

**Guidance, Control and Momentum Management of  
Spacecraft with Multiple Pointing Constraints**

by

**R. Calaon**

B.S., University of Padua, Italy, 2017

M.S., University of Padua, Italy, 2019

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy

Ann and H.J. Smead Department of Aerospace Engineering Sciences

2024

Committee Members:

Prof. Hanspeter Schaub

Dr. Cody Allard

Dr. Daniel Kubitschek

Prof. Jay McMahon

Prof. Morteza Lahijanlian

Calaon, R. (Ph.D., Aerospace Engineering Sciences)

Guidance, Control and Momentum Management of Spacecraft with Multiple Pointing Constraints

Thesis directed by Prof. Hanspeter Schaub

Developments in space technology have enabled new, excitement mission concepts that are allowing humankind to push the boundaries of the known universe. One of such advancements, electric propulsion, has changed the paradigm of deep space exploration, enabling more efficient mass utilization throughout long missions. Large and highly-efficient solar arrays are often combined with electric propulsion as the main source of power in deep space.

Each of these instruments are associated by pointing requirements that must be respected in order for the instruments to operate effectively. Accurate pointing is often made possible by articulated components that can move with respect to the rigid spacecraft hub. This, however, adds degrees of freedom to the problem of defining an optimal attitude for a spacecraft with multiple pointing requirements. While ensuring satisfaction of pointing requirements is typically a trivial challenge for each piece of technology considered individually, combining multiple elements into one spacecraft gives rise to higher-dimensional problems whose solution is not trivial. The highly constrained nature of this problem becomes even more challenging when considering the dynamics of a spacecraft, specifically issues such as momentum buildup on the system due to external perturbations, or maneuvering the spacecraft between two commanded attitudes while ensuring that pointing constraints remain satisfied at all times.

For these reasons, there is the need for new, more versatile and comprehensive algorithms that can intelligently deliver solutions that are tailored to spacecraft with complex dynamics and pointing requirements. This work analyzes the dynamics of one such spacecraft, proposing GN&C algorithms that leverage the spacecraft design to enhance the performance and reduce costs and human input, while ensuring stability, requirement satisfaction, and online execution.

## Dedication

*No story lives unless someone wants to listen.*

*The stories we love best do live in us forever.*

– J. K. Rowling

To all of you, who have listened to my stories.

## Acknowledgements

First and foremost, I would like to thank my advisor, Dr. Hanspeter Schaub. Your technical input has been invaluable during this Ph.D., but equally important was your guidance through the highs and lows of this journey. Thank you for always listening and believing in me.

I will forever be grateful to Cody Allard, Thibaud Teil, Pat Kenneally, and the whole EMA squad. What we accomplished in such little time was incredible. Working with you all was a joy, and stepping down from the mission to focus on graduating was a tough but necessary decision. I cannot wait to see the fruits of our labor shooting through the Asteroid Belt!

I would also like to thank the Fulbright Program and the Ermenegildo Zegna Founder's Scholarship, whose support for this Ph.D. was essential during the uncertainty of the pandemic.

To all of the AVS Lab, thank you for being my family and my home away from home. Thank you for the many laughs, thank you for the occasional disagreements. I cannot say whether I would have made it or not without your support, but I am sure it would not have been even nearly as much fun. I am sure we will all ski down a mountain together soon, like in the good old days.

To my real family, who is loud, chaotic, and messy, but never fails to remind me who I am and where I will forever belong.

To Alice and Eva, for being two of the three brain cells I have left. To Mattia, for always being my emotional punchball. To Eli & Addie, for being the kindest, most reliable friends. To Susan, because one whole page would not be enough to say thank you properly. To all the friends, old and new, who have been a part of this story: thank you. This dissertation marks the end of a chapter, but the story continues. I hope to see you tag along for the next adventures.

## Contents

<b>Chapter</b>	<b></b>
<b>1</b>	<b>Introduction</b> <span style="float: right;"><b>1</b></span>
1.1	Motivation . . . . . 1
1.2	Background and related work . . . . . 4
1.3	Summary of Proposed Research . . . . . 7
1.4	Publications . . . . . 9
1.4.1	Journal Publications . . . . . 9
1.4.2	Conference Papers . . . . . 10
<b>2</b>	<b>Attitude Dynamics and Control</b> <span style="float: right;"><b>12</b></span>
2.1	Attitude Descriptions . . . . . 12
2.1.1	Direction Cosine Matrix . . . . . 14
2.1.2	Principal Rotation Parameters . . . . . 15
2.1.3	Quaternions . . . . . 16
2.1.4	Classic Rodrigues Parameters . . . . . 17
2.1.5	Modified Rodrigues Parameters . . . . . 18
2.2	Attitude Dynamics and Momentum Exchange Devices . . . . . 20
2.2.1	Rigid Body Dynamics . . . . . 20
2.2.2	Momentum Exchange Devices . . . . . 21
2.3	Nonlinear Spacecraft Attitude Control . . . . . 23

2.3.1	Lyapunov Direct Method . . . . .	24
2.3.2	MRP-based nonlinear PID control . . . . .	24
2.4	Attitude Simulation Framework . . . . .	26
<b>3</b>	<b>Guidance for Spacecraft Attitude and Moving Components</b>	<b>32</b>
3.1	Platform Guidance . . . . .	33
3.1.1	Gimbaled Platform and Thruster Modeling . . . . .	34
3.1.2	Closed-form Guidance Solution . . . . .	35
3.2	Attitude Guidance . . . . .	39
3.2.1	Inertial Pointing with Maximum Power . . . . .	42
3.2.2	Inertial Pointing with Lower-bounded Power Generation . . . . .	50
3.3	Solar Array Guidance . . . . .	57
3.3.1	Maximum Power Generation . . . . .	57
3.3.2	Upper-bounded Power Generation . . . . .	58
<b>4</b>	<b>Continuous Momentum Management</b>	<b>61</b>
4.1	Continuous Momentum Management via Gimbaled SEP Thruster . . . . .	62
4.1.1	Control Law Derivation with CM Knowledge . . . . .	62
4.1.2	Stability Analysis . . . . .	65
4.1.3	Control Law Derivation without CM Knowledge . . . . .	65
4.2	Momentum Management via Rotating Solar Arrays . . . . .	67
4.2.1	Symmetric, Fully-absorbent Solar Arrays . . . . .	68
4.2.2	Stability Analysis . . . . .	70
4.2.3	Fully Reflective Surface . . . . .	70
4.3	Center of Mass Estimation . . . . .	73
4.3.1	Traditional Approaches . . . . .	74
4.3.2	Steady-state Reference Tracking . . . . .	75
4.3.3	Measurement Model . . . . .	75

4.3.4	Observability . . . . .	77
4.3.5	Recursive Least-Squares Algorithm . . . . .	78
4.3.6	Biased Measurement . . . . .	79
<b>5</b>	<b>Fully Articulated Spacecraft Dynamics Simulation</b>	<b>82</b>
5.1	Simulation Setup . . . . .	84
5.1.1	Spacecraft Hub Properties . . . . .	84
5.1.2	Reaction Wheel Configuration and Mass Properties . . . . .	84
5.1.3	Solar Arrays Configuration, Properties, and Control . . . . .	85
5.1.4	Gimbal Thruster Platform Modeling and Control . . . . .	86
5.1.5	Solar Radiation Pressure Modeling . . . . .	87
5.2	Simulation Results with Perfect CM Knowledge . . . . .	88
5.2.1	Perfect Thruster-CM Alignment, Short Time Window . . . . .	88
5.2.2	Perfect Thruster-CM Alignment, Long Time Window . . . . .	89
5.2.3	Thruster-based Momentum Management, Long Time Window . . . . .	89
5.2.4	Thruster-based Momentum Management with Swirl Torque and SRP Un- loading, Long Time Window . . . . .	92
5.3	Simulation Results with CM Uncertainty . . . . .	95
5.3.1	Platform Integral Feedback . . . . .	95
5.3.2	CM Estimation without External Perturbations . . . . .	95
5.3.3	CM Estimation with SRP . . . . .	97
5.3.4	CM Estimation with SRP and Off-nominal Thruster Firing . . . . .	100
5.4	Monte Carlo Results . . . . .	102
<b>6</b>	<b>Discrete Constrained Attitude Maneuvering</b>	<b>105</b>
6.1	Workspace discretization . . . . .	106
6.2	Obstacle representation . . . . .	108
6.3	Graph search: the A* algorithm . . . . .	111

6.3.1	Algorithm Review . . . . .	111
6.3.2	Nonsingular MRP-based A* implementation . . . . .	113
<b>7</b>	<b>MRP Path Smoothing via NURBS Curves</b>	<b>116</b>
7.1	NURBS Curves Overview . . . . .	117
7.2	Time-spacing for MRP-based Kinematic Trajectory . . . . .	118
7.3	Control Point Definition for MRP-based Kinematic Trajectory . . . . .	120
7.3.1	Two-point Boundary Value Compliance . . . . .	120
7.3.2	NURBS Interpolation . . . . .	121
7.3.3	NURBS Least Squares Fit . . . . .	124
7.4	Performance Study of NURBS Interpolation vs LS Approximation . . . . .	130
7.4.1	Scenario 1: Eigenaxis Rotation . . . . .	131
7.4.2	Scenario 2: Multiple Keep-out Zones . . . . .	133
7.4.3	Scenario 3: Mixed Keep-in and Keep-out Zones . . . . .	135
7.5	Closed-loop Performance of NURBS-based guidance for attitude path planning . . .	137
7.5.1	Incorporating Externally Generated Attitude Guidance . . . . .	138
7.5.2	Evaluating the Constrained Attitude Maneuver Performance . . . . .	139
7.5.3	Attitude Path Planners . . . . .	141
7.5.4	Benchmark Analysis . . . . .	144
7.5.5	Monte Carlo Analysis . . . . .	151
<b>8</b>	<b>Effort-based Constrained Attitude Maneuvering</b>	<b>155</b>
8.1	Effort-based A* graph search . . . . .	156
8.2	NURBS Comparison in Effort-based Graph Search Algorithm . . . . .	159
8.3	Actuator Dynamics Incorporation . . . . .	162
8.3.1	Effort-based graph search . . . . .	165
8.3.2	Numerical results . . . . .	167



<b>9</b>	<b>Conclusions</b>	<b>171</b>
9.1	Summary of Contributions . . . . .	171
9.2	Future Work . . . . .	173
9.2.1	Momentum Management via SRP Torque on Solar Arrays . . . . .	173
9.2.2	Dynamics Models in the CM Estimator . . . . .	174
9.2.3	Benchmarking the Effort-based Path Planning Algorithm . . . . .	174
	<b>Bibliography</b>	<b>176</b>
	<b>Appendix</b>	
<b>A</b>	Pseudocode for Algorithms of Chapter 3	<b>183</b>
<b>B</b>	Time-warping Routine for Constant Angular Rate Maneuvering	<b>187</b>

## Tables

### Table

5.1	Table of dispersed properties and initial conditions for MC runs . . . . .	102
8.1	Effort-based Path Planners; Simulation Result Data . . . . .	168

## Figures

### Figure

1.1	Artist's impression of the EMA's MBR Explorer spacecraft (Image credit: UAE Space Agency) . . . . .	3
1.2	Spacecraft conceptual illustration . . . . .	7
2.1	Body, reference, and inertial frames . . . . .	13
2.2	Basilisk simulation block diagram for Hill pointing with RWs . . . . .	27
2.3	Simulation results for Hill-point scenario with RWs . . . . .	31
3.1	Three consecutive rotations . . . . .	35
3.2	Sun incidence on array power-generating surface . . . . .	41
3.3	Optimal Sun incidence angle on array power-generating surface . . . . .	47
3.4	Sun incidence angle on array power-generating surface using the triad method . . . . .	49
3.5	$\gamma$ contour plot for $(\beta, \delta) = [0, \pi] \times [0, \pi]$ . . . . .	50
3.6	Solution space with $A = 0.777, B = -1.112, C = -0.826$ . . . . .	54
3.7	Solution space with $0 \leq K \leq 1$ . . . . .	55
3.8	Intersection of solution spaces with $0 \leq K \leq 1$ . . . . .	56
3.9	Body-fixed and array-fixed frames . . . . .	57
3.10	Solution space with $A = 0.777, B = -1.112, C = -0.826, 0 \leq K \leq 1$ . . . . .	59
3.11	Array reference angles . . . . .	60
4.1	Center of mass offset . . . . .	63

4.2	Array orientation for different net momentum configurations . . . . .	68
4.3	Control law in Equation (4.18) with $\sigma = 1$ . . . . .	69
4.4	$\bar{h}(\alpha_{\pm}(\xi, \zeta))$ . . . . .	73
4.5	Center of Mass bias . . . . .	81
5.1	Full Basilisk block diagram for SEP point scenario . . . . .	83
5.2	Four reaction wheels in pyramid configuration . . . . .	85
5.3	Perfect thruster-CM alignment, short time window . . . . .	90
5.4	Perfect thruster-CM alignment, long time window . . . . .	91
5.5	Thruster momentum management, long time window . . . . .	93
5.6	Thruster & solar array momentum management, long time window . . . . .	94
5.7	Momentum management with CM uncertainty and integral feedback . . . . .	96
5.8	CM estimation without external perturbations . . . . .	97
5.9	Momentum management with CM uncertainty, CM estimation, and SRP . . . . .	98
5.10	CM estimation with SRP . . . . .	99
5.11	Momentum management with CM uncertainty, CM estimation, SRP, and thruster misfire . . . . .	100
5.12	CM estimation with SRP and thruster misfire . . . . .	101
5.13	Monte Carlo Runs . . . . .	103
6.1	MRP grid with two minimum-distance paths: i) green: no MRP switching; ii) purple: MRP switching . . . . .	106
6.2	Neighboring nodes . . . . .	107
6.3	Three general keep-out constraints, $\beta = 20$ deg . . . . .	109
6.4	Two constraints, 0 deg and 180 deg apart from boresight, $\beta = 20$ deg . . . . .	110
6.5	Metric-based A* solution to the path planning query, ‘o’ is starting point and ‘x’ is target point . . . . .	114

7.1	Metric-based A* solution to the path planning query, ‘o’ is starting point and ‘x’ is target point . . . . .	122
7.2	Attitude, rates, acceleration, and torque over time for the trajectory in Figure 7.1 . . . . .	123
7.3	Interpolated metric-based A* solution involving MRP switching . . . . .	124
7.4	LS approximating NURBS, attitude-based, for varying numbers of control points $n$ . . . . .	127
7.5	LS approximating NURBS, attitude- and angular rate-based, for varying numbers of control points $n$ . . . . .	130
7.6	Scenario 1 - Eigenaxis Rotation . . . . .	132
7.7	Scenario 2 - Multiple Keep-out Zones . . . . .	134
7.8	Scenario 3 - Mixed Keep-in and Keep-out Zones . . . . .	136
7.9	Basilisk modular structure for the current simulation . . . . .	137
7.10	Tracking error for Planner #2 for different density of points $M$ sampled from the interpolated trajectory . . . . .	143
7.11	Tracking error for Planner #3 for different density of points $M$ sampled from the interpolated trajectory . . . . .	144
7.12	2D plots of inertial boresight directions; ‘o’: starting point, ‘x’ endpoint. . . . .	145
7.13	Angles between Star Tracker and Sun and between Sun Sensors and Sun . . . . .	147
7.14	Performance metrics of the four planners compared . . . . .	149
7.15	Sensitivity of Planners #1 - #3 to gain variations . . . . .	151
7.16	Inertial attitude in the MC runs . . . . .	152
7.17	Angular rates in the MC runs . . . . .	153
7.18	Attitude error w.r.t. reference trajectory in MC runs . . . . .	154
8.1	Interpolated effort-based A* solution . . . . .	158
8.2	Attitude, rates, acceleration and torque over time for the trajectory in Figure 8.1 . . . . .	159
8.3	Control effort and computational time of effort-based A*, NURBS comparison . . . . .	161
8.4	Example trajectory computed with metric-based A*, $N = 7$ . . . . .	164

8.5	Wheel speeds, wheel control torques and power required to track the trajectory in Figure 8.4 . . . . .	164
8.6	Inertial trajectories of the sensitive boresight in different simulation scenarios . . . .	169
B.1	Attitude, rates, acceleration and torque with constant $\ \boldsymbol{\omega}\ $ for the trajectory in Figures 7.1 and 7.2 . . . . .	189

## Introduction

### 1.1 Motivation

Recent advancements in spacecraft technology have enabled exciting new mission concepts to be designed. One of such advancements that has slowly taken over the design for space missions consists in the use of electric propulsion. Initially used primarily for station keeping in low and geosynchronous orbits, one of the major breakthroughs in electric propulsion was its successful application in raising the orbit from geostationary transfer orbit to GEO, and therefore establishing as the leading technology for GEO applications [1]. The main argument in favor of electric propulsion is the higher specific impulse, which is a parameter that describes the efficiency with which propellant particles are accelerated by the engine to provide thrust [2]. Despite the small instantaneous thrusts delivered by electric thrusters, the higher impulse-to-mass ratio that can be achieved thanks to the higher specific impulse has made electric propulsion a compelling design choice for deep space missions in recent years.

Successful application of electric thrusters in interplanetary missions in history are NASA's Deep Space 1, launched in 1998 [3], JAXA's Hayabusa 1 and 2 missions, launched in 2003 and 2014 respectively [4, 5], ESA's Smart-1 mission, launched in 2003 [6], NASA's Dawn mission, launched in 2007 [7], and the ESA-JAXA joint mission Bepi Colombo, launched in 2018 [8]. Another instances of deep-space mission that involve electric propulsion are NASA's Psyche mission, launched in late 2023 [9], and the Emirates Mission to the Asteroid Belt (EMA) [10]. Except for the Smart-1 and Bepi Colombo missions, which had the Moon and Mercury as their respective targets, all of the

aforementioned missions involve the exploration of bodies in the asteroid belt. This marks the bottleneck for the applicability of electric propulsion to interplanetary mission, primarily due to the high power that needs to be supplied to the thruster to operate. In all of the aforementioned missions, the power is supplied through solar arrays. However, the ability of the arrays to harvest solar power directly depends on solar irradiance, which decreases quadratically with the distance from the Sun [11]. As a consequence, solar energy is not a feasible option to power electric propulsion systems beyond the asteroid belt. Mission concepts exist that involve the exploration of the outer planets and their moons using radioisotope electric propulsion, where the thruster is supplied by radioisotope thermoelectric generators. Such mission designs remain, however, only theoretical at this stage [12, 13, 14].

Distance from the Sun is typically what regulates the sizing of the solar arrays. At large distances, these can reach up to  $40\text{m}^2$  of surface area each [15]. However, another parameter severely affects the ability of the spacecraft to generate power: the inclination of sunlight with the power-generating surface of the array. For this reason, designs involving gimbaled solar arrays are becoming more and more common. Connecting the array to a one-axis gimbal gives it an additional degree of freedom that allows the array to track the direction of sunlight [3, 6, 16].

Large solar arrays are a necessary design choice for space missions with large power requirements. Together with the extra mass that they add to the system, they also come with an additional downside: the large surface areas are subject to non-negligible solar radiation pressure (SRP) effects. This phenomenon is due to the photons emitted by the Sun, which hit the array surface area and exchange a certain amount of momentum with it, depending on the surface material's optical properties [17]. SRP can have an appreciable impact on the system's dynamics, as it can cause the spacecraft to have undesired orbital and attitude drifts [18], which clash with the need to hold attitude steady for pointing accuracy. As a result, the momentum imparted onto the spacecraft by SRP, and other external forces, is absorbed by momentum exchange devices such as reaction wheels (RWs): these store the momentum along their spin axis, allowing the spacecraft attitude to remain unperturbed. This, however, cannot be done indefinitely, and momentum needs to be



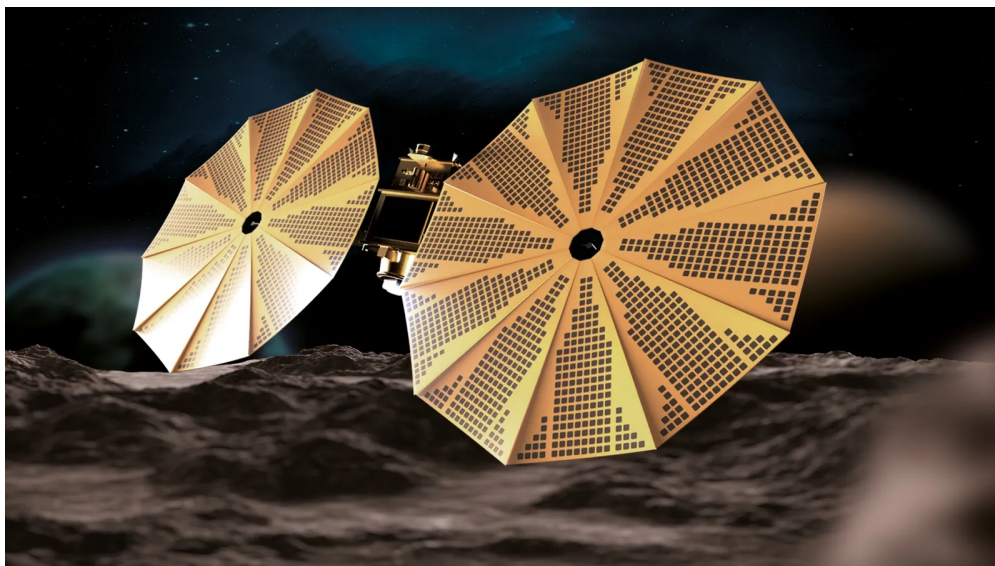


Figure 1.1: Artist's impression of the EMA's MBR Explorer spacecraft (Image credit: UAE Space Agency)

periodically dumped applying opposite external forces and torques to the spacecraft.

During deep-space missions, mode switches are often required. These are instances in which the spacecraft's main task changes. For a thrust spacecraft, for example, this can consist in, but is not limited to, switching from nominal thrust mode to Earth-pointing mode to enable telecommunications. Such mode switches are often characterized by a change in the spacecraft attitude, due to the change in the pointing requirements [6]. During Earth-pointing mode, the main driver is the ability to point the high gain antenna towards Earth, while the thruster is temporarily switched off. When this is the case, the high input power requirement can be relaxed, and the solar arrays might no longer need perfect alignment with sunlight. These attitude changes give rise to another problem, as the spacecraft slews from one pointing configuration to another. Because spacecraft are often equipped with high-sensitive cameras for data acquisition, or star trackers for attitude determination, it is necessary to keep the sun direction away from the field of view to avoid burning the optics as the spacecraft slews. Conversely, as attitude slews typically happen very slowly, it is desirable to maintain some level of illumination on the solar arrays, to avoid an abrupt decrease in power supply. These requirements translate into keep-out and keep-in

zones in the attitude space that pose a significant challenge to the attitude maneuver planning.

Developments in technology have enabled new, excitement mission concepts that are allowing humankind to push the boundaries of the known universe. Together with these new concepts and pieces of technology, however, there is the need for new, more versatile and comprehensive algorithms that can intelligently deliver solutions that are tailored to spacecraft with complex dynamics and pointing requirements. This work analyzes the dynamics of one such spacecraft, proposing GNC algorithms that leverage the spacecraft design to enhance the performance and reduce costs and human input, while ensuring stability, requirement satisfaction, and online execution.

## 1.2 Background and related work

Momentum management is a necessary task during space missions, and it consists in providing external torques to the system such that the resulting variation of net momentum brings down the angular velocities of the reaction wheels. In the most general case, this is done via a set of thrusters located on the main hub, typically in pairs to avoid producing net forces on the system that could cause trajectory deviations [19]. This choice is obviously associated with additional propellant mass, and also with the problem of high-frequency oscillation resulting from the firing, which can excite the vibrational modes of long appendages such as the solar arrays [20], and require accurate positioning of the payload [21]. Another problem related to using thrusters for momentum dumping is the fact that one or more of them might fail. For this reason, work exists in literature that addresses how to dump momentum using underactuated control torques [22]. In this case, however, momentum dumping happens over multiple phases that require attitude reorientation, thus losing the reference attitude temporarily. Other existing work investigates how to perform continuous momentum management exploiting environment features, such as the gravity gradient torque [23], or Earth’s magnetic field, by means of onboard magnetotorquers [24]. Both these approaches, however, are only applicable for Earth-orbiting satellites, and not during deep-space missions, due to the lack of gravity gradients or significant magnetic fields.

Some of the more recent mission designs have explored the idea of mounting the electric

thruster on a dual-gimbal mechanism that allows to control the direction of the thrust vector with respect to the spacecraft hub. In principle, this is motivated by the need to adjust the thrust direction based on the system’s center of mass, which changes over time [25, 7]. For the Psyche mission, the dual-gimbaled electric thrusters is offset with respect to the center of mass of the system, in order to perform continuous momentum management on the system leveraging the resulting thruster torque [9]. Utilization of gimbaled electric thrusters to perform momentum management, as well as leveraging external perturbations such as SRP, have been theorized before [26]. However, existing literature lacks detailed analytical derivation of the guidance and control laws required for such applications, as well as simulation results.

The knowledge of the system center of mass is a necessary piece of the puzzle to successfully control the gimbaled thruster. The center of mass location is known with a certain accuracy at the beginning of the mission, but it changes over time, primarily due to mass depletion. Typically, mass properties are estimated during calibration maneuvers, measuring the large angular rates and accelerations and using them to inform several Kalman filters that provide updated estimates of the inertia tensor, the location of the center of mass, and the total system mass [27, 28]. These filters typically combine measurements from on-board instruments with first principles, such as conservation of angular momentum [29]. However, such maneuvers need specific calibration procedures involving large slews that can interrupt the spacecraft’s nominal operations [30]. Moreover, large slews can excite the vibrational modes of flexible appendages such as solar arrays, which can further complicate the estimation process. Other methods to estimate inertia properties involve moving components: knowledge of the mass and position of the appendage with respect to the body, combined with velocity measurements, allows to estimate the inertia properties for the whole system [31].

Regarding constraint enforcement during slew maneuvers, the solutions that are found in literature can be broadly categorized into two groups: potential-function-based and path-planning-based. Solutions that rely on potential functions typically consist of an attractive potential, that causes the output control law to steer the spacecraft towards the desired attitude, and a repulsive

potential, whose output control component steers the spacecraft away from the keep-out zone or the boundaries of the keep-in zone [32, 33, 34, 35]. Such approaches are straightforward to implement and computationally cheap, but they can fail in the presence of complex geometries and/or overlapping constraints: whenever the resulting potential has local minima other than the desired target attitude, they can cause the spacecraft to remain stuck in the wrong configuration. Path-planning based approaches, on the other hand, usually rely on some sort of discretization of the attitude space to build a graph of constraint-compliant nodes, which can be navigated from the initial to the target attitude [36, 37, 38]. In these cases, graph-searching algorithms such as depth-first [39], breadth-first [40] or A\* [41] are applied to navigate such graphs. In different approaches, the attitude space sampling is not deterministic, but rather stochastic: for example, Probabilistic RoadMaps [42] sacrifice the completeness of the attitude space mapping in exchange for faster execution times. These latter approaches do not suffer from the problem of local minima. However, all they typically provide is a sequence of constraint-compliant attitude waypoints, which means that there is no further information about the attitude as a function of time or the angular rates required along the maneuver. Some recent contributions by Tan et al. [43] provide solutions based on attitude sampling, combined with some form of interpolation to obtain a smooth reference trajectory from a sequence of attitude waypoints. Other approaches, called metaheuristic, start from a baseline solution, which is further improved upon exploring the neighboring solution space. Differential evolution algorithms or particle swarm optimization are examples of such approaches. Their strength consists in the ability to refine the solution based on an optimality condition, while at the same time enforcing the constraints that such solution must not violate. Examples are found in References 44, 45, 46, 47, 48. Conceptually similar, but different in methodology is the approach by Dearing et al. [49], which consists in setting up a constrained minimization problem via the definition of a quadratic cost function, and performing iterations on the trajectory solution manifold until convergence is reached.

### 1.3 Summary of Proposed Research

The overarching purpose of this research proposal is to provide new analytical formulations and algorithms to tackle the problem of meeting the complex pointing requirements of a spacecraft equipped with articulating appendages and sensitive instruments. The goal is to develop algorithms that can autonomously compute solutions that meet the spacecraft's pointing requirements, while ensuring enough computational agility to run onboard a flight computer. The accuracy and performance of such algorithms is tested via high-fidelity simulations.

- (1) The first research thrust of this dissertation involves the **definition of static reference frames and angles for the spacecraft hub and the articulating appendages**. Figure 1.2 illustrates a spacecraft whose design is conceptually similar to those of the missions mentioned earlier in this introduction. Such design involves two solar arrays connected to the rigid hub via a single degree of freedom hinge, and an electric thruster mounted on a dual-gimbaled platform. In general, this attitude problem in all its parts is 7-dimensional. Because each component is characterized by its own pointing requirement, **a solid ana-**

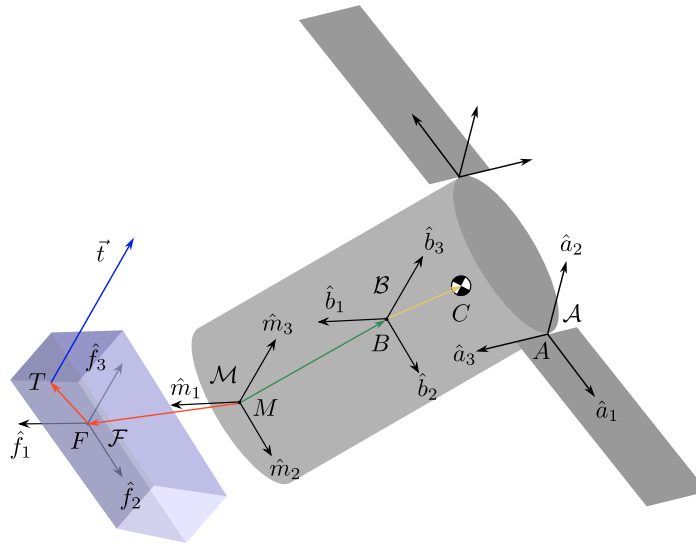


Figure 1.2: Spacecraft conceptual illustration

lytical formulation is required to describe the viable space of attitudes that can enforce certain constraints at any given time, depending on the order of priority of each constraint. Emphasis is given to the scenarios where the priority order affects the choice of the solution, as well as when it is impossible to satisfy all the requirements simultaneously and, therefore, constrained optimization solutions are required.

- (2) The second research thrust revolves around the aspect of **center of mass estimation and momentum management**. These two problems are connected by the fact that, to successfully perform continuous momentum management via the gimballed thruster, knowledge of the center of mass location is needed. Less intuitive is the inverse dependence, according to which the effect of the misalignment of the thruster with respect to the center of mass can be used as a source of information to estimate the center of mass location. The purpose of this second research goal is to **deliver a novel estimation algorithm that allows to locate the center of mass exploiting the effects that the thruster misalignment produces on the system, thus avoiding to interrupt the regular course of spacecraft operation to perform calibrations**. Consequently, a control law is designed, whose purpose is to intentionally **offset the thrust vector with respect to the center of mass, to successfully perform continuous momentum management**, again avoiding ad hoc maneuvers and costly thruster firings.
- (3) The third research thrust involves the constraint attitude maneuvering problem that arises during mode switching. The goal is to deliver an **online planner** that can compute an open-loop trajectory **that slews the spacecraft into place without violating any of the pointing constraints**. This is done via a path-planning algorithm, which allows to regulate the density of the sampled attitude space, thus **adjusting the computational load according to the memory availability**, while not suffering from the most known problems of potential field-based approaches. Emphasis is given to the **kinodynamic aspect of the open loop trajectory**, to ensure that closed-loop performance can match

the open-loop planning. **Control effort is blended into the pathfinding algorithm**, in order to deliver solutions that are not only feasible, but also effort-efficient. Effort accounts for both the hub dynamics, in case the spacecraft starts from an initial tumbling configuration, and actuators dynamics, in case the maneuver starts from a condition of actuators saturation.

## 1.4 Publications






All publications that originated during this Ph.D. career are listed below. Publications whose content does not appear in this dissertation are marked with a dagger symbol.

### 1.4.1 Journal Publications


- (6) **R. Calaon**, C. Allard and H. Schaub, “Momentum Management Of A Spacecraft Equipped With A Dual-Gimbaled Electric Thruster,” *Journal of Spacecraft and Rockets (in preparation)*
- (5) † T. Teil and **R. Calaon**, “Attitude Uncertainty Quantification of Rectilinear Asteroid Flybys,” *Journal of Guidance, Control, and Dynamics (submitted in Aug 2024)*
- (4) **R. Calaon**, C. Allard and H. Schaub, “Attitude Reference Generation for Spacecraft with Rotating Solar Arrays and Pointing Constraints,” *Astrodynamics (accepted in October 2024)*
- (3) **R. Calaon** and H. Schaub, “Constrained Attitude Path Planning via Least-Squares Modified-Rodrigues-Parameters-Based Smoothing Curves,” *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 11, November 2023, pp. 2207–2217. doi:10.2514/1.G007324
- (2) **R. Calaon**, H. Schaub and M. A. Trowbridge, “Basilisk-Based Benchmark Analysis of Different Constrained Attitude Dynamics Planners,” *Journal of Aerospace Information Systems*, Vol. 20, No. 2, February 2023, pp. 60–69. doi:10.2514/1.I011109



- (1) **R. Calaon** and H. Schaub, “Constrained Attitude Maneuvering via Modified Rodrigues Parameters Based Motion Planning Algorithms,” *Journal of Spacecraft and Rockets*, Vol. 59, No. 4, July–August 2022, pp. 1342–1356. doi:10.2514/1.A35294

#### 1.4.2 Conference Papers

- (9) **R. Calaon**, C. Allard and H. Schaub, “Attitude Reference Generation for Spacecraft with Rotating Solar Arrays and Pointing Constraints,” IEEE Aerospace Conference, Big Sky, Montana, March 2–9, 2024, Paper No. 2555. doi:10.1109/AERO58975.2024.10521153. 
- (8) † T. Teil and **R. Calaon**, “Attitude Uncertainty Quantification of Rectilinear Asteroid Flybys for the Emirates Mission to the Asteroid Belt,” AAS Rocky Mountain GN&C Conference, Breckenridge, CO, Feb. 2–7, 2024, Paper No. AAS 24-065
- (7) † **R. Calaon**, C. Allard and H. Schaub, “Solar Electric Propulsion GN&C Pointing State Overview For The Emirates Mission To The Asteroid Belt,” AAS Rocky Mountain GN&C Conference, Breckenridge, CO, Feb. 2–7, 2024, Paper No. AAS 24-056. 
- (6) **R. Calaon**, C. Allard and H. Schaub, “Continuous Center Of Mass Estimation For A Gimbaled Ion Thruster Equipped Spacecraft,” AAS/AIAA Astrodynamics Specialist Conference, Big Sky, Montana, August 13–17, 2023, Paper No. AAS 23-218. 
- (5) **R. Calaon**, L. Kiner, C. Allard and H. Schaub, “Momentum Management Of A Spacecraft Equipped With A Dual-Gimbaled Electric Thruster,” AAS Rocky Mountain GN&C Conference, Breckenridge, CO, Feb. 2–8, 2023, Paper No. AAS 23-178. 
- (4) **R. Calaon** and H. Schaub, “Optimal Actuator-Based Attitude Maneuvering Of Constrained Spacecraft Via Motion Planning Algorithms,” International Astronautical Congress, Paris, France, Sept. 18–22, 2022, Paper No. IAC-22,C1,2,5,x68843. 
- (3) **R. Calaon** and H. Schaub, “Constrained Attitude Path Planning Via Least Squares MRP-Based NURBS Curves,” AAS Astrodynamics Specialist Conference, Charlotte, North Car-



olina, August 7–10 2022, Paper No. AAS 22-541. 

- (2) **R. Calaon**, M. Trowbridge and H. Schaub, “A Basilisk-Based Benchmark Analysis Of Different Constrained Attitude Dynamics Planners,” AIAA SciTech Forum, San Diego, CA, January 3-7, 2022, Paper No. AIAA 2022-2277. [doi:10.2514/6.2022-2277](https://doi.org/10.2514/6.2022-2277), 
- (1) **R. Calaon** and H. Schaub, “Constrained Attitude Maneuvering Via Modified Rodrigues Parameters Based Motion Planning Algorithms,” AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, August 8–12, 2021, Paper No. AAS 21-527. 

## Attitude Dynamics and Control

The goal for this chapter is to build the theoretical foundation for the work described in the following chapters. The first section aims to give a comprehensive review of the most common descriptions used in spacecraft attitude and related kinematic equations. The second section outlines the equations that govern the dynamics of a rigid body, expanding on the Lyapunov method used to deliver nonlinear, asymptotically stable control for the attitude of a rigid body. Lastly, the third section removes the rigid body assumption to introduce the dynamics of momentum exchange devices into the model. These devices are actuators that are used to control the motion of the spacecraft hub by conservatively exchanging angular momentum with it.

### 2.1 Attitude Descriptions

The attitude of a spacecraft is the motion of the spacecraft about its center of mass. This motion is, in first approximation, independent from the orbital dynamics, which instead describes the motion of the spacecraft as a point mass while subject to gravity forces. Describing the attitude of a rigid body is inherently equivalent to describing the motion of a frame rigidly attached to this rigid body. Therefore, this work will typically refer to the body frame  $\mathcal{B}$  as the frame that, at any given instant in time, describes the orientation of the spacecraft main body, or hub. The attitude of a body, or frame, is expressed as a rigid body rotation performed with respect to another frame. Generally, this frame is the inertial frame  $\mathcal{N}$ , whose orientation is inertially fixed and well known. It is also possible to define a relative attitude between two non-inertially fixed frames: in this case,

we are generally referring to the attitude error between a rigid body and a desired reference frame  $\mathcal{R}$  that the spacecraft hub should target. Let's assume that the desire is to describe the orientation of the body frame  $\mathcal{B}$ , defined by a right-handed set of unit vectors  $\hat{\mathbf{b}}_1$ ,  $\hat{\mathbf{b}}_2$ , and  $\hat{\mathbf{b}}_3$ , with respect to the inertial frame  $\mathcal{N}$ , defined by  $\hat{\mathbf{n}}_1$ ,  $\hat{\mathbf{n}}_2$ , and  $\hat{\mathbf{n}}_3$ , and a reference frame  $\mathcal{R}$ , defined by  $\hat{\mathbf{r}}_1$ ,  $\hat{\mathbf{r}}_2$ , and  $\hat{\mathbf{r}}_3$ . Figure 2.1 shows these frames with respect to a rigid body. Attitude is inherently a relative measurement of a displacement between two frames, and this work uses the two-letter notation to identify the two frames between which this relative attitude is expressed. With respect to Figure 2.1, the definition of frame  $\mathcal{R}$  with respect to  $\mathcal{N}$  is typically referred to as ‘guidance’, i.e., the definition of the frame that the body should target. The definition of  $\mathcal{B}$  with respect to  $\mathcal{N}$  is referred to as ‘navigation’, that is, defining the current attitude of the body-frame with respect to the inertial frame. Lastly, driving the  $\mathcal{B}$  frame towards the  $\mathcal{R}$  frame is known as a ‘control’ problem. These three components typically come together under the collective definition of Guidance, Navigation, and Control (GNC). This work deals extensively with guidance and control problems in the attitude domain, and only marginally with navigation.

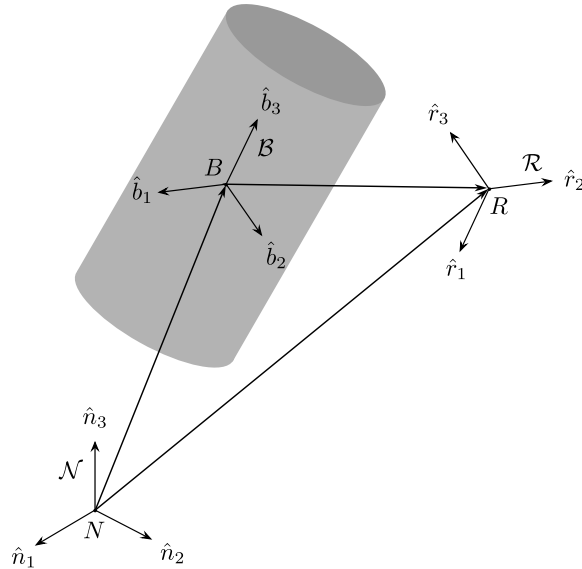


Figure 2.1: Body, reference, and inertial frames

Attitude can be described by no less than three independent coordinates. Any attitude description that relies on three coordinates is defined as a minimal set, and every minimal set presents orientations for which the associated coordinate set is singular. This poses a problem when integrating the attitude kinematic equations based on a given attitude representation, because such kinematic equations will also be singular. This issue is typically avoided with redundant sets: these are sets of attitude parameters that use four or more non-independent coordinates. While these sets are nonsingular and fully describe the attitude space, they typically require numerical constraints to be enforced during attitude propagation.

### 2.1.1 Direction Cosine Matrix

The Direction Cosine Matrix, or DCM, is the most basic description of relative attitude. The DCM  $[\mathcal{BN}]$  is a  $3 \times 3$  matrix that, when right-multiplied by a vector expressed in  $\mathcal{N}$ -frame coordinates, returns the description of the same vector in  $\mathcal{B}$ -frame coordinates. Using the left superscript to indicate the frame in which a vector is expressed, the DCM description is obtained as [50]:

$$[\mathcal{BN}] = \begin{bmatrix} \mathcal{B}\hat{\mathbf{n}}_1 & \mathcal{B}\hat{\mathbf{n}}_2 & \mathcal{B}\hat{\mathbf{n}}_3 \end{bmatrix} \quad [\mathcal{NB}] = \begin{bmatrix} \mathcal{N}\hat{\mathbf{b}}_1 & \mathcal{N}\hat{\mathbf{b}}_2 & \mathcal{N}\hat{\mathbf{b}}_3 \end{bmatrix}. \quad (2.1)$$

Every valid DCM is an orthonormal matrix. Therefore, despite the matrix representation requiring 9 elements, only 3 of these elements are truly linearly independent, while the remaining 6 are dependent on the fact that each row (column) of the DCM must have unity norm, and each pair of rows (columns) must be orthogonal [51]. For this reason, it is also true that:

$$[\mathcal{BN}] = [\mathcal{NB}]^T. \quad (2.2)$$

Moreover, DCMs can be ‘added’ together to obtain the relative attitude between two frames by means of the relative attitude of the two with respect to an intermediate frame:

$$[\mathcal{BN}] = [\mathcal{BR}][\mathcal{RN}]. \quad (2.3)$$

Let's define  ${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}$  the angular rate of the frame  $\mathcal{B}$  measured with respect to the inertial frame  $\mathcal{N}$ . The left superscript  $\mathcal{B}$  indicates that the vector is expressed in  $\mathcal{B}$ -frame coordinates, consistently with Reference 50. This notation will be used consistently throughout this work. Differentiating Equation (2.1) and applying the transport theorem [50] gives the kinematic equation for the DCM attitude representation:

$$[\dot{\mathcal{B}\mathcal{N}}] = -[{}^{\mathcal{B}}\tilde{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}][\mathcal{B}\mathcal{N}] \quad (2.4)$$

where the tilde operator denotes the skew-symmetric cross-product matrix obtained from a 3-dimensional vector.

