracked with an error $|\theta_{estim} - \theta_{ref}|$ that is less than $1.5°$ in dynamic conditions and less than $0.1°$ (estimation accuracy) at steady state;
- the angular velocity profile is tracked with an error $|\omega_{estim} - \omega_{ref}|$ that is less than $1°/s$ in dynamic conditions and less than $0.1°/s$ (estimation accuracy) at steady state.

Figure 8.52 represents the control torque time profile in dashed red line for the Matlab simulation and in dashed blue line for the real system, with the reference profile in continuous black line. The deviation of both the Matlab simulation and the real system behaviors from the reference one can be due to the following reasons:

- the control action computation is based on an estimation of the current attitude kinematic state of the system and not on the true state;
- the control action is actuated according to a Pulse Width Modulation (PWM) logic with a duty-cycle less than 35% and not as a Piece Wise Constant (PWC) profile;
- in the real system we have the contribution of the yaw joint friction that is not considered[10] neither in the Matlab simulator nor in the model used to compute the optimal reference trajectories.

These test results confirm that the MPC controller is able to overcome these kind of not-modeled disturbances (robustness), allowing to track the reference trajectory with a more than acceptable error for a coarse repointing maneuver.

After this first coarse phase, a fine attitude acquisition maneuver may start. This kind of fine maneuver has to be performed using both an instrumental setup and a control system, i.e. control law and actuator setup (e.g. a reaction wheel bench), that allow to estimate and control the attitude, respectively, with an higher accuracy. These types of systems generally have tighter operative ranges, e.g. limited control authority and thus slower maneuver time, and require higher on board resources in terms of computing power and therefore electric power.

[10]The friction at joints is very difficult to estimate experimentally since it strongly depends on temperature and humidity conditions, which are not kept within predefined intervals.

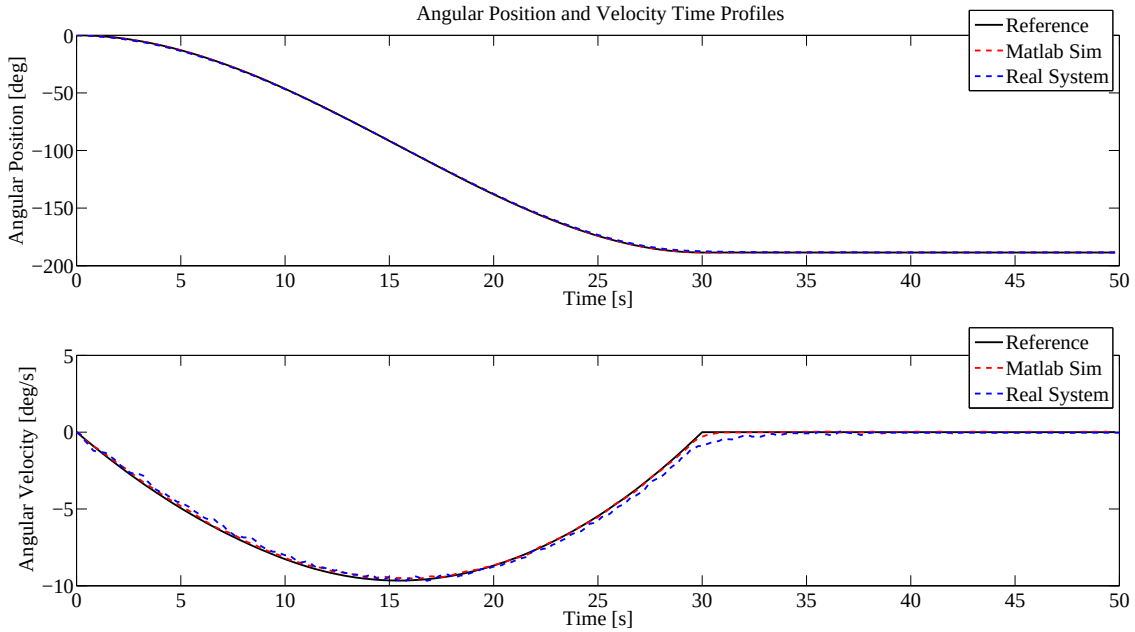


Figure 8.48: Angular position (top) and velocity (bottom) time profiles – Reference, Matlab Simulator and Real System comparison – Test 3.

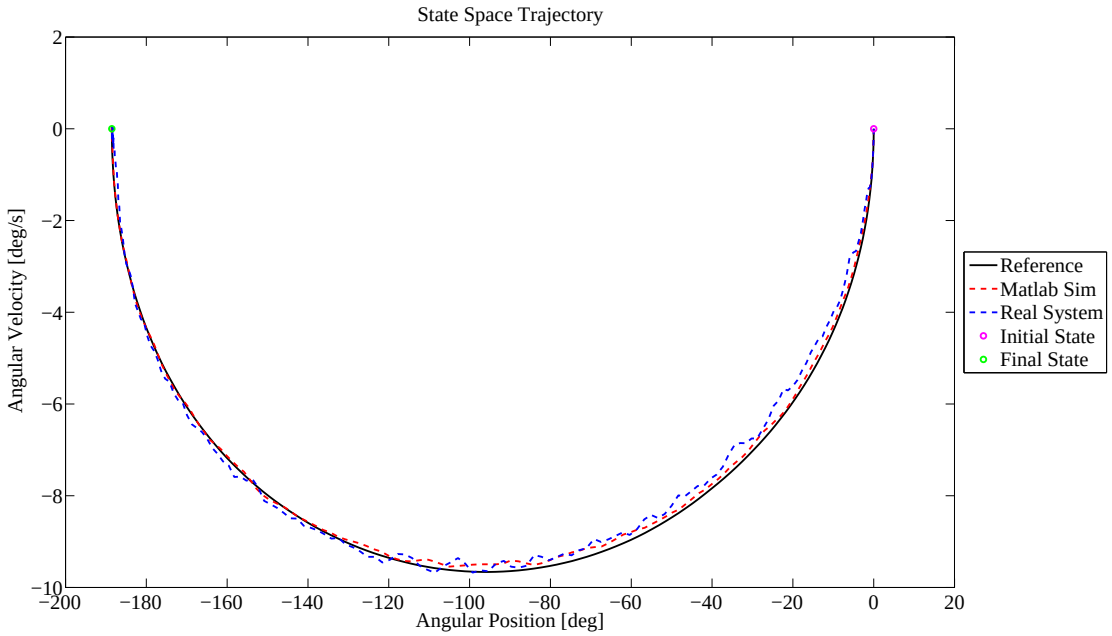


Figure 8.49: State space trajectory in dashed red line for the Matlab simulation, in dashed blue line for the real system and in continuous black line for the reference – Test 3.

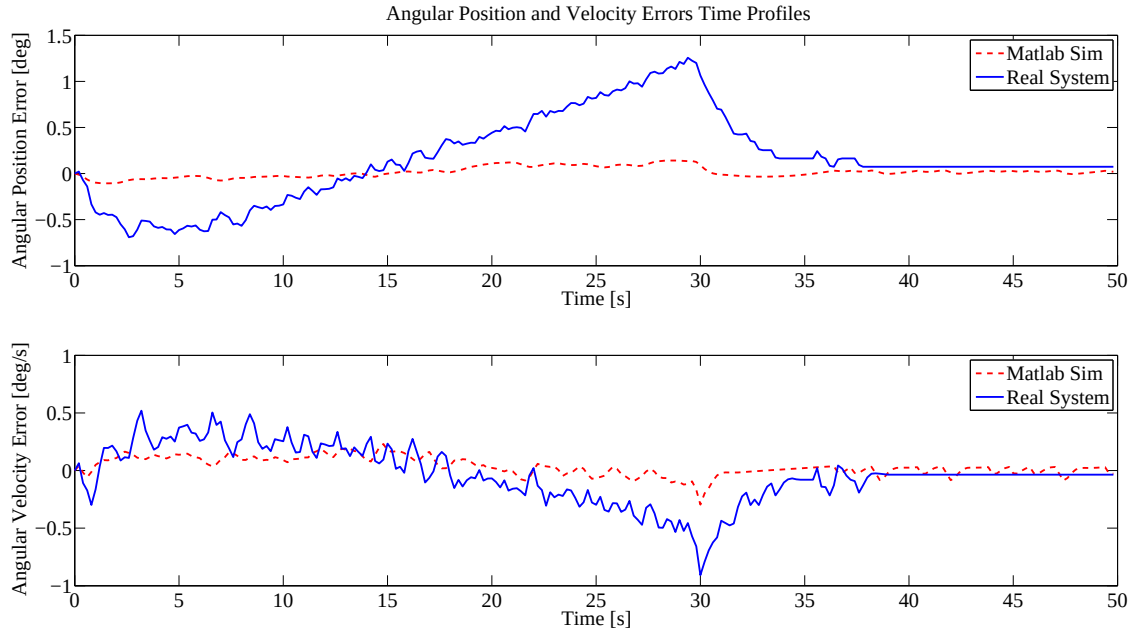


Figure 8.50: Angular position (top) and velocity (bottom) errors time profiles – Matlab Simulator and Real System comparison – Test 3.