### 2.1.2 Principal Rotation Parameters

The Principal Rotation Parameters (PRP) attitude description is based on Euler's Principal Rotation Theorem, according to which a rigid body rotation can be fully described by a Principal Rotation Angle (PRA)  $\phi$  and a Principal Rotation Vector (PRV)  $\hat{\mathbf{e}}$ , where the PRV has the same coordinates in both the initial and final rotation frame. It follows from this definition that the PRV is an eigenvector of the DCM describing the rigid body rotation, whose associated eigenvalue is 1. Because the PRP is a unit direction vector, and therefore is only defined by two independent variables, the PRP set is a minimal set: rotations of  $2n\pi, n \in \mathbb{Z}$  can in fact not be described by a proper PRP set. The mapping from PRPs to DCM is [52]:

$$[C] = \begin{bmatrix} e_1^2 \Sigma + \cos \phi & e_1 e_2 \Sigma + e_3 \sin \phi & e_1 e_3 \Sigma - e_2 \sin \phi \\ e_1 e_2 \Sigma - e_3 \sin \phi & e_2^2 \Sigma + \cos \phi & e_2 e_3 \Sigma + e_1 \sin \phi \\ e_1 e_3 \Sigma + e_2 \sin \phi & e_2 e_3 \Sigma - e_1 \sin \phi & e_3^2 \Sigma + \cos \phi \end{bmatrix} \quad \Sigma = 1 - \cos \phi \quad (2.5)$$

where  $\hat{\mathbf{e}} = (e_1, e_2, e_3)^T$ . The inverse mapping is:

$$\cos \phi = \frac{1}{2} (C_{11} + C_{22} + C_{33} - 1) \quad \hat{\mathbf{e}} = \frac{1}{2 \sin \phi} \begin{pmatrix} C_{23} - C_{32} \\ C_{31} - C_{13} \\ C_{12} - C_{21} \end{pmatrix} \quad (2.6)$$

with  $C_{ij}$  being the entries of the DCM. From Equation (2.5) it can be observed that multiple sets of PRPs describe the same rigid body rotation: defining  $\phi' = \phi - 2\pi$ , the four sets  $(\phi, \hat{e})$ ,  $(-\phi, -\hat{e})$ ,  $(\phi', \hat{e})$ , and  $(-\phi', -\hat{e})$  are equivalent. Equation (2.6) shows the singularity of the PRP description, which is undefined for  $\phi = 2n\pi$ . Defining the vector  $\boldsymbol{\gamma}_{B/N} = \phi \hat{e}$  allows to express the kinematic equation for the PRPs:

$$\dot{\boldsymbol{\gamma}}_{B/N} = \left[ \mathbf{I}_{3 \times 3} + \frac{1}{2}[\tilde{\boldsymbol{\gamma}}] + \frac{1}{\phi^2} \left( 1 - \frac{\phi}{2} \cot \frac{\phi}{2} \right) [\tilde{\boldsymbol{\gamma}}] \right] {}^B \boldsymbol{\omega}_{B/N} \quad (2.7)$$

$${}^B \boldsymbol{\omega}_{B/N} = \left[ \mathbf{I}_{3 \times 3} - \left( \frac{1 - \cos \phi}{\phi^2} \right) [\tilde{\boldsymbol{\gamma}}] + \left( \frac{1 - \sin \phi}{\phi^3} \right) [\tilde{\boldsymbol{\gamma}}]^2 \right] \dot{\boldsymbol{\gamma}}_{B/N}. \quad (2.8)$$

### 2.1.3 Quaternions

Likely the most popular attitude description, quaternions, also known as Euler parameters, are a redundant set consisting of a four-dimensional unit vector. The quaternion set can be defined starting from the PRP definition given in Section 2.1.2:

$$\hat{\boldsymbol{\beta}} = \begin{Bmatrix} \cos \frac{\phi}{2} \\ \hat{e} \sin \frac{\phi}{2} \end{Bmatrix} = \begin{Bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{Bmatrix}. \quad (2.9)$$

Similarly to PRPs, quaternions are a twofold mapping of the space of rigid body rotations, with both  $\hat{\boldsymbol{\beta}}$  and  $-\hat{\boldsymbol{\beta}}$  mapping to the same DCM. The mapping from quaternion vector to DCM is:

$$[C] = \begin{bmatrix} \beta_0^2 + \beta_1^2 - \beta_2^2 - \beta_3^2 & 2(\beta_1\beta_2 + \beta_0\beta_3) & 2(\beta_1\beta_3 - \beta_0\beta_2) \\ 2(\beta_1\beta_2 - \beta_0\beta_3) & \beta_0^2 - \beta_1^2 + \beta_2^2 - \beta_3^2 & 2(\beta_2\beta_3 + \beta_0\beta_1) \\ 2(\beta_1\beta_3 + \beta_0\beta_2) & 2(\beta_2\beta_3 - \beta_0\beta_1) & \beta_0^2 - \beta_1^2 - \beta_2^2 + \beta_3^2 \end{bmatrix} \quad (2.10)$$

and the inverse mapping is:

$$\begin{aligned}
\beta_0 &= \pm \frac{1}{2} \sqrt{C_{11} + C_{22} + C_{33} + 1} \\
\beta_1 &= \frac{C_{23} - C_{32}}{4\beta_0} \\
\beta_2 &= \frac{C_{31} - C_{13}}{4\beta_0} \\
\beta_3 &= \frac{C_{12} - C_{21}}{4\beta_0}.
\end{aligned} \tag{2.11}$$

The expression in Equation (2.11) is singular for  $\beta_0 \rightarrow 0$ . However, in such case it is possible to express the elements of the quaternion vector as a function of a different element, other than  $\beta_0$ , which is not singular. This method, known as Sheppard's method, allows to robustly compute the quaternion vector from the associated DCM [53]. The redundancy of the quaternion set makes it a compelling choice in attitude propagation algorithms, because using Sheppard's method, no singularities are ever encountered. Conversely, because only unit quaternion vectors accurately describe valid attitudes, the unity constraint  $\|\hat{\beta}\| = 1$  must be enforced at every integration step. Another point in quaternions' favor is the linear nature of the kinematic equations:

$$\dot{\beta}_{\mathcal{B}/\mathcal{N}} = \frac{1}{2} [B(\hat{\beta}_{\mathcal{B}/\mathcal{N}})]^{\mathcal{B}} \omega_{\mathcal{B}/\mathcal{N}} \tag{2.12}$$

$${}^{\mathcal{B}} \omega_{\mathcal{B}/\mathcal{N}} = 2 [B(\hat{\beta}_{\mathcal{B}/\mathcal{N}})]^T \dot{\beta}_{\mathcal{B}/\mathcal{N}} \tag{2.13}$$

with:

$$[B(\hat{\beta}_{\mathcal{B}/\mathcal{N}})] = \begin{bmatrix} -\beta_1 & -\beta_2 & -\beta_3 \\ \beta_0 & -\beta_3 & \beta_2 \\ \beta_3 & \beta_0 & -\beta_1 \\ -\beta_2 & \beta_1 & -\beta_0 \end{bmatrix}. \tag{2.14}$$

#### 2.1.4 Classic Rodrigues Parameters

Classic Rodrigues Parameters (CRPs), also known as Gibbs vector, are a three-dimensional attitude representation consisting in the stereographic projection of the quaternion set onto a three-dimensional hyperplane [54]. The CRP set can be expressed both in terms of PRPs and

quaternions:

$$\mathbf{q} = \frac{1}{\beta_0} \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{Bmatrix} = \tan \frac{\phi}{2} \hat{\mathbf{e}}. \quad (2.15)$$

The CRP set is a onefold mapping of the attitude space: it can be observed how both  $\hat{\beta}$  and  $-\hat{\beta}$ , which are two different quaternion descriptions of the same attitude, map to the same CRP set. Being a minimal representation, CRP sets present a singularity at  $\phi \rightarrow \pm\pi$ . The mapping from CRP to DCM is:

$$[C] = \frac{((1 - \mathbf{q}^T \mathbf{q})[\mathbf{I}_{3 \times 3}] + 2\mathbf{q}\mathbf{q}^T - 2[\tilde{\mathbf{q}}])}{1 + \mathbf{q}^T \mathbf{q}}, \quad (2.16)$$

while the kinematic equations are [54]:

$$\dot{\mathbf{q}}_{\mathcal{B}/\mathcal{N}} = \frac{1}{2} \left[ [\mathbf{I}_{3 \times 3}] + [\tilde{\mathbf{q}}_{\mathcal{B}/\mathcal{N}}] + \mathbf{q}_{\mathcal{B}/\mathcal{N}} \mathbf{q}_{\mathcal{B}/\mathcal{N}}^T \right] {}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \quad (2.17)$$

$${}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} = \frac{2}{1 + \mathbf{q}^T \mathbf{q}} ([\mathbf{I}_{3 \times 3}] - [\tilde{\mathbf{q}}]) \dot{\mathbf{q}}. \quad (2.18)$$

### 2.1.5 Modified Rodrigues Parameters

Modified Rodrigues Parameters (MRPs) are another minimal set of attitude parameters. Similarly to CRPs, they are obtained from a stereographic projection of the four-dimensional quaternion set onto a 3-dimensional hyperplane. The difference with CRPs is that the projection point is moved from the origin to (-1,0). As a result, the expression for the MRP set is [55, 52, 56]:

$$\boldsymbol{\sigma} = \frac{\sin \phi/2}{1 + \cos \phi/2} \hat{\mathbf{e}} = \tan \frac{\phi}{4} \hat{\mathbf{e}}. \quad (2.19)$$

The singularity in the MRP formulation in Equation (2.19) appears for  $\phi = \pm 2\pi$ . However, the singularity in the MRP formulation can be avoided by choosing the appropriate MRP set for attitude representation. MRPs are also a twofold mapping of the attitude space: therefore, for every principal rotation set  $(\mathbf{e}, \Phi)$  there exists a shadow set  $(\mathbf{e}, \Phi')$  with  $\Phi' = \Phi - 2\pi$  that represents the same attitude. Such shadow set is related to the equivalent MRP set via:

$$\boldsymbol{\sigma}^S = \frac{\sin \phi'/2}{1 + \cos \phi'/2} \hat{\mathbf{e}} = \frac{1 + \cos \phi/2}{1 - \cos \phi/2} \boldsymbol{\sigma} = -\frac{\boldsymbol{\sigma}}{\sigma^2} \quad (2.20)$$



where  $\sigma^2 = \|\boldsymbol{\sigma}\|^2$ . Another interesting property can be inferred from Equation (2.20) is that any MRP set whose norm is higher than 1 ( $\sigma > 1$ ) has a corresponding shadow set whose norm is smaller than one ( $\sigma^S < 1$ ): this means that for any MRP set represented by a point outside a unit 3D sphere centered in the origin, there exists a shadow MRP set within the same unit sphere that represents the same attitude. For points on the boundary of the unit sphere, such that  $\sigma = 1$ , it is  $\boldsymbol{\sigma}^S = -\boldsymbol{\sigma}$ . For all the above, the operational domain for the representation of the attitude of a rigid body can be restricted, without loss of generality, to the points in  $\mathbb{R}^3$  contained in a sphere of radius 1, including boundary points. This choice gets rid of the singularity altogether, at the price of introducing a discontinuity in the MRP attitude description when the PRA grows above  $\pi$ . The kinematic equations for MRPs are:

$$\dot{\boldsymbol{\sigma}}_{B/N} = \frac{1}{4}[B(\boldsymbol{\sigma}_{B/N})]^B \boldsymbol{\omega}_{B/N} \quad (2.21)$$

$${}^B \boldsymbol{\omega}_{B/N} = 4[B(\boldsymbol{\sigma}_{B/N})]^{-1} \dot{\boldsymbol{\sigma}}_{B/N} \quad (2.22)$$

$$\ddot{\boldsymbol{\sigma}}_{B/N} = \frac{1}{4} \left( [B(\boldsymbol{\sigma}_{B/N})]^B \dot{\boldsymbol{\omega}}_{B/N} + [\dot{B}(\boldsymbol{\sigma}_{B/N}, \dot{\boldsymbol{\sigma}}_{B/N})]^B \boldsymbol{\omega}_{B/N} \right) \quad (2.23)$$

$${}^B \dot{\boldsymbol{\omega}}_{B/N} = 4[B(\boldsymbol{\sigma}_{B/N})]^{-1} \left( \ddot{\boldsymbol{\sigma}}_{B/N} - [\dot{B}(\boldsymbol{\sigma}_{B/N}, \dot{\boldsymbol{\sigma}}_{B/N})][B(\boldsymbol{\sigma}_{B/N})]^{-1} \dot{\boldsymbol{\sigma}}_{B/N} \right) \quad (2.24)$$

with:

$$[B(\boldsymbol{\sigma}_{B/N})] = (1 - \sigma_{B/N}^2)[\mathbf{I}_{3 \times 3}] + 2[\tilde{\boldsymbol{\sigma}}_{B/N}] + 2\boldsymbol{\sigma}_{B/N}\boldsymbol{\sigma}_{B/N}^T \quad (2.25)$$

$$[B(\boldsymbol{\sigma}_{B/N})]^{-1} = \frac{1}{(1 + \sigma_{B/N}^2)^2} [B(\boldsymbol{\sigma}_{B/N})]^T \quad (2.26)$$

$$[\dot{B}(\boldsymbol{\sigma}_{B/N}, \dot{\boldsymbol{\sigma}}_{B/N})] = (-2\boldsymbol{\sigma}_{B/N}^T \dot{\boldsymbol{\sigma}}_{B/N})[\mathbf{I}_{3 \times 3}] + 2[\dot{\tilde{\boldsymbol{\sigma}}}_{B/N}] + 2(\boldsymbol{\sigma}_{B/N} \dot{\boldsymbol{\sigma}}_{B/N}^T + \dot{\boldsymbol{\sigma}}_{B/N} \boldsymbol{\sigma}_{B/N}^T). \quad (2.27)$$

Note that this analysis provides also the second-order derivative of the MRP set and its kinematic expression as function of angular rates and accelerations. These equations are provided because MRPs will be used extensively in the last chapters of this dissertation, where smooth, twice differentiable interpolating curves in MRP spaces are mapped into kinematically feasible attitude reference trajectories leveraging Equations (2.23) and (2.24). Another noteworthy relationship exists between the norm of the MRP derivative and the norm of the angular rate. From Equations (2.21) and (2.26)

it follows that:

$$\begin{aligned}
\dot{\sigma}_{B/N}^2 &= \dot{\boldsymbol{\sigma}}_{B/N}^T \dot{\boldsymbol{\sigma}}_{B/N} = \frac{1}{16} {}^B \boldsymbol{\omega}_{B/N}^T [B(\boldsymbol{\sigma}_{B/N})]^T [B(\boldsymbol{\sigma}_{B/N})] {}^B \boldsymbol{\omega}_{B/N} \\
&= \frac{(1 + \sigma_{B/N}^2)^2}{16} {}^B \boldsymbol{\omega}_{B/N}^T [B(\boldsymbol{\sigma}_{B/N})]^{-1} [B(\boldsymbol{\sigma}_{B/N})] {}^B \boldsymbol{\omega}_{B/N} \\
&= \frac{(1 + \sigma_{B/N}^2)^2}{16} \boldsymbol{\omega}_{B/N}^2
\end{aligned} \tag{2.28}$$

from which:

$$\|\dot{\boldsymbol{\sigma}}_{B/N}\| = \frac{1 + \sigma_{B/N}^2}{4} \|\boldsymbol{\omega}_{B/N}\|. \tag{2.29}$$

## 2.2 Attitude Dynamics and Momentum Exchange Devices

### 2.2.1 Rigid Body Dynamics

This section develops the equations that describe the rotational motion of a spacecraft that is subject to a set of internal and external torques. The fundamental building block for future discussion is the equation that describes the rotational dynamics of a rigid body, commonly known as Euler's equation [57]:

$$[I_C] \dot{\boldsymbol{\omega}}_{B/N} = -[\tilde{\boldsymbol{\omega}}_{B/N}] [I_C] \boldsymbol{\omega}_{B/N} + \mathbf{u} + \mathbf{L}_{\text{ext}} \tag{2.30}$$

where  $[I_C]$  is the inertia tensor of the rigid body with respect to its center of mass,  $\mathbf{u}$  is the control torque, and  $\mathbf{L}_{\text{ext}}$  is any other external torque acting on the system. In this work, the dot always denotes the inertial derivative  $\dot{\boldsymbol{\omega}}_{B/N} = \frac{\mathcal{N}}{d} \boldsymbol{\omega}_{B/N}$ , whereas local-frame time derivatives are denoted via the prime superscript  $\boldsymbol{\omega}'_{B/N} = \frac{\mathcal{B}}{d} \boldsymbol{\omega}_{B/N}$ . The distinction between control torque and external torque is made as the first is a torque that is intentionally delivered to the system to obtain a response (attitude slew, momentum dumping, etc), whereas external torques are typically the result of forces acting on the spacecraft that are outside of control, and can only be dealt with passively. It is important to reiterate that Equation (2.30) describes the dynamics of the spacecraft hub considered as a rigid body. This does not factor in the contribution of moving components such as momentum exchange devices, whose dynamic contributions are described in detail in the following subsection.

### 2.2.2 Momentum Exchange Devices

Momentum Exchange Devices (MED) are one class of actuators that are used to deliver the stabilizing control the attitude of the spacecraft. As the name suggests, these actuators are spun up or down in order to exchange momentum with the spacecraft hub, which rotates as a result. Because the exchange of torques between the spacecraft hub and the MEDs is internal to the system, it follows from conservation of angular momentum that the net momentum change on the system as a whole is zero. Different types of MEDs are reaction wheels (RWs), control moment gyros (CMGs), and variable-speed control moment gyros (VSCMGs). RWs have a spin axis that is fixed in the spacecraft body frame; each wheel exchanges momentum by changing its rotation speed about its spin axis. CMGs consist of a wheel spinning at a constant rate about an axis. The spin axis, however, is not fixed in the body frame, and its motion causes a torque along the direction orthogonal to the spin axis and the CMG gimbal axis. Lastly, VSCMGs are CMGs in which the spin rate is variable, therefore giving control on an additional degree of freedom. Both RWs and CMGs are subsets of VSCMGs.

This work only considers reaction wheels as the main actuators, therefore the following analysis does not generalize to the other MED types. For a comprehensive analysis of all MEDs, the reader is once again referred to Reference 50. Moreover, the reaction wheels are always assumed to be perfectly balanced about their spin axis and frictionless. Under these assumptions, let's consider the rotational dynamics of the hub first, which is given by Equation (2.30) removing any control torque or external torque:

$$\dot{\mathbf{H}}_{\text{hub}} = [I_{\text{hub}}]\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times [I_{\text{hub}}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}. \quad (2.31)$$

Let's now define a gimbal frame  $\mathcal{G}$ , with respect to which the single wheel's dynamics is expressed:

$$[\mathcal{B}\mathcal{G}] = \begin{bmatrix} \mathcal{B}\hat{\mathbf{g}}_s & \mathcal{B}\hat{\mathbf{g}}_t & \mathcal{B}\hat{\mathbf{g}}_g \end{bmatrix} \quad (2.32)$$

where  $\hat{\mathbf{g}}_s$  is the spin axis of the wheel, and the other two complete the frame. Note that each axis of the gimbal frame are fixed in the body frame  $\mathcal{B}$ . The angular momentum of the wheel alone can

be expressed as:

$$\mathbf{H}_w = [I_w](\boldsymbol{\Omega} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}) = I_{w_s}(\boldsymbol{\Omega} + \hat{\mathbf{g}}_s^T \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}) + I_{w_t} \hat{\mathbf{g}}_t^T \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + I_{w_t} \hat{\mathbf{g}}_g^T \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \quad (2.33)$$

where  $\boldsymbol{\Omega}$  is the relative speed of the wheel with respect of the hub, and it is entirely along  $\hat{\mathbf{g}}_s$ . The variables  $I_{w_s}$  and  $I_{w_t}$  are, respectively, the inertia of the wheel about the spin axis and the transverse axis, and Equation (2.33) shows the total wheel momentum decomposed along the axis of the gimbal frame. Taking the inertial time derivative gives:

$$\dot{\mathbf{H}}_w = I_{w_s}(\dot{\boldsymbol{\Omega}} + \hat{\mathbf{g}}_s^T \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}) + I_{w_t} \hat{\mathbf{g}}_t^T \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + I_{w_t} \hat{\mathbf{g}}_g^T \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times \mathbf{H}_w. \quad (2.34)$$

Because the total angular momentum of the system needs to be conserved, as the torque exchanged between hub and wheel is an internal torque, Equations (2.31) and (2.34) add up to zero. Let's now define the combined inertia:

$$[I_{\text{tot}}] = [I_{\text{hub}}] + I_{w_t} (\hat{\mathbf{g}}_t^T \hat{\mathbf{g}}_t + \hat{\mathbf{g}}_g^T \hat{\mathbf{g}}_g) \quad (2.35)$$

which is the inertia of the hub, plus the contribution of the inertia of the wheel as it rotates with the hub's angular rate. Adding Equations (2.31) and (2.34) together and factoring in this last equation yields the following:

$$[I_{\text{tot}}] \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([I_{\text{tot}}] \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + I_{w_s}(\boldsymbol{\Omega} + \hat{\mathbf{g}}_s^T \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}})) = \mathbf{u} \quad (2.36)$$

where  $\mathbf{u}$  is the control torque experienced by the hub. The torque experienced by the wheel is:

$$u_w = -\hat{\mathbf{g}}_s^T \mathbf{u} = I_{w_s}(\dot{\boldsymbol{\Omega}} + \hat{\mathbf{g}}_s^T \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}). \quad (2.37)$$

The analysis carried out so far applied to a rigid hub equipped with a single reaction wheel. However, it can easily be generalized to a case with  $M$  reaction wheels. Equation (2.35) becomes in this case:

$$[I_{\text{tot}}] = [I_{\text{hub}}] + I_{w_t} \sum_{m=1}^M (\hat{\mathbf{g}}_{t_m}^T \hat{\mathbf{g}}_{t_m} + \hat{\mathbf{g}}_{g_m}^T \hat{\mathbf{g}}_{g_m}) \quad (2.38)$$

and Equation (2.36) becomes:

$$[I_{\text{tot}}] \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([I_{\text{tot}}] \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [G_s] \mathbf{h}_s) = \mathbf{u} \quad (2.39)$$

where:

$$\mathbf{h}_s = I_{w_s} (\boldsymbol{\Omega} + [G_s]^T \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}) \quad (2.40)$$

is the total wheel spin momentum. In this case,  $\boldsymbol{\Omega} = \{\Omega_1, \dots, \Omega_M\}^T$  is the  $M$ -dimensional vector containing the wheel spin rates, and  $[G_s] = [\hat{\mathbf{g}}_{s_1}, \dots, \hat{\mathbf{g}}_{s_M}]$  is a  $3 \times M$  matrix containing the spin axes. The individual torques experienced by each wheel are:

$$\mathbf{u}_w = \dot{\mathbf{h}}_s = I_{w_s} (\dot{\boldsymbol{\Omega}} + [G_s]^T \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}) \quad (2.41)$$

however, in order to accurately propagate the dynamics of the wheels, it is necessary to characterize the relationship between  $\mathbf{u}$  and  $\mathbf{u}_w$ . The first is the control torque that is required to actuate the spacecraft: control laws for such purpose are derived in the following Section 2.3. The second is a vector whose dimension is the space generated by the spin axes of the reaction wheels. In the presence of at least three linearly-independent RWs,  $\mathbf{u}$  can be uniquely mapped into a set of wheel torques  $\mathbf{u}_w$ . With less RWs, this mapping is in general not possible. With more than 3 RWs, there exist infinite ways to map  $\mathbf{u}$  into  $\mathbf{u}_w$ , and the choice of the mapping is left to the user. One possible mapping is:

$$\mathbf{u}_w = -[G_s]^T ([G_s][G_s]^T)^{-1} \mathbf{u} \quad (2.42)$$

which yields the minimum-norm solution for  $\mathbf{u}_w$ . Other solutions exist that can, for instance, yield the set of wheel torques that result in the minimum instantaneous power consumption [58].

### 2.3 Nonlinear Spacecraft Attitude Control

The previous section described a number of common attitude parameterizations. Any one of such attitude parameterizations can be used to describe the relative attitude between an inertial frame  $\mathcal{N}$ , the spacecraft body frame  $\mathcal{B}$ , and a reference frame  $\mathcal{R}$ , which usually describes the frame that the body frame is required to track. This section expands on how to develop the controls to drive the spacecraft body frame onto the desired reference frame, which can generally be time-varying, ensuring global asymptotic stability behavior.

### 2.3.1 Lyapunov Direct Method

Without aiming to provide a detailed and exhaustive description of nonlinear control methods, the purpose of this section is to highlight the key aspects of Lyapunov's direct method, and highlight how this applies to the design of globally and asymptotically stabilizing controls for the rotational dynamics of a rigid body. A function  $V(\mathbf{x})$  is a candidate Lyapunov function for a system  $\dot{\mathbf{x}} = f(\mathbf{x})$  if it is continuous and if, in a neighborhood of  $\mathbf{x}_r$  [59]:

- (1)  $V(\mathbf{x})$  is positive definite about  $\mathbf{x}_r$ ;
- (2)  $V(\mathbf{x})$  has continuous partial derivatives;
- (3)  $\dot{V}(\mathbf{x})$  is negative semi-definite about  $\mathbf{x}_r$ .

If one such function exists, the system is stable in the sense of Lyapunov about  $\mathbf{x}_r$ . Moreover, if  $\dot{V}(\mathbf{x})$  is negative definite about  $\mathbf{x}_r$ , the system is asymptotically stable. Sometimes proving the negative-definiteness of the derivative can be challenging, in which case it is possible to still prove the asymptotic stability of the system analyzing higher order derivatives [60]. If the function  $V(\mathbf{x})$  is also globally positive definite and radially unbounded, the system is globally stable, which means it is stable for any set of initial conditions.

### 2.3.2 MRP-based nonlinear PID control

For a general application in attitude dynamics and control, the state consists of the chosen attitude parameterization and angular rates of the rigid body: this is because Equation (2.30), coupled with any of the kinematic equations described in Section 2.1, constitute a nonlinear second-order system. From this point onwards, the analysis focuses on MRP as the go-to attitude parameterization of choice. The same analysis can be carried out with any other attitude parameterization, only the resulting controls laws would result slightly different. The controlled state is therefore  $\mathbf{x} = \{\boldsymbol{\sigma}_{B/R}, {}^B\boldsymbol{\omega}_{B/R}\}$ . Let's consider the following Lyapunov function:

$$V(\boldsymbol{\sigma}_{B/R}, \boldsymbol{\omega}_{B/R}) = \frac{1}{2}\boldsymbol{\omega}_{B/R}^T [I_C] \boldsymbol{\omega}_{B/R} + 2K \ln\left(1 + \sigma_{B/R}^2\right) + \frac{1}{2}\mathbf{z}^T [K_I] \mathbf{z} \quad (2.43)$$

where  $[K_I]$  is a positive definite matrix and the vector  $\mathbf{z}$  is defined as [61]:

$$\mathbf{z} = K \int_{t_0}^t \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} dt + [I_C] \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}}, \quad (2.44)$$

where the left superscript of the angular rate vector is dropped for ease of notation. The function  $V(\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}}, \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}})$  is positive definite, and its derivative is:

$$\dot{V}(\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}}, \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}}) = (\boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}} + [K_I] \mathbf{z})^T \left( [I_C] \boldsymbol{\omega}'_{\mathcal{B}/\mathcal{R}} + K \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} \right), \quad (2.45)$$

where the prime superscript denotes the body-frame derivative  $\boldsymbol{\omega}'_{\mathcal{B}/\mathcal{R}} = {}^{\mathcal{B}} \frac{d}{dt} \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}}$ . The negative-semidefiniteness of the derivative can be enforced by equating it to the expression:

$$\dot{V}(\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}}, \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}}) = - (\boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}} + [K_I] \mathbf{z})^T [P] (\boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}} + [K_I] \mathbf{z}) \quad (2.46)$$

with  $[P]$  positive definite. Equating Equations (2.45) and (2.46) leads to the closed-loop dynamics:

$$[I_C] \boldsymbol{\omega}'_{\mathcal{B}/\mathcal{R}} + K \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} + [P] \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}} + [P][K_I] \mathbf{z} = 0. \quad (2.47)$$

Applying the transport theorem to  $\boldsymbol{\omega}'_{\mathcal{B}/\mathcal{R}}$  in Equation (2.47) and plugging in Equation (2.30) gives the stabilizing control:

$$\begin{aligned} \mathbf{u} = & -K \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} - [P] \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}} - [P][K_I] \mathbf{z} \\ & + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times [I_C] \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [I_C] (\dot{\boldsymbol{\omega}}_{\mathcal{R}/\mathcal{N}} - \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times \boldsymbol{\omega}_{\mathcal{R}/\mathcal{N}}) - \mathbf{L}_{\text{ext}}. \end{aligned} \quad (2.48)$$

The control law in Equation (2.48) is designed to zero the relative attitude ( $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}}$ ) and angular velocity ( $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}}$ ) between the body frame  $\mathcal{B}$  and a reference frame  $\mathcal{R}$ . The integral term  $[P][K_I] \mathbf{z}$  has the ability to zero the bias in the steady-state response due to a constant, unmodeled torque acting on the system. The remaining terms arise due to the nonlinearities intrinsic to the rotational dynamics of a rigid body. So far, the control law is only shown to be stabilizing: the reader is redirected to Reference 50 for a detailed proof of the asymptotic stability guarantee of Equation (2.48). This control law is valid for a rigid body, because the dynamics contained in Equation (2.30) is also valid for a rigid body. The control law that is implemented throughout this work is meant

to be asymptotically stabilizing for a spacecraft equipped with reaction wheels, whose dynamics contribute to the gyroscopic coupling of the system as a whole. With this idea in mind, it is possible to define analogous Lyapunov functions using, instead of the hub's inertia tensor, the combined inertia tensor  $[I_{\text{tot},C}]$  about the system center of mass from Equation (2.38). This allows to substitute Equation (2.39) into Equation (2.47) and obtain the final asymptotically-stabilizing control law for a spacecraft with reaction wheels:

$$\begin{aligned} \mathbf{u} = & -K\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} - [P]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}} - [P][K_I]\mathbf{z} \\ & + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([I_{\text{tot},C}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [G_s]\mathbf{h}_s) + [I_{\text{tot},C}] (\dot{\boldsymbol{\omega}}_{\mathcal{R}/\mathcal{N}} - \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times \boldsymbol{\omega}_{\mathcal{R}/\mathcal{N}}) - \mathbf{L}_{\text{ext}}. \end{aligned} \quad (2.49)$$

## 2.4 Attitude Simulation Framework

This work employs Basilisk Astrodynamics Simulation Framework<sup>1</sup> to simulate the attitude dynamics illustrated in the previous sections of this chapter, as well as the guidance and control laws that are going to be developed in the following chapters. Basilisk is an open-source software framework that can simulate complex spacecraft systems in the space environment. The dual nature of Basilisk consists in its C/C++ core software modules, which ensure speed of execution, combined with a Python interface, to allow for easy scriptability and reconfigurability. Basilisk's main strength relies in its modular structure, which allows for minimal coupling between different segments of code that simulate different spacecraft behaviors. The minimal coupling between modules is enabled by Basilisk's messaging system: each module reads in input messages from other modules and outputs its own message(s), thus decoupling the data flow between modules and removing explicit intermodule dependency [62, 63].

Figure 2.2 shows the block diagram for a relative simple Basilisk scenario, where the goal is to use a set of three RWs to drive the spacecraft attitude to track the Hill reference frame. Each block in the diagram represents a module in Basilisk. It can be immediately noticed how the modules are divided into two main tasks: the Dynamics Task, and the Flight Software Task. A task in

<sup>1</sup> <https://hanspeterschaub.info/basilisk>



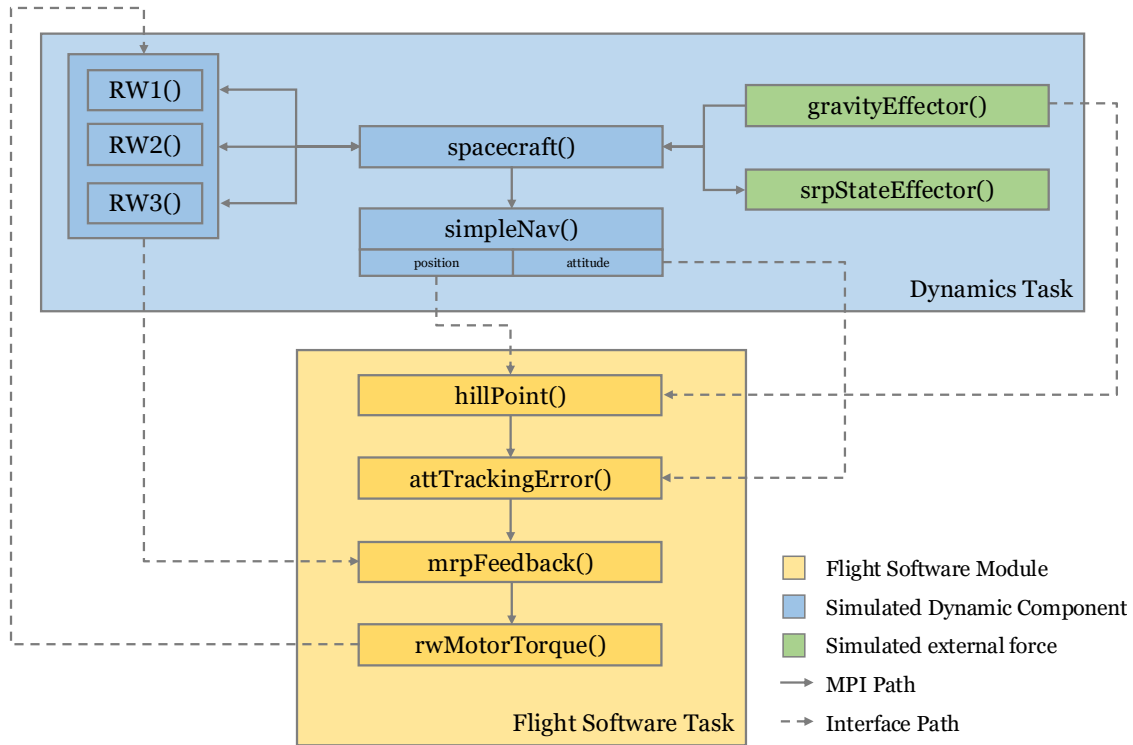


Figure 2.2: Basilisk simulation block diagram for Hill pointing with RWs

Basilisk is a grouping of modules, which are updated with a fixed integration rate. Multiple tasks can be used within a simulation, especially when different integration steps are required to capture the behavior of certain high-frequency phenomena. Tasks can be switched on and off according to the necessity [62]. Dynamics is usually updated at a higher frequency than flight software, as dynamics aims to simulate the infinite resolution of real physical phenomena. Flight software, on the contrary, consists of a set of modules that mirror the flight software onboard a spacecraft. The interfaces from dynamics to flight software modules are typically sensor measurements, whereas from flight software to dynamics they come in the form of controls.

The block diagram in Figure 2.2 is not meant to be a comprehensive description of the capabilities built into Basilisk, but rather to give an intuition about the complexity that exists behind a high-fidelity simulation of spacecraft behavior, and the number of smaller building blocks that need to come together to constitute the simulation as a whole.

- `spacecraft()`: this module solves the position and attitude differential equations of the spacecraft hub. It contains information such as spacecraft mass and inertia, and outputs a message containing information about the inertial position of the spacecraft and its center of mass, together with attitude, angular rates and accelerations of the body-fixed frame with respect to the inertial frame.
- `simpleNav()`: this module adds error on top of the message that it receives from the `spacecraft()` module. The motivation for this module is to provide a realistic navigation signal, in order to test the guidance and control modules in presence of signal errors. Such error is modelled as a Gauss-Markov process.
- `RW()`: this module creates a set of reaction wheels. These can be generated from a database of existing wheels, or they can be customized by the user. When generating a RW, the user must specify the body-frame direction of the spin axis of the wheel  ${}^B\hat{g}_s$ . Optional parameters can be provided such as initial wheel speed, maximum speed and/or maximum momentum.
- `gravityEffector()`: this module is used to create gravity bodies such as Earth and Sun. Earth is used as the primary center of gravity around which the spacecraft is orbiting. The Sun is generated to play the role of the bright celestial body about which the constraints are defined.
- `srpStateEffector()`: this module computes the forces and torques acting on the spacecraft due to solar radiation pressure (SRP). SRP is calculated based on the current relative distance and attitude with respect to the Sun, and information on the geometry of the spacecraft provided by the user.
- `hillPoint()`: this module computes the Hill reference frame for the spacecraft given its relative position to the gravity body.
- `attTrackingError()`: computes and outputs the relative attitude  $\sigma_{B/\mathcal{R}}$  of the spacecraft

with respect to the reference, the relative angular rates  ${}^B\boldsymbol{\omega}_{B/\mathcal{R}}$  and accelerations  ${}^B\dot{\boldsymbol{\omega}}_{B/\mathcal{R}}$  in body-frame components.

- `mrpFeedback()`: this module computes the required control torque on the spacecraft according to Equation (2.49). It receives the messages containing the relative attitude between spacecraft and reference frames, mass and inertia properties of the spacecraft, and RW states, and computes a commanded torque in body-frame components.
- `rwMotorTorque()`: this module maps the required torque into individual reaction wheel motor torques, depending on the RW configuration and availability of the wheels, according to Equation (2.6).

In Basilisk, the `spacecraft()` module is the cornerstone for the integration of the equations of motion of the whole system. All the other building blocks that contribute to the system dynamics are commonly referred to as “effectors”, and are divided into two categories. Dynamic effectors exchange forces and torques with the spacecraft, but do not require continuous integration of the effector’s own state to compute such forces and torques. In general terms, dynamic effectors are used to represent external perturbations such as gravity, SRP, or thruster firings. State effectors, on the other hand, require the continuous integration of the effector’s state in order to compute the forces and torques exchanged with the spacecraft. Examples of these effectors are MEDs and rotating or flexing appendages connected to the spacecraft hub. The integration of the fully-coupled system dynamics is performed inside the `spacecraft()` class, which receives the contributions from all the connected effectors. Within the `spacecraft()` class the back-substitution method is used to decouple the spacecraft mass matrix from that of all the effectors, while preserving the fully coupled nature of the dynamical system. This method allows to solve for the spacecraft’s own states first (position, velocity, attitude, and angular rates), and later back-substitute these values into the actuators’ equations of motion to integrate the actuators’ states. The back-substitution method and its application in Basilisk has been extensively discussed in past contributions [64, 65, 66, 67].

The modular, open-source nature of Basilisk allows the user to rapidly prototype modules

that can be connected at any level within the block diagram and solve additional, customizable tasks. An example of this is complex attitude reference trajectory tracking thanks to the subdivision of the attitude tracking segment of code into modular components, consisting in a base pointing reference, an attitude offset and a dynamic reference trajectory relative to the base that is to be tracked [68].

Figure 2.3 shows some simulation results for the Hill-point scenario described by the block diagram in Figure 2.2. Figure 2.3a compares the spacecraft attitude  $\sigma_{B/N}$  with the reference attitude  $\sigma_{R/N}$  computed in the `hillPoint()` flight software module. Figure 2.3b shows the relative MRP error  $\sigma_{B/R}$  between these two frames drop to zero asymptotically. It is interesting to see that, while Figure 2.3a shows a discontinuity due to MRP switching, in order to maintain the norm  $\sigma \leq 1$ , this discontinuity does not appear in Figure 2.3b, because the relative error is always computed about the short rotation. As a result, the spacecraft is steered towards the reference attitude following the shortest rotation. Figures 2.3c and 2.3d show analogous results for the angular rates. It is possible to observe that one angular rate component in Figure 2.3c is driven to a constant value to compensate for the orbital motion of the spacecraft as it orbits around Earth. Figures 2.3e and 2.3f show, respectively, RW relative speeds and commanded torques. Because the RWs are aligned with the principal inertia axes of the spacecraft, these torques are opposite to the torque components experienced by the spacecraft along these axes. Figure 2.3f shows the torques requested as a result of `mrpFeedback()` in dashed lines, compared to the actual torques delivered by the RWs in solid. The RW along the third axis reaches the maximum allowable torque of 0.2 N and therefore cannot provide the requested torque, however this does not prevent the spacecraft to eventually converge to the desired attitude. Because the spacecraft is actuated from a rest position to a non-rest reference, the RW along the component of the orbital angular velocity converges to a non-zero value, as a result of the conservation of the zero net momentum of the system. Lastly, Figure 2.3g shows the SRP torque acting on the system, which varies with the spacecraft attitude, as a result of the surfaces changing relative orientation with respect to the Sun.

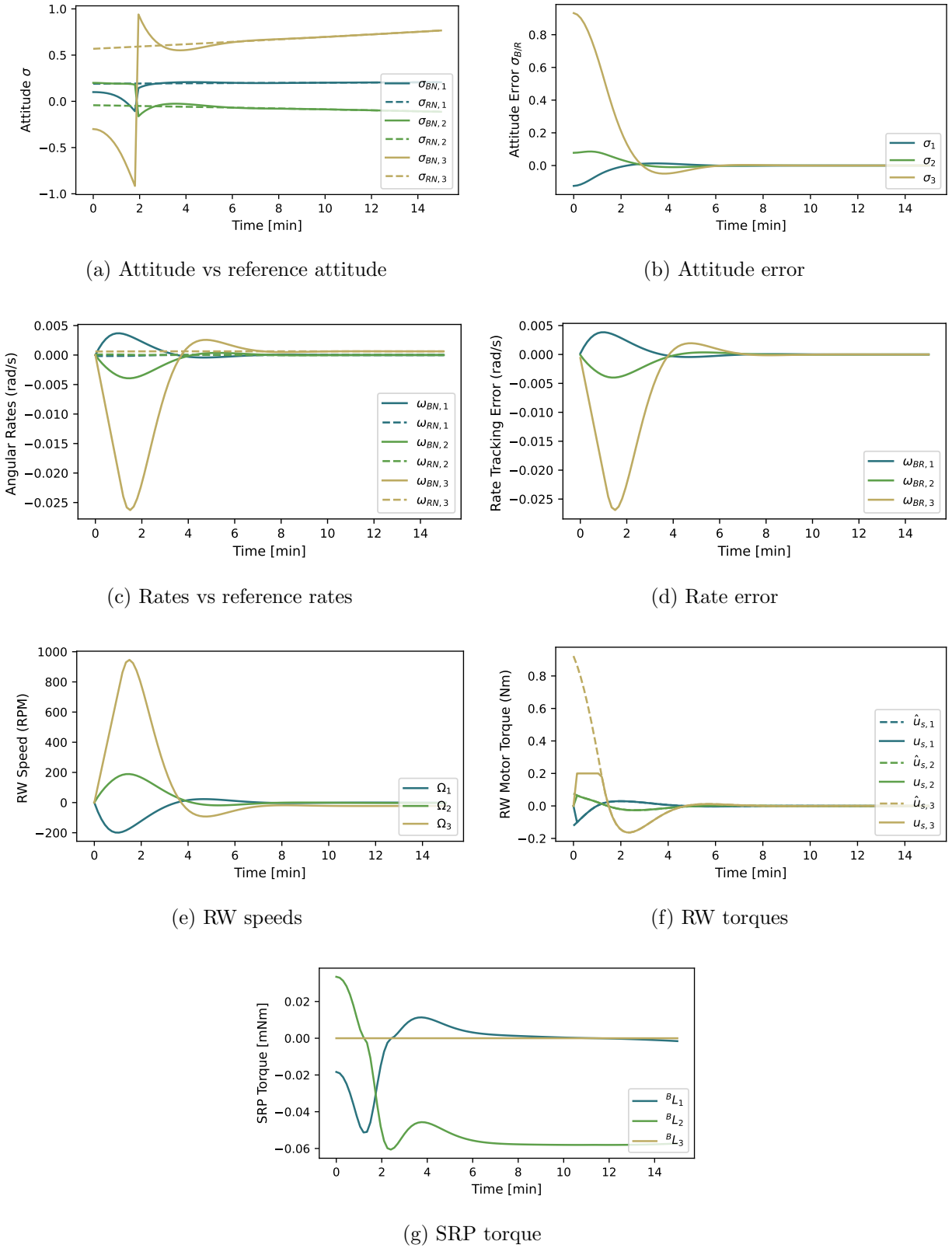


Figure 2.3: Simulation results for Hill-point scenario with RWs

## Guidance for Spacecraft Attitude and Moving Components

Multiple spacecraft attitude pointing constraints can pose a challenge to the definition of the optimal guidance for the spacecraft attitude and its subcomponents: with attitude being an inherently 3-dimensional variable, enforcing multiple pointing constraints can quickly turn into an overdetermined problem. Additionally, when some of these constraints are time-varying, it is not trivial to determine whether multiple pointing constraints can be enforced simultaneously at all times. Moreover, spacecraft often present additional, softer constraints such as thermal-sensitive surfaces or equipment that must be kept away from the Sun, and which add another layer of complexity to the optimal attitude reference definition problem.

Literature documents on ESA's SMART-1 mission cite that during nominal operations the thruster is aligned with the inertial requested direction, while the SADA axes are maintained at a 'shallow angle from the Sun direction' [25], signaling that the thruster alignment constitutes the main driver in defining the attitude reference, whereas the SA pointing does not seem to be the object of any rigorous optimization process. Due to the nature of the mission, which targeted Earth's Moon, it is reasonable to assume that suboptimal illumination on the arrays would have been acceptable, due to the proximity to the Sun and the fact that the heliocentric distance of the spacecraft does not vary much throughout the mission. More interesting information regarding the reference frame definitions are found for the Psyche spacecraft. During thrust cruise the spacecraft is first maneuvered to a configuration in which the thruster is inertially aligned. Then, the spacecraft is rolled about the thrust line and the SAs articulated about their axis until maximum

illumination is achieved [69]. This strategy appears to be time-inefficient and heavily reliant on sensor measurements, and lacks the ability to deterministically define the optimal reference frame for the spacecraft hub that satisfies both constraints simultaneously. A more thorough description is found on the GNC flight software that computes the reference attitude for the Dawn spacecraft, where the same logical steps are taken as for Psyche, but the full 3-axis compliant solutions are computed analytically beforehand, and in case of multiple solutions, one is chosen in order to optimize Sun exposure on a thermal sensitive surface of the hub [70]. Still, the mathematical formulation of the attitude reference generation algorithm is not presented, and the description does not provide a thorough description of the solution spaces as functions of the spacecraft design and mission parameters.

This section develops a series of guidance laws for the spacecraft and its articulating components, such as the dual-gimbaled thruster platform and the rotating solar arrays. These guidance laws are based on the assumption that the pointing requirements can be reduced to a set of geometrical equalities and inequalities between vectors in 3D-space. An assumption made in this work is that, in the presence of moving components that participate in determining the pointing requirements for the spacecraft, the guidance laws that govern the motion of such components can be decoupled from the guidance algorithm to compute the optimal reference attitude itself. The flow of information between the different guidance modules for the different subcomponents is therefore reduce to the essential [62], which in this work takes the form of a series of headings in body-frame coordinates. Ultimately, this approach aims to ensure the computation of the correct references for each spacecraft subcomponent, while maintaining each guidance module conceptually simple, easy to maintain, verify, and validate individually [68].

### 3.1 Platform Guidance

The platform supporting the electric thruster is modeled as a rigid body attached to the main spacecraft hub, as shown in Figure 1.2. To describe the relative motion between the two, two frames are defined: frame  $\mathcal{M} = \{\hat{\mathbf{m}}_1, \hat{\mathbf{m}}_2, \hat{\mathbf{m}}_3\}$  is a hub-fixed frame, whose origin  $M$  coincides

with the joint through which the platform and hub are connected and exchange forces. Frame  $\mathcal{F} = \{\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \hat{\mathbf{f}}_3\}$ , with origin  $F$ , is a platform-fixed frame. The complexity of the problem involving the articulation of the platform lies in the fact that the thrust vector is expressed in  $\mathcal{F}$  frame coordinates and applied through point  $T$ , which may not coincide with the origin  $F$  of the frame. Additionally, the origins of the two frames  $M$  and  $F$ , in general, do not coincide either. This section describes how to compute the reference gimbal angles  $\nu_{R1}$  and  $\nu_{R2}$  that align the thrust direction vector  $\hat{\mathbf{t}}$ , expressed in  $\mathcal{F}$ -frame coordinates, through a specific point  $C$  in the hub. This point  $C$  can coincide with the system's center of mass, if the desire is to have the thrust not produce a torque on the system. Conversely, it is possible to intentionally thrust at an offset distance from the center of mass to exploit such torque to perform momentum dumping. This section does not consider uncertainty on the location of point  $C$  as, for this analysis, the assumption is that the coordinates of point  $C$  are known. The final direction cosine matrix that aligns the thrust with the desired point  $C$  is obtained via a series of three consecutive rigid body rotations that take advantage of intermediate frames.

### 3.1.1 Gimbaled Platform and Thruster Modeling

The relative motion between the platform and the hub consists of 2 degrees of freedom tip-and-tilt rotations. Such rotation angles  $\nu_1$  and  $\nu_2$  are defined about the  $\hat{\mathbf{m}}_1$  axis and the intermediate  $\hat{\mathbf{f}}_2$  axis, respectively, via two consecutive rotations. The direction cosine matrix that defines the mapping from  $\mathcal{M}$  to  $\mathcal{F}$  is therefore:

$$[\mathcal{FM}] = \begin{bmatrix} \cos \nu_2 & \sin \nu_1 \sin \nu_2 & -\cos \nu_1 \sin \nu_2 \\ 0 & \cos \nu_1 & \sin \nu_1 \\ \sin \nu_2 & -\sin \nu_1 \cos \nu_2 & \cos \nu_1 \cos \nu_2 \end{bmatrix} \quad (3.1)$$

where emphasis is put on the fact that the element (2,1) of the  $[\mathcal{FM}]$  direction cosine matrix must be zero. This ensures that the rotation is, in fact, a tip-and-tilt type of rotation that is compliant with the constrained motion of the platform. Moreover, it can be observed that when  $\nu_1 = \nu_2 = 0$  the two frames align and the mapping becomes an identity.



To solve the problem in its most general form, the following offset vectors are defined:

- ${}^{\mathcal{M}}\mathbf{r}_{C/M}$ : position of  $C$  with respect to  $M$ , expressed in  $\mathcal{M}$ -frame coordinates;
- ${}^{\mathcal{F}}\mathbf{r}_{F/M}$ : position of  $F$  with respect to  $M$ , expressed in  $\mathcal{F}$ -frame coordinates;
- ${}^{\mathcal{F}}\mathbf{r}_{T/F}$ : position of  $T$  with respect to  $F$ , expressed in  $\mathcal{F}$ -frame coordinates.

For ease of notation, the following vector are defined and used in the following subsections:

$$\mathbf{a} = \mathbf{r}_{F/M} + \mathbf{r}_{T/F} \qquad \mathbf{b} = \mathbf{r}_{C/M}. \quad (3.2)$$

### 3.1.2 Closed-form Guidance Solution

#### 3.1.2.1 First Rotation

The first rotation is defined starting from the hub-fixed frame  $\mathcal{M}$ , and it maps to an intermediate frame  $\mathcal{D}_1$  such that, when  $\mathcal{F} \equiv \mathcal{D}_1$ , the thrust vector  $\mathbf{t}$  is parallel to  $\mathbf{b}$ . The DCM  $[\mathcal{D}_1\mathcal{M}]$  is defined by means of principal rotation angle and principal rotation vector  $(\phi, \hat{\mathbf{e}}_\phi)$ . These quantities are computed as follows:

$$\phi = \arccos\left(\frac{{}^{\mathcal{F}}\mathbf{t} \cdot {}^{\mathcal{M}}\mathbf{b}}{\|{}^{\mathcal{F}}\mathbf{t}\| \cdot \|{}^{\mathcal{M}}\mathbf{b}\|}\right) \qquad \hat{\mathbf{e}}_\phi = \frac{{}^{\mathcal{F}}\mathbf{t} \times {}^{\mathcal{M}}\mathbf{b}}{\|{}^{\mathcal{F}}\mathbf{t} \times {}^{\mathcal{M}}\mathbf{b}\|}. \quad (3.3)$$

From which  $[\mathcal{D}_1\mathcal{M}]$  is computed using Equation (2.5). Figure 3.1a visually shows the rotation of vectors performed by the first DCM. It should be noted that the matrix  $[\mathcal{D}_1\mathcal{M}]$  is, in general, not

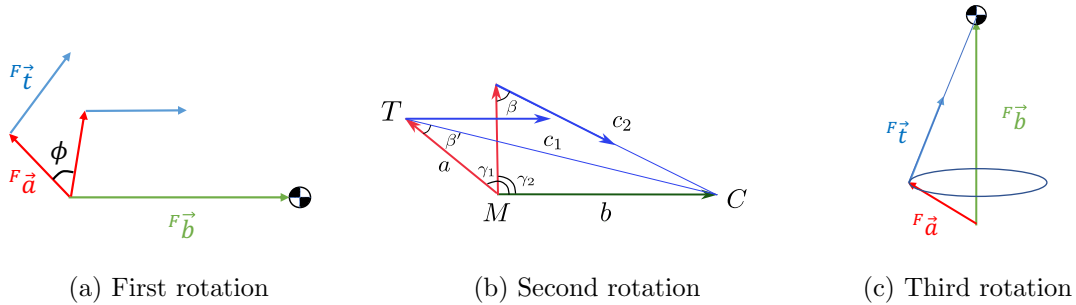


Figure 3.1: Three consecutive rotations

constraint compliant, i.e., it does not describe a tip-and-tilt rotation like the DCM outlined in Equation (3.1).

### 3.1.2.2 Second Rotation

The second rotation puts the thrust direction vector through point  $C$ . The principal rotation vector in this case is again  $\hat{\mathbf{e}}_\psi = \hat{\mathbf{e}}_\phi$ , so the rotation is a direct continuation of the previous one. The derivation of the second principal rotation angle  $\psi$ , however, is significantly less intuitive. When the thrust application point  $T$  coincides with the origin of the hub-fixed frame  $M$ , therefore  $\|\mathbf{a}\| = 0$ , it is also  $\psi = 0$ , and the first rotation alone is sufficient to put the thrust through point  $C$ . When this is not the case, the nonzero value of  $\psi$  must be computed. Figure 3.1b shows the second rotation that aligns the thrust vector, in blue and initially parallel to  $\mathbf{b}$ , to point  $C$ . The principal rotation angle  $\psi$  is obtained as  $\psi = \gamma_1 - \gamma_2$ . The angles  $\beta$ ,  $\beta'$  and  $\gamma_1$  are known from the geometry and the relative position of the vectors at the beginning of the second rotation. The segments  $c_1$  and  $c_2$ , respectively the distance between the thrust application point  $T$  and point  $C$  before and after the rotation, are computed as follows:

$$c_1 = \sqrt{a^2 + b^2 - 2ab \cos \gamma_1} \qquad c_2 = a \cos \beta \pm \sqrt{b^2 - a^2 \sin^2 \beta}. \quad (3.4)$$

It can be observed that  $c_2$  only exists when the relation  $b \geq a |\sin \beta|$  is satisfied. This means that it might not always be possible to align the thrust vector with point  $C$  for any choice of vectors  $\mathbf{a}$  and  $\mathbf{b}$ . Taking the product of the two solutions for  $c_2$  given in Equation (3.4) gives:

$$c_{2,1} \cdot c_{2,2} = a^2 \cos^2 \beta - b^2 + a^2 \sin^2 \beta = a^2 - b^2 \quad (3.5)$$

which is negative when  $a < b$ . This is considered a valid assumption for the following reasons: vector  $\mathbf{b}$  indicates the position of point  $C$ , which in general coincides with the CM of the system, with respect to point  $M$ . For a massive spacecraft, the system CM is located close to the geometric center of the system, thus away from the point  $M$ , which is at the lower end of the hub where the platform is attached. Vector  $\mathbf{a}$ , instead, is the offset between the thrust application point  $T$  and

point  $M$ . Assuming that the platform is relatively small compared to the spacecraft hub, it makes sense to assume that  $a < b$ . With these considerations, the choice for  $c_2$  is:

$$c_2 = a \cos \beta + \sqrt{b^2 - a^2 \sin^2 \beta}. \quad (3.6)$$

The principal rotation angle  $\gamma$  is derived from trigonometric relations:

$$\sin \psi = \sin(\gamma_1 - \gamma_2) = \cos \gamma_2 \sin \gamma_1 - \cos \gamma_1 \sin \gamma_2 = \frac{1}{b}(c_1 \cos \gamma_2 \sin \beta' - c_2 \cos \gamma_1 \sin \beta) \quad (3.7)$$

with:

$$\cos \gamma_i = \frac{a^2 + b^2 - c_i^2}{2ab} \quad \text{for} \quad i = 1, 2. \quad (3.8)$$

Ultimately, the second DCM is derived as  $[\mathcal{D}_2 \mathcal{D}_1] = f(\psi, \hat{e}_\psi)$  using Equation (2.5) once more.

Multiplying the first and second rotation matrices gives the DCM:

$$[\mathcal{D}_2 \mathcal{M}] = [\mathcal{D}_2 \mathcal{D}_1][\mathcal{D}_1 \mathcal{M}] \quad (3.9)$$

where again, in general,  $[\mathcal{D}_2 \mathcal{M}]$  does not describe a rotation that is compliant with the constraints expressed by Equation (3.1).

### 3.1.2.3 Third Rotation

The first two rotations show that it is possible, under reasonable assumptions, to rotate the thrust vector in order to align it with point  $C$  in the hub. However, it is not yet possible to do so through a rotation that is also compliant with the platform's constraints, which only allow tip-and-tilt kind of rotations. This subsection computes a third rotation DCM that maintains the alignment condition achieved by the first two rotations, and simultaneously enforces the constraint compliance on the final solution. Equation (3.9) provides a rotation matrix that puts the thrust vector through point  $C$ , but this solution is not unique. In fact, there exists a family of infinite DCMs that accomplish the same. Of all such DCMs, the purpose of this section is to find the one that is also constraint-compliant. Let's define such DCM as  $[\mathcal{D}_3 \mathcal{M}]$ . Figure 3.1c shows that

rotations about the  $\mathbf{b}$  vector do not break the desired alignment of the thrust vector. For this reason, the principal rotation vector of the third rotation,  $\hat{\mathbf{e}}_\theta$ , is defined as:

$$\hat{\mathbf{e}}_\theta = \frac{\mathbf{b}}{\|\mathbf{b}\|}. \quad (3.10)$$

The unknown variable in this problem is the principal rotation angle  $\theta$ . Let's define the CRP set:

$$\mathbf{q} = \hat{\mathbf{e}}_\theta \tan\left(\frac{\theta}{2}\right) \quad (3.11)$$

from which the third DCM  $[\mathcal{D}_3\mathcal{D}_2]$  is obtained applying Equation (2.16). The goal is to obtain a final DCM  $[\mathcal{F}\mathcal{M}]$  such that:

$$[\mathcal{F}\mathcal{M}] = [\mathcal{D}_3\mathcal{M}] = [\mathcal{D}_3\mathcal{D}_2][\mathcal{D}_2\mathcal{D}_1][\mathcal{D}_1\mathcal{M}] \quad (3.12)$$

is of the same form as Equation (3.1). Defining the following quantities component-wise:

$$\hat{\mathbf{e}}_\theta = (e_1, e_2, e_3)^T \quad [\mathcal{D}_2\mathcal{M}] = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix}, \quad (3.13)$$

carrying out the product  $[\mathcal{D}_3\mathcal{M}] = [\mathcal{D}_3\mathcal{D}_2][\mathcal{D}_2\mathcal{M}]$  and equating the element (2, 1) of  $[\mathcal{D}_3\mathcal{M}]$  to zero to meet the rotational constraint, gives the following equation:

$$\frac{At^2 + Bt + C}{1 + t^2} = 0 \quad \text{with} \quad t = \tan\left(\frac{\theta}{2}\right) \quad (3.14)$$

where:

$$\begin{aligned} A &= 2(d_{21}e_2^2 + d_{11}e_1e_2 + d_{31}e_2e_3) - d_{21} \\ B &= 2(d_{31}e_1 - d_{11}e_3) \\ C &= d_{21}. \end{aligned} \quad (3.15)$$

Equation (3.14) has solutions when  $\Delta = B^2 - 4AC \geq 0$ . In such case, it is:

$$t = \frac{-B \pm \sqrt{\Delta}}{2A} \quad (3.16)$$

which can be plugged back into Equation (3.11). Finally, Equation (3.12) yields the rotation  $[\mathcal{FM}]$  that complies with all the requirements. Defining  $f_{ij}$  for  $i, j = 1, 2, 3$  the elements of the  $[\mathcal{FM}]$  matrix, the tip and tilt reference angles  $\nu_{R1}$  and  $\nu_{R2}$  in Equation (3.1) are obtained as:

$$\nu_{R1} = \arctan\left(\frac{f_{23}}{f_{22}}\right) \quad \nu_{R2} = \arctan\left(\frac{f_{31}}{f_{11}}\right). \quad (3.17)$$

### 3.2 Attitude Guidance

The goal of this section is to define a reference frame  $\mathcal{R}$  for the spacecraft, which satisfies a series of pointing requirements. Each individual pointing requirement analyzed in this paper can be expressed as an inequality that describes the desired minimum or maximum angle between a certain unit direction vector in body/reference-frame coordinates, and another unit direction vector in inertial frame coordinates:

$$\begin{cases} \mathcal{R}\hat{e}_1 \cdot [\mathcal{RN}]^{\mathcal{N}}\hat{e}_2 \geq \cos \theta_1 \\ \mathcal{R}\hat{e}_1 \cdot [\mathcal{RN}]^{\mathcal{N}}\hat{e}_2 \leq \cos \theta_2 \end{cases} \iff \cos \theta_1 \leq \mathcal{R}\hat{e}_1 \cdot [\mathcal{RN}]^{\mathcal{N}}\hat{e}_2 \leq \cos \theta_2 \quad (3.18)$$

Equation (3.18) describes a generic constraint on the angle between a certain body-fixed unit vector  $\hat{e}_1$  and an inertially-defined vector  $\hat{e}_2$ . The dot product between the two vectors is taken after the inertial unit direction vector is mapped onto the reference frame. The constraint defined by Equation (3.18) means is that, when the body frame tracks the reference frame, the angle  $\theta$  between vectors  $\hat{e}_1$  and  $\hat{e}_2$  is such that  $\cos \theta_1 \leq \cos \theta \leq \theta_2$ . Because the central expression in Equation (3.18) is the dot product between two unit vectors, it is always true that:

$$-1 \leq \mathcal{R}\hat{e}_1 \cdot [\mathcal{RN}]^{\mathcal{N}}\hat{e}_2 \leq 1. \quad (3.19)$$

If  $\theta_1 = \theta_2 = \theta$ , Equation (3.18) can collapse into:

$$\mathcal{R}\hat{e}_1 \cdot [\mathcal{RN}]^{\mathcal{N}}\hat{e}_2 = \cos \theta. \quad (3.20)$$

The following types of pointing requirements, ordered by priority, are considered in this paper:

- (1) Vector alignment: this is the case in which, for example, the body-frame thrust direction must be aligned with an inertial requested direction. Alternatively, it can be the case

where the body-fixed high-gain antenna (HGA) must be aligned along the direction of Earth relative to the spacecraft for telecommunication purposes. Referring with  $\hat{\mathbf{h}}_1$  to the body vector, and with  $\hat{\mathbf{h}}_{R1}$  to the inertial request vector, this requirement is expressed as:

$${}^{\mathcal{R}}\hat{\mathbf{h}}_1 \cdot [\mathcal{RN}]^{\mathcal{N}}\hat{\mathbf{h}}_{R1} = 1, \quad (3.21)$$

where  ${}^{\mathcal{R}}\hat{\mathbf{h}}_1 = {}^{\mathcal{B}}\hat{\mathbf{h}}_1$  under the assumption that the body frame will ultimately converge to the reference frame. This vector alignment alone locks two out of the three DoFs that characterize an attitude problem. Therefore, when present, it only leaves the space of rotations about the  $\hat{\mathbf{h}}_{R1}$  axis to optimize for other potential pointing requirements.

- (2) Solar array Sun incidence: under the assumption of rotating solar arrays, this requirement is dictated by power requirements. Let's define  $\hat{\mathbf{s}}$  as the unit direction vector of the Sun's position relative to the spacecraft, and with  $\hat{\mathbf{a}}$  the unit direction of the solar array drive axis (SADA) in the body frame, about which the arrays can rotate. In the presence of two SAs whose drive axes are opposite with respect to one-another, it is sufficient to pick either as a valid description for  $\hat{\mathbf{a}}$  without invalidating any of the results shown in this paper. The description does however fail when the SADAs are not collinear. The pointing requirement for the solar arrays is described by:

$$|{}^{\mathcal{R}}\hat{\mathbf{a}} \cdot [\mathcal{RN}]^{\mathcal{N}}\hat{\mathbf{s}}| \leq \sin \gamma, \quad (3.22)$$

where  $\gamma$  indicates the incidence angle of sunlight, measured from the normal to the power-generating surface of the solar arrays, as shown in Figure 3.2. While it is out of the scope of this paper to go into the details of what the optimal  $\gamma$  is for the different phases of a space mission, it is relevant to acknowledge that this parameter is likely to vary depending on the relative position of the spacecraft and the Sun. Because irradiance decreases quadratically with the distance from the Sun [11],  $\gamma \rightarrow 0$  is the desired outcome in outer regions of the

Solar System. Substituting this condition into Equation (3.22) results in:

$$\mathcal{R}\hat{\mathbf{a}} \cdot [\mathcal{R}\mathcal{N}]^{\mathcal{N}}\hat{\mathbf{s}} = 0, \quad (3.23)$$

which is the condition for maximum power generation. This condition is equivalent to stating that the SADA must be perpendicular to the direction of incoming sunlight, and it locks one degree of freedom for the spacecraft attitude. In the inner regions of the Solar System it is possible to generate more power with suboptimal illumination conditions, due to the higher irradiance experienced by the spacecraft. In this case, the condition  $0^\circ < \gamma < 90^\circ$  substituted into Equation (3.22) results in an even larger solution space, which can accommodate additional pointing requirements.

- (3) Keep-in/out zone: this third constraint is conceptually analogous to the first, but the alignment requirement is relaxed. This category describes a variety of requirements, such as thermally-sensitive surfaces or optical instruments that need to be kept at a certain angular distance from the Sun, or a low-gain antenna (LGA) that must point at Earth within an certain angular distance. The problem is formulated similarly, with a body-frame direction  $\hat{\mathbf{h}}_2$ , representing the instrument, and an inertial-frame direction  $\hat{\mathbf{h}}_{R2}$  that describes the axis of the conical keep-in or keep-out zone. The pointing requirement is

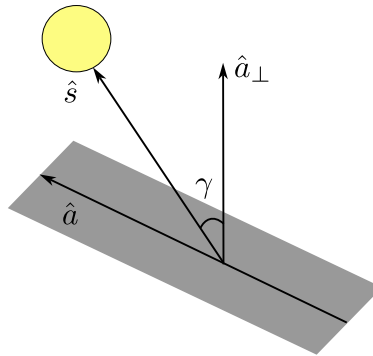


Figure 3.2: Sun incidence on array power-generating surface

described by:

$${}^{\mathcal{R}}\hat{\mathbf{h}}_2 \cdot [{}^{\mathcal{R}}\mathcal{N}] {}^{\mathcal{N}}\hat{\mathbf{h}}_{R2} \geq \cos \theta, \quad (3.24)$$

where  $\theta$  describes the half angle of the cone. Note that Equation (3.24) most intuitively describes a keep-in zone, but it can equivalently describe a keep-out zone when substituting  $\hat{\mathbf{h}}_{R2}$  with the opposite vector  $-\hat{\mathbf{h}}_{R2}$ , and  $\theta$  with  $\pi - \theta$ .

### 3.2.1 Inertial Pointing with Maximum Power

This subsection investigates the compatibility of requirements 1, 2, and 3, when applied simultaneously. This problem describes the case of a spacecraft like Psyche, where the thrust must be aligned at all times with the inertial requested vector, and at the same time the spacecraft must be able to generate as much power as possible. Additionally, it is desirable to point a thermally sensitive surface away from the Sun, to ensure heat dissipation. The requirements 1, 2, and 3 are therefore presented in order of importance.

The goal of this section is to define a reference frame  $\mathcal{R}$  that satisfies the pointing constraints. This is done by means of two consecutive rigid body rotations involving an intermediate frame  $\mathcal{D}$ :

$$[{}^{\mathcal{R}}\mathcal{N}] = [{}^{\mathcal{R}}\mathcal{D}][{}^{\mathcal{D}}\mathcal{N}]. \quad (3.25)$$

#### 3.2.1.1 Thruster Alignment

Enforcing requirement 1 is relatively straightforward. The principal rotation parameters (PRPs) are a set composed of a unit direction principal rotation vector (PRV) and a principal rotation angle (PRA), and they fully describe the relative attitude between two frames. For the intermediate frame DCM  $[{}^{\mathcal{D}}\mathcal{N}]$  they are obtained as:

$${}^{\mathcal{N}}\hat{\mathbf{e}}_\phi = \frac{{}^{\mathcal{N}}\hat{\mathbf{h}}_1 \times {}^{\mathcal{N}}\hat{\mathbf{h}}_{R1}}{|{}^{\mathcal{N}}\hat{\mathbf{h}}_1 \times {}^{\mathcal{N}}\hat{\mathbf{h}}_{R1}|} \quad (3.26a)$$

$$\phi = \arccos \left( {}^{\mathcal{N}}\hat{\mathbf{h}}_1 \cdot {}^{\mathcal{N}}\hat{\mathbf{h}}_{R1} \right), \quad (3.26b)$$



where the body-frame heading is preemptively mapped from current  $\mathcal{B}$ -frame coordinates to  $\mathcal{N}$ -frame coordinates. From Equation (3.26) the DCM  $[\mathcal{DN}]$  is readily computed. See Ref. [50] for a full description of PRPs. The DCM  $[\mathcal{DN}]$  so obtained describes a rotation from the inertial frame to a frame  $\mathcal{D}$  in which the pointing requirement on the thruster is satisfied.

### 3.2.1.2 Maximum Power Generation

The space of rotations about the thrust axis describes a set of frames, all compliant with the thruster pointing requirement. It is within this space that the second intermediate DCM  $[\mathcal{RD}]$  is to be found, in order to maximize the incidence of sunlight on the solar arrays. The PRV for the second intermediate rotation  $[\mathcal{RD}]$  coincides with the thruster heading:

$${}^{\mathcal{R}}\hat{\mathbf{e}}_{\psi} = {}^{\mathcal{R}}\hat{\mathbf{h}}_1 = {}^{\mathcal{R}}\hat{\mathbf{h}}_{R1}, \quad (3.27)$$

from which it is possible to define the CRP set (or Gibbs Vector):

$$\mathbf{q} = \tan\left(\frac{\psi}{2}\right) {}^{\mathcal{R}}\hat{\mathbf{e}}_{\psi} = t \cdot {}^{\mathcal{R}}\hat{\mathbf{e}}_{\psi}. \quad (3.28)$$

The DCM  $[\mathcal{RD}]$  is computed using Equation (2.16), and it is ultimately a function of the variable  $t = \tan(\psi/2)$ . Maximizing power generation is equivalent to requiring that the sunlight is perpendicular to the SADAs, or as close to perpendicular as possible. This translates into minimizing the expression on the left-hand side of equation Equation (3.22)

$$|f| = |{}^{\mathcal{R}}\hat{\mathbf{a}} \cdot [\mathcal{RD}][\mathcal{DN}]^{\mathcal{N}}\hat{\mathbf{s}}| = |{}^{\mathcal{R}}\hat{\mathbf{a}} \cdot [\mathcal{RD}]^{\mathcal{D}}\hat{\mathbf{s}}|. \quad (3.29)$$

Using the DCM expression of  $[\mathcal{RD}]$  as function of  $t$ , the quantity  $f$  can be expressed as:

$$f(t) = \frac{At^2 + Bt + C}{1 + t^2} \quad (3.30)$$

where:

$$A = 2({}^{\mathcal{D}}\hat{\mathbf{s}} \cdot {}^{\mathcal{R}}\hat{\mathbf{e}}_{\psi}) ({}^{\mathcal{R}}\hat{\mathbf{a}} \cdot {}^{\mathcal{R}}\hat{\mathbf{e}}_{\psi}) - {}^{\mathcal{D}}\hat{\mathbf{s}} \cdot {}^{\mathcal{R}}\hat{\mathbf{a}} \quad (3.31a)$$

$$B = 2{}^{\mathcal{R}}\hat{\mathbf{a}} \cdot ({}^{\mathcal{D}}\hat{\mathbf{s}} \times {}^{\mathcal{R}}\hat{\mathbf{e}}_{\psi}) \quad (3.31b)$$

$$C = {}^{\mathcal{D}}\hat{\mathbf{s}} \cdot {}^{\mathcal{R}}\hat{\mathbf{a}}. \quad (3.31c)$$

In Equation (3.31) it is, again,  $\mathcal{R}\hat{\mathbf{a}} = {}^B\hat{\mathbf{a}}$  and  $\mathcal{R}\hat{\mathbf{e}}_\psi = {}^B\hat{\mathbf{e}}_\psi$ , assuming that the body frame will eventually converge to the reference frame. The coefficients  $A$ ,  $B$ , and  $C$  are not entirely independent from one-another.  $C$  is the scalar product between two direction vectors, therefore it is:

$$-1 \leq C \leq 1. \quad (3.32)$$

$B$  is two times the triple product of three unit direction vectors, which is known to be equal to the volume of parallelepiped which has such vectors as its sides. Because the angle between two of these vectors is already bounded by  $C$ , it is:

$$-2\sqrt{1-C^2} \leq B \leq 2\sqrt{1-C^2}. \quad (3.33)$$

Lastly,  $A$  is bounded by functions of both  $B$  and  $C$ , due to the scalar products that appear in its formulation. It can be proved that:

$$\frac{B^2}{2} - C - 2 \leq A \leq 2 - C - \frac{B^2}{2}. \quad (3.34)$$

The denominator in Equation (3.30) is never zero, making the expression always non-singular, and the numerator is a quadratic expression. Different types of solutions are sought depending on the discriminant  $\Delta = B^2 - 4AC$  of this quadratic expression:

- $\Delta \geq 0$ : the equation  $f(t) = 0$  has two solutions. This means that maximum power condition is achievable, with incoming sunlight exactly perpendicular to the solar arrays' power-generating surface. The corresponding principal rotation angles (PRA) are obtained as:

$$\psi_{1/2} = 2 \arctan \left( \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \right). \quad (3.35)$$

Equation (3.35) is singular for  $A = 0$ . It should be noted, however, that even in such case, two solutions exist for  $\psi$ . The first trivial, and it is  $\psi_1 = -2 \arctan(C/B)$ . The second is  $\psi_2 = \pm\pi$ , and it corresponds to the singular configuration where  $t \rightarrow \pm\infty$ .

- $\Delta < 0$ : the equation  $f(t) = 0$  has no real solutions. The optimal attitude is given by the PRA for which  $|f(t)|$  is minimum. To find such value, let's consider the derivative of  $f(t)$

with respect to  $t$ :

$$f'(t) = \frac{-Bt^2 + 2(A - C)t + B}{(1 + t^2)^2}. \quad (3.36)$$

The absolute minimum and maximum for the expression  $f(t)$  are found equating  $f'(t)$  to zero, which gives the two solutions:

$$\psi_{1/2} = 2 \arctan \left( \frac{A - C \pm \sqrt{(A - C)^2 + B^2}}{B} \right). \quad (3.37)$$

Note that the term under the square root in Equation (3.37) is always non-negative, therefore the two solutions are always real. In this case, however, the two solutions  $\psi_1$  and  $\psi_2$  are not equivalent: one corresponds to the maximum for  $f(t)$ , and the other one to the minimum. In order to maximize power generation, only the solution for which  $|f(t(\psi))|$  is minimum should be chosen. Equation (3.37) is singular for  $B = 0$  which, under the assumption that  $\Delta < 0$ , necessarily implies that  $A$  and  $C$  are nonzero and have opposite signs. From these assumptions it follows that the optimal PRA is either  $\psi_1 = 0$  or  $\psi_2 = \pi$ , depending on the sign of  $A - C$ .

The optimal value for  $\psi$  can be combined into Equations (2.16) and (3.28) in order to compute  $[\mathcal{RD}]$ , and ultimately the DCM between inertial frame and optimal reference  $[\mathcal{RN}]$ .

Attention should be paid to the fact that there is no a priori knowledge on the coefficients  $A$ ,  $B$ , and  $C$ , and while their definition is always non-singular, singularities can be encountered in Equations (3.35) and (3.37) when some of these coefficients are zero. Algorithm 2 shows how to compute the correct PRA  $\psi$  in all circumstances, including cases in which Equations (3.35) and (3.37) are singular. Moreover, Algorithm 2 inherently addresses the singularity in the CRP formulation, which arises when  $\psi \rightarrow \pm\pi$ . The logic implemented in the algorithm analyzes the coefficients  $A$ ,  $B$ , and  $C$  such that when the absolute minimum of  $|f(t)|$  is at  $t \rightarrow \pm\infty$ , the solution returned is  $\psi = \pi$ . In such case, the only caveat is to compute the relative DCM  $[\mathcal{RD}]$  using the formulation in Equation (2.5) instead of Equation (2.16), leveraging the principal rotation parameter set  $(\psi, \mathcal{R}\hat{e}_\psi)$  instead of the CRP set, thus avoiding the singularity.

### 3.2.1.3 Keep-out Zone

The keep-out zone for the thermal-sensitive panel is expressed by Equation (3.24), where  ${}^{\mathcal{R}}\hat{\mathbf{h}}_2$  is the panel-normal heading in body/reference frame coordinates, and  ${}^{\mathcal{N}}\hat{\mathbf{h}}_{R2} = -{}^{\mathcal{N}}\hat{\mathbf{s}}$ . Following the same procedure outlined for Equations (3.30) and (3.31), one obtains the expression:

$$g(t) = \frac{Dt^2 + Et + F}{1 + t^2} \geq \cos \theta \quad (3.38)$$

where:

$$D = {}^{\mathcal{D}}\hat{\mathbf{s}} \cdot {}^{\mathcal{R}}\hat{\mathbf{h}}_2 - 2 ({}^{\mathcal{D}}\hat{\mathbf{s}} \cdot {}^{\mathcal{R}}\hat{\mathbf{e}}_\psi) ({}^{\mathcal{R}}\hat{\mathbf{h}}_2 \cdot {}^{\mathcal{R}}\hat{\mathbf{e}}_\psi) \quad (3.39a)$$

$$E = 2 {}^{\mathcal{R}}\hat{\mathbf{h}}_2 \cdot ({}^{\mathcal{R}}\hat{\mathbf{e}}_\psi \times {}^{\mathcal{D}}\hat{\mathbf{s}}) \quad (3.39b)$$

$$F = -{}^{\mathcal{D}}\hat{\mathbf{s}} \cdot {}^{\mathcal{R}}\hat{\mathbf{h}}_2. \quad (3.39c)$$

However, the previous subsection describes how to compute the PRA in order to ensure that the resulting reference attitude yields the maximum power. It has been shown that there can be two different scenarios, depending on whether  $\Delta = B^2 - 4AC$  is negative or not. In the first case, there is only one  $\psi$  value that maximizes the generated power. In this circumstance it is not possible to further optimize the reference attitude to accommodate the requirement on the thermal-sensitive platform without compromising the ability to generate power, which is already suboptimal. Conversely, when  $\Delta > 0$ , two solutions exist. Because these two solutions are equivalent in terms of power generation, the solution for which  $g(t(\psi)) \geq \cos \theta$  should be chosen. In the scenario when neither solution satisfies the keep-out constraint, the best solution is the one for which  $g(t(\psi))$  is higher.

### 3.2.1.4 Performance Discussion

An interesting result is derived exploring the space of possible relative orientations between the unit direction vectors involved in this analysis. Let's define  $\beta$  and  $\delta$  as the angles between the two body-frame directions, and the two inertial-frame directions, respectively, and  $\gamma$  as the

incidence angle on the solar arrays, as defined in Figure 3.2:

$$\beta = \arccos \left( {}^{\mathcal{B}}\hat{\mathbf{a}} \cdot {}^{\mathcal{B}}\hat{\mathbf{h}}_1 \right) \quad (3.40a)$$

$$\delta = \arccos \left( {}^{\mathcal{N}}\hat{\mathbf{s}} \cdot {}^{\mathcal{N}}\hat{\mathbf{h}}_{R1} \right) \quad (3.40b)$$

$$\gamma = \arccos \left( {}^{\mathcal{R}}\hat{\mathbf{a}} \cdot {}^{\mathcal{R}}\hat{\mathbf{s}} \right). \quad (3.40c)$$

Figure 3.3 shows the angle  $\gamma(\beta, \delta)$  for every combination of  $(\beta, \delta) \in [0, \pi] \times [0, \pi]$ , where  $\gamma$  is derived after computing the optimal reference  $\mathcal{R}$  applying Algorithm 2. The central plateau where  $\gamma = 0$  represents the cases for which  $\Delta \geq 0$  and two solutions exist for the optimal reference attitude. Conversely, the four regions in which  $0 < \gamma \leq \pi/2$  represent suboptimal configurations in which the optimal illumination on the solar arrays cannot be achieved simultaneously with the thruster pointing constraint ( $\Delta < 0$ ). The relation between the angles expressed in Equation (3.40) is:

$$\gamma = \begin{cases} \frac{\pi}{2} - \beta - \delta & \text{if } 0 \leq \beta \leq \frac{\pi}{2} \text{ and } \delta < \frac{\pi}{2} - \beta \\ -\frac{\pi}{2} - \beta + \delta & \text{if } 0 \leq \beta \leq \frac{\pi}{2} \text{ and } \delta > \beta + \frac{\pi}{2} \\ -\frac{\pi}{2} + \beta - \delta & \text{if } \frac{\pi}{2} < \beta \leq \pi \text{ and } \delta < \beta - \frac{\pi}{2} \\ \beta + \delta - \frac{3}{2}\pi & \text{if } \frac{\pi}{2} < \beta \leq \pi \text{ and } \delta > \frac{3}{2}\pi - \beta \\ 0 & \text{otherwise.} \end{cases} \quad (3.41)$$

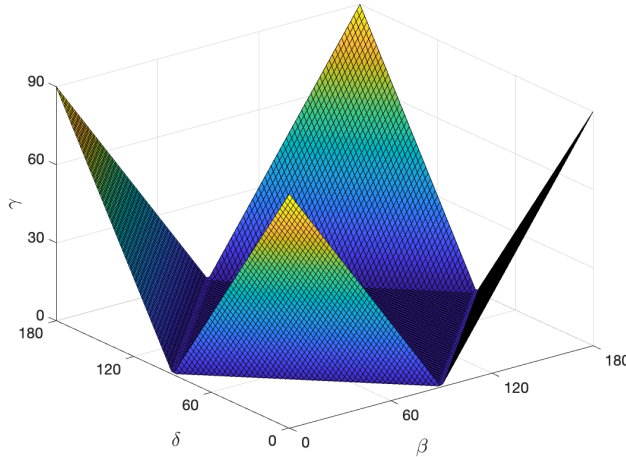


Figure 3.3: Optimal Sun incidence angle on array power-generating surface

Given the parameterization of the problem showed in Equation (3.31), where the coefficients  $A$ ,  $B$ , and  $C$ , depend on the relative headings of two vectors  ${}^{\mathcal{R}}\hat{\mathbf{a}}$  and  ${}^{\mathcal{D}}\hat{\mathbf{s}}$  with respect to  ${}^{\mathcal{R}}\hat{\mathbf{e}}_\psi$ .