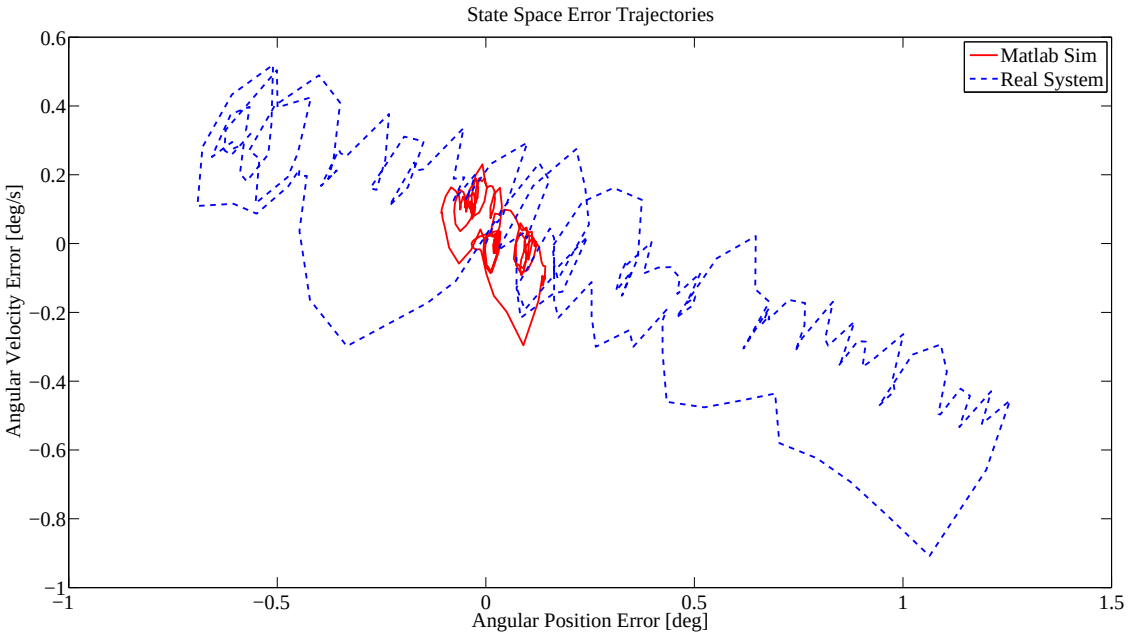


Figure 8.51: State space error trajectory – Reference, Matlab Simulator and Real System – Test 3.

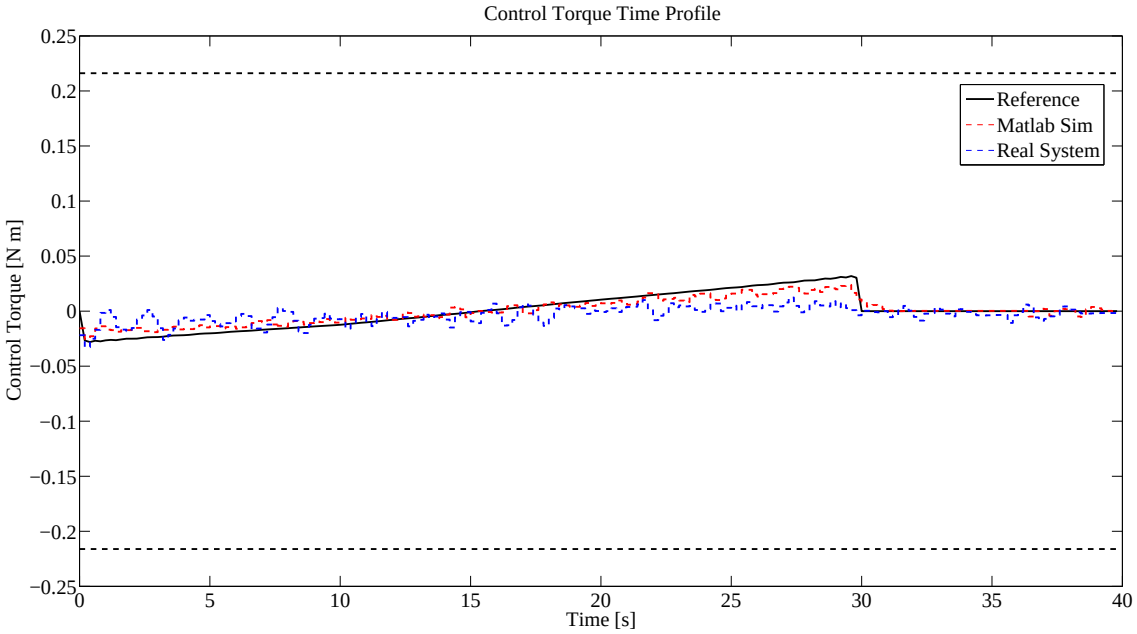


Figure 8.52: Control torque time profiles – Reference, Matlab Simulator and Real System – Test 3.

8.7 Future Activities and Perspectives

In this period our team is working to complete the AM characterization an the on board software development for a 3 DOF motion, in order to execute some tests on 3 DOF attitude estimation (a quaternion-based Kalman Filter will probably be used) and 3 DOF attitude control strategies.

Two main milestones for the SFF test bed improvement have been identified. The first one consists on designing, building and testing of the TM with definition of the interface requirements between the AM and the TM. This milestone requires a preliminary experimental activity whose main tasks are: (a) to design the skids that allow to create the air cushion between the TM and a glass table; and (b) to determine the minimum pressure and the mass flow of the air at the skid inlet that guarantee a good (low friction) movement of the Simulators on the glass table. The last activity also provides useful data to design the on board pneumatic system that feeds the air cushion skids.

The second milestone concerns the development and testing of a 2D positioning system that allows to identify the Simulators position on the glass table. The following two possible systems has been taken into account.

1. An ultra-sound (US) system with external US beacons and US receivers on board each Simulator. Taking measurements of the US signal time of flight it is possible to estimate both the absolute position and the attitude of each Simulator using a GPS-like algorithm. This kind

of system is currently used for example in the MIT SPHERES testbed.