Because each relative heading is characterized by two DoFs, the entire formulation is inherently four-dimensional. However, according to the results in Figure 3.3, the power performance of the spacecraft parameterized by the resulting incidence angle  $\gamma$  only depends on the two relative angles  $\beta$  and  $\delta$ .

Based on this results, considerations can be made on the geometry of the spacecraft. For example, Figure 3.3 shows that when  $\beta = \pi/2$  the optimal illumination condition  $\gamma = 0$  is always achievable. In a design where the thruster is fixed with respect to the body frame, it is desirable to have the thrust vector perpendicular to the SADA axis: this ensures that the spacecraft can generate the maximum amount of power at all times while thrusting in the right direction. However, when the pointing requirement is driven by a movable component in the  $\mathcal{B}$ -frame, like the gimballed thruster,  $\beta$  becomes a function of time, and consequently the optimum  $\gamma$  depends also on  $\delta$ , i.e. the angle between the requested pointing direction and the Sun.

It is interesting to compare the performance of the approach outlined in the previous sections with a simpler, non-optimal algorithm to compute the reference frame  $[\mathcal{RN}]$ . The triad method, for example, is a widely known algorithm used to identify the orientation of the body frame with respect to the inertial frame given two non-parallel vectors whose coordinates are well known in both frames [71]. In this application, the input arguments are two non-parallel vectors in the body frame ( $\hat{\mathbf{h}}_1$  and  $\hat{\mathbf{a}}_1$ ) and two non-parallel vector in the inertial frame ( $\hat{\mathbf{h}}_{R1}$  and  $\hat{\mathbf{s}}$ ). The main difference with the canonical triad method is that the desired DCM  $[\mathcal{RN}]$  is the one that aligns  $\hat{\mathbf{h}}_1$  with  $\hat{\mathbf{h}}_{R1}$ , but drives  $\hat{\mathbf{a}}_1$  orthogonal to  $\hat{\mathbf{s}}$ . This requirement is achieved with a variation in the standard triad method, where the definitions of the unit vectors with index 1 and 3 are swapped

in Equations (3.42b) and (3.42c):

$$\mathcal{R}\hat{\mathbf{r}}_2 = \mathcal{R}\hat{\mathbf{h}}_1 \qquad \mathcal{N}\hat{\mathbf{n}}_2 = \mathcal{N}\hat{\mathbf{h}}_{R1} \qquad (3.42a)$$

$$\mathcal{R}\hat{\mathbf{r}}_3 = \frac{\mathcal{R}\hat{\mathbf{a}} \times \mathcal{R}\hat{\mathbf{h}}_1}{\left| \mathcal{R}\hat{\mathbf{a}} \times \mathcal{R}\hat{\mathbf{h}}_1 \right|} \qquad \mathcal{N}\hat{\mathbf{n}}_1 = \frac{\mathcal{N}\hat{\mathbf{s}} \times \mathcal{N}\hat{\mathbf{h}}_{R1}}{\left| \mathcal{N}\hat{\mathbf{s}} \times \mathcal{N}\hat{\mathbf{h}}_{R1} \right|} \qquad (3.42b)$$

$$\mathcal{R}\hat{\mathbf{r}}_1 = \mathcal{R}\hat{\mathbf{r}}_2 \times \mathcal{R}\hat{\mathbf{r}}_3 \qquad \mathcal{N}\hat{\mathbf{n}}_3 = \mathcal{N}\hat{\mathbf{n}}_1 \times \mathcal{N}\hat{\mathbf{n}}_2 \qquad (3.42c)$$

$$[\mathcal{RD}] = [\mathcal{R}\hat{\mathbf{r}}_1, \mathcal{R}\hat{\mathbf{r}}_2, \mathcal{R}\hat{\mathbf{r}}_3] \qquad [\mathcal{ND}] = [\mathcal{N}\hat{\mathbf{n}}_1, \mathcal{N}\hat{\mathbf{n}}_2, \mathcal{N}\hat{\mathbf{n}}_3] \qquad (3.42d)$$

$$[\mathcal{RN}] = [\mathcal{RD}][\mathcal{ND}]^T. \qquad (3.43)$$

The algorithm outlined in Equations (3.42) and (3.43) is significantly simpler and less computationally expensive. However, the DCM computed using this method is only successful at aligning  $\hat{\mathbf{h}}_1$  with  $\hat{\mathbf{h}}_{R1}$ . With respect to the SADA axis  $\hat{\mathbf{a}}_1$ , this is only correctly placed orthogonal to  $\hat{\mathbf{s}}$  in the special case where  $\hat{\mathbf{a}}_1 \perp \hat{\mathbf{h}}_1$ . The an equivalent plot for  $\gamma(\beta, \delta)$  to that in Figure 3.3 can be generated using the triad method, and the result is shown in Figure 3.4. The reader can observe that, in this case, the incidence on the arrays is only optimal ( $\gamma = 0$ ) if and only if  $\hat{\mathbf{a}}_1 \perp \hat{\mathbf{h}}_1$  ( $\beta = 90$  deg) or  $\hat{\mathbf{s}} \perp \hat{\mathbf{h}}_{R1}$  ( $\delta = 90$  deg). These results can be further expanded on considering the maximum allowable incidence angle on the SAs based on the power requirements. For EMA, for example, the requirement is to maintain the array surface normal to the sunline within  $\pm 5$  deg [10].

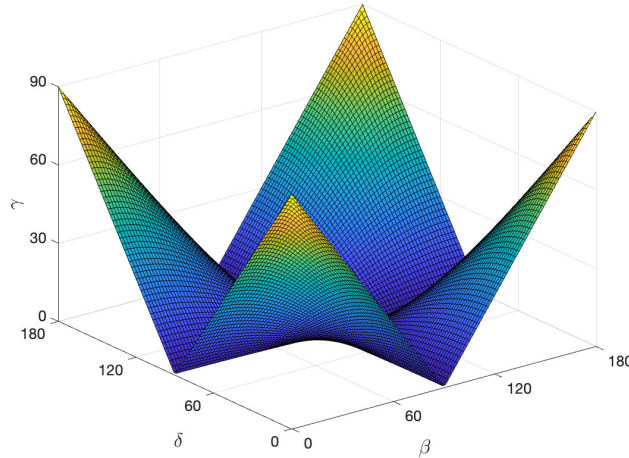


Figure 3.4: Sun incidence angle on array power-generating surface using the triad method

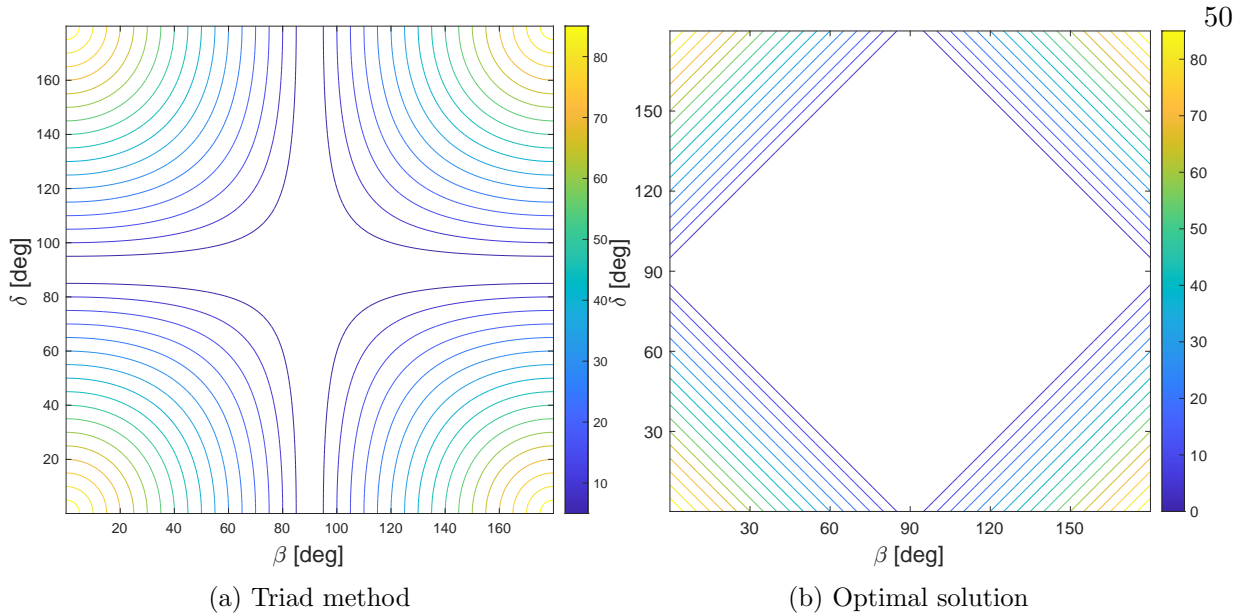


Figure 3.5:  $\gamma$  contour plot for  $(\beta, \delta) = [0, \pi] \times [0, \pi]$

Taking Figure 3.3 and the corresponding plot obtained with the triad method, and slicing them with horizontal planes at different  $\gamma$  values produces the plots in Figure 3.5. In both subfigures, the innermost purple line represents the  $\gamma = 5$  deg bounding curve, within which the pointing requirement is satisfied. Because the angle  $\delta$  is a parameter that depends on circumstances such as the relative orientation of the Sun with respect to the thruster pointing requirement, the only control variable is the angle  $\beta$ . Comparing the two plots it is evident how the optimal solution ensures a wider admissible range for  $\beta$  which, in the context of the EMA mission, means that there is a much larger margin for the thruster gimbals, which are allowed to perform larger angles without compromising the mission pointing requirements.

### 3.2.2 Inertial Pointing with Lower-bounded Power Generation

This section expands on the earlier results in the following manner. The same set of constraints is considered, but the strict requirement on maximum power generation is relaxed. For a mission to the outer Solar System, the power requirements would be defined based on the arrival conditions, where illumination is worse. However, during flybys of the inner planets, it may not be



necessary to use the full power generation capability of the spacecraft and, conversely, it might be more important to ensure that the spacecraft is dissipating heat efficiently. This section starts from the same initial assumptions of the former one, where the thruster alignment is the main driving requirement for attitude reference generation. Once again, the solution is obtained as the product of two intermediate DCMs as shown in Equation (3.25). Because power-related considerations do not affect the thruster alignment, the first intermediate DCM  $[\mathcal{DN}]$  is obtained analogously.

### 3.2.2.1 Lower-bounded Power Generation

When the spacecraft is closer to the Sun, the irradiance on the arrays increases. As a result, more power can be harvested even if the incidence angle of the light on the arrays is not optimal. In such case, the requirement in Equation (3.22) becomes:

$$-K \leq f(t) \leq K \quad (3.44)$$

where Equations (3.30) and (3.31) still apply, and  $K = \sin \gamma$  is a parameter that describes how far off from perpendicular the array surface and sunlight can be while still harvesting a sufficient amount of power. In practice,  $K$  constitutes the lower admissible bound on the amount of power that can be generated by the spacecraft in that position in orbit. Because  $0 \leq \gamma \leq \pi/2$ , it is  $0 \leq K \leq 1$ , where  $K = 0$  means that the lower bound is the maximum power requirement described in the previous section. In practical terms, a limit on how much  $K$  should be increased is posed by the shadowing of the spacecraft onto the arrays, which occurs when the SADAs are not perfectly orthogonal to the sunline and further degrades the power-generating performance. Because this issue is specific to the spacecraft design, it is not considered in the following analysis.

Before diving into the discussion of the problem presented by Equation (3.44), a simplifying assumption is made. No a priori knowledge is given on the coefficients  $A$ ,  $B$ , and  $C$ . However, it is possible to assume, without loss of generality, that  $A > 0$ . This is true because, even when  $A < 0$ , Equation (3.44) can be rewritten in an equivalent form where the coefficient of the  $t^2$  term

is positive. With this consideration, Equation (3.44) is rewritten as the system:

$$\begin{cases} (A + K)t^2 + Bt + (C + K) \geq 0 \\ (A - K)t^2 + Bt + (C - K) \leq 0 \end{cases} \quad (3.45)$$

with  $A \geq 0$  and  $0 \leq K \leq 1$ . To analyze the solutions of this system it is useful to define the functions:

$$y(t) = At^2 + Bt + C \quad (3.46a)$$

$$y^+(t) = (A + K)t^2 + Bt + (C + K) \quad (3.46b)$$

$$y^-(t) = (A - K)t^2 + Bt + (C - K) \quad (3.46c)$$

which describe three parabolas as functions of  $t$ . It is easy to show that these three parabolas never intersect, except for the case  $K = 0$ , for which they coincide. Moreover, it is true for any real value of  $t$ , that:

$$y^-(t) \leq y(t) \leq y^+(t) \quad (3.47)$$

Analytical methods can be derived to compute the solution space for the system in Equation (3.45). These methods are differentiated depending on the sign of the discriminant of the base parabola  $y(t)$ . The following discriminants are defined for the three parabolas in Equation (3.46):

$$\Delta = B^2 - 4AC \quad (3.48a)$$

$$\Delta^+ = B^2 - 4(A + K)(C + K) \quad (3.48b)$$

$$\Delta^- = B^2 - 4(A - K)(C - K). \quad (3.48c)$$

- Case  $\Delta \geq 0$ : the equation  $y(t) = 0$  has two solutions. It is interesting to determine for which values of  $K$   $y^+(t) = 0$  and  $y^-(t) = 0$  also have solutions. Such values are determined by the solution of the following inequality:

$$\begin{aligned} \Delta^\pm &= B^2 - 4(A \pm K)(C \pm K) \geq 0 \\ &= -[4K^2 \pm 4(A + C)K - B^2 + 4AC] \geq 0 \end{aligned} \quad (3.49)$$

whose discriminant, as a function of the variable  $K$ , is:

$$\Delta_K = 16 [(A - C)^2 + B^2] \geq 0 \quad (3.50)$$

and whose critical  $K$  values are:

$$K_{1/2}^+ = \frac{-(A + C) \pm \sqrt{(A - C)^2 + B^2}}{2} \quad (3.51a)$$

$$K_{1/2}^- = \frac{+(A + C) \pm \sqrt{(A - C)^2 + B^2}}{2}. \quad (3.51b)$$

For both these pairs of solutions it is true that:

$$K_1^\pm \cdot K_2^\pm = -\frac{1}{4}(B^2 - 4AC) \leq 0, \quad (3.52)$$

from which it can be concluded that each pair of solutions consists of a positive and a negative solution. Based on this result, it is concluded that the conditions for  $y^+(t) = 0$  and  $y^-(t) = 0$  to have each two solutions are, respectively:

$$K \leq \frac{\sqrt{(A - C)^2 + B^2} - (A + C)}{2} \quad (3.53a)$$

$$K \leq \frac{\sqrt{(A - C)^2 + B^2} + (A + C)}{2}. \quad (3.53b)$$

Defining the following zeroes of Equation (3.46):

$$t_1 = \frac{-B - \sqrt{\Delta}}{2A} \quad t_2 = \frac{-B + \sqrt{\Delta}}{2A} \quad (3.54a)$$

$$t_1^+ = \frac{-B - \sqrt{\Delta^+}}{2(A + K)} \quad t_2^+ = \frac{-B + \sqrt{\Delta^+}}{2(A + K)} \quad (3.54b)$$

$$t_1^- = \frac{-B - \sqrt{\Delta^-}}{2(A - K)} \quad t_2^- = \frac{-B + \sqrt{\Delta^-}}{2(A - K)} \quad (3.54c)$$

it is possible to visualize the problem represented by Equation (3.45) in Figure 3.6, where the solution space is highlighted in green. Attention should be paid to the fact that the parabola  $y^-(t)$  changes its concavity depending on the sign of  $(A - K)$ , and the solution space changes accordingly, as highlighted by the differences between Figures 3.6a and 3.6b. In the event of  $K = A$ ,  $y^-(t)$  degenerates into a line, and  $t_2^- \rightarrow \infty$ . Mapping the solution space in  $t$  to the respective PRA gives the range of angles  $\psi$  that satisfy Equation (3.45).

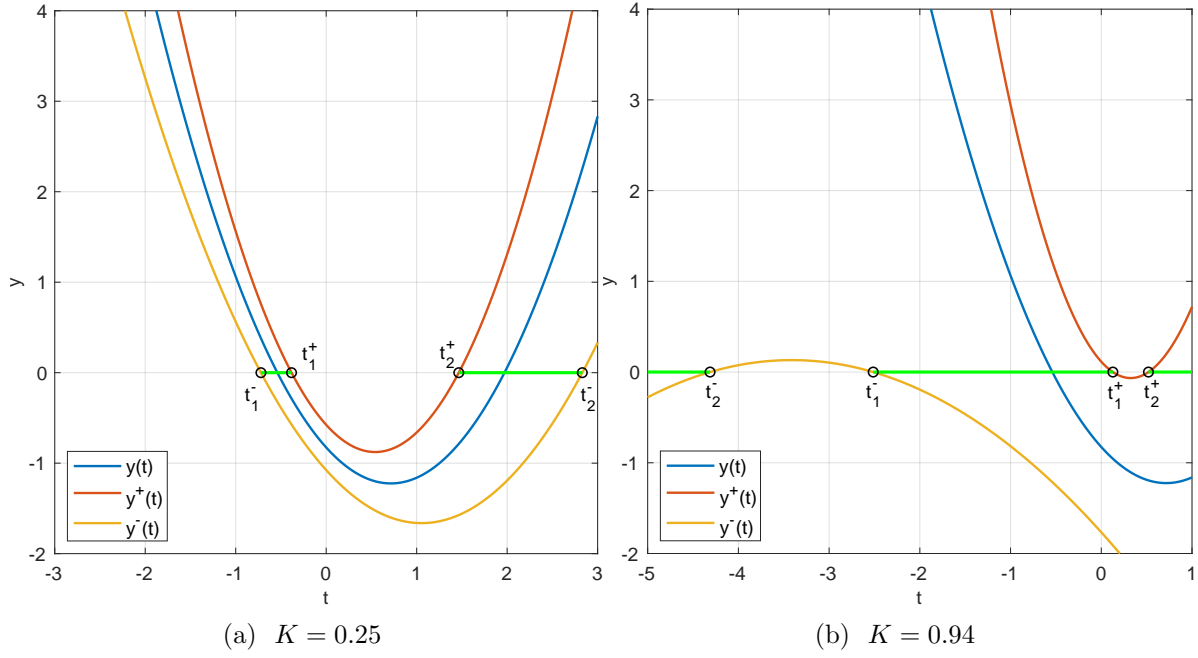


Figure 3.6: Solution space with  $A = 0.777$ ,  $B = -1.112$ ,  $C = -0.826$

Figure 3.7a shows such solution space for varying values of  $K$ , where it is observed that, for  $K = 0$ , the solution space degenerates into the two individual solutions presented in Equation (3.35).

- Case  $\Delta < 0$ : the equation  $y(t) = 0$  has no real solutions. Based on Equation (3.47) it is true that  $0 < y(t) \leq y^+(t)$ , therefore the system in Equation (3.45) reduces to:

$$y^-(t) = (A - K)t^2 + Bt + (C - K) \leq 0. \quad (3.55)$$

The discriminant of this equation is expressed by Equation (3.49), with the negative sign. The two solutions  $K_{1/2}^-$  are those in Equation (3.51b). However, in this case it is:

$$K_1^- \cdot K_2^- = -\frac{1}{4}(B^2 - 4AC) > 0, \quad (3.56)$$

from which it follows that the two critical  $K$  values have the same sign. It can be concluded that  $y^-(t) = 0$  has no solutions for  $K < K_1^-$ , two solutions for  $K_1^- \leq K \leq K_2^-$ , and infinite solutions for  $K > K_2^-$ . This result allows to solve Equation (3.55) and, mapping

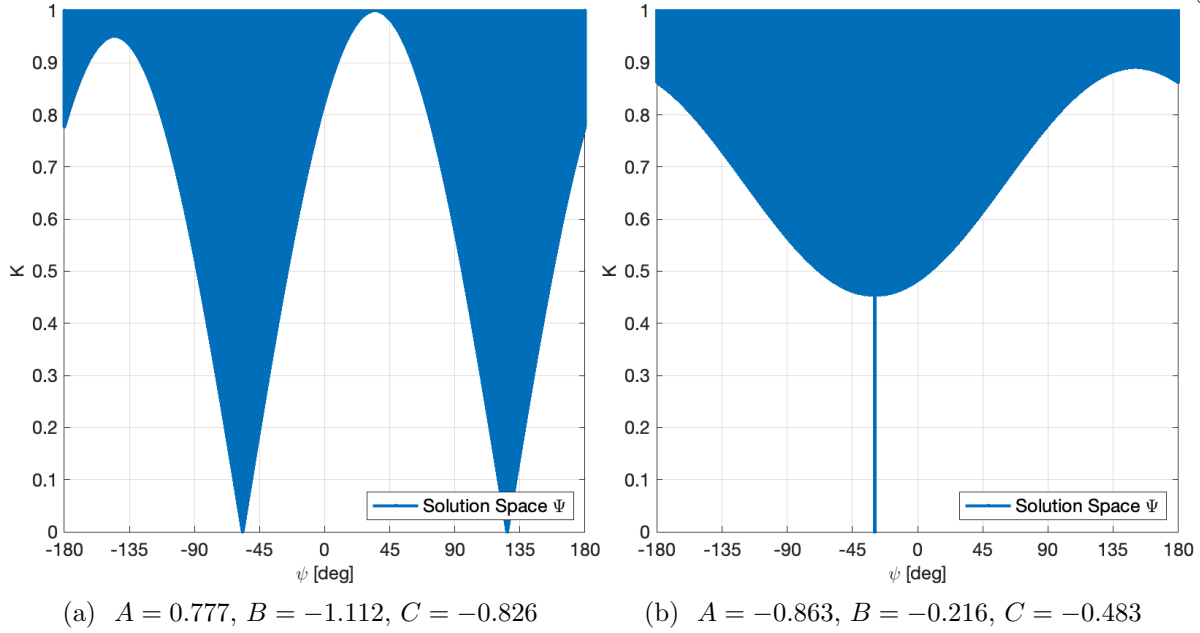


Figure 3.7: Solution space with  $0 \leq K \leq 1$

the solutions for  $t$  to the respective PRV values  $\psi$ , the solution space in Figure 3.7b is obtained. Note that, for  $K \leq K_1^- = 0.454$ , the solution from Equation (3.37) is chosen to ensure that  $|f(t(\psi))|$  is minimum. Analogously, this choice drives the SADA drive axis as close to perpendicular to sunlight as possible. For  $K > K_2^- = 0.891$ , infinite solutions exist.

Algorithm 3 shows how to derive the solution space  $\Psi$  for any combination of  $A$ ,  $B$ ,  $C$ , and  $K$ , and obtain the solution space plots in Figures 3.7a and 3.7b.

### 3.2.2.2 Keep-out Zone

For the keep-out constraint, the Equations (3.38) and (3.39) still apply. Specifically, Equation (3.38) can be rewritten in the form:

$$z(t) = (D - Q)t^2 + Et + (F - Q) \geq 0, \quad (3.57)$$

where  $Q = \cos \theta$ . The solution set  $\Theta$  of this inequality can be computed using Algorithm 4. When the set that satisfies Equation (3.57) is the empty set, the solution computed by Algorithm 4

coincides with the value for which  $z(t)$  is maximum, i.e., the computed PRA gets the heading  $\hat{h}_2$  as far from the Sun direction as possible.

The final solution of the attitude reference generation problem as a whole is, in this case, the intersection space of  $\Psi$  and  $\Theta$ . A non-empty intersection space contains the solutions that best satisfy all three pointing requirements in order of priority. Conversely, an empty solution space means that it is not possible to satisfy the second and third constraint simultaneously. In this circumstance, it is recommendable to choose  $\psi$  such that:

$$\psi \in \Psi : \psi = \arg \min(\text{dist}(\Psi, \Theta)) \quad (3.58)$$

which means,  $\psi$  is the element of the set  $\Psi$  that is closest to the set  $\Theta$ . The resulting set is visualized in yellow in Figure 3.8, where the yellow band corresponds to a range of solutions for which Equation (3.38) is satisfied. Conversely, for low enough values of  $K$  the solution is unique, and it is the solution for which  $z(t)$  is maximum, within the bounds that satisfy Equation (3.44).

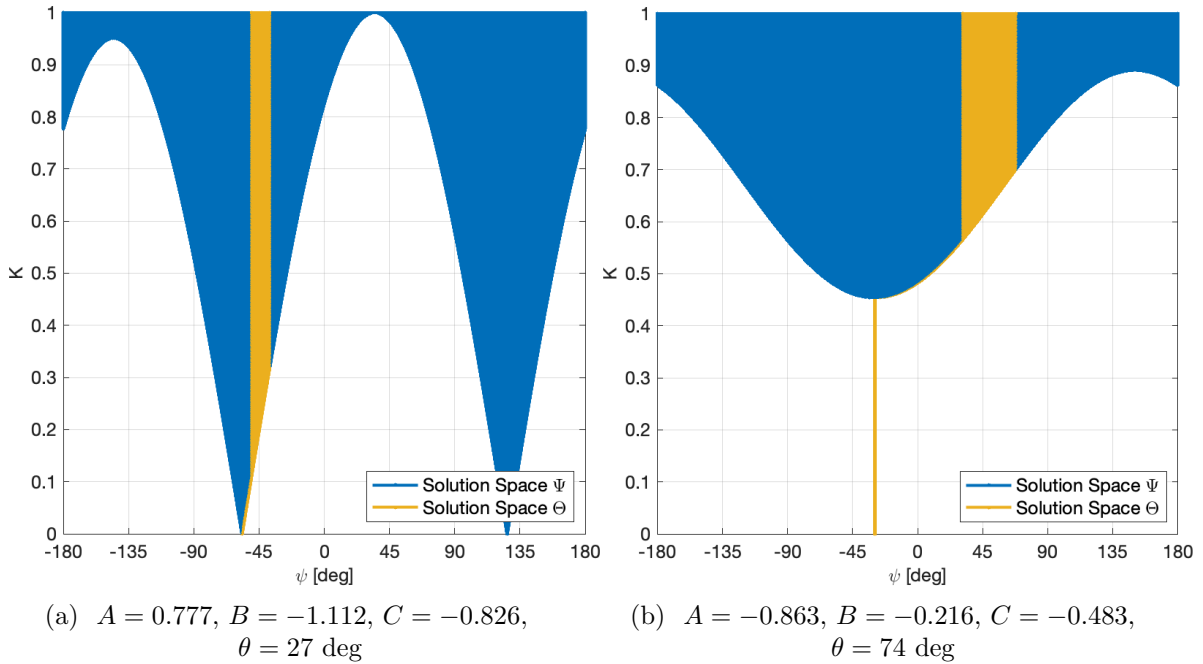


Figure 3.8: Intersection of solution spaces with  $0 \leq K \leq 1$

### 3.3 Solar Array Guidance

Once the reference attitude  $\mathcal{R}$  is computed, a guidance algorithm needs to be defined for the solar arrays. Recalling the assumption on the modularity of the guidance laws for the different subcomponents, this section assumes that the only information needed to compute the optimal reference for the arrays is the knowledge of the optimal attitude reference  $\mathcal{R}$ , computed in the previous sections, plus information on the Sun direction coming from sensor measurements. The rotation angle of the arrays  $\alpha$  is defined with respect to a zero direction, for which  $\alpha = 0$ . For this purpose, let's define two frames: a frame  $\mathcal{A} = \{\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \hat{\mathbf{a}}_3\}$  that is hub-fixed, and a frame  $\mathcal{A}' = \{\hat{\mathbf{a}}'_1, \hat{\mathbf{a}}'_2, \hat{\mathbf{a}}'_3\}$  that is array-fixed. The two frames are separated by a rotation angle  $\alpha$  about the first axis ( $\hat{\mathbf{a}}_1 = \hat{\mathbf{a}}'_1$ ) and they coincide for  $\alpha = 0$ . Figure 3.9 illustrates these two frames.

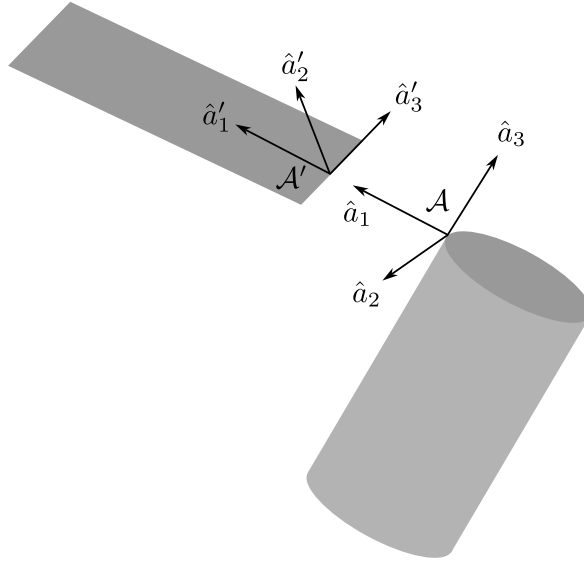


Figure 3.9: Body-fixed and array-fixed frames

#### 3.3.1 Maximum Power Generation

The goal of this section is to identify the reference direction  $\hat{\mathbf{a}}'_{R2}$  along which the power-generating surface of the solar arrays needs to be pointed. Under the assumption that the requirement is to generate as much power on the solar arrays as possible, this is equivalent to maximizing

the function

$$p(\hat{\mathbf{a}}'_{R2}) = \hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_{R2}. \quad (3.59)$$

The constraints on  $\hat{\mathbf{a}}'_{R2}$  are that it must be orthogonal to  $\hat{\mathbf{a}}_1$  and that it has unit norm. Considering the Lagrange multipliers  $\lambda_1$  and  $\lambda_2$ , the function to be maximized becomes:

$$P(\hat{\mathbf{a}}'_{R2}) = \hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_{R2} + \lambda_1 \hat{\mathbf{a}}_1 \cdot \hat{\mathbf{a}}'_{R2} + \lambda_2 (\|\hat{\mathbf{a}}'_{R2}\|^2 - 1). \quad (3.60)$$

Defining  $\hat{\mathbf{a}}'_{R2} = \{a'_1, a'_2, a'_3\}^T$ ,  $\hat{\mathbf{a}}_1 = \{a_1, a_2, a_3\}^T$ , and  $\hat{\mathbf{s}} = \{s_1, s_2, s_3\}^T$ , the maximum of the function in Equation (3.60) is obtained solving the system of equation that is obtained equating to zero the gradient of  $P(\hat{\mathbf{a}}'_{R2})$  derived with respect to the augmented state  $\mathbf{x} = \{a'_1, a'_2, a'_3, \lambda_1, \lambda_2\}^T$ , which gives:

$$\begin{cases} s_1 + \lambda_1 a_1 + 2\lambda_2 a'_1 = 0 \\ s_2 + \lambda_1 a_2 + 2\lambda_2 a'_2 = 0 \\ s_3 + \lambda_1 a_3 + 2\lambda_2 a'_3 = 0 \\ a_1 a'_1 + a_2 a'_2 + a_3 a'_3 = 0 \\ a_1'^2 + a_2'^2 + a_3'^2 = 1 \end{cases} \quad (3.61)$$

which, remembering that  $\hat{\mathbf{s}}$  and  $\hat{\mathbf{a}}_1$  also have unit norm, gives  $\lambda_1 = -(\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}_1)$  and  $\lambda_2 = -(\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_{R2})/2$ .

Substituting the Lagrange multipliers gives:

$$\hat{\mathbf{a}}'_{R2} = \frac{\hat{\mathbf{s}} - (\hat{\mathbf{a}}_1 \cdot \hat{\mathbf{s}})\hat{\mathbf{a}}_1}{\sqrt{1 - (\hat{\mathbf{a}}_1 \cdot \hat{\mathbf{s}})^2}}. \quad (3.62)$$

From Equation (3.62) it is easy to show that  $\hat{\mathbf{a}}'_{R2} \perp \hat{\mathbf{a}}_1$  and  $\hat{\mathbf{a}}'_{R2} \cdot \hat{\mathbf{s}} = \cos \gamma$ , where  $\gamma$  is consistent with the definition given in Equation (3.22). Therefore for  $\gamma = 0$ , assuming that a solution that satisfies the attitude requirement exists, results in  $\hat{\mathbf{a}}'_{R2} \parallel \hat{\mathbf{s}}$ , that is the arrays are directly facing the Sun. The reference angle for the solar arrays is:

$$\alpha_R = \arccos(\hat{\mathbf{a}}_2 \cdot \hat{\mathbf{a}}'_{R2}). \quad (3.63)$$

### 3.3.2 Upper-bounded Power Generation

The case may exist where, due to close proximity to the Sun, the desire is to limit the exposure of the arrays to the Sun. This can happen, for example, to avoid overheating the spacecraft.



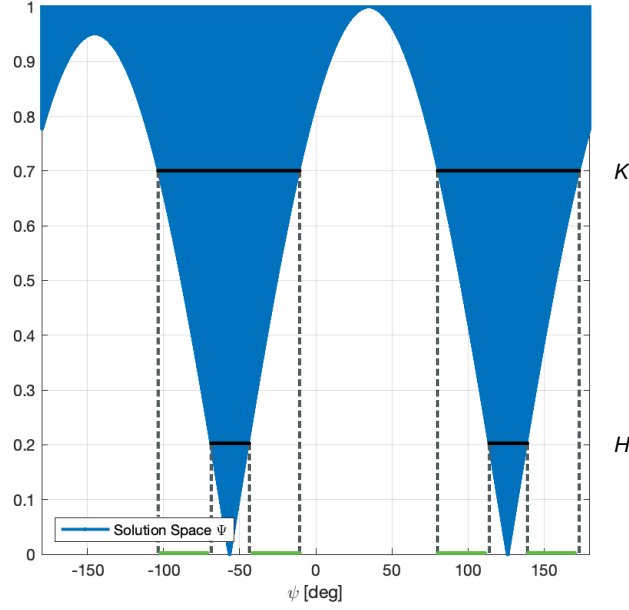


Figure 3.10: Solution space with  $A = 0.777$ ,  $B = -1.112$ ,  $C = -0.826$ ,  $0 \leq K \leq 1$

Effectively, this translates into the problem of upper-bounding the amount of power that can be generated by the arrays. This can be accomplished via two different approaches. With respect to Equation (3.44), it is possible to redefine the requirement as:

$$H \leq |f(t)| \leq K. \quad (3.64)$$

This equation can be translated to a problem similar to that analyzed in Section 4, with the difference that, in this case, the solution space would be a subset of the respective solution spaces shown in Figures 3.7a and 3.7b. For a  $\Delta > 0$  scenario, this would produce the solution space corresponding to the green lines in Figure 3.10. Besides the analytical complexity of determining the solution of this problem, this is not the desirable approach because it would ultimately involve computing a new reference frame and slewing the spacecraft to it. The better approach consists in only articulating the solar arrays in order to face them away from sunlight whenever needed. This allows to hold the hub's attitude steady and at the same time change the condition of illumination while only rotating the arrays. Let's define  $\sigma$  as the desired angle between the Sun direction and the normal to the array surface, as opposed to  $\gamma$ , which indicates the smallest such angle possible

given the current spacecraft attitude. The relation between the two is:

$$\cos \sigma = \cos \gamma \cos \epsilon, \quad (3.65)$$

which implies that  $\sigma \geq \gamma$ . Combining this relation with the result of the previous subsection gives:

$$\alpha_R = \arccos(\hat{\mathbf{a}}_2 \cdot \hat{\mathbf{a}}'_{R2}) \pm \epsilon = \arccos(\hat{\mathbf{a}}_2 \cdot \hat{\mathbf{a}}'_{R2}) \pm \arccos\left(\frac{\cos \sigma}{\cos \gamma}\right) \quad (3.66)$$

where the same reference angle computed in Equation (3.63) is offset by an amount  $\epsilon$  depending on the desired performance. It can be observed that for  $\sigma = \gamma$ , i.e., when the requirement coincides with the best performance, Equations (3.63) and (3.66) coincide. The angles  $\gamma$ ,  $\epsilon$ ,  $\sigma$  and  $\alpha_R$  are visualized in Figure 3.11.

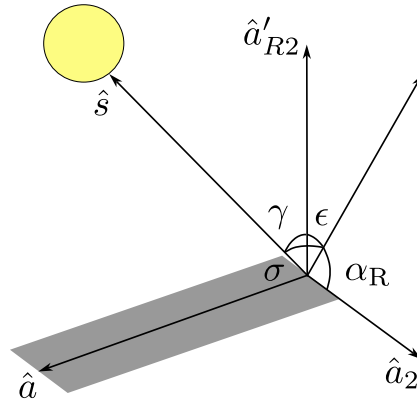


Figure 3.11: Array reference angles

## Continuous Momentum Management

Chapter 2 describes the dynamics of a spacecraft equipped with reaction wheels, and how to use reaction wheels as actuators to drive the spacecraft to the desired reference attitude. Chapter 3 presents guidance algorithms to determine the optimal spacecraft reference attitude for a spacecraft with common pointing constraints. This chapter dives into momentum management, which is a complementary aspect of the relation that exists between momentum exchange devices and attitude pointing requirements. Spacecraft are typically subject to ambient forces and torques, the most significant of which, for a deep-space mission, is solar radiation pressure (SRP). Depending on the orientation of the spacecraft surfaces with respect to the incoming sunlight, SRP can produce torques on the spacecraft that over long periods of time can cause attitude and orbital drifts. Such effects decrease with the distance from the Sun, but increase with the surface of the exposed surfaces, specifically solar arrays. Because the success of the mission is dependent on the spacecraft's ability to hold the attitude at the optimal reference within a certain accuracy, and potentially for long periods of time, the small but steady contribution of SRP needs to be counterbalanced. MEDs offer the type of high-resolution control that can compensate for small external disturbances and hold the attitude in position. In exchange, the momentum that would have otherwise been absorbed by the hub, is stored in the MEDs instead. As a result, MEDs spin up over time due to the progressive accumulation of momentum. This can only be done up to a certain point, because significant problems arise when RWs or CMGs are spun at very high angular rates. First of all, this causes mechanical stress on the bearings and significant power consumption. Second, a high

net momentum on the MEDs makes it increasingly harder to control the spacecraft due to the gyroscopic effect shown in Equation (2.39), requiring a larger and larger control torque from the actuators, which might not be able to spin up further to deliver it.

This chapter considers the spacecraft described in Chapter 1, which is equipped with a solar electric propulsion (SEP) thruster mounted on a gimbaled platform, which gives the ability to control the direction of the thrust vector with respect to the hub. The large solar arrays are also able to perform 360 degree rotations about their respective body-fixed axis. This section explores the usage of these moving components to continuously manage the momentum building up on the system due to SRP action. First, this chapter provides an enhanced guidance law that computes a reference for the motion platform such that the resulting thruster torque acting on the system counters the action of the SRP torque. Furthermore, an enhanced guidance law is defined for the solar arrays, when the desire is to point the arrays in order to reduce the momentum build-up on the reaction wheels. Lastly, this chapter proposes a technique to continuously estimate the CM location of the system while thrust is continuously being applied. The motion platform supporting the thruster is gimbaled periodically to align the thrust vector based on the current CM estimate. Of interest is how these small platform orientation changes can be used to make the three-dimensional CM location observable.

## 4.1 Continuous Momentum Management via Gimbaled SEP Thruster

### 4.1.1 Control Law Derivation with CM Knowledge

Section 3.1 shows how, by means of three consecutive rigid body rotations, it is possible to align the thrust vector with a point  $C$  on the hub. This point  $C$  can coincide with the CM of the whole system: in such case, the thrust does not produce any torque on the system. Conversely, it is possible to intentionally offset the thrust vector from the CM to use the resulting torque to perform momentum offloading.

Let's define  $\mathbf{H}_{RW}$  as the amount of RW momentum that needs to be dumped. This can be a

fraction of the momentum on the wheels that exceeds a certain threshold, or it can coincide with the total momentum on the wheels. In this application the goal is to dump the net momentum on the wheels entirely, until the wheel speeds are driven down to zero. The amount of momentum to be dumped is therefore:

$$\mathbf{H}_{\text{RW}} = \sum_{j=1}^N I_{w_s} \Omega_j \hat{\mathbf{u}}_{w_j}. \quad (4.1)$$

Note that  $\mathbf{H}_{\text{RW}}$  is the total net momentum of the reaction wheels with respect to the hub, and not an inertial observer. Before proceeding, let's recall the description of the thruster and platform assembly given in Section 3.1, where the coordinates of the thrust vector  $\mathbf{t}$  are considered known in the platform frame  $\mathcal{F}$ , and its coordinates in the mount and body frames  $\mathcal{M}$  and  $\mathcal{B}$  are pending on the knowledge of the platform gimbal angles, according to Equation (3.1).