2. An optical mouse like system placed at the TM base [95]. Every optical mouse is indeed equipped with a small size and low cost IC that directly provides information about the mouse 2D displacement. Placing an array of these systems at the TM base edge and properly combining their data it is possible to estimate both the TM motion on the glass table and its rotation about an axis perpendicular to the table (vertical axis).

With respect to the first US system, the last setup is simpler, cheaper, the estimate performance does not depends on temperature/humidity conditions but is dependent on lighting and focusing conditions (the last condition is directly dependent on the air cushion thick stability), and only a relative position is provided.

At the moment the team is oriented toward the second solution. In order to realize suck a system, some preliminary design and experimental activities are required with the aim of identifying: (a) the components of each mouse-like unit (a mouse IC, a lens and some proximity electronics); (b) the estimation accuracy achievable with each single unit as a function of IC pixel number, lighting and focusing conditions for a set of imaging surface type; (c) the number and the position of these mouse-like units to improve the TM 2D positioning accuracy and also to provide its rotation about the vertical axis. A final experimental activity will be then carried out to verify and eventually to calibrate the whole system. This activity will consist on the comparison between the 3 DOF estimation provided by the mouse-like system and the one provided by a more accurate system.

With a system of two or more Simulators each with 5 DOF, many formation flight and rendezvous and docking scenarios can be tested, both for research and didactic purposes. The SFF team has identified the following first complete maneuver to be executed with the SFF testbed: rendezvous and docking maneuver with an external docking mechanism representing a reference target and a controlled at least 3 DOF Simulator with an on board compatible docking mechanism and a stereo camera system used to estimate the relative dynamic state with respect to the target.

Chapter 9

CONCLUSIONS

The final aim of my Ph.D. activities was to develop and test modern optimal control strategies for Spacecraft Formation Flying applications. This task was accomplished by means of both computer simulations and experimental tests conducted on both the MIT SPHERES testbed and the SFF hardware simulator that we are developing at the University of Padova.

Satellite Formation Flying is a key technology for several future missions, since, with respect to a single spacecraft, it allows better performances, new capabilities, more flexibility and robustness to failure and cost reduction. Despite these benefits, however, the SFF concept poses several significant design challenges and requires new technologies. The Guidance, Navigation and Control system is a key element in the SFF concept and it must be reliable in coordinating all the satellites flying in formation and at the same time efficient in using the limited on board resources. Model Predictive Control (MPC), also referred to as Receding Horizon Control, is a modern optimal control technique that seems to be suitable for these purposes because of its three main features: model-based control scheme, constraints handling ability and replanning nature.

MPC capabilities were first tested in computer simulations in carrying out a formation acquisition maneuver for two space vehicles, taking into account two scenarios: a Leader-Follower (LF) formation and Projected Circular Orbit (PCO) formation. The performances of the MPC-based controller were compared with those of a LQR-based controller in the presence of active constraints on the maximum control acceleration, evaluating also the effects of the gravitational harmonics J_2 and J_3 and atmospheric drag perturbations on the proposed maneuvers. Simulation results showed that, with similar performances in tracking the same reference state trajectory in terms of settling time, the MPC controller can reduce the total Δv requirement by 40 % in the LF formation scenario and by 30 % in the PCO formation scenario. Simulation results also showed that in the perturbed case the MPC controller is more efficient than the LQR controller with just a small reduction (< 1 %) on the performances with respect to the unperturbed scenarios.

The next activity concerned the development of some guidance and control strategies for a Collision-Avoidance scenario in which a free-flying chief spacecraft follows temporary off-nominal conditions and a controlled deputy spacecraft performs a collision avoidance maneuver. The proposed strategy consists on a first Separation Guidance that, using a computationally simple, deterministic and closed-form algorithm, takes charge of avoiding a predicted collision. When some safe conditions on the relative state vector are met, a subsequent Nominal Guidance takes over. Genetic Algorithms are used to compute a pair of reference state trajectories in order to place the deputy spacecraft in a bounded safe or "parking" trajectory, while minimizing the propellant consumption and avoiding the formation evaporation. The performances of a LQR and a MPC in tracking these reference trajectories were compared, and simulation results showed that a MPC controller reduces the total Δv requirement by $5 - 10\,\%$ with respect to a LQR.

MPC capabilities were also evaluated on the MIT SPHERES testbed in simulating the close-proximity phase of the rendez-vous and capture maneuver for the Mars Orbital Sample Return (MOSR) scenario. We first performed a comparison between a MPC and a PD controller in executing this maneuver both in a Matlab simulator and in the MIT SPHERES software simulator, which is a more representative environment in terms of SPHERES properties and sensors/actuators noises. These preliminary simulation results confirmed the better performances of MPC with respect to PD, with a reduction of the total Δv requirement by $10 - 15\,\%$. The proposed control strategy was then tested in the SPHERES Flat Floor facility at the MIT Space System Laboratory. The on-line MPC computation is performed by an external laptop and then transmitted to the SPHERES satellite for the actuation.

The last part of my research activities was devoted to the SFF hardware simulator of the University of Padova. This experimental testbed will allow to conduct on ground experimental research on both formation flight and rendez-vous and docking. My contributions to this project are summarized as follows:

- conclusion of the designing, building and testing of the five main subsystems of the hardware simulator;
- software development for the hardware simulator, including the software for both the on board micro-controller and the control station, and a Matlab software simulator;
- preparatory experimental activities aimed at determining the hardware simulator on board thrusters force and estimating the hardware simulator moment of inertia about its z-Body axis;
- test of some one degree of freedom attitude control maneuvers with the use of predictive controllers.

In particular, three main tests were carried out. The first one aimed at tuning a Kalman Filter to properly estimate the yaw axis angular velocity using a double-integrator as dynamic model and angular position measurements provided by the yaw quadrature encoder. With the use of a simple Kalman Filter, the yaw angular position and velocity could be estimated with an error less than 0.1° and $0.1^\circ/s$, respectively. In the second test, an explicit MPC was used to perform a 170° slew maneuver of the hardware simulator attitude module about the yaw axis. The final target angular position was reached with an error less than 0.5° in $20\,s$. In the third test, a 3 DOF attitude reference trajectory was first computed using pseudospectral optimization methods for a repointing maneuver with active constraints on the attitude trajectory. The state trajectory was then projected along the z-Body axis and tracked in the hardware simulator using an explicit MPC. In the last two tests, an explicit MPC was run at $5\,Hz$ on a micro-controller with a clock speed of $29\,MHz$. Experimental results of the last test showed that with an explicit MPC the reference trajectories can be tracked with an error less that 1.5° for the angular position and less than $1^\circ/s$ for the angular velocity, both in dynamic conditions. The final target state was reached with an error less than the estimation accuracy.

The SFF Hardware Simulator is a ground-based testbed for the development and verification of GNC algorithms that in the present configuration allows the development and testing of advanced controls for attitude motion and in its final form will enable the derivation of control strategies for Formation Flight and Automated Rendezvous and Docking.

Bibliography

[1] G. W. Hill, "Researches in the Lunar Theory," *American Journal of Mathematics*, vol. 1, no. 1, pp. 5–26, 1878.

[2] W. H. Clohessy and R. S. Wiltshire, "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 1, pp. 653–658, 1960.

[3] J. Leitner, "Formation Flying - The Future of Remote Sensing from Space," Proceedings of the 18th International Symposium on Space Flight Dynamics (ESA SP-548), 11-15 October 2004, Munich, Germany.

[4] D. Scharf, F. Hadaegh, and S. R. Ploen, "A Survey of Spacecraft Formation Flying Guidance and Control (Part I): Guidance," Proceedings of the American Control Conlerence, 4-6 June 2003, Denver, Colorado.

[5] D. P. Scharf, F. Y. Hadaegh, and S. Ploen, "A Survey of Spacecraft Formation Flying Guidance and Control (Part II): Control," Proceedings of 2004 the American Control Conlerence, June 30 - July 2 2004, Boston, Massachusetts.

[6] G. Krieger, M. Zink, M. Bachmann, B. Bräutigam, D. Schulze, M. Martone, P. Rizzoli, U. Steinbrecher, J. W. Antony, F. De Zan, I. Hajnsek, K. Papathanassiou, F. Kugler, M. R. Cassola, M. Younis, S. Baumgartner, P. López-Dekker, P. Prats, and A. Moreira, "TanDEM-X: a radar interferometer with two formation-flying satellites," *Acta Astronautica*, vol. 89, no. 0, pp. 83 – 98, 2013.

[7] J. S. Llorente, A. Agenjo, C. Carrascosa, C. de Negueruela, A. Mestreau-Garreau, A. Cropp, and A. Santovincenzo, "PROBA-3: Precise formation flying demonstration mission," *Acta Astronautica*, vol. 82, pp. 38–46, 2013.

[8] S. Cesare, G. Biondetti, S. Mottini, M. Parisch, G. Sechi, E. Canuto, M. Pisani, M. Aguirre, B. Leone, L. Massotti, and P. Silvestrin, "Satellite-to-Satellite Laser Tracking for a Next Generation Gravimetric Mission," Proceedings of the 3rd International Symposium on Formation

Flying, Missions and Technologies (ESA SP-654), 23-25 April 2008, Noordwijk, The Netherlands.

[9] H. Schaub, "Relative Orbit Geometry Through Classical Orbit Element Differences," *Journal of Guidance, Control and Dynamics*, vol. 27, no. 5, pp. 839–848, 2004.

[10] S. A. Schweighart and R. J.Sedwick , "High-Fidelity Linearized J2 Model for Satellite Formation Flight," *Journal of Guidance, Control and Dynamics*, vol. 25, no. 6, pp. 1073–1080, 2002.

[11] M. Sabatini and G. B. Palmerini, "Linearized formation-flying dynamics in a perturbed orbital environment," in *Aerospace Conference, 2008 IEEE*, pp. 1–13, March 2008.

[12] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems.* American Institute of Aeronautics and Astronautics Inc., 2003.

[13] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, *Spacecraft Formation Flying: Dynamics, control and navigation.* Butterworth-Heinemann, 1st edition 2010.