To compensate the action of SRP it is therefore necessary to point the thruster at an offset that results in a thrust torque that opposes the current RW net momentum in Equation (4.1). Let's define such offset vector  $\mathbf{d}$ , and the resulting thruster torque  $\mathbf{L}_{\text{thr}}$ . Over an infinitesimal time interval  $\Delta\tau$  it is:

$$-\Delta\mathbf{H} = \mathbf{L}_{\text{thr}}\Delta\tau = \mathbf{d} \times \mathbf{t}\Delta\tau. \quad (4.2)$$

To obtain  $\mathbf{d}$ , both sides of the previous equation are cross-multiplied by  $\mathbf{t}$ . With the simplifying

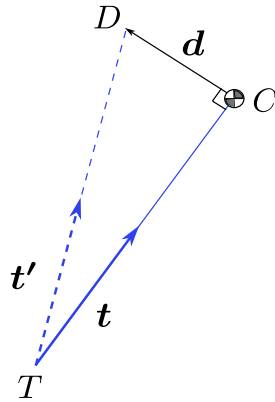


Figure 4.1: Center of mass offset

assumption that  $\mathbf{d} \perp \mathbf{t}$ , this gives:

$$-\mathbf{t} \times \Delta \mathbf{H} = \mathbf{t} \times (\mathbf{d} \times \mathbf{t}) \Delta \tau = [t^2 \mathbf{d} - (\mathbf{t} \cdot \mathbf{d}) \mathbf{t}] \Delta \tau = t^2 \mathbf{d} \Delta \tau \quad (4.3)$$

which ultimately yields the law:

$$\mathbf{d} = -\frac{\kappa}{t^2} (\mathbf{t} \times \mathbf{H}_{\text{RW}}) \quad (4.4)$$

where  $\kappa = 1/\Delta \tau$  [Hz] is a proportional control gain. A few assumptions are made in this derivation. The coordinates of  $\mathbf{t}$  are always known in the  $\mathcal{F}$  frame, which is the frame in which the thrust vector is, at least in first approximation, constant. To know the coordinates of  $\mathbf{t}$  in the  $\mathcal{M}$  frame, it is necessary to compute the tip and tilt angles  $\nu_1$  and  $\nu_2$  that derive from applying the guidance equations outlined in Section 3.1. For this purpose, let's choose to align  $\mathbf{t}$  with the system's CM, which is here assumed to be known. That is to say that the  $\mathbf{t}$  vector that appears in Equation (4.4) is aligned with the system's CM, and the  $\mathbf{d}$  vector computed accordingly is perpendicular to such direction, as shown in Figure 4.1. Offsetting  $C$  by  $\mathbf{d}$  gives point  $D$ , which is the point with which the thrust vector must be aligned in order to produce a torque that dumps the desired amount of momentum. Doing so requires reorienting the platform, thus aligning the thrust vector along  $\mathbf{t}'$  and computing new gimbal reference angles  $\nu_{R1}$  and  $\nu_{R2}$ . It is safe to assume that the SEP thruster, placed at a reasonable moment arm from the system CM, can produce a very large torque compared to the SRP torque. For this reason, a small offset  $\mathbf{d}$  is sufficient to compensate the momentum buildup due to SRP. Under this assumption it is admissible to state that  $\mathbf{d} \perp \mathbf{t} \approx \mathbf{t}'$ .

A drawback of this momentum management technique is that, at least with a single SEP thruster, it is underactuated. This is evident when considering the fact that the thruster torque:

$$\mathbf{L}_{\text{thr}} = \mathbf{d} \times \mathbf{t}' \approx \frac{\kappa}{t^2} \mathbf{t} \times (\mathbf{t} \times \mathbf{H}_{\text{RW}}) \quad (4.5)$$

is the result of a cross product, therefore it cannot have a component parallel to the thrust vector  $\mathbf{t}'$ . In other words, any amount of momentum along  $\mathbf{t}'$  cannot be dumped, which is analogous to say that any component of SRP torque (or whatever external torque acting on the spacecraft) along  $\mathbf{t}'$  cannot be compensated by the SEP thruster.

### 4.1.2 Stability Analysis

It is possible to show that the control law for thruster pointing in Equation (4.4) can, under certain assumptions, drive the net wheel momentum in Equation (4.1) to zero. This is done applying Lyapunov’s direct method presented in Section 2.3.1 defining the candidate Lyapunov function:

$$V(\mathbf{H}) = \frac{\mathbf{H} \cdot \mathbf{H}}{2} = \frac{H^2}{2} \quad (4.6)$$

which is clearly positive definite. Let’s assume simplified spacecraft dynamics, where the spacecraft hub is not moving ( $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} = 0$ ) and the only external torque acting on the system is the thruster torque in Equation (4.5). Let’s also assume a very fast response from the RWs, such that all the thruster torque is stored in the form of RW net momentum, i.e.  $\dot{\mathbf{H}} = \dot{\mathbf{H}}_{\text{RW}} = \mathbf{L}_{\text{thr}}$ . Under these assumptions, the derivative of the Lyapunov function becomes:

$$\dot{V}(\mathbf{H}) = \mathbf{H} \cdot \dot{\mathbf{H}} = \frac{\kappa}{t^2} \mathbf{H} \cdot [\mathbf{t} \times (\mathbf{t} \times \mathbf{H})] = -\kappa H^2 [1 - (\hat{\mathbf{t}} \cdot \hat{\mathbf{H}})^2]. \quad (4.7)$$

The Lyapunov derivative  $\dot{V}(\mathbf{H})$  is negative definite, except in the case in which the RW net momentum  $\mathbf{H}$  is aligned with the thrust vector. Note that while this analysis shows asymptotic stability, some assumptions are made. Equation (4.5) assumes  $\mathbf{t} \approx \mathbf{t}'$ , which is true within some error. Carrying out the math without this assumption would result in additional terms in the expression for  $\dot{V}(\mathbf{H})$  that would not allow to claim negative definiteness. In practical terms, this control law is asymptotically stabilizing provided that the net wheel momentum  $\mathbf{H}_{\text{RW}}$  has no component along the “zero-offset” thruster orientation, i.e., when the thruster is aligned with the system CM. When this is not the case, residual momentum will remain on the wheels once the system has reached a steady state.

### 4.1.3 Control Law Derivation without CM Knowledge

In the previous derivation, an assumption was made on the a priori knowledge of the location of the system CM. Generally speaking, the CM location can be known within a certain tolerance, and it changes significantly over the course of a multi-year mission. For the purpose of using the

SEP thruster for momentum management, a wrong CM estimate can lead to an inaccurate steady state thruster pointing that results in parasitic torques on the system, contributing to the growth of net RW momentum over time. For this reason, Equation (4.4) can be enhanced with the addition of an integral term as follows:

$$\mathbf{d} = -\frac{\mathbf{t}}{t^2} \times (\kappa \mathbf{H} + \kappa_I \mathbf{H}_I) \quad (4.8)$$

where:

$$\mathbf{H}_I = \int_0^t \mathbf{H} d\tau. \quad (4.9)$$

and the units of  $\kappa_I$  are Hz<sup>2</sup>. When the net momentum grows over time, the integral feedback term corrects the offset  $\mathbf{d}$  to align the thruster with the real CM location. The stability analysis can be performed defining a new Lyapunov function:

$$V(\mathbf{H}, \mathbf{H}_I) = \frac{H^2}{2} + \kappa_I \frac{H_I^2}{2} \quad (4.10)$$

which is positive definite. The derivative is:

$$\begin{aligned} \dot{V}(\mathbf{H}, \mathbf{H}_I) &= \mathbf{H} \cdot (\dot{\mathbf{H}} + \kappa_I \dot{\mathbf{H}}_I) \\ &= \mathbf{H} \cdot \left\{ \frac{\mathbf{t}}{t^2} \times [\mathbf{t} \times (\kappa \mathbf{H} + \kappa_I \mathbf{H}_I)] + \kappa_I \mathbf{H}_I \right\} = -\kappa H^2 \end{aligned} \quad (4.11)$$

where the final result is obtained with the assumption that  $\mathbf{t} \cdot \mathbf{H} = 0$ , i.e., the net momentum has no components along the thrust vector at zero offset. In this case, the result of Equation (4.11) only proves that the derivative of the Lyapunov function is negative semidefinite, because there is no trace of  $\mathbf{H}_I$  in the expression. This proves Lyapunov stability, but to prove asymptotic stability higher-order derivatives must be computed and evaluated on the set in which  $\dot{V}(\mathbf{H}, \mathbf{H}_I) = 0$  [60], which is  $\mathbf{H} = 0$ . For the second-order derivative it is:

$$\begin{aligned} \ddot{V}(\mathbf{H}, \mathbf{H}_I) &= -2\kappa \mathbf{H} \cdot \dot{\mathbf{H}} = -2\kappa \mathbf{H} \cdot \left\{ \frac{\mathbf{t}}{t^2} \times [\mathbf{t} \times (\kappa \mathbf{H} + \kappa_I \mathbf{H}_I)] + \kappa_I \mathbf{H}_I \right\} \\ &= 2\kappa \mathbf{H} \cdot (\kappa \mathbf{H} + \kappa_I \mathbf{H}_I) \end{aligned} \quad (4.12)$$

which is once again obtained with the assumption that  $\mathbf{t} \cdot \mathbf{H} = 0$ . The Lyapunov stability of the system implies that  $\mathbf{H}$  and  $\mathbf{H}_I$  are bounded, therefore it follows that  $\ddot{V}_{\mathbf{H}=0} = 0$ . The third



derivative gives:

$$\begin{aligned}
\ddot{V}(\mathbf{H}, \mathbf{H}_I) &= 2\kappa \dot{\mathbf{H}} \cdot (\kappa \mathbf{H} + \kappa_I \mathbf{H}_I) + 2\kappa \mathbf{H} \cdot (\kappa \dot{\mathbf{H}} + \kappa_I \mathbf{H}) \\
&= 2\kappa (2\kappa \mathbf{H} + \kappa_I \mathbf{H}_I) \cdot \left\{ \frac{\mathbf{t}}{t^2} \times [\mathbf{t} \times (\kappa \mathbf{H} + \kappa_I \mathbf{H}_I)] + k_I \mathbf{H}_I \right\} + 2\kappa \kappa_I H^2 \\
&= 2(\kappa_I - 2\kappa^2) \kappa H^2 - 6\kappa^2 \kappa_I \mathbf{H} \cdot \mathbf{H}_I - 2\kappa \kappa_I^2 H_I^2 \left[ 1 - (\hat{\mathbf{t}} \cdot \hat{\mathbf{H}}_I)^2 \right]
\end{aligned} \tag{4.13}$$

which evaluated on the set  $\mathbf{H} = 0$  gives:

$$\ddot{V}_{\mathbf{H}=0} = -2\kappa \kappa_I^2 H_I^2 \left[ 1 - (\hat{\mathbf{t}} \cdot \hat{\mathbf{H}}_I)^2 \right]. \tag{4.14}$$

This result is analogous to that of Equation (4.11), where the function is negative definite except for the case in which the integral of the net momentum is aligned with the thrust vector. With this caveat, the third derivative being negative definite on the set  $\mathbf{H} = 0$  proves that the control law in Equation (4.8) is indeed asymptotically stabilizing. This is true within the assumption made in this analysis, where no unmodeled external torques are acting on the system and the thruster torque is all absorbed by the reaction wheels.

## 4.2 Momentum Management via Rotating Solar Arrays

Section 3.3 shows how to compute the optimal reference angle for the solar arrays, ensuring maximum sun incidence on the power-generating surface, given the hub reference frame  $\mathcal{R}$ . This section aims to expand on that analysis to suggest an alternative solar array articulation that leverages SRP torque to perform momentum offloading. It is important to notice that this strategy might not be suitable to all missions, as it requires to depart from the condition of optimal power generation. However, during segments of a mission in which the SEP thruster is not active, it is reasonable to assume that the spacecraft does not need to operate at full power. Similarly, when the spacecraft flies closer to the inner regions of the Solar System, more power is generated as a consequence to the proximity to the Sun, so suboptimal array configuration might be sufficient.

### 4.2.1 Symmetric, Fully-absorbent Solar Arrays

This analysis applies to a pair of symmetric solar array that can rotate about antiparallel, body-fixed spacecraft axes. The spacecraft attitude is considered given, with the body frame  $\mathcal{B}$  aligned with the reference frame  $\mathcal{R}$ , which is dictated by pointing requirements. The aim is to articulate the arrays differentially in order to result in a net SRP torque in the opposite direction to the current RW net momentum. This analysis considers only the SRP acting on the arrays, assumed fully absorbent, and it neglects secondary effects such as the SRP on the spacecraft hub, or secondary reflections of photons. The arrays are modeled as two-dimensional panels with infinitesimally small thickness. The same array frames  $\mathcal{A}$  and  $\mathcal{A}'$  described in Section 3.3 and illustrated in Figure 3.9 are considered. With respect to these frame vectors, let's define a simplified SRP force model with full absorption:

$$\mathbf{F}_{\text{SRP,abs}} = -PS (\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2) \hat{\mathbf{s}}. \quad (4.15)$$

where  $S$  is the surface of each array. A complete description of the SRP force model is provided in Reference 72. It is easy to show that for  $\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2 = 1$  the force is maximum, and it vanishes for  $\hat{\mathbf{s}} \perp \hat{\mathbf{a}}'_2$ . The goal of this analysis is to deflect the array whose resulting SRP torque feeds positively into the net RW momentum. For this purpose, let's consider deflection angles  $\theta$  with respect to

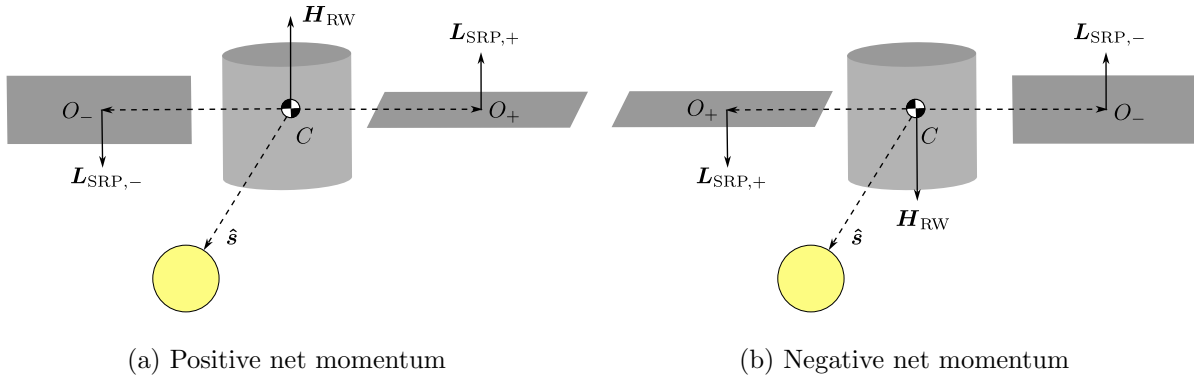


Figure 4.2: Array orientation for different net momentum configurations

the Sun direction. The net SRP torque acting on the arrays can be expressed as:

$$\begin{aligned} \mathbf{L}_{\text{SRP,net}} &= \mathbf{r}_{O/C,+} \times \mathbf{F}_{\text{SRP,+}} + \mathbf{r}_{O/C,-} \times \mathbf{F}_{\text{SRP,-}} \\ &= -PS(\cos \theta_+ - \cos \theta_-) \mathbf{r}_{O/C} \times \hat{\mathbf{s}} \end{aligned} \quad (4.16)$$

where  $\mathbf{r}_{O/C,i}$  are the distance between the system CM and the center of pressure of each array, which for symmetric geometries corresponds to the geometric center. Let's also make the simplifying assumption that  $\mathbf{r}_{O/C} = \mathbf{r}_{O/C,+} = -\mathbf{r}_{O/C,-}$ . Let's define the quantities:

$$\nu_i = -(\mathbf{r}_{O/C,i} \times \hat{\mathbf{s}}) \cdot \mathbf{H}_{\text{RW}}, \quad (4.17)$$

which are proportional to the net RW momentum as defined in Equation (4.1), from which it is possible to consider the following control law for the deflection angles of the arrays:

$$\theta_i = \begin{cases} \arctan(\sigma \nu_i^n) & \text{if } \nu_i > 0 \\ 0 & \text{if } \nu_i \leq 0 \end{cases}. \quad (4.18)$$

The control law in Equation (4.18) ensures that the array that produces a negative feedback SRP torque ( $\nu_-$ ) is not deflected, whereas the one which produces a positive feedback torque ( $\nu_+$ ) is deflected proportionally to the net RW momentum, as illustrated in Figure 4.2. The arctangent is used in order to bound the deflection angle in the  $[-\pi/2, \pi/2]$  domain. The factor  $\sigma$  is effectively a proportional gain, whereas the exponent  $n$  is introduced because, for  $n > 1$ , it is possible to obtain a deadband in the control law that has the benefit of limiting the chattering around  $\theta = 0$ , as shown in Figure 4.3.

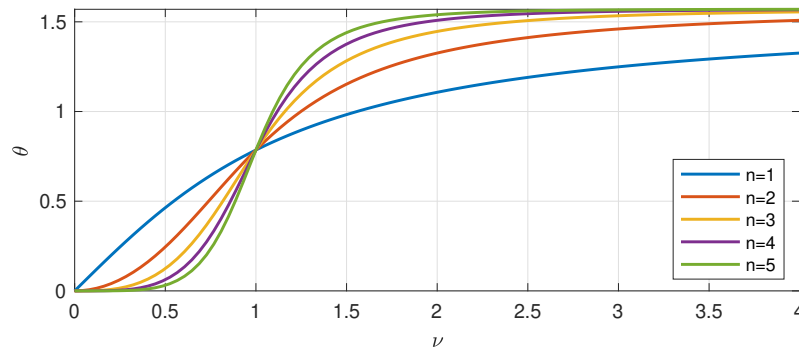


Figure 4.3: Control law in Equation (4.18) with  $\sigma = 1$

### 4.2.2 Stability Analysis

To prove the stability of the control law in Equation (4.18), it is useful to start from the Taylor-series expansion of  $\cos(\theta(\nu))$  under the assumption of small deflection angles. This gives:

$$\cos(\theta(\nu)) = \cos(\arctan(\sigma\nu^n)) \approx 1 - \frac{\sigma^2\nu^{2n}}{2} + o(\nu^{2n+1}). \quad (4.19)$$

Plugging this result back into the net SRP force expression allows to compute a simplified dynamics model, under the assumption that the only torque acting on the system is the SRP torque:

$$\dot{\mathbf{H}} = \mathbf{L}_{\text{SRP,net}} = -PS \left( 1 - \frac{\sigma^2\nu_+^{2n}}{2} - 1 \right) \mathbf{r}_{O/C} \times \hat{\mathbf{s}} = PS \frac{\sigma^2\nu_+^{2n}}{2} \mathbf{r}_{O/C} \times \hat{\mathbf{s}}. \quad (4.20)$$

Considering the positive definite candidate Lyapunov function  $V(\mathbf{H}) = H^2/4$  and under the assumption  $\dot{\mathbf{H}} = \dot{\mathbf{H}}_{\text{RW}} = \mathbf{L}_{\text{SRP,net}}$ , the Lyapunov derivative is:

$$\dot{V}(\mathbf{H}) = PS\sigma^2\nu_+^{2n} (\mathbf{r}_{O/C} \times \hat{\mathbf{s}}) \cdot \mathbf{H}_{\text{RW}} = -PS\sigma^2\nu_+^{2n+1}. \quad (4.21)$$

Per hypothesis, the only array to deflect is the one whose SRP torque feeds positively into the net momentum, i.e.  $\nu_+ = -(\mathbf{r}_{O/C,+} \times \hat{\mathbf{s}}) \cdot \mathbf{H}_{\text{RW}} > 0$ , therefore under this hypothesis, the Lyapunov derivative is negative definite. The case may exist where the vectors  $\mathbf{r}_{O/C}$ ,  $\hat{\mathbf{s}}$ , and  $\mathbf{H}_{\text{RW}}$  are colinear. If the net RW is parallel to the Sun direction, for instance, it is  $\nu_+ = \nu_- = 0$ , in which case the Lyapunov derivative becomes identically zero. In this case, analogous stability considerations can be made as to those in Section 4.1: this control method is locally asymptotically stable for the components of momentum that are not orthogonal to the cross product  $\mathbf{r}_{O/C} \times \hat{\mathbf{s}}$ .

### 4.2.3 Fully Reflective Surface

It is interesting to study the case of a rotating, fully-reflecting surface connected to the spacecraft, for momentum management purposes. The idea is to take advantage of the higher amount of momentum exchanged between the reflected photons and the rotating surface. The resulting SRP force is orthogonal to the surface and depends on the cosine of the angle between surface normal and the Sun squared. This significantly complicates the analytical model, however

it might give rise to an optimal orientation angle to maximize the momentum offloading torque. This analysis applies to an individual surface that can rotate about a body-fixed spacecraft axis. Besides considering an individual surface in this subsection, all the assumptions made in the fully absorbent solar array case still apply.

Let's define a simplified SRP force model:

$$\mathbf{F}_{\text{SRP}} = -F \operatorname{sgn}(\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2) (\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2)^2 \hat{\mathbf{a}}'_2. \quad (4.22)$$

In this simplified model, the assumptions are that all the photons emitted by the Sun are perfectly reflected by the surface, therefore the resulting force is antiparallel to the surface normal  $\hat{\mathbf{a}}'_2$ . All the other constants that contribute to SRP, such as distance from the Sun, area of the surfaces, reflectance of the surfaces, are lumped into the constant  $F$  in the expression in Equation (4.22). A more accurate SRP model is found in Reference 72. Let  $\mathbf{r}_{O/C}$  be the distance between the system CM and the center of pressure of the surface, which for symmetric geometries corresponds to the geometric center. Moreover, let  $\mathbf{H}_{\text{RW}}$  be the net wheel momentum as defined in Equation (4.1). In order to maximize momentum offloading, the component of SRP torque antiparallel to  $\mathbf{H}_{\text{RW}}$  should be maximized. This is equivalent to finding the minimum of the function:

$$\begin{aligned} h(\alpha) &= (\mathbf{r}_{O/C} \times \mathbf{F}_{\text{SRP}}) \cdot \mathbf{H}_{\text{RW}} = -F \operatorname{sgn}(\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2) (\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2)^2 (\mathbf{r}_{O/C} \times \hat{\mathbf{a}}'_2) \cdot \mathbf{H}_{\text{RW}} \\ &= F \|\boldsymbol{\eta}\| \operatorname{sgn}(\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2) (\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2)^2 \hat{\boldsymbol{\eta}} \cdot \hat{\mathbf{a}}'_2 \end{aligned} \quad (4.23)$$

where  $\boldsymbol{\eta} = \mathbf{r}_{O/C} \times \mathbf{H}_{\text{RW}}$  for convenience, and  $\alpha$  is the rotation angle between frames  $\mathcal{A}$  and  $\mathcal{A}'$  as defined in Equation (3.63). Note that  $h(\alpha)$  is a periodic function of  $\alpha$  of period  $\pi$ . The vectors  $\hat{\mathbf{s}}$  and  $\boldsymbol{\eta}$  are known and the constant  $F$  does not depend on  $\alpha$ . Without loss of generality, it is possible to cast the problem onto the  $\mathcal{A}$  frame: this simplifies the computations when considering that the vector  $\hat{\mathbf{a}}'_2$  is by definition orthogonal to  $\hat{\mathbf{a}}_1$ , therefore has coordinates  ${}^{\mathcal{A}}\hat{\mathbf{a}}'_2 = \{0, \cos \alpha, \sin \alpha\}^T$ . Let's also define the other relevant vectors component-wise:  ${}^{\mathcal{A}}\hat{\mathbf{s}} = \{s_1, s_2, s_3\}^T$  and  ${}^{\mathcal{A}}\hat{\boldsymbol{\eta}} = \{\eta_1, \eta_2, \eta_3\}^T$ . This allows to reformulate the expression for  $h(\alpha)$  as:

$$\bar{h}(\alpha) = \operatorname{sgn}(s_2 \cos \alpha + s_3 \sin \alpha) (s_2 \cos \alpha + s_3 \sin \alpha)^2 (\eta_2 \cos \alpha + \eta_3 \sin \alpha) \quad (4.24)$$

with  $h(\alpha) = F\|\boldsymbol{\eta}\|\bar{h}(\alpha)$ . It is easy to prove that  $-1 < \bar{h}(\alpha) < 1$ . The derivative with respect to  $\alpha$  is:

$$\begin{aligned} \bar{h}'(\alpha) = (s_2 \cos \alpha + s_3 \sin \alpha) [(2s_3\eta_2 + s_2\eta_3) \cos^2 \alpha \\ + 3(s_3\eta_3 - s_2\eta_2) \cos \alpha \sin \alpha - (2s_2\eta_3 + s_3\eta_2) \sin^2 \alpha] \end{aligned} \quad (4.25)$$

where the term  $\text{sgn}(s_2 \cos \alpha + s_3 \sin \alpha)$  is dropped for ease of notation. Equating  $\bar{h}'(\alpha)$  to zero and dividing by  $\cos \alpha$  gives the following equation in  $t = \tan \alpha$ :

$$(s_3 t + s_2) [(2s_2\eta_3 + s_3\eta_2)t^2 + 3(s_2\eta_2 - s_3\eta_3)t - (2s_3\eta_2 + s_2\eta_3)] = 0. \quad (4.26)$$

The first solution is:

$$\alpha_0 = -\text{atan} \left( \frac{s_2}{s_3} \right), \quad (4.27)$$

however it can be noticed that this is also the solution for which  $\bar{h}(\alpha) = 0$ . The discriminant of the second-order equation in  $t$  is:

$$\begin{aligned} \Delta &= 9(s_2\eta_2 - s_3\eta_3)^2 + 4(2s_2\eta_3 + s_3\eta_2)(2s_3\eta_2 + s_2\eta_3) \\ &= 9(s_2\eta_2 + s_3\eta_3)^2 + 8(s_2\eta_3 - s_3\eta_2)^2 \end{aligned} \quad (4.28)$$

which ensures that two solutions always exist. These are:

$$\alpha_+ = \text{atan} \left( \frac{3(s_3\eta_3 - s_2\eta_2) + \sqrt{\Delta}}{4s_3\eta_2 + 2s_2\eta_3} \right) \quad \alpha_- = \text{atan} \left( \frac{3(s_3\eta_3 - s_2\eta_2) - \sqrt{\Delta}}{4s_3\eta_2 + 2s_2\eta_3} \right). \quad (4.29)$$

To correctly choose between  $\alpha_+$  and  $\alpha_-$ , a bit more manipulation is required. The unit vectors  $\hat{\mathbf{s}}$  and  $\hat{\boldsymbol{\eta}}$  can be expressed in terms of their respective azimuth angles  $\xi$  and  $\zeta$  measured from vector  $\hat{\mathbf{a}}_1$ . This gives:

$$\hat{\mathbf{s}} \cdot \hat{\mathbf{a}}'_2 = \hat{\mathbf{s}} \cdot \hat{\mathbf{a}}_1 (\cos \alpha \cos \xi + \sin \alpha \sin \xi) = \hat{\mathbf{s}} \cdot \hat{\mathbf{a}}_1 \cos(\alpha - \xi) \quad (4.30a)$$

$$\hat{\boldsymbol{\eta}} \cdot \hat{\mathbf{a}}'_2 = \hat{\boldsymbol{\eta}} \cdot \hat{\mathbf{a}}_1 (\cos \alpha \cos \zeta + \sin \alpha \sin \zeta) = \hat{\boldsymbol{\eta}} \cdot \hat{\mathbf{a}}_1 \cos(\alpha - \zeta). \quad (4.30b)$$

Substituting Equation (4.30) into Equation (4.24) and rederiving gives:

$$\alpha_{\pm} = \frac{-3 \cos(\xi + \zeta) \pm \sqrt{8 + \cos^2(\xi - \zeta)}}{2 \sin \xi \sin \zeta + 2 \sin(\xi + \zeta)}. \quad (4.31)$$

Equation (4.31) is computationally equivalent to Equation (4.29), however it allows to express the candidate minimum and maximum  $\alpha$  angles in terms of two other angles which are unknown in principle, but bounded in the domain  $(\xi, \zeta) \in [0, 2\pi] \times [0, 2\pi]$ . It is therefore possible to compute the value of  $\bar{h}(\alpha_{\pm})$  for all the possible combination of  $\xi$  and  $\zeta$ . This gives the plots in Figure 4.4, where it is shown that  $0 \leq \bar{h}(\alpha_{+}) \leq 1$  and  $-1 \leq \bar{h}(\alpha_{-}) \leq 0$ , from which it is possible to conclude that, in order to maximize the momentum offloading effect of SRP, the optimal surface angle choice is:

$$\alpha_{R,SRP} = \text{atan} \left( \frac{3(s_3\eta_3 - s_2\eta_2) - \sqrt{9(s_2\eta_2 + s_3\eta_3)^2 + 8(s_2\eta_3 - s_3\eta_2)^2}}{4s_3\eta_2 + 2s_2\eta_3} \right). \quad (4.32)$$

Due to the arctangent being periodic of period  $\pi$ ,  $\alpha_i \pm n\pi \forall n \in \mathbb{Z}$  are also solutions of Equation (4.26): this happens because two configurations with the same array 180 degrees apart generate the same SRP torque.

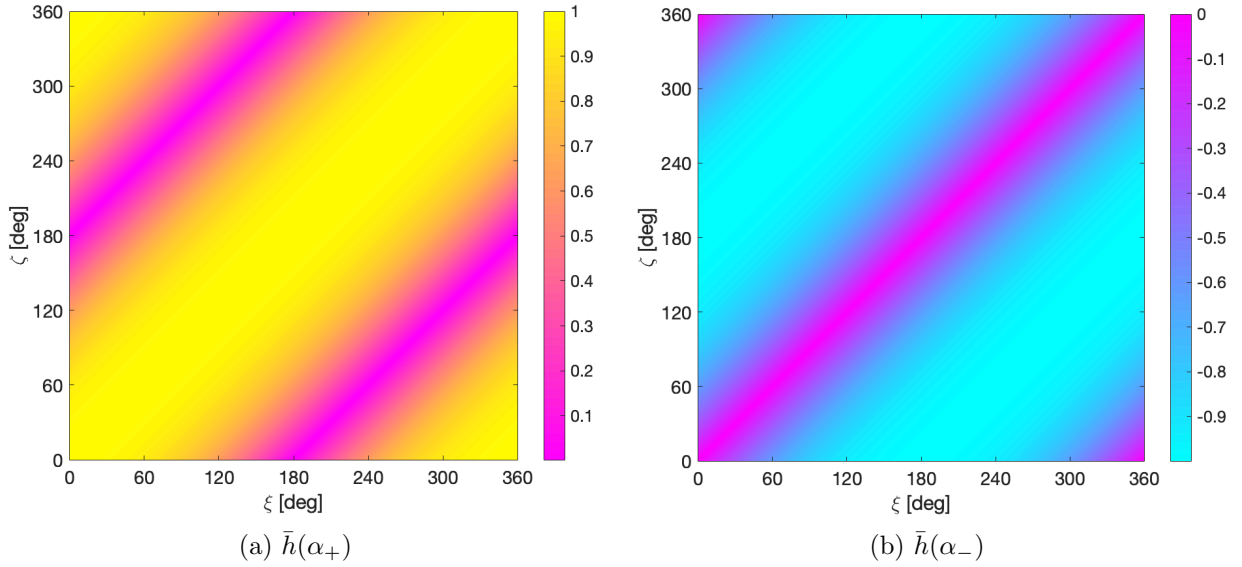


Figure 4.4:  $\bar{h}(\alpha_{\pm}(\xi, \zeta))$

### 4.3 Center of Mass Estimation

This section is motivated by the need to perform continuous CM estimation over the extended periods of time during which the electric thruster is firing, without interrupting the nominal course

of operations to perform calibration slew maneuvers to compute the exact location of the CM. Therefore, the aim for this section is to remove the assumption that the location of the system's CM is known, and rather estimate the CM location based on measurements of the system's states when the hub converges to the desired reference frame.

### 4.3.1 Traditional Approaches

Traditional CM estimation algorithms, such as that outlined in Reference 27, rely on accelerometers mounted on the spacecraft hub. The measurement model is given by the relative acceleration theorem:

$$\mathbf{a} = \mathbf{a}_{CM} + \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} \times \mathbf{r} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times (\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times \mathbf{r}) + 2\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times \dot{\mathbf{r}} \quad (4.33)$$

where  $\mathbf{a}$  is the measured quantity, and  $\mathbf{r}$  is the location of the center of mass to be estimated. Because the CM location is assumed to not vary over time, or vary very slowly allowing us to assume  $\dot{\mathbf{r}} \approx 0$ , and the system is not subject to external forces ( $\mathbf{a}_{CM} = 0$ ), the equation above simplifies to the linear model:

$$\mathbf{a} = [C(\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}, \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}})]\mathbf{r} \quad (4.34)$$

where the angular rates and accelerations of the hub are obtained from IMU measurements. These arguments, however, are not valid in the case of a thrust system such as the one considered in this paper. Because the external force acting on the system is not zero, the acceleration of the CM cannot be neglected:

$$\mathbf{a}_{CM} = \mathbf{t}/m \quad (4.35)$$

where  $\mathbf{t}$  is the thrust vector, and  $m$  is the mass of the system, which are both subject to uncertainty. Moreover, because the thruster must be aligned with a certain inertial direction, the angular rates and accelerations of the hub are typically zero, except during transient control responses. This makes it such that the linear model in Equation (4.34) is very weakly dependent on the CM location  $\mathbf{r}$ , and therefore very weakly observable.



### 4.3.2 Steady-state Reference Tracking

It is interesting to take a closer look at the control law Equation (2.49) when the system converges to the desired reference frame  $\mathcal{R}$ . It is reasonable to assume that this reference is, in first approximation, static, in order to ensure inertial pointing of the thruster and the solar arrays. It is therefore  $\boldsymbol{\omega}_{\mathcal{R}/\mathcal{N}} \approx \dot{\boldsymbol{\omega}}_{\mathcal{R}/\mathcal{N}} \approx 0$ . Under these assumptions, Equation (2.49) is reduced to a regulator control law:

$$\mathbf{u} = -K\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{N}} - [P]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} - [P][K_I]\mathbf{z} \quad \text{with} \quad \mathbf{z} = K \int_{t_0}^t \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} dt + [\mathbf{I}_{\text{tot},C}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \quad (4.36)$$

where the gyroscopic term  $\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([\mathbf{I}_{\text{tot},C}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [G_s]\mathbf{h}_s)$  is dropped without affecting the asymptotic stability properties of the control law [50], and the inertial frame  $\mathcal{R}$  is made coincide with the inertial frame  $\mathcal{N}$ , as it remains fixed in time. Additionally, once convergence to the reference frame is achieved, it is by definition  $\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} \approx \boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}} \approx 0$ . When this steady-state condition is met, the control torque is further reduced to:

$$\mathbf{u}_{\text{ss}} = -PK_I\mathbf{z} = -PK_IK \int_{t_0}^t \boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}} dt \quad (4.37)$$

where effectively the control torque coincides with the integral feedback term, and any dependence on the inertia of the system is lost. This means that, once steady state is achieved, the control torque required to stabilize the system is not affected by potential errors in the inertia tensor, which in contrast would only affect the transient response.

### 4.3.3 Measurement Model

The dynamics of the system when solar arrays and platform reach steady-state response are:

$$[\mathbf{I}_{\text{tot},C}]\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} = -\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([\mathbf{I}_{\text{tot},C}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [G_s]\mathbf{h}_s) + \mathbf{L}_{\text{thr}} + \mathbf{L}_{\text{ext}} + \mathbf{u} \quad (4.38)$$

where  $\mathbf{L}_{\text{ext}}$  is any general external unmodeled torque. With respect to Figure 1.2, the thruster torque on the system can be modeled as:

$$\mathbf{L}_{\text{thr}} = -\mathbf{r}_{C/T} \times \mathbf{t} = [\tilde{\mathbf{t}}]\mathbf{r}_{C/T} \quad (4.39)$$

where point  $T$  is the thrust application point, and it is known, and point  $C$  is the location of the CM, which is to be estimated.

Taking into consideration Equations (4.37) to (4.39), and once again making the assumption that the system has converged to the reference frame, the following correlation can be highlighted between the integral feedback term and the thruster torque:

$$\mathbf{Z} = [\tilde{\mathbf{t}}]\mathbf{r}_{C/T} \quad (4.40)$$

where  $\mathbf{Z} = PK_I \mathbf{z}$ . Equation (4.40) constitutes a linear measurement model for the CM location, where the thrust direction vector  $\mathbf{t}$  is, in first approximation, known, and the integral term  $\mathbf{Z}$  is extracted from the flight software code segment that computes the torque delivered to the system by the reaction wheels at steady state. Because the platform orientation changes with time, so does the thrust application point  $T$ . For this reason, the measurement model can be better reformulated introducing point  $B$ , the origin of the body frame  $\mathcal{B}$ , whose coordinates are known and do not change over time. The problem is therefore formulated in terms of the position of the CM with respect to point  $B$ :

$$\mathbf{Z} + [\tilde{\mathbf{t}}]\mathbf{r}_{T/B} = [\tilde{\mathbf{t}}]\mathbf{r}_{C/B} \quad (4.41)$$

where  $\mathbf{r}_{T/B}$  is the location of point  $T$  with respect to point  $B$ , and it is known accurately thanks to the reliable knowledge of the gimbal angles of the platform.

The main source of uncertainty is the thrust vector  $\mathbf{t}$ . While the mapping between the thruster-platform frame  $\mathcal{F}$  and the body frame  $\mathcal{B}$  is known accurately, the knowledge of the thrust vector in  $\mathcal{B}$ -frame coordinates  ${}^{\mathcal{B}}\mathbf{t}$  frame is ultimately dependent on how accurately the thruster performance is known in its own frame  ${}^{\mathcal{F}}\mathbf{t}$ . This means that if the thruster is misaligned with respect to the nominal direction, or if the magnitude of the thrust vector is off-nominal, the reliability of the measurement model is degraded. This work assumes, in first approximation, exact knowledge of the thrust vector  ${}^{\mathcal{F}}\mathbf{t}$ . Uncertainties in the model are discussed later this chapter.

#### 4.3.4 Observability

Based on the considerations made in the previous subsection, it is possible to define the following system:

$$\begin{cases} \dot{\mathbf{x}} = [A]\mathbf{x} \\ \mathbf{y} = [C]\mathbf{x} \end{cases} \quad (4.42)$$

where  $\mathbf{x} = \mathbf{r}_{C/B}$  is the quantity to be estimated,  $\mathbf{y} = \mathbf{Z} - [\tilde{\mathbf{t}}]\mathbf{r}_{T/B}$  constitutes the measurement, and  $[C] = [\tilde{\mathbf{t}}]$  the linear model. With the assumptions made in this work, the CM location does not change over time, which means that  $[A] = 0_{3 \times 3}$ . Analyzing the observability for a static configuration, i.e., a case in which the platform is not moving with respect to the hub, gives the following observability matrix [73]:

$$[\mathcal{O}] = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = [\tilde{\mathbf{t}}]. \quad (4.43)$$

Because  $[\tilde{\mathbf{t}}]$  is the skew-symmetric cross product matrix, it has rank 2. This means that, in such static configuration, the CM location is not fully observable. This could be inferred thinking about the problem from a physical standpoint: when no torque is acting on the system, i.e.,  $\mathbf{Z} = 0$ , it means that the thruster is being fired exactly through the system's CM. However, the exact position of the CM remains unknown, as it could lie anywhere along the thrust line.

However, the thrust vector does not remain constant in body-frame coordinates when the platform is actuated in order to perform momentum offloading. As a result, the measurement model is not static, but rather time-varying. In such case, the general observability matrix is defined as [74]:

$$[\mathcal{O}] = \begin{bmatrix} C(t) \\ \frac{d}{dt}C(t) \\ \vdots \\ \frac{d^{n-1}}{dt^{n-1}}C(t) \end{bmatrix}. \quad (4.44)$$

What this means in practical terms is that if the platform is held at a constant configuration, the problem is not fully observable. However, articulating the platform and taking measurements from different thruster-platform configurations, makes the problem fully observable even when the CM location has no dynamics.

#### 4.3.5 Recursive Least-Squares Algorithm

The recursive least-squares (LS) algorithm implemented to estimate the location of the CM, effectively, coincides with the correction step of a Kalman filter. In other words, the algorithm is equivalent to a Kalman filter where the state has no dynamics, and therefore there is no prediction.

At every  $n$ -th measurement update, the following quantities are given:

- $\epsilon_n = \sqrt{|\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}}|^2 + |\boldsymbol{\omega}_{\mathcal{B}/\mathcal{R}}|^2}$  : convergence error
- $[C_n] = [\tilde{\mathbf{t}}]$  : linear model
- $\mathbf{y}_n = \mathbf{Z} + [\tilde{\mathbf{t}}]\mathbf{r}_{T/B}$
- $[K_n]$  : optimal gain
- $\mathbf{x}_n = \mathbf{r}_{C/B}$  : CM location estimate
- $[P_n]$  : covariance of the state estimate.

The equations that define how to update each quantity at every step are, once again, those of the correction step in a regular Kalman filter [73]:

$$[K_n] = [P_n][C_n]^T ([C_n][P_n][C_n]^T + [R])^{-1} \quad (4.45)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + [K_n](\mathbf{y}_n - [C_n]\mathbf{x}_n) \quad (4.46)$$

$$[P_{n+1}] = ([I] - [K_n][C_n])[P_n]. \quad (4.47)$$

$[R]$  is the measurement noise covariance, and it is constant for every measurement. Because the integral feedback torque is not a conventional measurement, defining the measurement noise is

difficult. In this work, it is modeled as the variance of the integral feedback torque after the system has reached steady state according to the convergence error threshold. It is interesting to also consider the case where the platform gimbal angles are themselves affected by measurement noise: this affects the mapping  $[FB]$  between the platform frame and body frame, which results in an uncertain thrust vector estimate. Because this quantity affects the measurement  $\mathbf{y}_n$ , its uncertainty should be combined with the uncertainty of the integral feedback term to produce  $[R]$ . Uncertainty in the gimbal angles is not modeled in this work.

The convergence error  $\epsilon_n$  is used as a measure of whether the system has reached the steady state. If  $\epsilon_n$  is large it means that the system has not converged to the reference, therefore the integral term  $\mathbf{Z}$  is not representative of the torque that the thrust is applying on the system. For this reason, measurements are only processed when the convergence error drops below a user-defined threshold  $\epsilon_n < \bar{\epsilon}$ , which is a tuning parameter for the algorithm and depends on the gains of the attitude control law and the dynamic properties of the system.