[14] D. A. Vallado, *Fundamentals of Astrodynamics and Applications.* Microcosm Press/Springer, 3rd edition 2007.

[15] D. F. Lawden, *Optimal Trajectories for Space Navigation.* Butterworths, London, 1963.

[16] J. Tschauner and P. Hempel, "Rendezvous zu eine in elliptischer bahn umlaufenden ziel," *Astronautica Acta*, vol. 11, no. 2, pp. 104–109, 1965.

[17] G. Inalhan, M. Tillerson, and J. P. How, "Relative Dynamics and Control of Spacecraft Formations in Eccentric Orbits," *Journal of Guidance, Control and Dynamics*, vol. 27, no. 9, pp. 48–59, 2002.

[18] S. Gaulocher, J. Chretien, C. Pittet, M. Delpech, and D. Alazard, "Closed-loop Control of Formation Flying Satellites: Time and Parameters varing Framework," 2nd International Symposium on Formation Flying Missions & Technologies, 2004.

[19] K. Yamanaka and F. Ankersen, "New State Transition Matrix for Relative Motion on an Arbitrary Elliptical Orbit," *Journal of Guidance, Control and Dynamics*, vol. 25, no. 1, pp. 60–66, January-February 2002.

[20] R. G. Melton, "Time-Explicit Representation of Relative Motion Between Elliptical Orbits," *Journal of Guidance, Control and Dynamics*, vol. 23, no. 4, pp. 604–610, July-August 2000.

[21] D. Gim and K. Alfriend, "State Transition Matrix of Relative Motion for the Perturbed Noncircular Refernce Orbit," *Journal of Guidance, Control and Dynamics*, vol. 26, no. 6, pp. 956–971, 2003.

[22] K. Alfriend and H. Yan, "Evaluation and Comparison of Relative Motion Theories," *Journal of Guidance, Control and Dynamics*, vol. 28, no. 2, pp. 254–261, March-April 2005.

[23] H. Schaub, S. R. Vadali, J. L. Junkins, and K. T. Alfriend, "Spacecraft Formation Flying Control using Mean Orbital Elements," *Journal of the Astronautical Sciences*, vol. 48, pp. 69–87, 2000.

[24] L. Breger and J. P. How, "Gauss' Variational Equation-based Dynamics and Control for Formation Flying Spacecraft," *Journal of Guidance, Control and Dynamics*, vol. 30, pp. 437–448, 2007.

[25] S. Vaddi, S. R. Vadali, and K. Alfriend, "Formation Flying: Accommodating Nonlinearity and Eccentricity Perturbations," *Journal of Guidance, Control and Dynamics*, vol. 26, no. 2, pp. 214–223, 2003.

[26] J. W. Mitchell and D. L. Richardson, "A Third-Order Analytical Solution for Relative Motion with a Circular Reference Orbit," *Journal of the Astronautical Sciences*, vol. 51, no. 1, pp. 1–12, 2003.

[27] D. Massonet, "The interferometric cartwheel: a constellation of passive satellites to produce radar image to be coherently combined," *International Journal of Remote Sensing*, vol. 22, no. 12, pp. 2413–2430, 2001.

[28] D. Massonet, "Capabilities and Limitations of the Interferometric Cartwheel," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, 2001.

[29] G. Krieger, J. Fiedler, J. Mittermayer, K. Papathanassiou, and A. Moreira, "Analysis of Multistatic Configurations for Spaceborne SAR Interferometry," *IEE Proceedings Radar, Sonar and Navigation*, vol. 150, no. 3, 2003.

[30] A. Moccia and G. Fasano, "Analysis of Spaceborne Tandem Configurations for Complementing COSMO with SAR Interferometry," *EURASIP Journal on Applied Signal Processing*, vol. 20, pp. 3304–3315, 2005.

[31] N. Sneeuw and H. Schaub, "Satellite clusters for future gravity field missions," International Symposium on Gravity, Geoid and Space Missions, Aug. 30–Sept. 3, 2004, Porto, Portugal.

[32] D. N. Wiese, W. M. Folkner, and R. S. Nerem, "Alternative mission architectures for a gravity recovery satellite mission," *Journal of Geodesy*, vol. 83, no. 6, pp. 569–581, 2009.

[33] M. A. Sharifi, N. Sneeuw, and W. Keller, "Gravity recovery capability of four generic satellite formations," Institute of Geodesy, University of Stuttgart.

[34] D. P. Scharf, F. J. Hadaegh, and R. S. Ploen, "A Survey of Spacecraft Formation Flying guidance and control (part i): Guidance," Progeeding of the American Control Conference, June 4–6, 2003, Denver, Colorado.

[35] P. Ferguson and J. How, "Decentralized Estimation Algorithms for Formation Flying Spacecraft," in *the AIAA Guidance, Navigation and Control Conference*, pp. 2003–5442, 2003.

[36] F. Busse and J. P. How, "Real-time Experimental Demonstration of Precise Decentralized Relative Navigation for Formation Flying Spacecraft," AIAA GNC, 2002.