#### 4.3.6 Biased Measurement

In the presence of unmodeled external perturbations such as SRP torque, a bias is implicitly added to the measurement model. This happens because the integral feedback control law in Equation (2.49) drives the spacecraft to the desired attitude regardless of the nature of the unmodeled perturbations, whether these are due to a thruster offset, SRP, or both. As a consequence, when a biased external perturbation acts on the system, a bias in the estimated CM location can be observed even when coherent measurement noise  $[R]$  is added to the LS algorithm. This steady-state bias in the CM estimate can be explained factoring the external torque into the model, and introducing point  $C^*$ :

$$\begin{aligned}\mathbf{Z} &= [\tilde{\mathbf{t}}]\mathbf{r}_{C/T} + \mathbf{L}_{\text{ext}} \\ \mathbf{Z} &= [\tilde{\mathbf{t}}] (\mathbf{r}_{C/C^*} + \mathbf{r}_{C^*/B} + \mathbf{r}_{B/T}) + \mathbf{L}_{\text{ext}} \\ \mathbf{Z} + [\tilde{\mathbf{t}}] (\mathbf{r}_{T/B} + \mathbf{r}_{C^*/C}) - \mathbf{L}_{\text{ext}} &= [\tilde{\mathbf{t}}]\mathbf{r}_{C^*/B}\end{aligned}\tag{4.48}$$

where, at steady state, the external torque becomes constant. Due to the nature of the control law that actuates the platform, at steady state the thruster continuously dumps the momentum on the wheels. Consequently, the thruster is not aligned with the center of mass, but rather with point  $C^*$ . This ensures that the resulting thruster torque is opposite to the external torque, minus its component along the thrust direction:

$$\mathbf{L}_{\text{net}} = \mathbf{L}_{\text{ext}} - (\mathbf{L}_{\text{ext}} \cdot \hat{\mathbf{t}}) \hat{\mathbf{t}} + \mathbf{r}_{C^*/C} \times \mathbf{t} = 0. \quad (4.49)$$

Combining Equations (4.48) and (4.49) gives:

$$\underbrace{\mathbf{Z} + [\tilde{\mathbf{t}}] \mathbf{r}_{T/B} - (\mathbf{L}_{\text{ext}} \cdot \hat{\mathbf{t}}) \hat{\mathbf{t}}}_{\mathbf{y}} = [\tilde{\mathbf{t}}] \underbrace{\mathbf{r}_{C^*/B}}_{\mathbf{x}} \quad (4.50)$$

which constitutes the effective measurement model in the presence of external perturbations. In the absence of these, Equation (4.50) coincides with Equation (4.41), and the result of the estimation is the actual CM location. Conversely, when external perturbations act on the system, the result of the estimation is the location of point  $C^*$ . The estimation bias error can be computed taking the cross product between  $\mathbf{t}$  and Equation (4.49):

$$\begin{aligned} \mathbf{t} \times (\mathbf{L}_{\text{ext}} - (\mathbf{L}_{\text{ext}} \cdot \hat{\mathbf{t}}) \hat{\mathbf{t}} + \mathbf{r}_{C^*/C} \times \mathbf{t}) &= \mathbf{t} \times 0 \\ \mathbf{t} \times \mathbf{L}_{\text{ext}} + \mathbf{t} \times (\mathbf{r}_{C^*/C} \times \mathbf{t}) &= 0 \\ \mathbf{t} \times \mathbf{L}_{\text{ext}} + t^2 \mathbf{r}_{C^*/C} - (\mathbf{t} \cdot \mathbf{r}_{C^*/C}) \mathbf{t} &= 0 \\ \mathbf{r}_{C^*/C} &= -\frac{\mathbf{t} \times \mathbf{L}_{\text{ext}}}{t^2} + (\hat{\mathbf{t}} \cdot \mathbf{r}_{C^*/C}) \hat{\mathbf{t}}. \end{aligned} \quad (4.51)$$

With respect to Figure 4.5,  $C^*$  could lie anywhere along the thrust line  $\mathbf{t}$ , resulting in the same thruster torque with respect to the real CM. From Equation (4.51) it is possible to observe that the bias has a component that is orthogonal to the thrust vector and one that is parallel. The orthogonal component is inversely proportional in magnitude to the thrust. The parallel component is a product of the poor observability of the problem which, at steady state, causes a loss of accuracy in the CM estimate along the thrust vector.

When fired through point  $C^*$ , the thruster compensates for two out of three components of the external torque, so long as this remains smaller or comparable in magnitude to the torque

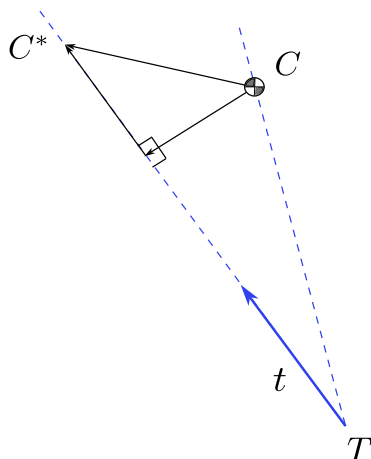


Figure 4.5: Center of Mass bias

that the thruster can generate, and Equation (4.50) does not deviate too much from linearity. Particularly interesting is the case in which the thruster output deviates from nominal expected behavior: in such case the unmodeled perturbation consists in the torque error with respect to nominal behavior. This error also results in a bias Equation (4.50), but it ultimately does not prevent the system from converging to a stable configuration.

## Fully Articulated Spacecraft Dynamics Simulation

The purpose of this chapter is to combine the results outlined in Chapters 2 to 4 and implement them in simulation to show the expected behavior. This chapter considers a spacecraft similar to the one sketched in Figure 1.2, on a heliocentric orbit, propelled by a SEP thruster. The spacecraft is at a heliocentric radius of approximately  $2/3$  AU, which roughly corresponds to the orbit of Venus, in a circular orbit. The thruster is required to be aligned with an in-plane direction that remains constant throughout all the scenarios without loss of generality of the following results. The reason for this choice is to simulate the behavior of the spacecraft in a scenario where SRP plays a significant role into the dynamics. The thruster is mounted on a platform that is connected to the spacecraft hub via a two-axes gimbal that enables tip-and-tilt type of rotations. Additionally, the spacecraft features two solar arrays attached to the hub via one-degree-of-freedom hinges, which allow the arrays to rotate about the hinge axis.

The results shown in this chapter are presented incrementally, where many assumptions and simplifications are made at first to more easily show the expected behavior of individual, incremental additions. Further down, these assumptions are removed to show the behavior of the system as simulation complexity increases.

Figure 5.1 shows a more comprehensive block diagram than the simpler scenario shown in Figure 2.2, which captures the full complexity of the flight software architecture needed in order to implement the guidance and control for a spacecraft with articulated appendages like the one discussed in the previous chapters and sketched in Figure 1.2. This work does not simulate any form



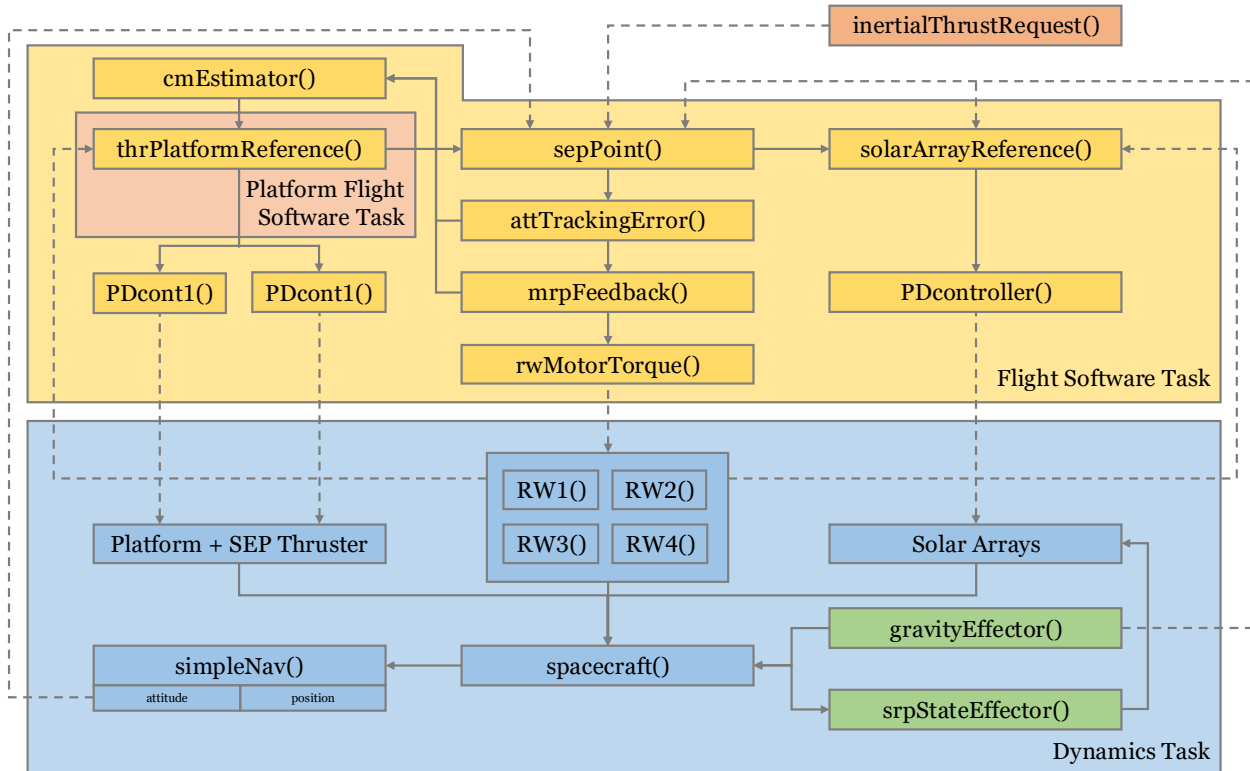


Figure 5.1: Full Basilisk block diagram for SEP point scenario

of navigation algorithm besides the `cmEstimator()` module, which implements the CM estimation algorithm outlined in Section 4.3. Knowledge of the spacecraft states such as attitude, angular rates, inertial position, and inertial velocity, are obtained through orbit and attitude determination filters that combine sensor measurements with the system dynamics. Because no novel navigation algorithms are developed in this dissertation, the navigation aspect is removed from the simulation, and relevant spacecraft states are provided as inputs to the flight software modules in terms of simulated noisy estimates and measurements.

One peculiar aspect of Figure 5.1 is the introduction of a separate flight software task for the `thrusterPlatformReference()` module, which is run at a different frequency with respect to the main flight software task. This is motivated by the desire to have the thruster platform change its orientation once every hour, as opposed to the other guidance and control components which are updated at 0.5 Hz. The reason is that, for the CM estimation to work accurately, the system needs

time for the attitude to converge to the reference within a certain tolerance. The dynamics task is updated at 2 Hz.

The optical properties used for the spacecraft hub and solar array surfaces are the same as for the Lucy spacecraft [75].

## 5.1 Simulation Setup

### 5.1.1 Spacecraft Hub Properties

The spacecraft hub has a mass  $m_{\text{hub}} = 2500$  kg and the following inertia tensor, expressed with respect to the principal frame  $\mathcal{B}$ :

$${}^{\mathcal{B}}[\mathbf{I}_{\text{hub}}] = \begin{bmatrix} 1725 & -5 & -12 \\ -5 & 5525 & 43 \\ -12 & 43 & 4810 \end{bmatrix} \text{ kg m}^2. \quad (5.1)$$

The location of the center of mass of the hub is  ${}^{\mathcal{B}}\mathbf{r}_{C_{\text{hub}}/B} = \{0.008, -0.010, 1.214\}^T$  m with respect to the origin  $B$  of the body frame. The body frame is centered at the bottom surface of the spacecraft.

The following optical coefficients are used for the surfaces of the hub:

$$\lambda_{\text{hub}} = 0.525 \qquad \rho_{\text{hub}} = 0.336 \qquad \delta_{\text{hub}} = 0.139. \quad (5.2)$$

### 5.1.2 Reaction Wheel Configuration and Mass Properties

Figure 5.2 shows the reaction wheel setup for the spacecraft hub. The RWs are installed such that two reaction wheels can provide torque about the first principal body axis  ${}^{\mathcal{B}}\hat{\mathbf{b}}_1 = (1, 0, 0)^T$ , two about the second principal body axis  ${}^{\mathcal{B}}\hat{\mathbf{b}}_2 = (0, 1, 0)^T$ , and all four contribute to the torques about the third principal body axis  ${}^{\mathcal{B}}\hat{\mathbf{b}}_3 = (0, 0, 1)^T$ . This is achieved by a 40 deg upwards tilt angle in the direction of the wheels' spin axes  $\hat{\mathbf{u}}_j$ , and it is done to ensure that control about each axis

is achievable even in the case of a single wheel failure. This results in the reaction wheel mapping:

$$[G_s] = [{}^B\hat{u}_1, {}^B\hat{u}_2, {}^B\hat{u}_3, {}^B\hat{u}_4] = \begin{bmatrix} \cos 45^\circ \cos 40^\circ & \cos 135^\circ \cos 40^\circ & \cos 225^\circ \cos 40^\circ & \cos 315^\circ \cos 40^\circ \\ \sin 45^\circ \cos 40^\circ & \sin 135^\circ \cos 40^\circ & \sin 225^\circ \cos 40^\circ & \sin 315^\circ \cos 40^\circ \\ \sin 40^\circ & \sin 40^\circ & \sin 40^\circ & \sin 40^\circ \end{bmatrix} \quad (5.3)$$

The RWs have an inertia  $I_{w_s} = 0.16 \text{ kg m}^2$ , can deliver a torque up to 0.2 Nm, and can spin up to 6000 rpm. Under the assumption that the center of mass of each wheel is aligned with the respective spin axis, the RW motion does not affect the CM of the system.

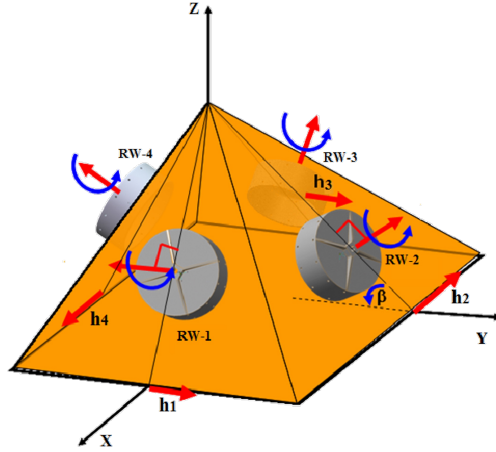


Figure 5.2: Four reaction wheels in pyramid configuration

### 5.1.3 Solar Arrays Configuration, Properties, and Control

The solar arrays are modeled as discs with a diameter of 7 m, whose center of mass is located at 3.75 m from the hinges, along the respective rotation axes  ${}^B\hat{a}_1 = \{\pm 1, 0, 0\}^T$ . The locations of the hinges in the hub are  ${}^B\mathbf{r}_{A/B} = \{\pm 0.75, 0, 0.45\}^T$  m. The mass of the solar arrays is  $m_{sa} = 85$  kg and the inertia tensor is:

$${}^A[\mathbf{I}_{sa}] = \begin{bmatrix} 250 & 0 & 0 \\ 0 & 250 & 0 \\ 0 & 0 & 500 \end{bmatrix} \text{ kg m}^2 \quad (5.4)$$

with respect to the solar array frame  $\mathcal{A}$  described in Figure 3.9. The following optical coefficients are used for the power-generating surface of the arrays:

$$\lambda_{\text{sa}}^+ = 0.680 \quad \rho_{\text{sa}}^+ = 0.160 \quad \delta_{\text{sa}}^+ = 0.160 \quad (5.5)$$

and for the back-surface of the arrays:

$$\lambda_{\text{sa}}^- = 0.440 \quad \rho_{\text{sa}}^- = 0.0 \quad \delta_{\text{sa}}^- = 0.560. \quad (5.6)$$

The equations of motion of the solar arrays are solved via the back-substitution method, as explained in Section 2.4. The relative rotation between the hub and the arrays is controlled via a torque input. This is the torque exchanged between the hub and each array, and it has the form of a PD control law:

$$\mathbf{u}_{\text{SA}} = -[K_{\text{SA}}(\alpha - \alpha_R) + P_{\text{SA}}(\dot{\alpha} - \dot{\alpha}_R)] \hat{\mathbf{a}}_1 \quad (5.7)$$

where  $\alpha$  is the displacement angle between frames  $\mathcal{A}$  and  $\mathcal{A}$ , and  $\alpha_R$  is the desired reference angle. In this work, the center of mass of the arrays is aligned with the spin axis, therefore the motion of the arrays does not affect the location of the CM of the system.

#### 5.1.4 Gimbal Thruster Platform Modeling and Control

The thruster platform assembly is modeled as a rigid body with negligible mass, attached to the bottom surface of the spacecraft. The platform frame  $\mathcal{F}$  and its motion with respect to the body frame  $\mathcal{B}$  are those described in Figure 1.2 and in Section 3.1. A two axis gimbal connects the platform to the main hub: the two gimbal angles are controlled via torque inputs about the gimbal axes, once again according to PD control laws:

$$\mathbf{u}_{\text{PL}} = -[K_{\text{PL}}(\nu_1 - \nu_{R1}) + P_{\text{PL}}(\dot{\nu}_1 - \dot{\nu}_{R1})] \hat{\mathbf{m}}_1 - [K_{\text{PL}}(\nu_2 - \nu_{R2}) + P_{\text{PL}}(\dot{\nu}_2 - \dot{\nu}_{R2})] \hat{\mathbf{f}}_2. \quad (5.8)$$

The thrust vector  $\mathbf{t}$  is defined in platform frame coordinates, as it is applied to the platform. However, applying the back-substitution method, the force and torque resulting from the thruster's action are mapped into resulting force and torque on the system as a whole. The thruster is

assumed to fire continuously, with a specific impulse  $I_{sp} = 1600$  s and a thrust output of  $t = 0.27$  N. In nominal conditions, the thrust is along the platform axis  ${}^{\mathcal{F}}\hat{\mathbf{f}}_z = \{0, 0, 1\}^T$ . With respect to Figure 1.2, the mount frame  $\mathcal{M}$  coincides with the body frame  $\mathcal{B}$ . The origin of the platform-fixed frame  $F$  also coincides with the origin  $B$  of the body frame, and for zero tip-and-tilt platform rotation angles  $\nu_1 = \nu_2 = 0$ , the frames  $\mathcal{F}$  and  $\mathcal{B}$  coincide. SEP thrusters are typically characterized by a swirl torque, which is the result of the rotational motion resulting from the Lorentz forces acting on the ions as they transit through the thruster’s magnetic field [76]. This swirl torque is modeled as a torque along the thrust direction vector. The magnitude of the torque is proportional to the magnitude of the thrust, with the proportionality constant set to  $10^{-3}$  m.

Mass depletion is not simulated in this work: that means that, even though the thruster is fired for extended periods of time, the mass of the system does not change. As a consequence, the location of the CM also does not change throughout the simulation. This is obviously a simplifying assumption. However, because the shift in the CM is primarily due to mass depletion, and because mass flow rate of electric thrusters is typically small, it is reasonable to assume that significant CM shifts happen over time scales that are much bigger than the time windows simulated in this work.

### 5.1.5 Solar Radiation Pressure Modeling

For the purpose of SRP modeling, the spacecraft is represented as a collection of 10 facets with negligible thickness. Six square facets and four circular facets are used to model the rigid hub and solar arrays, respectively. The platform and thruster assembly are not considered for the SRP evaluation due to their relatively small size compared to the spacecraft hub and solar arrays.

Each facet is characterized by an area  $A$ , a vector normal to its surface  $\hat{\mathbf{n}}$ , a position vector from the spacecraft center of mass to the facet center of pressure,  $\mathbf{r}_{PC}$ , and three optical coefficients representing the interaction of impinging photons with the facet surface. The fraction of specularly reflected, diffusely scattered, and absorbed photons are represented using the coefficients  $\rho$ ,  $\delta$ , and  $\lambda$ , respectively, with  $\lambda + \rho + \delta = 1$ .

A faceted force model is used to estimate the SRP force acting on the spacecraft [72]:

$$\mathbf{F}_{\text{SRP}} = \sum_{i=1}^{10} \mathbf{F}_{\text{SRP}_i} = -P(|\mathbf{r}_{\text{sc}/\odot}|) \sum_{i=1}^{10} A_i \cos(\theta_i) \left[ (1 - \rho_i) \hat{\mathbf{s}} + 2 \left( \frac{\delta_i}{3} + \rho_i \cos(\theta_i) \right) \hat{\mathbf{n}}_i \right] \quad (5.9)$$

where  $\hat{\mathbf{s}}$  is the unit direction vector pointing radially towards the Sun from the spacecraft body-frame origin,  $\theta$  is defined as the incidence angle between each facet normal vector and the Sun-direction vector, and  $P(|\mathbf{r}_{\text{sc}/\odot}|)$  is the pressure acting on the spacecraft scaled by the spacecraft heliocentric distance. The vector quantities in Equation (5.9) are expressed in spacecraft principal body-frame components. The total torque acting about the spacecraft center of mass due to SRP is calculated by summing the torque contributions over all 10 facets:

$$\mathbf{L}_{\text{SRP},C} = \sum_{i=1}^{10} \mathbf{L}_{\text{SRP},C_i} = \sum_{i=1}^{10} \mathbf{r}_{P_i C} \times \mathbf{F}_{\text{SRP}_i}. \quad (5.10)$$

## 5.2 Simulation Results with Perfect CM Knowledge

### 5.2.1 Perfect Thruster-CM Alignment, Short Time Window

This first set of simulation results displayed in Figure 5.3 aims to show the response that happens when the spacecraft is actuated and maneuvered to the desired reference. For this purpose, the results are simulated over a relatively short simulation time window of 6 hours, given that over longer periods of time the high-frequency responses become harder to visualize. The gains  $\kappa$  and  $\kappa_I$  of the platform control law in Equation (4.8) are set to zero, to show the correct alignment of the thruster with the CM. The SEP swirl torque is set to zero in this first simulation. Moreover, the `cmEstimator` module is disconnected from the simulation, and the `platformReference` module is informed with the exact location of the system CM: while this is not a realistic assumption for a real onboard computer, it serves the purpose of showing the performance of the guidance and control algorithms for the thruster platform. Figure 5.3a shows the response of the spacecraft hub to the commanded reference. The response is underdamped, mainly due to the integral feedback term in the attitude control law. The desire is for the hub to reach convergence within a certain tolerance in less than an hour. Figure 5.3b shows that the wheels spin up and then back to zero

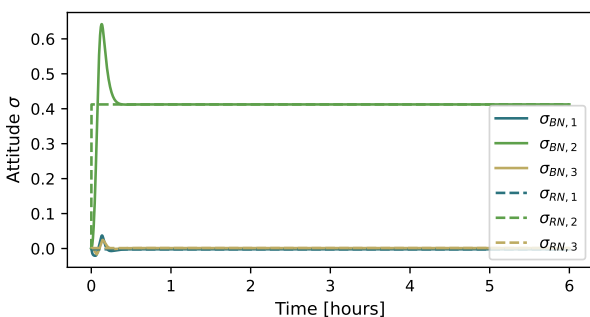
for this rest-to-rest maneuver. Because the thruster is correctly aligned with the CM, as shown in Figures 5.3d and 5.3h, no momentum is being added to the wheels. The SRP torque is however contributing to the momentum buildup, although this cannot be appreciated over the simulated 6 hour time window. Lastly, Figures 5.3e and 5.3f show the correct tracking of the requested solar array angles, and the angle between the array normals and the Sun dropping to zero as the spacecraft correctly tracks its reference.

### 5.2.2 Perfect Thruster-CM Alignment, Long Time Window

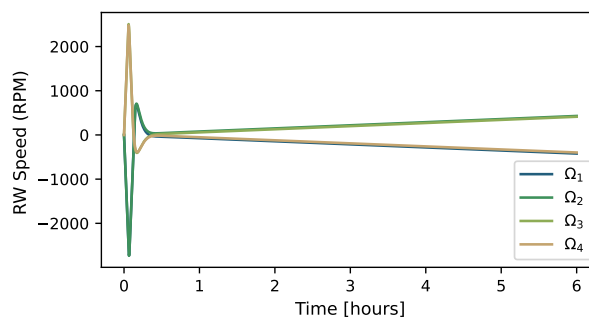
The simulation results in Figure 5.4 show the same problem setup as in Figure 5.3, however the simulation is run for a longer time window of 3 days to tease out the long term effects. Figure 5.4a shows that the spacecraft attitude is essentially held inertially fixed. However, because the spacecraft is orbiting the Sun, the relative position with respect to it changes over time, thus the arrays need to be constantly rotated to maintain perfect illumination: this is shown in Figures 5.4e and 5.4f. While the thruster is maintained aligned with the CM and thus not producing any torques, Figure 5.4g shows the evolution of the SRP torque due to the changes in relative orientation between the hub and the arrays, causing a progressive increase in torque about the  $\hat{\mathbf{b}}_1$  axis. As a result, the SRP torque is stored in the RWs in order to hold the attitude inertially fixed, as shown in Figure 5.4b. The momentum accumulated in the RWs is only due to the SRP torque: because no swirl torque is simulated here, and the thrust vector is aligned with the CM, the thruster does not produce any net torque on the system.

### 5.2.3 Thruster-based Momentum Management, Long Time Window

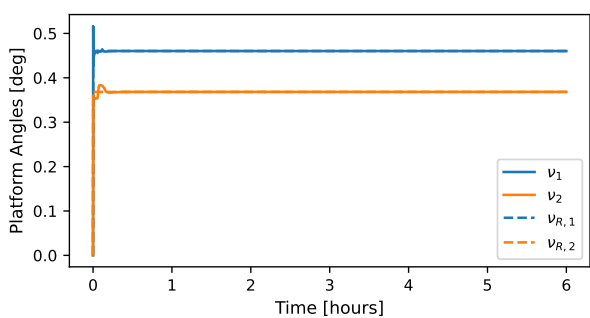
The simulation results in Figure 5.5 show the effect of the gimbaled-thruster-based momentum management technique when the control gain is set to  $\kappa = 2.5 \cdot 10^{-4}$  Hz. The swirl torque is still not being simulated. The most noticeable effect is in the RW speeds in Figure 5.5b which, thanks to the momentum being constantly offloaded by the thruster, do not spin up over time despite the persisting SRP torque. Figures 5.5c and 5.5d show the evolution of the platform angles over time



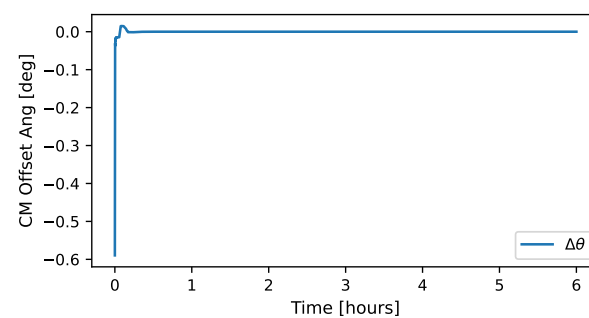
(a) Attitude vs reference attitude



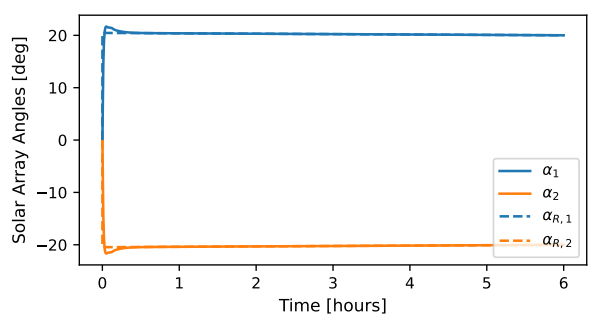
(b) Reaction wheel speeds



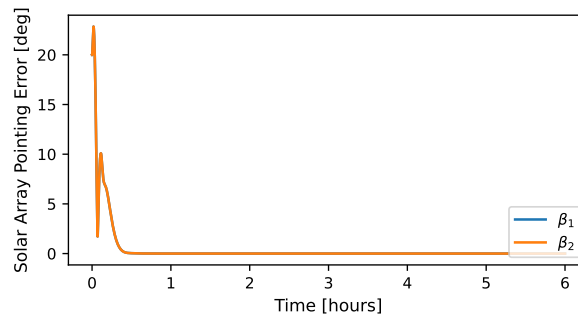
(c) Platform angles vs reference



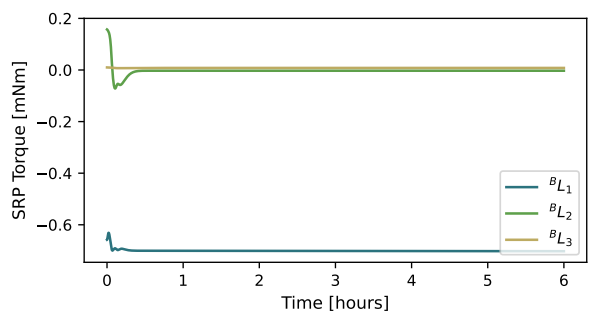
(d) Thruster-CM offset



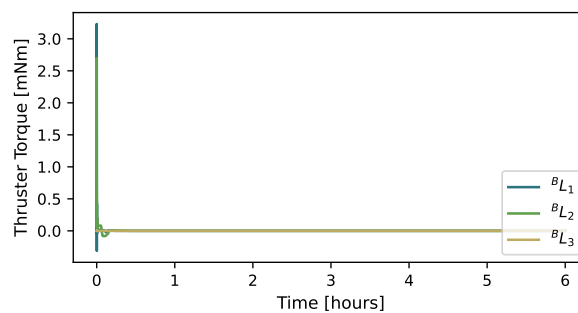
(e) Solar array angles and reference



(f) Solar array Sun pointing accuracy



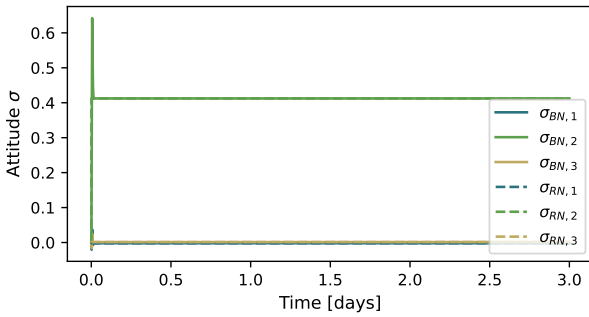
(g) SRP torque



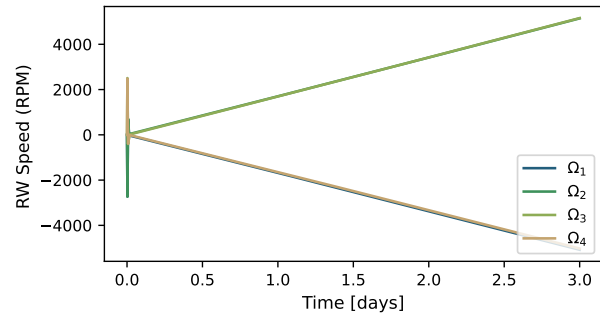
(h) Thruster torque

Figure 5.3: Perfect thruster-CM alignment, short time window

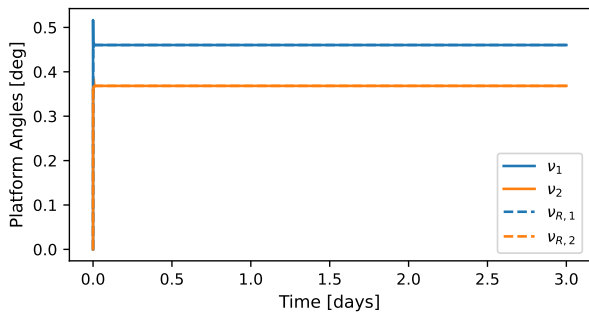




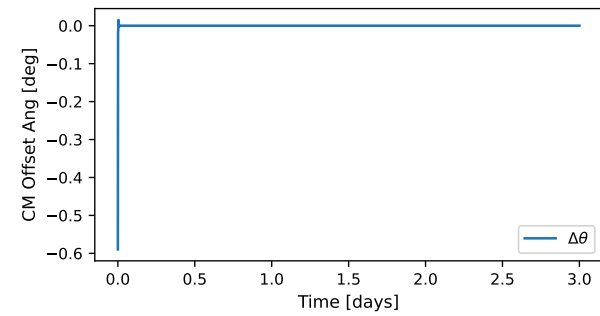
(a) Attitude vs reference attitude



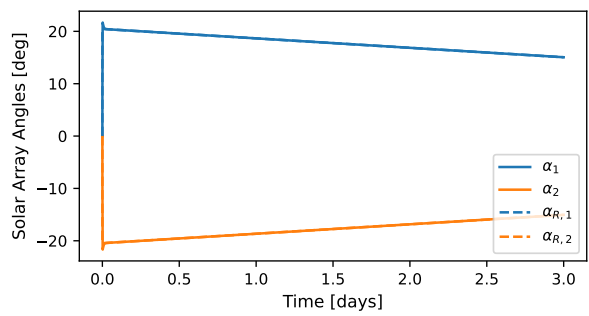
(b) Reaction wheel speeds



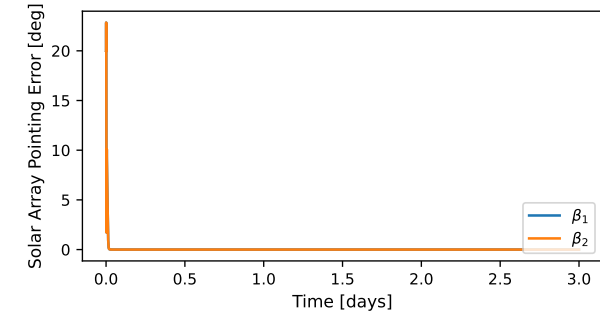
(c) Platform angles vs reference



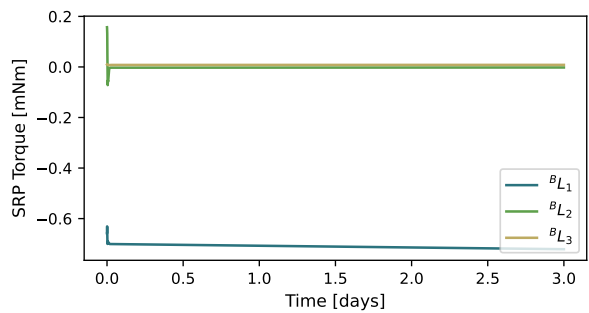
(d) Thruster-CM offset



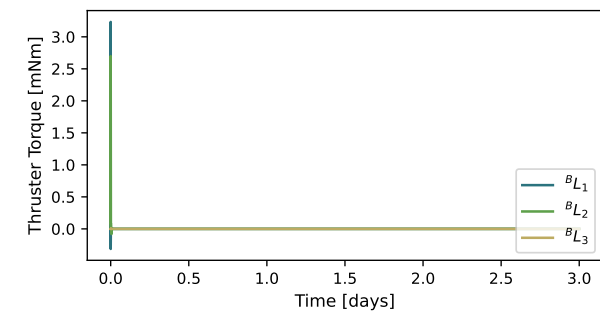
(e) Solar array angles and reference



(f) Solar array Sun pointing accuracy



(g) SRP torque



(h) Thruster torque

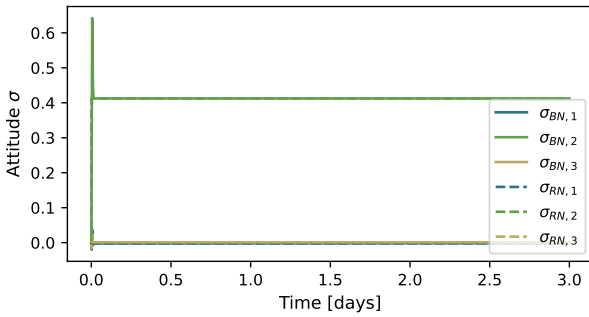
Figure 5.4: Perfect thruster-CM alignment, long time window

and the offset between the thrust vector and the CM. As a result, the thruster torque in Figure 5.5h develops a component along the  $\hat{\mathbf{b}}_1$  axis to counteract the SRP torque. Because very small gimbaling angles are sufficient to offload the wheels, there are no appreciable variations in the attitude plots in Figure 5.5a. However, as will be more obvious in the following simulations, every change in gimbaling angles is associated in a change in the reference attitude, in order to maintain the thruster pointed along the inertially-fixed direction.

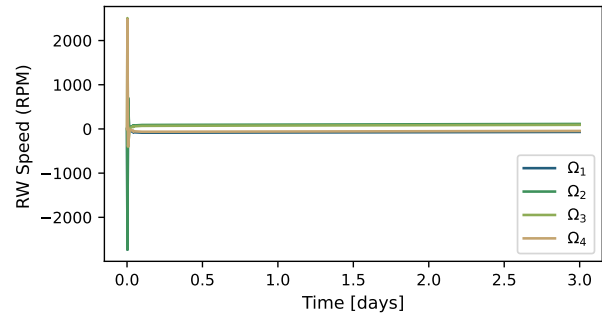
#### 5.2.4 Thruster-based Momentum Management with Swirl Torque and SRP Unloading, Long Time Window

Figure 5.6 shows the simulation results obtained when the SEP thruster is simulated including the effects of the swirl torque. After 1.5 days from ignition the arrays are switched from power-generating mode to momentum management mode, as described in Section 4.2. Figure 5.6b shows the most evident results of this simulation: in the first 1.5 days, the wheel momentum is building up, mainly along the positive  $\hat{\mathbf{b}}_3$  direction, due to the swirl torque. The thruster is still unloading the momentum building up due to SRP, but because the swirl torque is by definition along the thrust direction, this component of momentum cannot be unloaded. The change in solar array pointing is evident in Figures 5.6e and 5.6f, where variations in pointing are shown. At  $t = 1.5$  days, Figure 5.6b shows that the net momentum starts to decrease as a result of the SRP torque caused by the reorientation of one solar array. The law in Equation (4.18) is used to compute the array deflection angle, which is scaled by a factor  $\Theta_{\max} = 2/3$  to ensure that the incident sunlight is never more than 60 deg with respect to the array normal. The other parameter in the control law are  $\sigma = 5 \cdot 10^{-5}$  and  $n = 2$ .

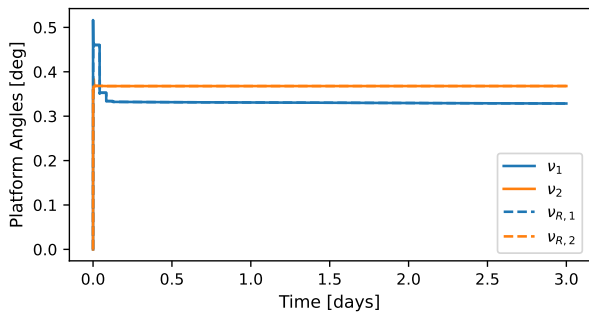
Figures 5.6g and 5.6h show that in the first part of the simulation the thruster torque is matching the SRP torque. In the second part, one array deflects to correct the momentum buildup due to the swirl torque. Because of the deadband in the control law, the array settles at 30 offset with respect to the Sun direction. After the array deflection, the thruster is reoriented to ensure that the thruster torque matches the SRP components along  $\hat{\mathbf{b}}_1$  and  $\hat{\mathbf{b}}_2$ .



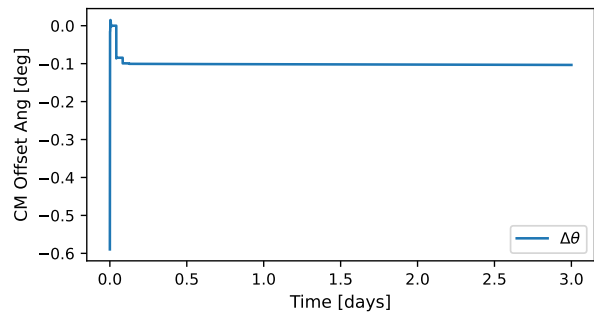
(a) Attitude vs reference attitude



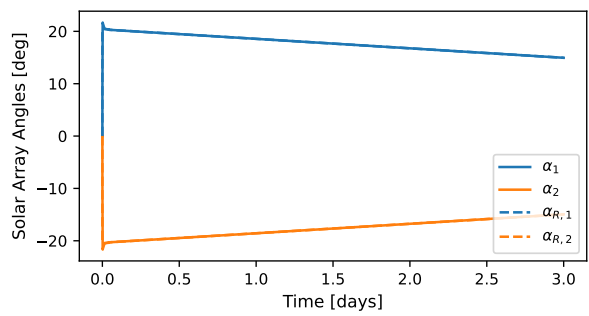
(b) Reaction wheel speeds



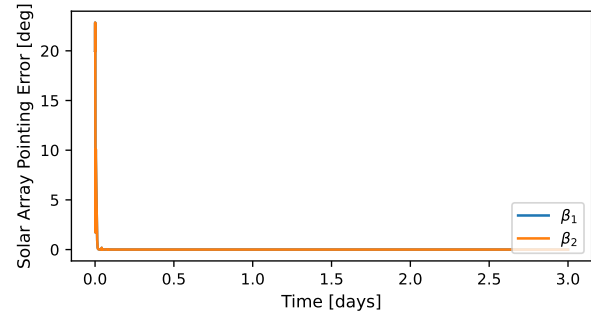
(c) Platform angles vs reference



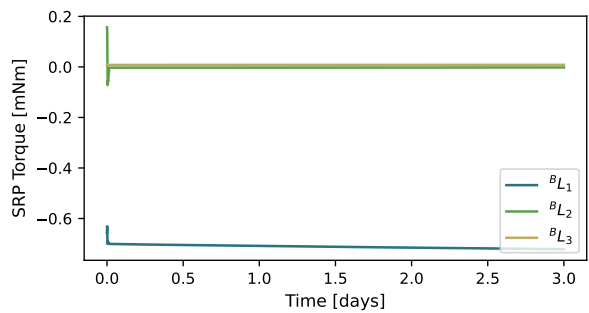
(d) Thruster-CM offset



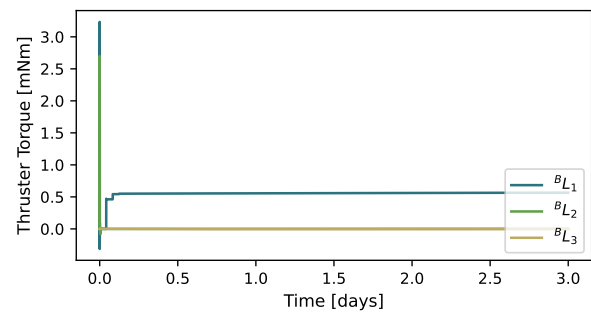
(e) Solar array angles and reference



(f) Solar array Sun pointing accuracy

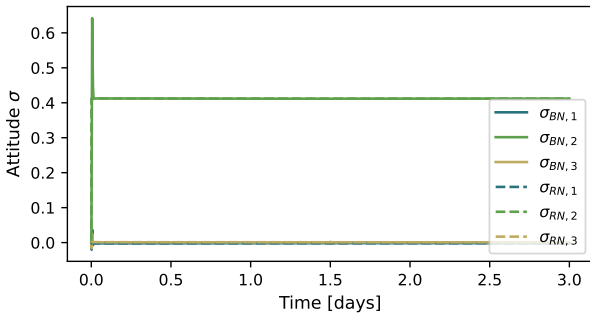


(g) SRP torque

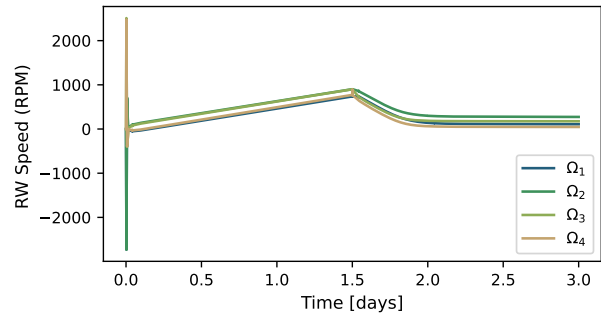


(h) Thruster torque

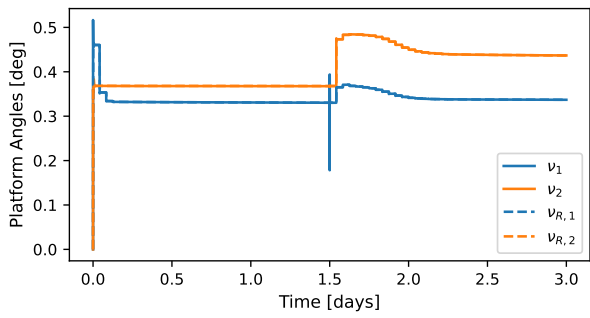
Figure 5.5: Thruster momentum management, long time window



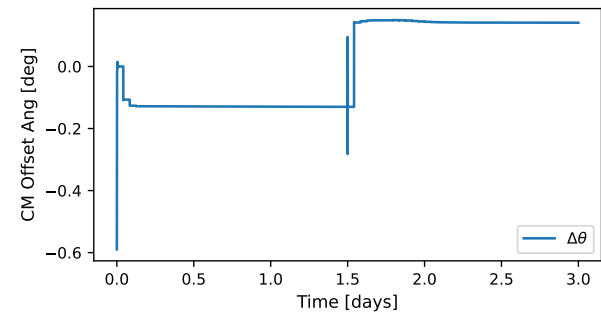
(a) Attitude vs reference attitude



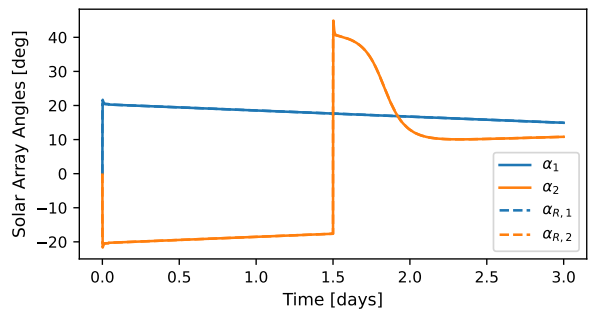
(b) Reaction wheel speeds



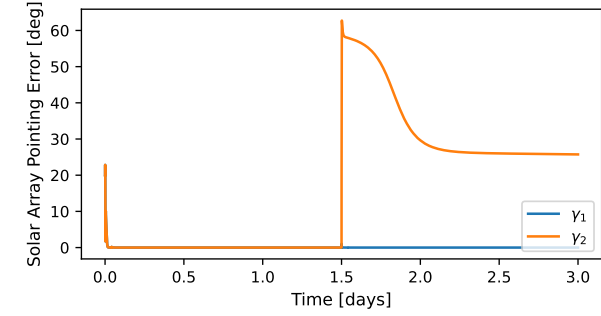
(c) Platform angles vs reference



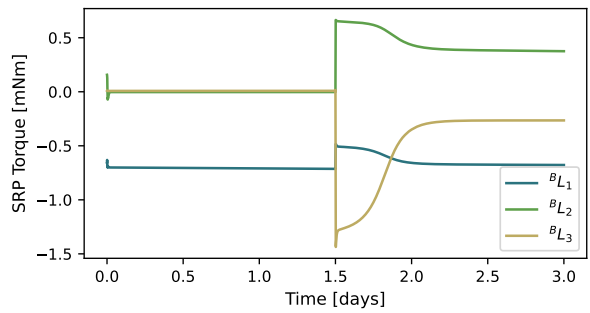
(d) Thruster-CM offset



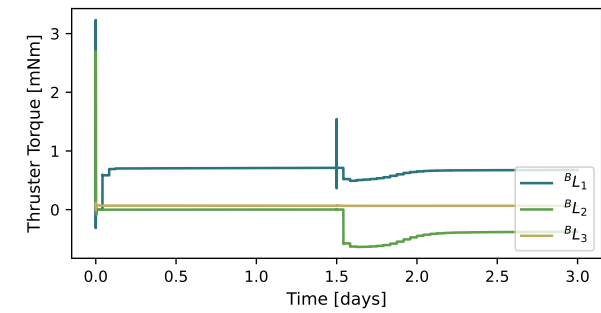
(e) Solar array angles and reference



(f) Solar array Sun pointing accuracy



(g) SRP torque



(h) Thruster torque

Figure 5.6: Thruster & solar array momentum management, long time window

### 5.3 Simulation Results with CM Uncertainty

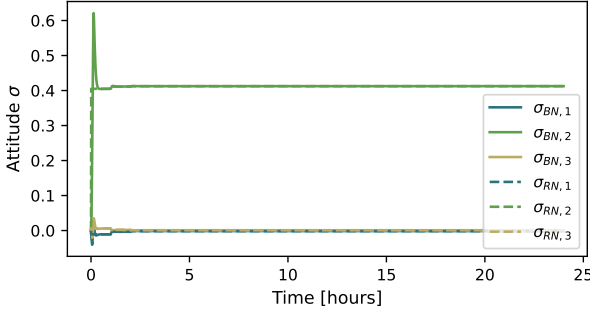
This last subsection shows the combined performance of the momentum management strategy combined with uncertain CM knowledge. The robustness of the approach is tested with a relatively inaccurate initial guess on the CM location: the flight software is seeded with the initial guess  $\mathbf{r}_{C/B,0} = \{0.04, -0.05, 1.25\}$  and initial state covariance  $\mathbf{P}_0 = 0.0025[\mathbf{I}]$ , corresponding to a standard deviation of 5 cm along each component of the CM location.

#### 5.3.1 Platform Integral Feedback

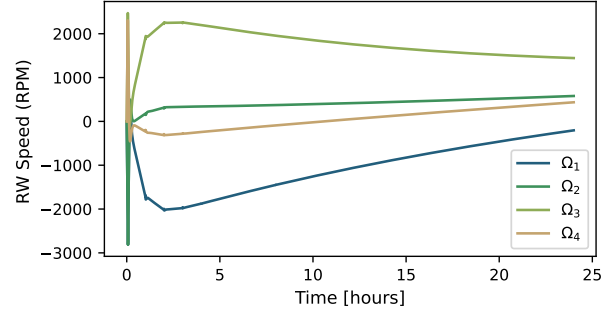
This first set of results is obtained implementing an integral feedback term with gain  $\kappa_I = 3 \cdot 10^{-9} \text{ Hz}^2$  in the platform control law in Equation (4.8). Figure 5.7 shows the simulated results, where the plots for the solar arrays are removed, as the information that they provide is not interesting for the purpose of this section. Figures 5.7c and 5.7d show that platform quickly converges to a steady-state equilibrium, provided that the platform reference gains  $\kappa$  and  $\kappa_I$  are tuned properly. As in the previous section, this equilibrium consists in a configuration in which the SRP torque is compensated almost exactly by the thruster torque, as shown in Figures 5.7e and 5.7f. The main issue is that the introduction of the integral feedback term causes the momentum management control loop to become significantly underdamped. As a result, Figure 5.7b shows that the reaction wheels grow quickly during the transient response due to the misalignment between the thruster and CM. At steady state, the wheel speeds are progressively being reduced by the continuous momentum management action of the thruster.

#### 5.3.2 CM Estimation without External Perturbations

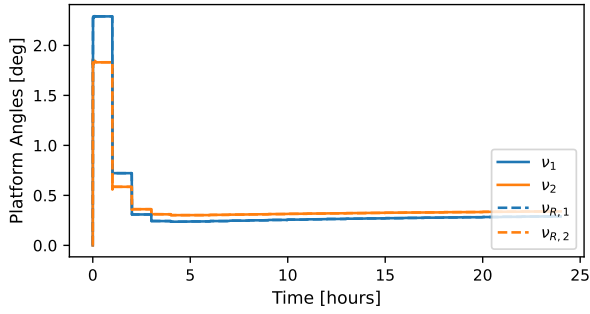
This subsection presents the performance of the `cmEstimator` module for which the mathematical foundation was laid out in Section 4.3. This simulation is run removing the SRP effects on the spacecraft, in order to non have any external perturbations acting on the system. This is obviously a non-realistic scenario, but it is run to validate the effectiveness of the `cmEstimator`



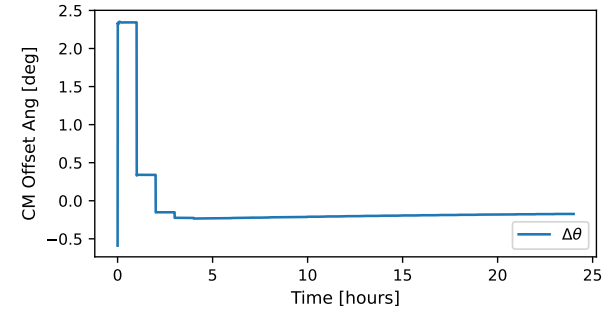
(a) Attitude vs reference attitude



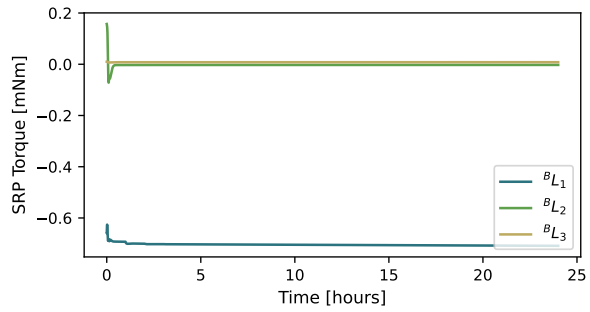
(b) Reaction wheel speeds



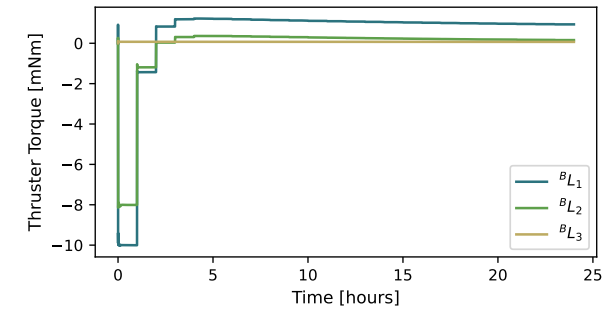
(c) Platform angles vs reference



(d) Thruster-CM offset



(e) SRP torque



(f) Thruster torque

Figure 5.7: Momentum management with CM uncertainty and integral feedback

module. Because no external disturbances are added to the system, the only source of error to the measurement model is due to the convergence error between actual attitude and reference attitude. The measurement noise covariance is set to  $[R] = 10^{-10}[I] \text{ Nm}^2$ . To avoid presenting multiple, substantially identical simulation results, this subsection only presents the results relative to the CM estimation. Figure 5.8a shows that the estimation error on  $\mathbf{r}_{C/B}$  drops to zero after two updates:

this is consistent with the analysis in Section 4.3, in which it is stated that two linearly-independent thruster torque observations are sufficient to fully resolve the 3D location of the CM. The estimated CM location is accurate to the tenth of a millimeter.

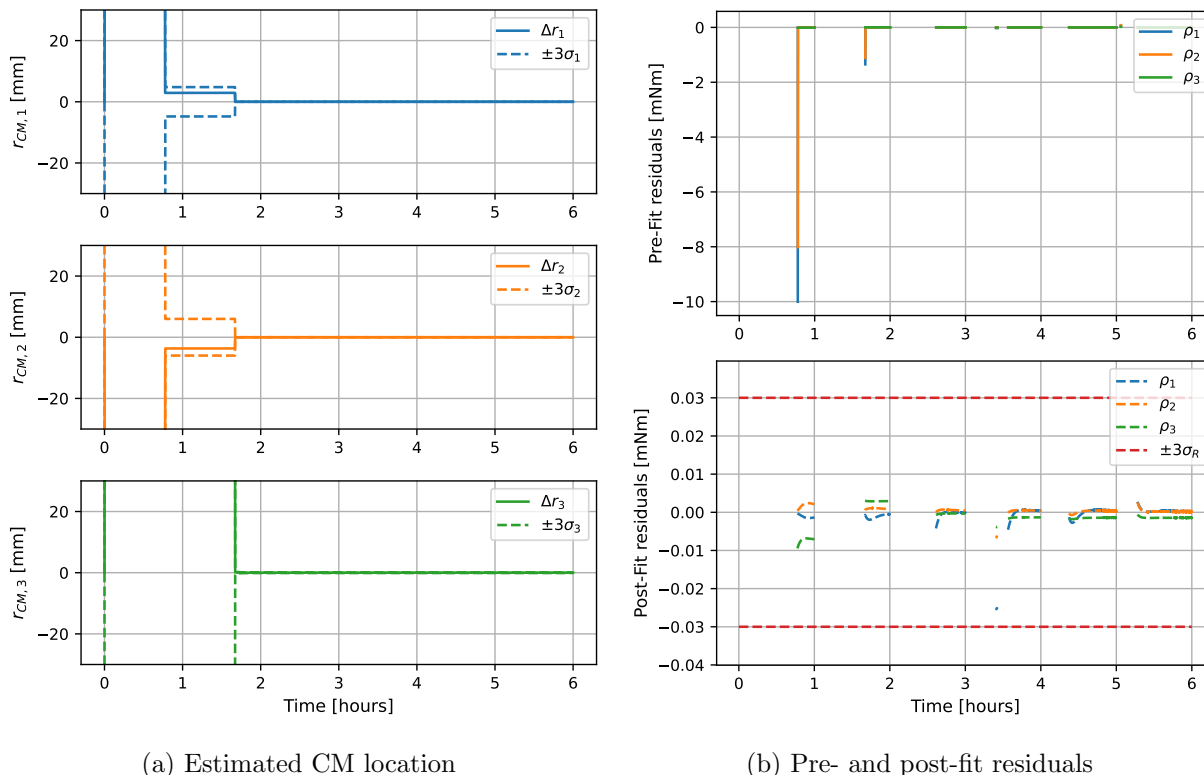
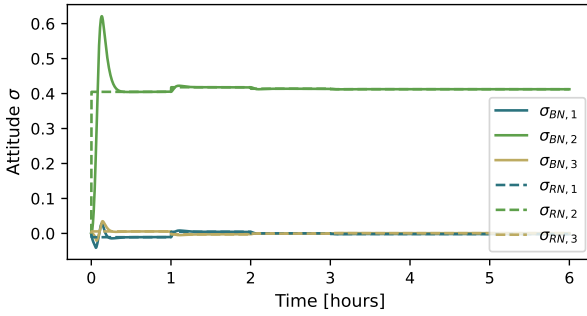


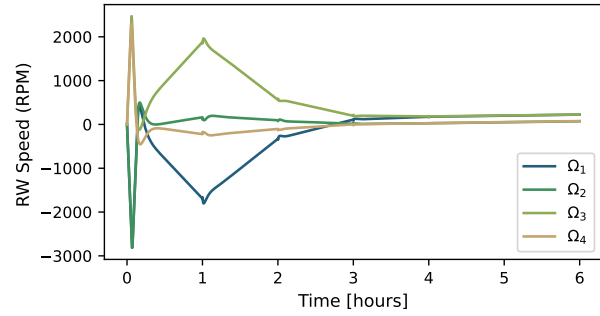
Figure 5.8: CM estimation without external perturbations

### 5.3.3 CM Estimation with SRP

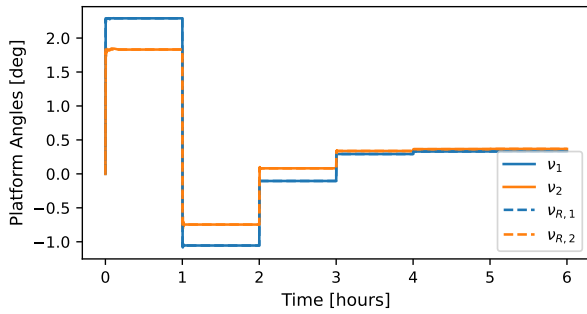
This subsection presents the same conditions as the previous one, except in this case SRP is added back to the simulation. The results are shown in Figures 5.9 and 5.10. In Figure 5.9 it is possible to observe that as a result of using the `cmEstimator` module, the system converges to the same steady-state response shown in Figure 5.7, but faster. The same platform angles are achieved, the same steady-state offset between thruster and CM in order to achieve the same balance between thruster torque and external disturbances, in this case SRP. However, quick convergence on an effective center of mass allows to reduce transient behavior. This is particularly evident in



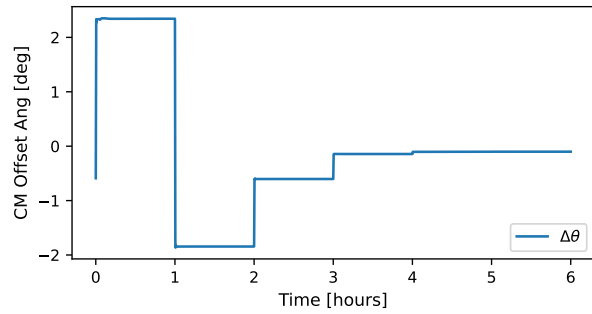
(a) Attitude vs reference attitude



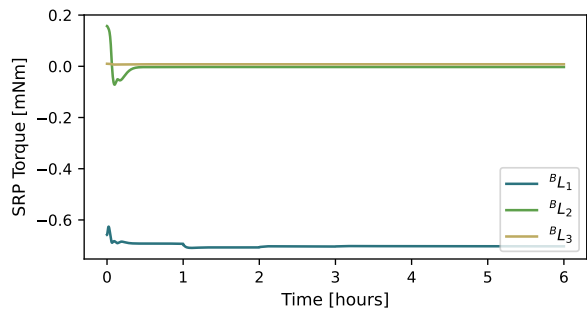
(b) Reaction wheel speeds



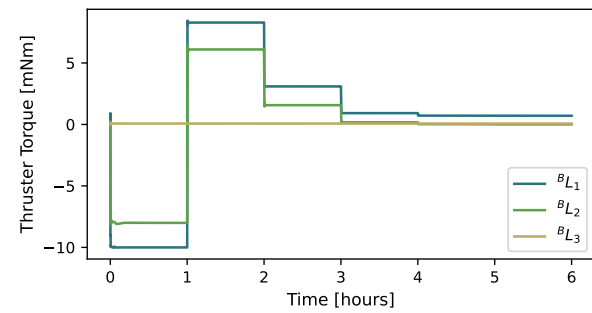
(c) Platform angles vs reference



(d) Thruster-CM offset



(e) SRP torque



(f) Thruster torque

Figure 5.9: Momentum management with CM uncertainty, CM estimation, and SRP



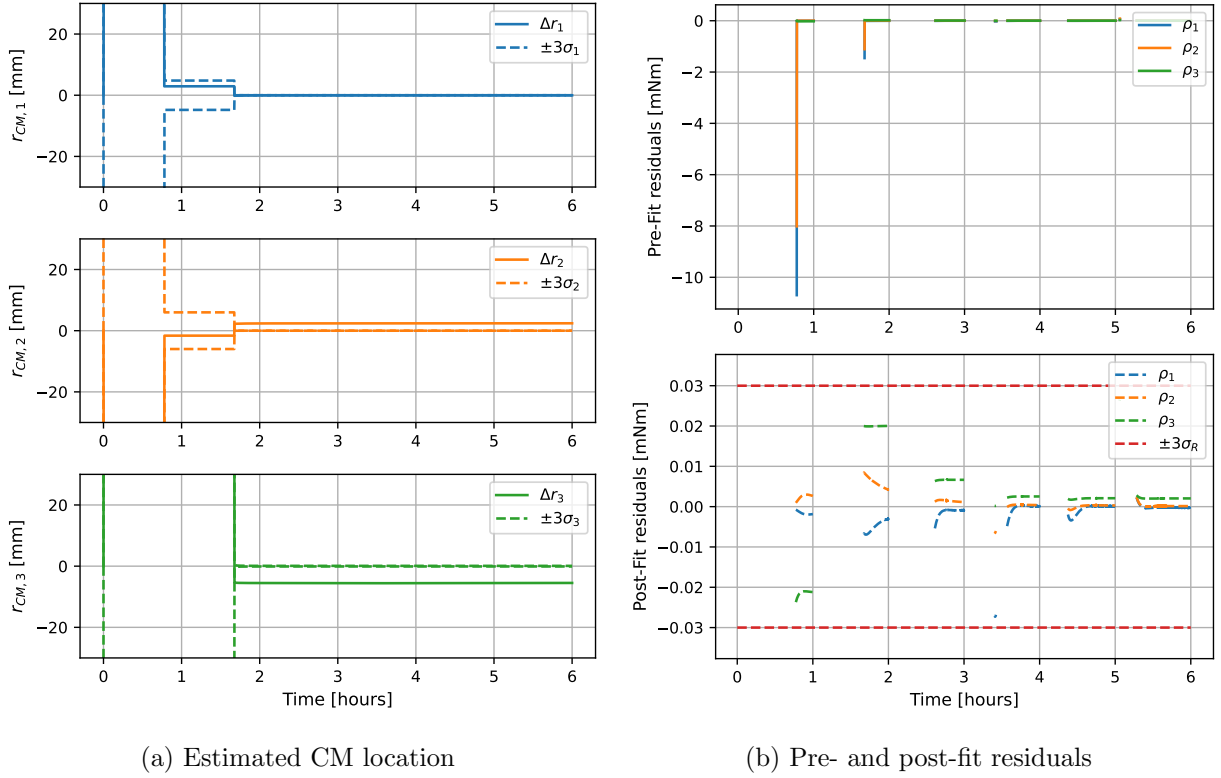


Figure 5.10: CM estimation with SRP

Figure 5.9b in which the RWs are observed to drop to almost zero in just three hours. This happens because the `cmEstimator` is decoupled from the momentum management control law, which in this instance is simply proportional as per Equation (4.4). Consequently, CM estimation is fast, and momentum management is no longer underdamped. However, Figure 5.10a shows that the CM estimate is no longer accurate, but rather, precision is lost along the  $\hat{\mathbf{b}}_2$  and  $\hat{\mathbf{b}}_3$  axes. With respect to Equation (4.51), the loss of precision along  $\hat{\mathbf{b}}_3$  is primarily due to the lack of observability at steady state, with the thruster  $\mathbf{t}$  mostly aligned with the  $\hat{\mathbf{b}}_3$  axis. The bias along the negative  $\hat{\mathbf{b}}_2$  axis, on the other hand, is due to the external SRP torque. With respect to Figure 5.9e, the main SRP torque component is along the negative  $\hat{\mathbf{b}}_1$  axis. With  $\hat{\mathbf{t}} \approx \hat{\mathbf{b}}_3$ , according to Equation (4.51), the resulting bias along the negative  $\hat{\mathbf{b}}_2$  direction is consistent with Equation (4.51).

### 5.3.4 CM Estimation with SRP and Off-nominal Thruster Firing

The last scenario aims to show the performance of the spacecraft when a thruster misfire is added to the scenario presented in the previous subsection. In this scenario, the thruster is fired at 90% of its maximum output. The thrust direction is set to  $\hat{t} = \{0.0099975, -0.019995, 0.9997501\}^T$ ,

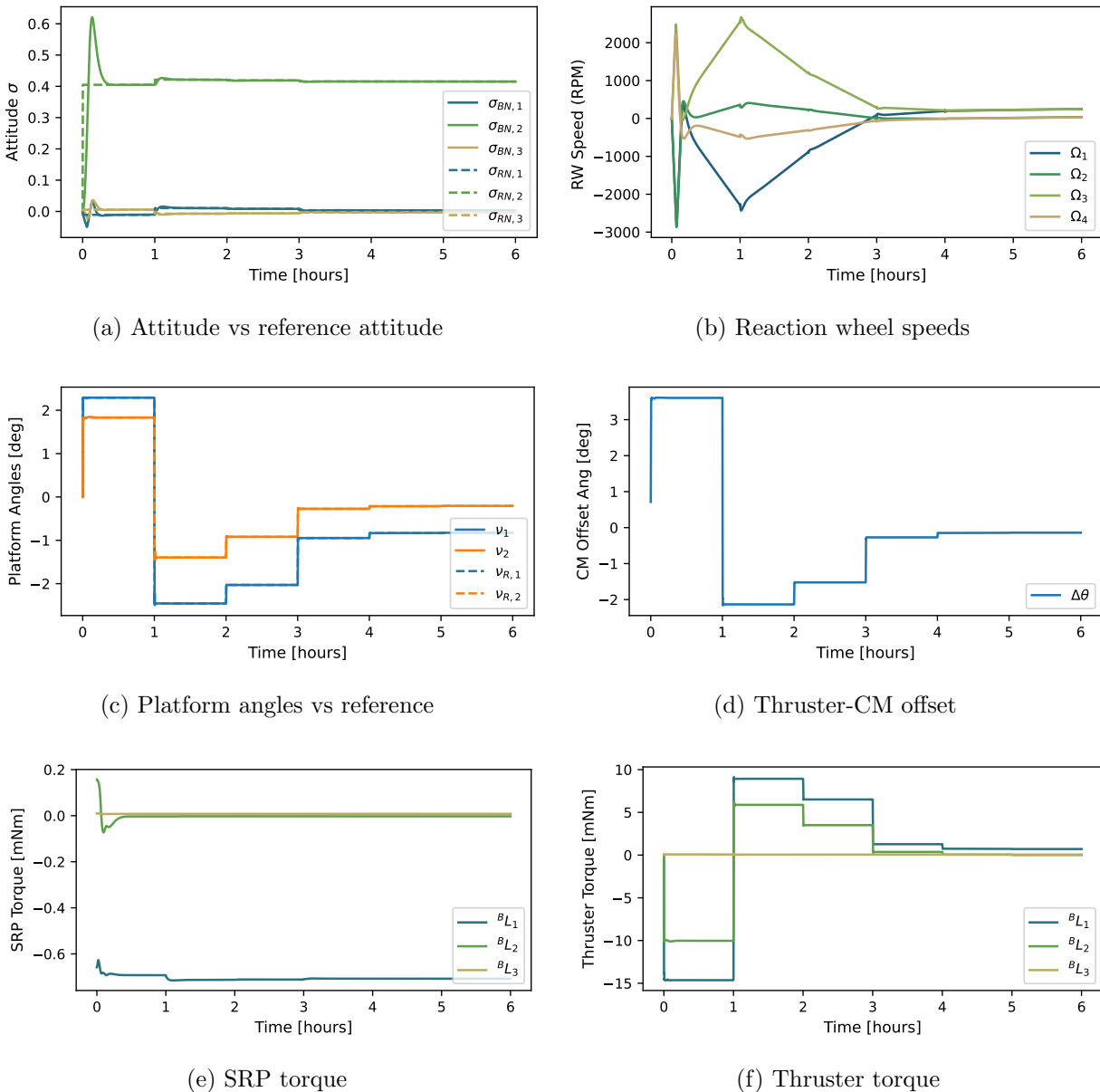


Figure 5.11: Momentum management with CM uncertainty, CM estimation, SRP, and thruster misfire

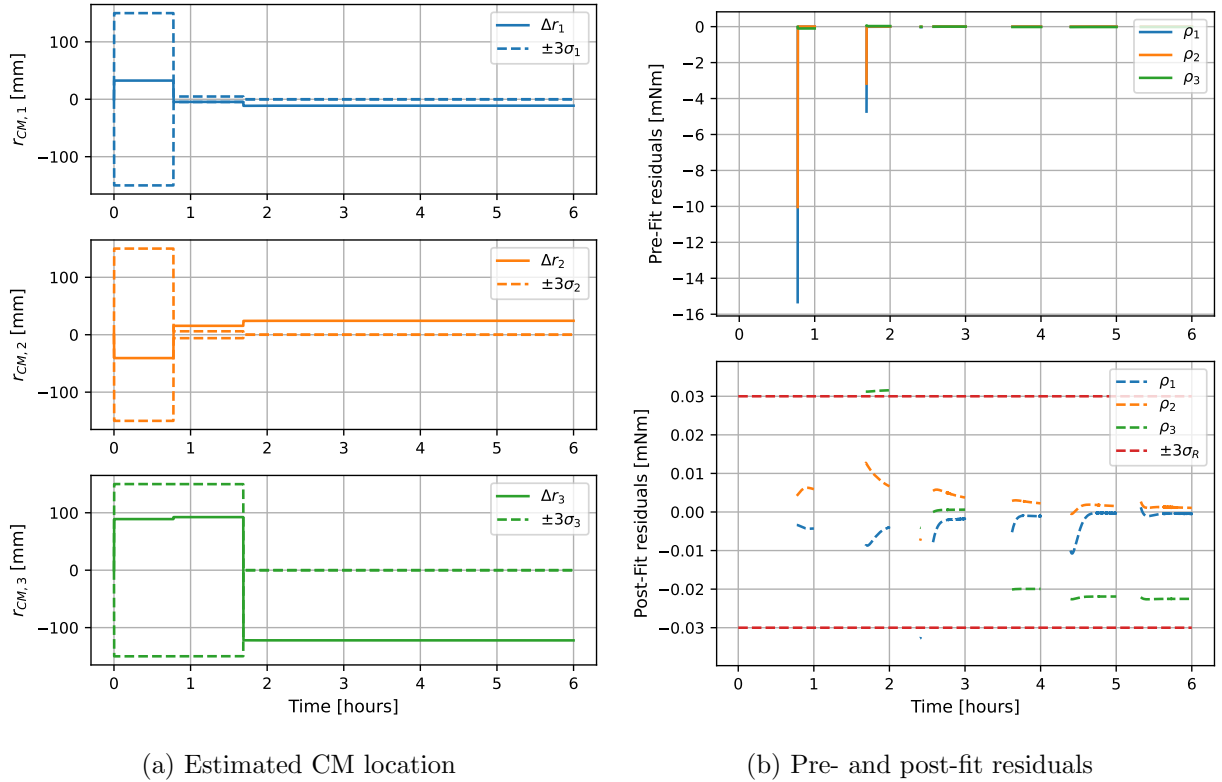


Figure 5.12: CM estimation with SRP and thruster misfire

which roughly corresponds to a misalignment of 1.3 degrees from nominal behavior. These deviations from nominal behavior are injected at simulation dynamics level, but the flight software is not informed of these changes, in order to test the GNC robustness against unmodeled errors. The results are presented in Figures 5.11 and 5.12. Figure 5.11 shows a similar performance to Figure 5.9, with only small differences in the transient responses. Figure 5.12, however, shows significant errors in the CM estimate, which in this case is off by over 10 cm. This bad estimate is due to rather large errors in the model, which result in a misalignment between thruster and CM. Consequently, the thruster misalignment acts as an unmodeled external torque, which makes the CM estimate even worse. A thruster misfire introduces a deviation from linearity assumed in Equation (4.48). However, as long as the thruster misfire is not too far from nominal, the estimator is still able to drive the system to a stable configuration despite the CM estimate being off.

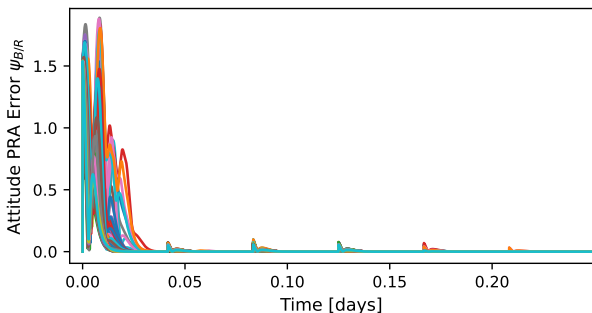
## 5.4 Monte Carlo Results

This final section is provided to show the robustness of the simulation to dispersed initial conditions and with variations in mass and inertia properties not modeled in the flight software. The following initial conditions are dispersed in 500 Monte Carlo runs, with respect to the nominal values already presented at the beginning of this chapter:

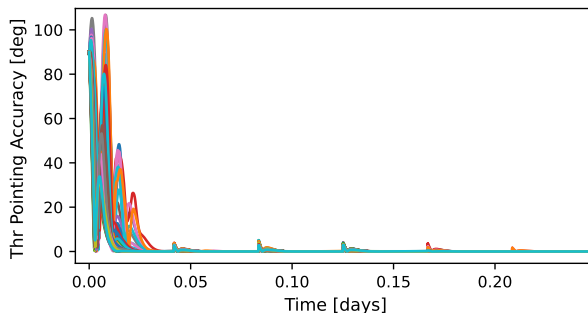
variable	distribution	bounds / std
$m_{\text{hub}}$	uniform	[2375, 2625] kg
$r_{C/B,\text{hub}}$	normal	5 cm - $3\sigma$
$[I_{\text{hub}}]$	normal	5% - $3\sigma$
${}^{\mathcal{B}}\hat{\mathbf{u}}_i$	normal	2 deg - $3\sigma$
$\Omega_i(0)$	uniform	[-1000, 1000] rpm
$I_{w_s,i}$	uniform	[0.1512, 0.1671] kg m <sup>2</sup>
$m_{\text{sa},i}$	uniform	[80.75, 89.25] kg m <sup>2</sup>
$[I_{\text{sa},i}]$	normal	5% - $3\sigma$
$\ \mathbf{t}\ $	uniform	[0.513, 0.567] N
$\hat{\mathbf{t}}$	normal	0.5 deg - $3\sigma$
$\mathbf{r}_{C/B}(0)$	uniform	5 cm - $3\sigma$

Table 5.1: Table of dispersed properties and initial conditions for MC runs

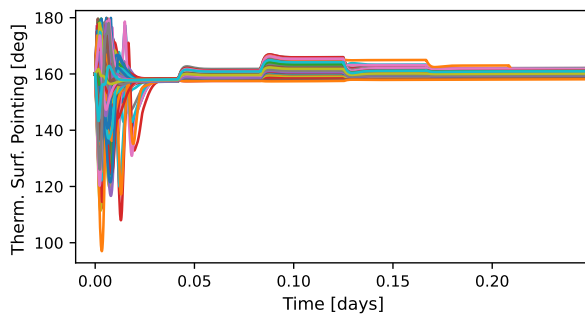
Figure 5.13 summarizes the results of the Monte Carlo runs. Figure 5.13a shows the attitude in terms of PRA with respect to the optimal reference frame. Figure 5.13b shows the angle between the requested and nominal thrust direction vector. Note that this figure does not express thruster misfires: because the angle is computed with respect to the nominal thruster direction, all runs show that this directions converges to the requirement at steady state. Figure 5.13c shows the angle between a thermally-sensitive platform located on the spacecraft's  $-\hat{\mathbf{b}}_2$  side and the Sun: in all runs, this angle is higher than 90 degrees, signaling that this surface is kept in the shade. Figures 5.13a to 5.13c are shown for 6 hours of simulation time to highlight the different transient behaviors. The steady-state behavior is not interesting, as it does not change. Figures 5.13d and 5.13e, on the



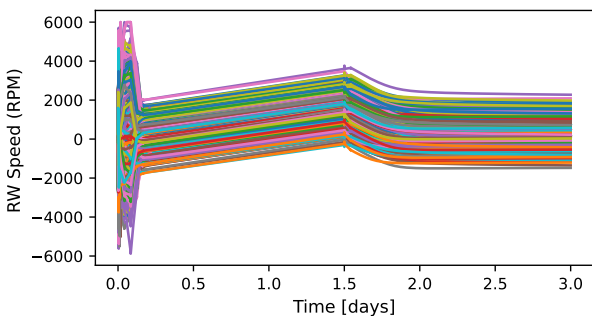
(a) Attitude PRA error



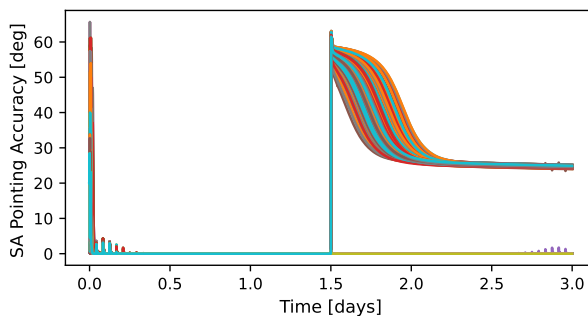
(b) Thruster Pointing Accuracy



(c) Thermal-sensitive platform Pointing



(d) Reaction Wheel Speeds



(e) Solar Array Pointing

Figure 5.13: Monte Carlo Runs

other hand, are shows for the full length of the simulation of 3 days. At 1.5 days of simulation time, the arrays are switched from pointing generation mode to momentum management mode. This is shown in Figure 5.13e because one of the arrays changes its orientation with respect to the Sun, and in Figure 5.13d where the reaction wheel momentum starts being offloaded. This is due to a change

in the total net momentum acting on the system: initially this is exclusively due to the thruster swirl torque, but at  $t = 1.5$  days the net momentum is reduced due to the momentum-dumping action of one of the arrays.

## Discrete Constrained Attitude Maneuvering

This chapter starts to dive into the constrained attitude maneuvering part of this dissertation. The problem analyzed here consists in driving a rigid spacecraft from an initial attitude to a final attitude while imposing pointing constraints that need to be respected at all times. This work considers pointing constraints that involve boresight vectors mounted on the spacecraft hub and therefore rotate with the spacecraft. Pointing constraints in this and in the following sections come in the form of keep-out and keep-in constraints: the former usually arising in the presence of sensitive instruments that must be kept away from bright objects; the latter usually related to solar array or antenna related pointing requirements.

As discussed in the introduction, solutions found in literature can vary in approach, ease of implementation, and computational complexity. The reader is referred to Chapter 1 for several references to different, existing solutions to this problem. The approach chosen in this work is mostly analogous to a Probabilistic RoadMap [42]. In the solution presented here, the attitude space is sampled within the MRP unit sphere centered around the origin of the inertial frame  $\mathcal{N}$ . The choice is motivated mainly by two factors. Lyapunov-based solutions are avoided based on the nature of the pointing constraints being analyzed: multiple, often overlapping constraints can easily give rise to local minima in the domain, which would cause the spacecraft to get stuck in the wrong configuration. Secondly, the attitude space and the constraints are usually well-known ahead of maneuver execution because they are an immediate consequence of sensor measurements. Within the context of a PRM, this results in the need to sample a very well-known and well-characterized

configuration space. Moreover, because the location of the constraints is known beforehand, the “probabilistic” aspect of the PRM is removed in this work, and the workspace sampling is instead done deterministically and with a level of granularity that suits the implementation, depending on the number of pointing constraints and computational availability onboard.

## 6.1 Workspace discretization

The workspace for the present analysis can be reduced, without loss of generality, to the points contained in a unit sphere centered in the origin. Such workspace is discretized using a 3D cartesian grid consisting of equally spaced nodes. The discretization level  $N$  of the grid is defined as the number of equally spaced nodes contained along each principal cartesian semiaxis. A visual representation of the cartesian grid, with a discretization level  $N = 6$ , is reported in Figure 6.1: such representation is two dimensional, for ease of visualization, and describes an obstacle-free workspace. As shown in Figure 6.1, non-equally-spaced nodes are added on the surface of the unit sphere, in substitution of the neighboring nodes that exceed the limit distance  $\sigma = 1$  from the

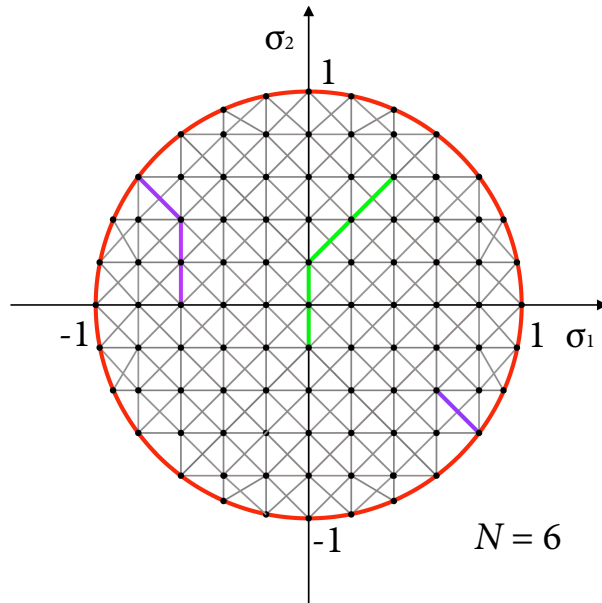


Figure 6.1: MRP grid with two minimum-distance paths: i) green: no MRP switching; ii) purple: MRP switching



origin, and are therefore excluded. The gray lines connect each node with its immediate neighbors: the definition of neighboring nodes is vital for the pathfinding algorithm, as it defines the directions in which it is allowed to explore the workspace from the current location. In absence of obstacles, all internal nodes (i.e. far enough from the surface of the sphere) have 26 neighboring nodes. A scheme of an internal, 26-connected node is reported in Figure 6.2: in purple is the internal node, in red are the neighboring nodes located at a distance of  $\frac{1}{(N-1)}$ , in orange the neighbors at a distance of  $\frac{\sqrt{2}}{(N-1)}$  and in yellow the neighbors at a distance of  $\frac{\sqrt{3}}{(N-1)}$ . Additionally, each node on the  $\sigma = 1$  boundary surface is linked to its own shadow set node, which is the node on the opposite side of the sphere with respect to the origin. This allows the pathfinding algorithm to automatically switch to the shadow set when necessary, avoiding principal rotations from the zero reference larger than 180 deg and, therefore, avoiding the singularity in the MRP formulation. This discretization is uniform in MRP space, but it is not uniform when the MRP sets are mapped to the respective principal rotation sets. Equation (2.19) shows the nonlinear relation between MRP and principal rotation set, for which any two pairs of nodes separated by the same distance in MRP space, are not necessarily characterized by the same PRA  $\Phi$ . However, since the workspace is bounded to the unit sphere, the PRA is also bounded in the domain  $-\pi \leq \Phi \leq \pi$ , for which

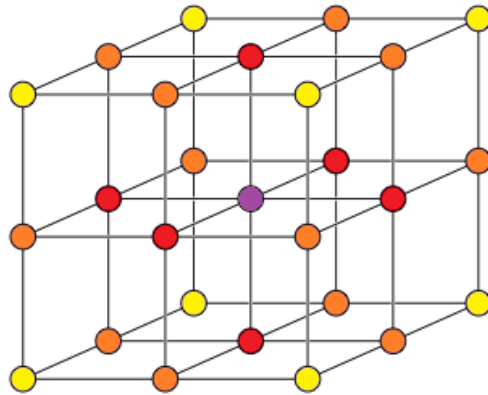


Figure 6.2: Neighboring nodes

Equation (2.19) is fairly linear. The nonlinearity can be further reduced by using higher-order Rodrigues parameters [77], or higher-order map projections for attitude representation [78, 79], at the expenses of having to perform multiple set switching to avoid the singularities introduced by more complex formulations. Reference 78 shows that MRPs have better linearity properties than other popular attitude representation sets. Strides have been made to sample the special orthogonal group  $SO(3)$  of rigid body rotations with grid structures that have a high degree of uniformity and coverage of the space [80]. The following work accepts the small deviation from linearity introduced by MRPs. Such nonlinearity introduced by Equations (2.19) and (2.21) is handled later in this paper, where a constant angular rate magnitude  $\|\boldsymbol{\omega}\|$  is imposed. This allows to track the reference trajectory with a constant rate of variation of attitude over time.