[37] D. P. Scharf, F. J. Hadaegh, and R. S. Ploen, "A Survey of Spacecraft Formation Flying guidance and control (part ii): Control," Progeeding of the 2004 American Control Conference, June 30 – July 2, 2003, Boston Massachusetts.

[38] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A Coordination Architecture for Spacecraft Formation Control," *IEEE transactions on Control Systems Technology*, vol. 9, no. 6, November 2001.

[39] E. Gill and H. Runge, "A Tight Formation for Along-track SAR Interferometry," 2nd International Symposium on Formation Flying Missions & Technologies, September 14–16, 2004, Washington DC.

[40] A. Visioli, *Practical PID Control*. Springer, 2006.

[41] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[42] J. A. Rossiter, *Model-Based Predictive Control - A Practical Approach*. CRC Press, 2003.

[43] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained Model Predictive Control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[44] Y.-G. Xi, D.-W. LI, and S. LIN, "Model Predictive Control – Status and Challenges," *Acta Automatica Sinica*, vol. 39, no. 3, 2013.

[45] J. H. Lee, "Model Predictive Control: Review of the Three Decades of Development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, pp. 415–424, 2011.

[46] J. B. Rawlings, "Tutorial Overview of Model Predictive Control," *IEEE Control System Magazine*, pp. 38–52, June 2000.

[47] M. Tillerson, G. Inalhan, and J. P. How, "Co-ordination and control of distributed spacecraft systems using convex optimization techniques," *International Journal of Robust and Nonlinear Control*, vol. 12, pp. 207–242, 2002.

[48] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 755–764, 2002.

[49] M. Tillerson, *Coordination and control of multiple spacecraft using convex optimization techniques*. Master's thesis, Massachusetts Institute of Technology, 2002.

[50] A. Richards and J. P. How, "Performance evaluation of Rendezvous using Model Predictive Control," AIAA Guidance, Navigation, and Control Conference and Exhibit, 2003.

[51] A. Richards and J. P. How, "Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility," Proceedings of the American Control Conference, 2003.

[52] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 16, no. 7, pp. 333–351, 2006.

[53] L. Breger and J. P. How, "Safe Trajectories for Autonomous Rendezvous of Spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 31, pp. 1478–1489, 2008.

[54] L. Breger, J. P. How, and A. Richards, "Model predictive control of spacecraft formations with sensing noise," Proceedings of the American Control Conference, 2005.

[55] M. Rossi and M. Lovera, "A multirate predictive approach to orbit control of small spacecraft," Proceedings of the American Control Conference, 2002.

[56] D. Prieto and Z. Ahmed, "A drag free control based on model predictive technics," Proceedings of the American Control Conference, 2005.

[57] R. Larsson, S. Berge, P. Bodin, and U. Jonsson, "Fuel efficient orbit control strategies for formation flying and rendezvous within prisma," 29th Annual ASS Guidance and Control Conference, 2006.

[58] L. Breger, *Model predictive control for formation flying spacecraft*. Master's thesis, Massachusetts Institute of Technology, 2004.

[59] M. Saponara, V. Barrena, A. Bemporad, E. N. Hartley, J. Maciejowski, and A. Richards, "Model Predictive Control application to spacecraft rendezvous in Mars Sample Return scenario," in *Proceedings of the 4th European Conference for Aero-Space Sciences (EUCASS)*, July, 2011.

[60] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski, "Model predictive control system design and implementation for spacecraft rendezvous," *Control Engineering Practice*, vol. 20, pp. 695–713, 2012.

[61] V. Manikonda, P. O. Arambel, M. Gopinathan, R. K. Mehra, and F. Y. Hadaegh, "A model predictive-based approach for spacecraft formation keeping and attitude control," Proceedings of the American Control Conference, 1999.

[62] W. R. van Soest, P. Q. Chu, and J. A. Mulder, "Combined feedback linearization and constrained model predictive control for entry flight," *Journal of Guidance, Control, and Dynamics*, vol. 29, pp. 427–434, 2006.

[63] E. R. van Oort, P. Q. Chu, and J. A. Mulder, "Nonlinear robust model predictive control for lifting body re-entry flight attitude control," 17th IFAC Symposium on Automatic Control in Aerospace, 2007.

[64] J. Shi and A. G. Kelkar, "Feedback linearization based generalized predictive control of jupiter icy moons orbiter," *ASME Journal of Dynamic Systems, Measurements, and Control*, vol. 131, 2009.

[65] J. L. Crassidis, F. L. Markley, T. C. Anthony, and S. F. Andrews, "Nonlinear predictive control of spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 20, pp. 1096–1103, 1997.

[66] O. Hegrenaes, J. T. Gravdahl, and P. Tondel, "Spacecraft attitude control using explicit model predictive control," *Automatica*, vol. 41, pp. 2107–2114, 2005.

[67] E. Silani and M. Lovera, "Magnetic spacecraft attitude control: a survey and some new results," *Control Engineering Practice*, vol. 13, pp. 357–371, 2005.

[68] M. Wood and W.-H. Chen, "Regulation of magnetically actuated satellites using model predictive control with disturbance modelling," Proceedings of the IEEE International Conference of Networking, Sensing and Control, 2008.

[69] M. Wood and W.-H. Chen, "Model predictive control of low earth-orbiting satellites using magnetic actuation," vol. 222, pp. 619–631, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2008.

[70] G. B. Dantzig, *Linear Programming and Extensions.* Princeton University Press, Princeton, 1963.

[71] A. Valmorbida, "Test of Model Predictive Control (MPC) strategies on the SPHERES test bed for the Mars Orbital Sample Return (MOSR) scenario - Internal report," 2013.

[72] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.

[73] G. Baù, C. Bombardelli, E. C. Lorenzini, and A. Francesconi, "A variational method for the propagation of spacecraft relative motion," 20th AAS/AIAA Space Flight Mechanics Meeting, Vol. 136 of Advances in the Astronautical Sciences, pp. 2459–2474, February 14–17, 2010, San Diego, CA, Paper AAS 10–266.

[74] J. Peláez, J. M. Hedo, and P. R. de Andreéz, "A special perturbation method in orbital dynamics," *Celestial Mechanics and Dynamical Astronomy*, vol. 97, no. 2, pp. 131–150, 2007.

[75] G. Baù, A. Valmorbida, E. C. Lorenzini, and A. Francesconi, "Fast and accurate numerical integration of relative motion in spacecraft formation flying," Proceedings of 3rd CEAS Air and Space Conference, 21st AIDAA Congress, October 24–28, 2011, Venezia, Italy.

[76] D. Kim, B. Woo, S. Park, and K. Choi, "Hybrid optimization for miultiple-impulsive reconfiguration trajectories of satellite formation flying," *Advances in Space Research*, vol. 44, no. 1, pp. 1257–1269, 1 December 2009.

[77] J. Tian, N. Cui, and R. Mu, "Optimal Formation Reconfiguration using Genetic Algorithms," 2009 International Conference on Computer Modeling and Simulation, 2009.

[78] R. Larsson, J. Mueller, S. Thomas, B. Jakobsson, and P. Bodin, "Orbit Constellation Safety on the PRISMA In-orbit Formation Flying Test Bed," Proceedings of the 3rd International Symposium on Formation Flying, Missions and Technologies (ESA SP-654), 23-25 April 2008, Noordwijk, The Netherlands.

[79] J. B. Mueller and R. L. Karsson, "Collision Avoidance Maneuver Planning with Robust Optimization," 7th International ESA Conference on Guidance, Navigation and Control Systems, 23-25 April 2008, Noordwijk, The Netherlands.

[80] R. Mattingly, "Continuing evolution of mars sample return," 2004 IEEE Aerospace Conference Proceedings, 2004.

[81] A. F. Sciences, "SPHERES MOSR RDOS Phase 2 - NASA Proposal," 2010.

[82] V. Jyothindran, *A Vision Based Control System for Autonomous Rendezvous and Capture of a Mars Orbital Sample.* Master's thesis, Massachusetts Institute of Technology, 2013.

[83] J. L. Schwartz, M. A. Peck, and C. D. Hall, "Historical Review of Air-Bearing Spacecraft Simulators," *Journal of Guidance, Control and Dynamics*, vol. 26, no. 4, pp. 513–522, July-August 2003.

[84] J. P. M. O. Hilstad and A. G. Richards, *The SPHERES Guest Scientist Program.* Space Systems Laboratory, Massachusetts Institute of Technology, 2009.

[85] L. N. Trefethen, *Spectral Methods in MATLAB.* SIAM, Philadelphia, 2000.

[86] W. Kang and N. Bedrossian, "Pseudospectral optimal control theory makes debut flight, saves nasa 1m dollars in under three hours," *SIAM News*, vol. 40, 2007.

[87] F. Scarpa, A. Valmorbida, A. Aboudan, and E. C. Lorenzini, *Design and testing with a torque balance of a formation flying simulator attitude control.* Department of Industrial Engineering, University of Padova, 2013.

[88] C. Schedlinski and M. Link, "A Survey of Current Inertia Parameter Identification Methods," *Mechanical Systems and Signal Processing*, vol. 15, no. 1, pp. 189 – 211, 2001.

[89] L. Tang and W. Shangguan, "An improved pendulum method for the determination of the center of gravity and inertia tensor for irregular-shaped bodies," *Measurement*, vol. 44, no. 10, pp. 1849 – 1859, 2011.

[90] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME - Journal of Basic Engineering*, pp. 35–45, 1960.

[91] A. Bemporad, "Hybrid Toolbox - User's Guide," 2004. `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox`.

[92] M. Kvasnica, P. Grieder, M. Baotic, and F. J. Christophersen, "Multi-parametric toolbox (mpt)," 2006.

[93] V. M. Becerra, "Solving complex optimal control problems at no cost with psopt," Proceedings of the IEEE Multi-conference on Systems and Control, September 7-10, 2010, Yokohama, Japan.

[94] V. M. Becerra, *PSOPT Optimal Control Solver User Manual.* Release 3, 2010.

[95] S. Bell, "High-Precision Robot Odometry Using an Array of Optical Mice," 2011.