## 6.2 Obstacle representation

To perform the path-planning query it is essential to map the constraints into the workspace. The constraints are known in terms of the direction of the celestial object  ${}^{\mathcal{N}}\hat{\mathbf{s}}_i$ , the body-fixed direction of the instrument  ${}^{\mathcal{B}}\hat{\mathbf{b}}_i$ , and the minimum/maximum angle between the two  $\beta_i$ , according to:

$${}^{\mathcal{B}}\hat{\mathbf{b}}_i \cdot [{}^{\mathcal{B}\mathcal{N}}]{}^{\mathcal{N}}\hat{\mathbf{s}}_i \begin{matrix} \leq \\ \geq \end{matrix} \cos \beta_i \quad (6.1)$$

where the  $<$  sign is used for keep-out constraints, whereas the  $\geq$  for keep-in constraints. Each of these constraints, in their simplest form, can be visualized as a conical region in the unit sphere. For keep-out constraints, the boresight vector of the sensitive instrument must avoid such conical regions as the spacecraft executes the maneuver. On the contrary, for keep-in constraints, the boresight should remain within the conical region during maneuvering.

For every node in the grid, compliance is verified with respect to all the constraints. The nodes that do not satisfy Equation (6.1) are marked as not collision-free and removed from the grid. As a consequence, the path-finding algorithm is blocked from exploring nodes that are not constraint compliant. It should be noted that with this approach, the segment that connects two

constraint-compliant nodes that are very close to an obstacle may itself not be constraint-compliant in all its parts. However, this paper only aims at analyzing simple, convex geometrical constraints, for which this phenomenon can be avoided choosing a discretization level  $N$  that is sufficiently high.

It is interesting to show how such convex, geometric constraints are mapped into obstacles in MRP space. Figure 6.3a shows the  $\mathcal{B}$  frame aligned with the inertial frame  $\mathcal{N}$ , i.e., with the principal axes aligned with the inertial directions  $x$ ,  $y$ , and  $z$ . The  $x$  body axis is shown in red, as it coincides with the boresight of the sensitive instrument with a field of view  $\beta = 20$  deg. The directions of the three keep-out cones are contained in the  $(y, z)$  plane, at an angle of 120 deg from one another. Figure 6.3b shows the keep out cones mapped in MRP space, where the dot in the origin represents the current attitude  $\sigma = \{0, 0, 0\}^T$ , which coincides with the inertial frame. Since the attitude is constraint-compliant, with the boresight vector outside of all the keep-out cones, so is the dot representing the initial configuration collision-free. It can be observed, for example, that the red keep-out cone blocks negative rotations around the body axis  ${}^{\mathcal{B}}\hat{\mathbf{b}}_y = \{0, 1, 0\}^T$ : in MRP space, this causes the appearance of an obstacle along the negative  $\sigma_2$  direction. Similar considerations

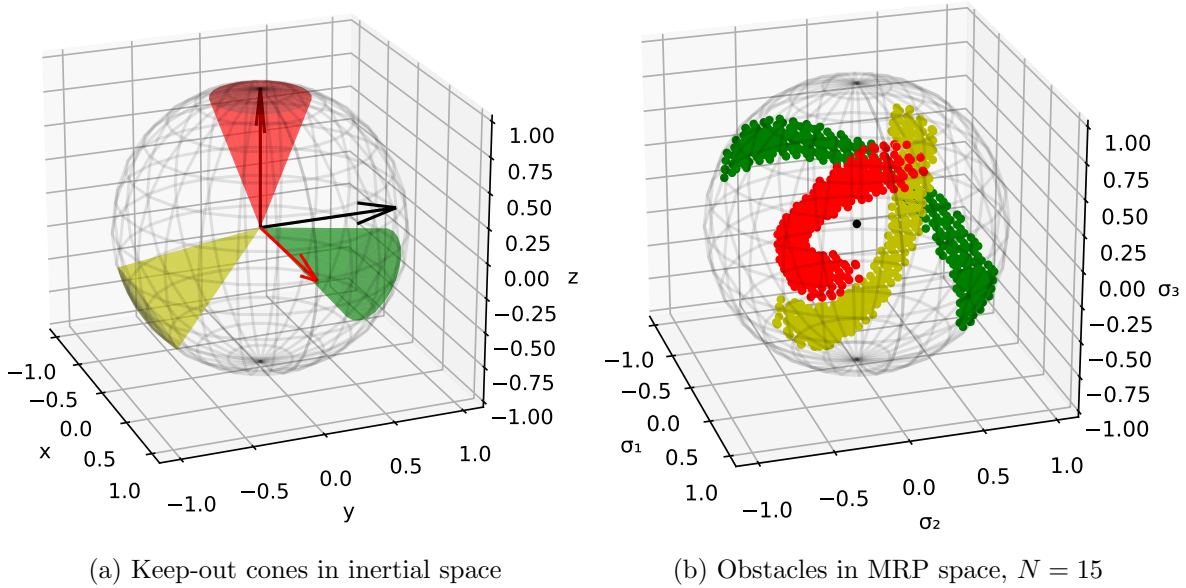


Figure 6.3: Three general keep-out constraints,  $\beta = 20$  deg

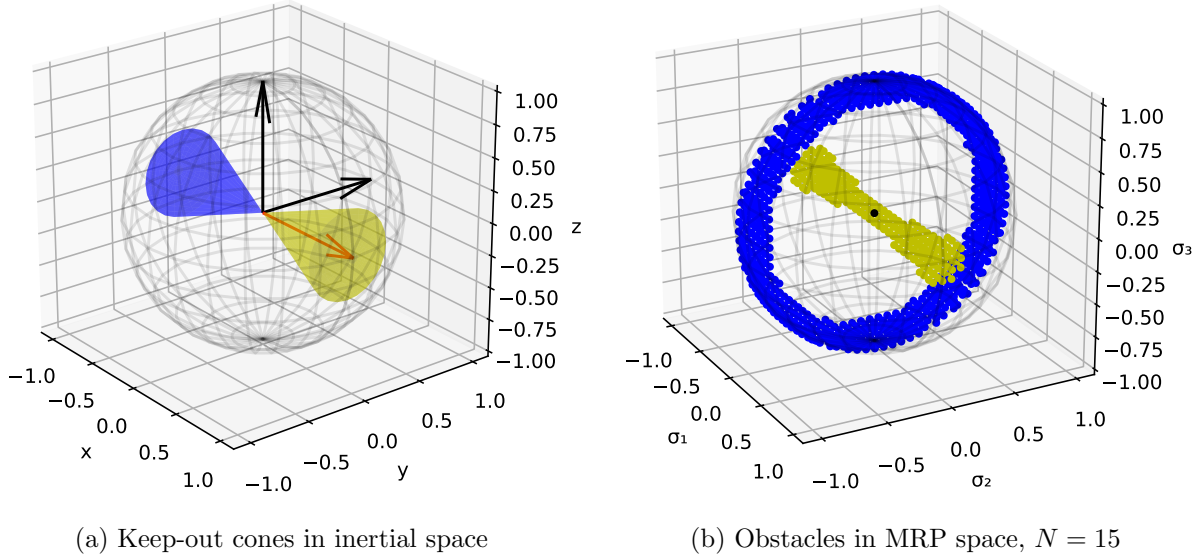


Figure 6.4: Two constraints, 0 deg and 180 deg apart from boresight,  $\beta = 20$  deg

can be made for the other two keep-out cones. General keep-out cones almost always generate tubular structures as those in Figure 6.3b: such structures are thicker or thinner according to the half-angle of the respective cone, and always go from one side of the unit sphere to the opposite, because opposite ends of the unit surface represent the same attitudes.

Some qualitatively different obstacles are reported in Figure 6.4, where the keep-out cones are 0 deg (yellow) and 180 deg (blue) apart from the boresight direction. It can be observed, first of all, that the boresight vector is inside the yellow cone: as a consequence, the black dot representing the initial attitude in MRP space falls inside the yellow obstacle in Figure 6.4b. Since any rotation about  ${}^B\hat{\mathbf{b}}_x = \{1, 0, 0\}^T$  maintains the boresight within the cone, the corresponding obstacle in MRP space encompasses the whole  $\sigma_1$  cartesian axis. As for the blue keep-out cone, any rotation of  $\Phi \geq 160$  deg about any body axis perpendicular to  ${}^B\hat{\mathbf{b}}_x$  causes a constraint violation, for this reason the blue obstacle in MRP space takes the shape of a ring distributed around the intersection between the unit sphere and the  $(\sigma_2, \sigma_3)$  plane.

The results in Figures 6.3 and 6.4 can be generalized to a problem with multiple sensitive instruments and/or sensors characterized by a keep-in constraint. For multiple sensitive instru-

ments, there are multiple boresight vectors: this causes each keep-out cone to generate an obstacle in MRP space for every sensitive instrument. Such obstacles can have different orientations and sizes, especially if the different instruments have different fields of view, and may overlap in points that represent attitudes for which multiple instruments are facing a keep-out direction. For keep-in constraints, the only difference is the sign in Equation (6.1). For this reason, the result are analogous to those of Figure 6.3b and Figure 6.4b, only in this case the colored regions represent constraint-compliant zones, whereas empty regions are not constraint-compliant. Obviously, in the presence of combined keep-out and keep-in constraints, the compliant regions are those where both the constraints are satisfied simultaneously.

## 6.3 Graph search: the A\* algorithm

### 6.3.1 Algorithm Review

The primary purpose of this chapter is to maneuver a spacecraft from an initial orientation to a final, user-specified orientation. Given the grid discretization and the obstacle representation in MRP space described in the previous section, the maneuvering problem is now transformed into a path-finding query. The aim is to find a path in MRP space that connects the starting point, representing the initial attitude, with the final point, representing the target attitude, avoiding the obstacles that represent the geometric constraints. Such path must be searched in the directed graph that has the grid nodes as graph nodes, and the segments connecting each nodes to its neighbors as graph edges. While this paper uses the A\* method to develop a set of discrete attitude path points, the paper’s methodology of using discrete MRP waypoints to develop a smooth, effort-optimized solution is not tied to the A\* method generating these points.

First of all, start and goal nodes  $n_S$  and  $n_G$  must be added to the graph. The graph is searched for the two nodes whose cartesian distances from  $n_S$  and  $n_G$ , respectively, are the lowest. To avoid redundancies, such two nodes are replaced by  $n_S$  and  $n_G$ , which are immediately connected to the neighbors of the nodes they replaced.

Different algorithms exist that can construct a tree (acyclic graph) rooted in  $n_S$ , which explores the neighboring nodes until reaching the leaf node  $n_G$ . There are two broad categories of such algorithms: breadth-first and depth-first. Breadth-first algorithms proceed exploring, at every iteration, all the nodes located at the same link length from the root [40]. Depth-first algorithms, on the contrary, start from the root node and explore nodes at increasing link lengths at every iteration until a leaf node is found [39]. Breadth-first algorithms have the advantage of exploring larger regions of the graph while running the query, therefore they can find the best path to goal in terms of the smallest link length. Depth-first algorithms, on the contrary, explore much less of the graph, but can find a suboptimal path to goal in less computational time.

Another class of algorithms exists, called greedy algorithms, which at every iteration explore the graph in the direction that appears to be the closest to the goal: an algorithm that belongs to this class is A\*. The explored node is chosen as the node having the smallest total cost  $f(n) = g(n) + h(n)$ , where  $g(n)$  is the cost of the path from  $n_S$  to  $n$  and  $h(n)$  is a heuristic estimate of the distance between  $n$  and  $n_G$  [41]. A\* is guaranteed to find the optimal solution in terms of the metric used to define the cost of a path. Moreover, A\* is also efficient, in the sense that its priority-driven nature allows to solve the query exploring the minimum number of graph nodes. For A\* to be successful and efficient, the chosen heuristic  $h(n)$  should be less than or equal to the actual cost of the path from  $n$  to  $n_G$  [81]. Two lists are used by the algorithm as it explores the graph: a closed list  $C$  containing the nodes that have already been processed, and an open list  $O$  containing the nodes that are yet to be processed. The algorithm receives the graph as input, and the open list initially contains only  $n_S$ . The pseudo-algorithm for A\* is provided in Algorithm 1.

Two things should be noted about A\* implementation: i) the algorithm does not stop when  $n_G$  is encountered, but only when  $n_G$  is encountered as the node with the smallest total cost; ii) when a node is encountered that is already in the open list with a higher total cost, it means that the algorithm found a more cost-efficient path to that node, therefore that node and its cost should be updated.

For a more comprehensive discussion about A\* and other graph search algorithms, the reader is redirected to References 81, 82.

---

**Algorithm 1** Path = A\*( $n_S, n_G, \text{Graph}$ )

---

```

Pick  $n_{\text{best}}$  such that  $f(n_{\text{best}}) \leq f(n) \forall n$  in  $O$ 
while  $n_{\text{best}} \neq n_G$  do
  Move  $n_{\text{best}}$  from  $O$  to  $C$ 
  for  $m$  in  $\text{Neighbors}(n)$  do
    if  $m \notin O$  then
      Add  $m$  to  $O$ 
    else
      if  $g(n_{\text{best}}) + \text{cost}(n_{\text{best}}, m) < g(m)$  then
        Change backpointer of  $m$  to point to  $n_{\text{best}}$ 

  Pick  $n_{\text{best}}$  such that  $f(n_{\text{best}}) \leq f(n) \forall n$  in  $O$ 
Backtrack Path from  $n_G$  to  $n_S$ 
return Path

```

---

### 6.3.2 Nonsingular MRP-based A\* implementation

A first, simple implementation of the pathfinding algorithm can be based on A\*, as described in the previous section. In this implementation, the cost of paths, edges and heuristics are based on the distance between nodes in MRP space. Specifically, defining  $d(\sigma_1, \sigma_2)$  the cartesian distance between nodes, the distance metric implemented in the algorithm is the following:

$$\hat{d}(\sigma_1, \sigma_2) = \min \{d(\sigma_1, \sigma_2), d(\sigma_1^S, \sigma_2), d(\sigma_1, \sigma_2^S), d(\sigma_1^S, \sigma_2^S)\}. \quad (6.2)$$

This metric allows to explore the entire discrete domain while avoiding the singularity introduced by MRPs, by automatically performing MRP switching. According to this metric, nodes on the  $\sigma = 1$  boundary have a distance  $\hat{d} = 0$  from their respective shadow set, therefore allowing to recognize a path that exits the unit sphere and reenters it from the opposite side as, potentially, the minimum-cost path. In Figure 6.1, the green and purple paths are both minimum-distance paths, although the first one is entirely contained in the unit sphere, whereas the latter one is obtained via MRP switching. The cost of a path  $p = [\sigma_S, \sigma_1, \dots, \sigma_n]$  is defined, in the following

implementation, as:

$$g(p) = \sum_{i=0}^{n-1} \hat{d}(\sigma_i, \sigma_{i+1}) \quad (6.3)$$

and the heuristic:

$$h(\sigma_i) = \hat{d}(\sigma_i, \sigma_G). \quad (6.4)$$

The following test scenario features a spacecraft with a sensitive instrument mounted along the  $x$  body axis  ${}^{\mathcal{B}}\hat{\mathbf{b}}_x = \{1, 0, 0\}^T$ , which has a field of view of 20 deg. Three bright objects are present, whose inertial directions are  ${}^{\mathcal{N}}\hat{\mathbf{s}}_1 = \{0, -0.981, -0.196\}^T$ ,  ${}^{\mathcal{N}}\hat{\mathbf{s}}_2 = \{-1, -0, 0\}^T$  and  ${}^{\mathcal{N}}\hat{\mathbf{s}}_3 = \{0.958, 0, 0.287\}^T$ . Initial and final attitude, with respect to the inertial frame  $\mathcal{N}$ , are  $\sigma_S = \{0, 0, 0.25\}^T$  and  $\sigma_G = \{0, 0, -0.75\}^T$ . Figure 6.5 shows the computed path in inertial space and in MRP space. For ease of visualization, the inertial space is projected onto a 2D plane in terms of longitude and latitude with respect to the inertial frame  $\mathcal{N}$ . The computed path is constraint compliant. However, it is only provided in terms of a sequence of waypoints, or attitudes, but it does not give any information on how to navigate between such waypoints. Angular rates and accelerations during the maneuver are unknown. One idea could be to treat every segment in the

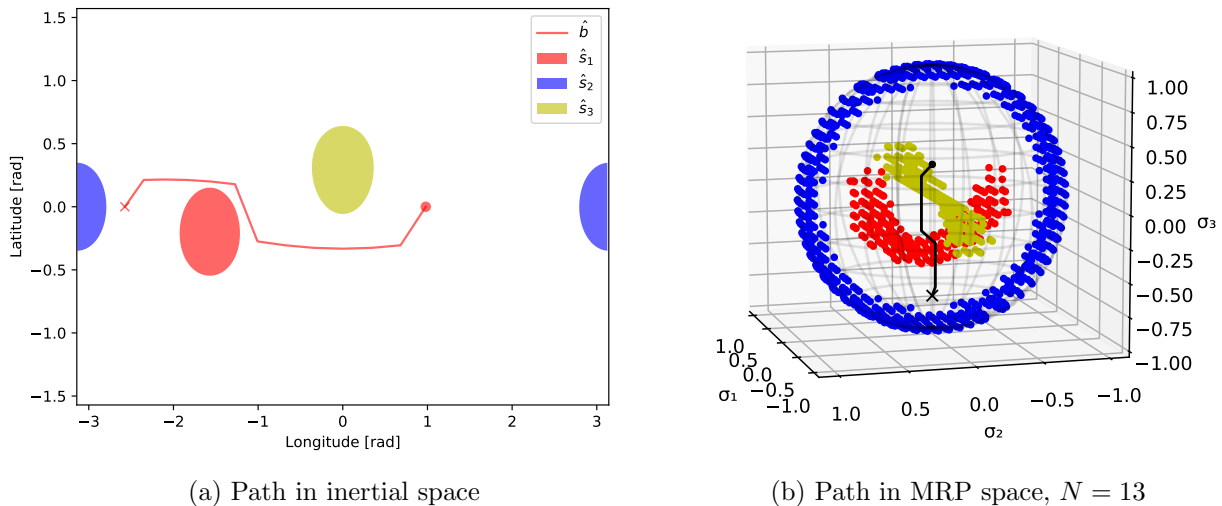


Figure 6.5: Metric-based A\* solution to the path planning query, ‘o’ is starting point and ‘x’ is target point



path as a rest-to-rest rigid body rotation. This approach would ensure constraint compliance, but would likely require a long time to repeatedly accelerate and slow down the spacecraft. Moreover, the computed path is only optimal with respect to the metric used, but it is agnostic to the dynamic properties of the spacecraft: the same path might, in fact, not be equally desirable for spacecraft with different inertia tensors, or different initial angular rates and/or target angular rates. The two-point boundary value problem could be solved to provide guidance between the intermediate points, provided that the angular rates are specified at each waypoint. However, this solution would present discontinuities in the required torque between one segment and the following one, and for a reaction wheel-controlled spacecraft this is not ideal, as an infinite acceleration would technically be required for an immediate change in the wheel angular speed. Ideally, the path should be transformed into a trajectory that interpolates between the waypoints: such trajectory should be differentiable and sufficiently smooth, and should match the required kinematic constraints (angular rates) at both the starting attitude and the target attitude. Such trajectory would allow to provide not just attitude as a function of time, but also rates and accelerations, from which, knowing the inertia properties of the spacecraft, the required torque can be computed.

## MRP Path Smoothing via NURBS Curves

Chapter 6 describes the attitude configuration space, the constraint representation, the graph construction, singularity avoidance using MRPs, and the graph-search technique. However, the solution found is not yet complete, because it simply consists of a series of constraint-compliant attitude waypoints, with no indication of the kinematic properties of the spacecraft as it navigates through these waypoints.

In robot navigation it is quite common to have to deal with paths that present sharp turns, that are not suitable for the robot's motion. A strategy that is often used to smooth such paths consists in waypoint interpolation. For this work, such interpolating function should yield a trajectory that is parametric-continuous, i.e., the support of the curve in MRP space should be continuous, and each component  $\sigma_i$  of the MRP set should also be continuous with respect to the parameter of the curve, namely the time  $t$  [83]. Additionally, it is desirable to have the ability to enforce some kind of upper bound on the maximum rate experienced by the spacecraft [84], while also trying to ensure a fast maneuver execution, and prescribing endpoint angular rates through some kind of two-point boundary value problem.

Moreover, the trajectory should be kinematically feasible, i.e., it should follow Equations (2.21) and (2.23). Failure to comply with these equations results in a violation of the equations of rotational motion, thus yielding poor performance of the closed-loop control in Equation (2.48), potentially causing the motion of the spacecraft to deviate from the reference trajectory and violate the pointing constraints.

This chapter presents a family of curves that are particularly suitable to fitting increasingly large sets of data with parametric functions without diverging, and also offer several knobs to tune in order to approximate the desired kinodynamic properties of the reference trajectory. This chapter presents the general formulation of these curves and then proceeds to highlight two different approaches that yield compliant solutions. These solutions are later benchmarked against one another. Finally, closed-loop performance is analyzed to test robustness.

## 7.1 NURBS Curves Overview

A family of functions that satisfy all the requirements for the interpolation of attitude waypoints is Non-Uniform Rational B-Splines (NURBS). NURBS curves are parametric, piecewise-polynomial functions in the intervals between waypoints. This makes them particularly suitable for this application, as they do not suffer from Runge's phenomenon of oscillation between data points when trying to fit too many data with a single high-order polynomial [85, 86]. NURBS functions have the form:

$$\boldsymbol{\sigma}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i \quad (7.1)$$

where  $u$  is a parameter in the range  $u \in [0, 1]$ ,  $\mathbf{P}_i$  are called control points, and  $N_{i,p}(u)$  are the  $p$ -th degree spline basis functions, which are linearly combined to construe the polynomial arcs between waypoints. The degree  $p$  of such polynomials is defined by the user, and it can be higher depending on the continuity and differentiability that the user wishes to enforce on the curve. The basis functions are defined over the knot vector  $U$  containing the  $m + 1$  scalar knots:

$$U = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{1, \dots, 1}_{p+1}\} \quad (7.2)$$

where the  $p + 1$  knots equal to 0 and 1 at the beginning and end of the knot vector are required to ensure that the curve passes exactly through the control points  $\mathbf{P}_0$  and  $\mathbf{P}_n$ . The intermediate knots correspond to the value of the parameter  $u$ , from now on defined as the dimensionless time, at which two different polynomial segments of the curve are joined. The basis functions are computed

with the De Boor recursive algorithm [87]:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (7.3)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u).$$

The derivative of a NURBS curve is obtained as:

$$\boldsymbol{\sigma}'(u) = \sum_{i=0}^n N'_{i,p}(u) \mathbf{P}_i \quad (7.4)$$

where  $\boldsymbol{\sigma}'(u) = \frac{d\boldsymbol{\sigma}}{du}$  denotes the derivative with respect to the dimensionless time  $u$ . The derivatives of the basis functions  $N_{i,p}(u)$  are computed using the formula [86]:

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (7.5)$$

## 7.2 Time-spacing for MRP-based Kinematic Trajectory

To build a NURBS curve it is necessary to define the time tags  $\bar{u}_k$  for  $k = 0, \dots, q$ , which correspond to the values of the dimensionless time  $u$  for which the curve passes through the  $k$ -th waypoint, with  $q + 1$  being the total number of waypoints. This is strictly true for an interpolating NURBS curve, where  $\boldsymbol{\sigma}(\bar{u}_k) = \boldsymbol{\sigma}_k$ , whereas it is not when the curve waypoint fit is done via a least-squares (LS) approach. In a generic application, Reference 86 suggests to compute the  $\bar{u}_k$ 's proportionally to a valid distance metric between two consecutive waypoints, defined in the space where such waypoints exist. In this work, such metric is defined as:

$$\theta(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2) = 4 \arctan \left[ \frac{(1 - |\boldsymbol{\sigma}_2|^2)\boldsymbol{\sigma}_1 - (1 - |\boldsymbol{\sigma}_1|^2)\boldsymbol{\sigma}_2 + 2\boldsymbol{\sigma}_1 \times \boldsymbol{\sigma}_2}{1 + |\boldsymbol{\sigma}_1|^2 |\boldsymbol{\sigma}_2|^2 + 2\boldsymbol{\sigma}_1 \cdot \boldsymbol{\sigma}_2} \right] \quad (7.6)$$

which consists in the principal rotation angle between the frames defined by  $\boldsymbol{\sigma}_1$  to  $\boldsymbol{\sigma}_2$ , assuming that the shortest rotation is performed [50]. This distance metric is defined in order to account for the nonlinear relation between principal rotation angle and distance between attitude points in MRP space. Alternatively, this nonlinearity could be mitigated using higher-order, more linear Rodrigues parameters [78, 77]. Such representation, however, would present multiple singularities and introduce additional layers of complexities to the problem.

A better sampling of the  $\bar{u}_k$ 's time tags is obtained when the desired properties of the NURBS curve are incorporated into the appropriate formulation. In this implementation, one of the goals is to obtain a reference trajectory with a constant angular rate norm. With respect to the final trajectory, the total angular distance swept by the spacecraft in its rotational motion is given by:

$$S(t) = \int_0^t \|\boldsymbol{\omega}\| d\tau. \quad (7.7)$$

A second-order Taylor series expansion of  $S(t)$  can be performed around the generic time instant  $t_k$ , giving:

$$\begin{cases} S_{k+1} = S_k + \dot{S}_k(t_{k+1} - t_k) + \frac{1}{2}\ddot{S}_k(t_{k+1} - t_k)^2 \\ \dot{S}_{k+1} = \dot{S}_k + \ddot{S}_k(t_{k+1} - t_k) \end{cases}. \quad (7.8)$$

where  $S_k = S(t_k)$ . Eliminating  $\ddot{S}_k$  from Equation (7.8), yields the expression:

$$S_{k+1} - S_k = \frac{\dot{S}_{k+1} + \dot{S}_k}{2}(t_{k+1} - t_k). \quad (7.9)$$

According to the Fundamental Theorem of Calculus, it is  $\dot{S}_k = \|\boldsymbol{\omega}(t_k)\| = \omega_k$ . The difference  $S_{k+1} - S_k$  can be approximated, assuming that the waypoints  $\boldsymbol{\sigma}_k$  and  $\boldsymbol{\sigma}_{k+1}$  are close enough together, with  $S_{k+1} - S_k \approx \theta(\boldsymbol{\sigma}_k, \boldsymbol{\sigma}_{k+1})$ . This gives the algorithm to calculate the time spacing between the waypoints:

$$t_{k+1} = t_k + k \frac{\theta(\boldsymbol{\sigma}_k, \boldsymbol{\sigma}_{k+1})}{\omega_k + \omega_{k+1}} \quad (7.10)$$

with  $k = 2$ . The constant  $k = 2$  is a direct consequence of approximating  $S(t)$  with a second-order polynomial, as implied by Equation (7.8), and consequently, the angular rate norm  $\|\boldsymbol{\omega}\|$  is a piecewise-linear function. This approximation holds quite well for the intervals between internal waypoints, where the angular rate norm nominally does not vary. However, when the NURBS curve is required to quickly ramp up and down from/to zero at the endpoints, a better result is obtained assuming a polynomial of higher order  $p$  for  $S(t)$  in the first and last interval. With zero endpoint angular rates, this gives  $k = \frac{p}{p-1}$ . Using a 4th order NURBS for  $\boldsymbol{\sigma}(t)$  suggests the choice for 4th order polynomial for the endpoint time intervals, which gives  $k = 4/3$ . With zero endpoint rates, and constant central angular rate norm  $\|\boldsymbol{\omega}\| = \omega^*$ , the algorithm to calculate the time spacing

between the  $q + 1$  waypoints is:

$$\begin{aligned} t_0 &= 0 & t_1 &= t_0 + \frac{4}{3} \frac{\theta(\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_1)}{\omega^*} \\ t_k &= t_{k-1} + \frac{\theta(\boldsymbol{\sigma}_k, \boldsymbol{\sigma}_{k+1})}{\omega^*} \text{ for } k = 2, \dots, q-1 & t_q &= t_{q-1} + \frac{4}{3} \frac{\theta(\boldsymbol{\sigma}_{q-1}, \boldsymbol{\sigma}_q)}{\omega^*} \end{aligned} \quad (7.11)$$

The dimensionless time tags  $\bar{u}_k$ 's are obtained normalizing the  $t_k$ 's by  $t_q$ , i.e., the time tag corresponding to the final waypoint:

$$\bar{u}_k = \frac{t_k}{t_q}. \quad (7.12)$$

The linear relation between dimensional time  $t$  and dimensionless time  $u$  allows to define a mapping between the derivatives of  $\boldsymbol{\sigma}$  with respect to one or the other:

$$\frac{d\boldsymbol{\sigma}}{du} = \frac{dt}{du} \cdot \frac{d\boldsymbol{\sigma}}{dt} = t_q \frac{d\boldsymbol{\sigma}}{dt} \quad \frac{d^2\boldsymbol{\sigma}}{du^2} = \frac{d^2t}{du^2} \cdot \frac{d\boldsymbol{\sigma}}{dt} + \left(\frac{dt}{du}\right)^2 \cdot \frac{d^2\boldsymbol{\sigma}}{dt^2} = t_q^2 \frac{d^2\boldsymbol{\sigma}}{dt^2}. \quad (7.13)$$

### 7.3 Control Point Definition for MRP-based Kinematic Trajectory

#### 7.3.1 Two-point Boundary Value Compliance

It is desirable to fit the curve while also imposing endpoint derivatives: this allows to account for the kinematic conditions of the spacecraft, i.e., initial angular speed and desired angular speed upon arrival at target attitude. This problem is solved separately from the problem of fitting the intermediate waypoints, and it is complimentary to both the methodologies that are presented in the following subsections. The approach followed in this subsection only focuses on interpolating the first and last attitude waypoints  $\boldsymbol{\sigma}_0$  and  $\boldsymbol{\sigma}_q$ , since the initial and final attitudes are two known pieces of information that the reference trajectory must match exactly. Moreover, it is also desired to obtain a reference trajectory that matches the required angular rates at the beginning and at the end of the maneuver. This translates into endpoint constraints also on the derivatives  $\dot{\boldsymbol{\sigma}}_0$  and  $\dot{\boldsymbol{\sigma}}_q$ , which are computed from the boundary conditions on the angular rates  $\boldsymbol{\omega}_0$  and  $\boldsymbol{\omega}_q$  according to Equation (2.21).

For a NURBS curve consisting of  $n + 1$  control points  $\mathbf{P}_n$  and  $q + 1$  waypoints  $\boldsymbol{\sigma}_q$ , the first two control points,  $\mathbf{P}_0$  and  $\mathbf{P}_1$ , and the last two,  $\mathbf{P}_{n-1}$  and  $\mathbf{P}_n$ , are determined imposing these

boundary conditions through Equations (7.1) and (7.4). This gives the linear,  $4 \times 4$  system:

$$\begin{bmatrix} N'_{0,p}(0) & N'_{1,p}(0) & 0 & 0 \\ N_{0,p}(0) & 0 & 0 & 0 \\ 0 & 0 & 0 & N_{n,p}(1) \\ 0 & 0 & N'_{n-1,p}(1) & N'_{n,p}(1) \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \mathbf{P}_{n-1} \\ \mathbf{P}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\sigma}'_0 \\ \boldsymbol{\sigma}_0 \\ \boldsymbol{\sigma}_q \\ \boldsymbol{\sigma}'_q \end{bmatrix}. \quad (7.14)$$

Note that it is possible to define also the endpoint accelerations of the NURBS curve, by adding two additional control points. In such case, Equation (7.14) becomes a  $6 \times 6$  linear system. This approach often proves not beneficial, because in order to match the multiple endpoint constraints, the resulting NURBS often presents significant oscillations.

### 7.3.2 NURBS Interpolation

Equation (7.1) is leveraged to impose the exact passage of the NURBS curve through the intermediate  $k$  waypoints, for  $k = 1, \dots, q-1$ . This yields the linear system of  $q-1$  equations:

$$\boldsymbol{\sigma}(\bar{u}_k) = \sum_{i=0}^n N_{i,p}(\bar{u}_k) \mathbf{P}_i \quad \text{for } k = 1, \dots, q-1 \quad (7.15)$$

Combining Equations (7.14) and (7.15) results in the linear system:

$$\begin{bmatrix} N'_{0,p}(0) & N'_{1,p}(0) & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ N_{0,p}(\bar{u}_1) & N_{1,p}(\bar{u}_1) & \dots & N_{n-1,p}(\bar{u}_1) & N_{n,p}(\bar{u}_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ N_{0,p}(\bar{u}_{q-1}) & N_{1,p}(\bar{u}_{q-1}) & \dots & N_{n-1,p}(\bar{u}_{q-1}) & N_{n,p}(\bar{u}_{q-1}) \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & N'_{n-1,p}(1) & N'_{n,p}(1) \end{bmatrix} \begin{bmatrix} \mathbf{P}_0 \\ \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_{n-1} \\ \mathbf{P}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\sigma}'_0 \\ \boldsymbol{\sigma}_0 \\ \vdots \\ \boldsymbol{\sigma}_q \\ \boldsymbol{\sigma}'_q \end{bmatrix} \quad (7.16)$$

where  $N_{0,p}(\bar{u}_0) = 1$ ,  $N_{i \neq 0,p}(\bar{u}_0) = 0$ ,  $N_{i \neq n,p}(\bar{u}_q) = 0$ , and  $N_{n,p}(\bar{u}_q) = 1$ . The system in Equation (7.16) has  $q+3$  rows and  $n+1$  columns. For the resulting NURBS to provide exact interpolation the system must have an exact solution, therefore  $n+1 = q+3$  control points are needed. Two

control points more than the number of waypoints are required in order to enforce the endpoint derivatives.

To populate the stiffness matrix in Equation (7.16) it is necessary to define a knot vector  $U$ , as described in Equation (7.2). This knot vector has size  $m + 1$ , where  $m = n + p + 1$ . The intermediate knots  $u_i$  can be computed averaging the  $\bar{u}_k$ 's [86]:

$$\begin{aligned} u_0 = \dots = u_p &= 0 & u_{m-p} = \dots = u_m &= 1 \\ u_{p+j+1} &= \frac{1}{p} \sum_{i=j+1}^{j+p} \bar{u}_i & \text{for } j &= 0, \dots, m - 2p - 2. \end{aligned} \quad (7.17)$$

The path computed with the basic A\* implementation at the end of Chapter 6 is interpolated with a 4-th order NURBS curve, with endpoint derivatives  ${}^{\mathcal{B}}\boldsymbol{\omega}_S = {}^{\mathcal{B}}\boldsymbol{\omega}_G = \{0, 0, 0\}^T$ , and an average angular rate  $\omega^* = 0.03$  rad/s. The angular rates and accelerations are obtained from  $\dot{\boldsymbol{\sigma}}$  and  $\ddot{\boldsymbol{\sigma}}$  applying Equations (2.22) and (2.24). Given the inertia tensor  $[\mathbf{I}]$  of the spacecraft, the required control torque  $\mathbf{L}$  is obtained from Equation (2.30).

The interpolated path is shown in Figure 7.1. Figure 7.2 shows the evolution of attitude, rates, accelerations and torque, along the maneuver, which is performed in about 150 seconds. The

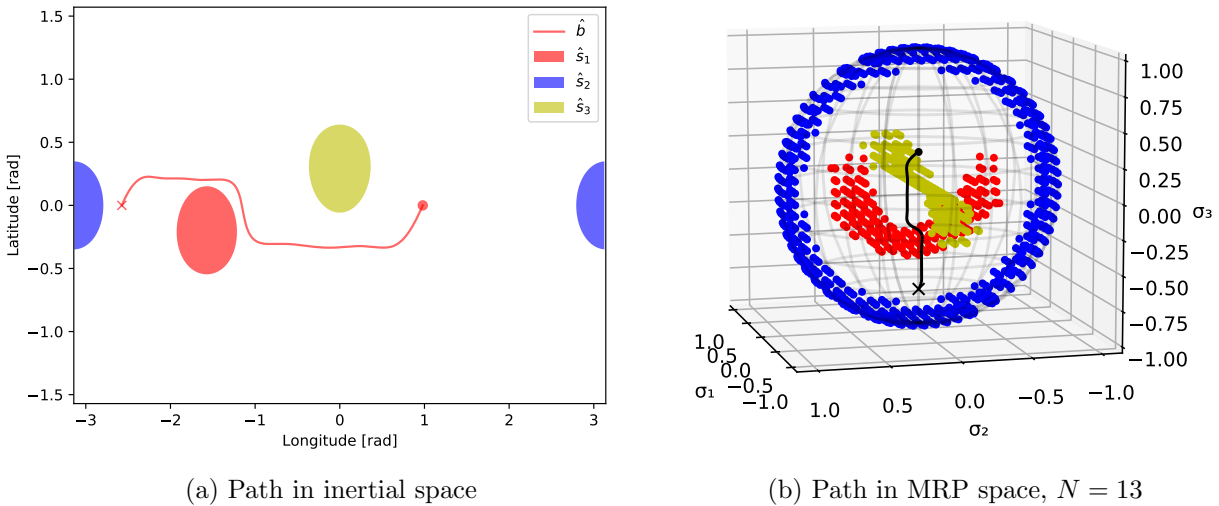


Figure 7.1: Metric-based A\* solution to the path planning query, 'o' is starting point and 'x' is target point



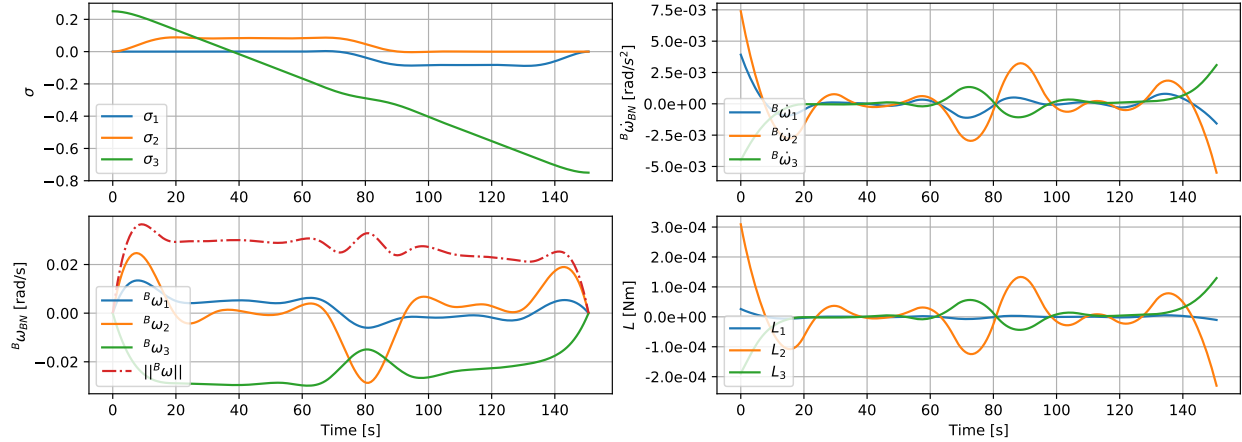


Figure 7.2: Attitude, rates, acceleration, and torque over time for the trajectory in Figure 7.1

order  $p = 4$  is chosen to ensure that the second order derivative  $\ddot{\sigma}$  is at least  $\mathcal{C}^1$  [88], which allows to achieve a smooth torque profile. The inertia tensor used to compute the torques corresponds to a three-unit cubesat with a uniformly distributed mass  $m = 4.2$  kg, expressed with respect to the principal body frame:

$${}^{\mathcal{B}}[I] = \begin{bmatrix} 6.67 & 0 & 0 \\ 0 & 41.87 & 0 \\ 0 & 0 & 41.87 \end{bmatrix} \cdot 10^{-3} \text{ kg} \cdot \text{m}^2. \quad (7.18)$$

The angular rate plot shows also the norm of the angular rate vector, which maintains itself fairly close to the desired  $\omega^* = 0.03$  rad/s. The largest accelerations, and torques, are required at the endpoints, where the spacecraft needs to be accelerated/decelerated from/to the rest condition.

A challenge is presented for the interpolating spline when the sequence of waypoints involves MRP switching, because the interpolating  $\sigma(t)$  function would have to present a discontinuity. The routine described above cannot handle such case directly. However,  $\sigma \leq 1$  is a soft constraint: it is therefore possible to exploit the shadow-sets of a subset of the waypoints, which lie outside of the unit sphere, to interpolate the function in a continuous domain. Once the continuous spline is interpolated, the part of the function that lies outside of the  $\sigma \leq 1$  constraint can be mapped back to the unit sphere. The angular rates and accelerations  ${}^{\mathcal{B}}\omega_{\mathcal{B}\mathcal{N}}$  and  ${}^{\mathcal{B}}\dot{\omega}_{\mathcal{B}\mathcal{N}}$  are computed using  $\dot{\sigma}$  and  $\ddot{\sigma}$  from either set. An example of this is reported in Figure 7.3: the black curve in Figure 7.3b

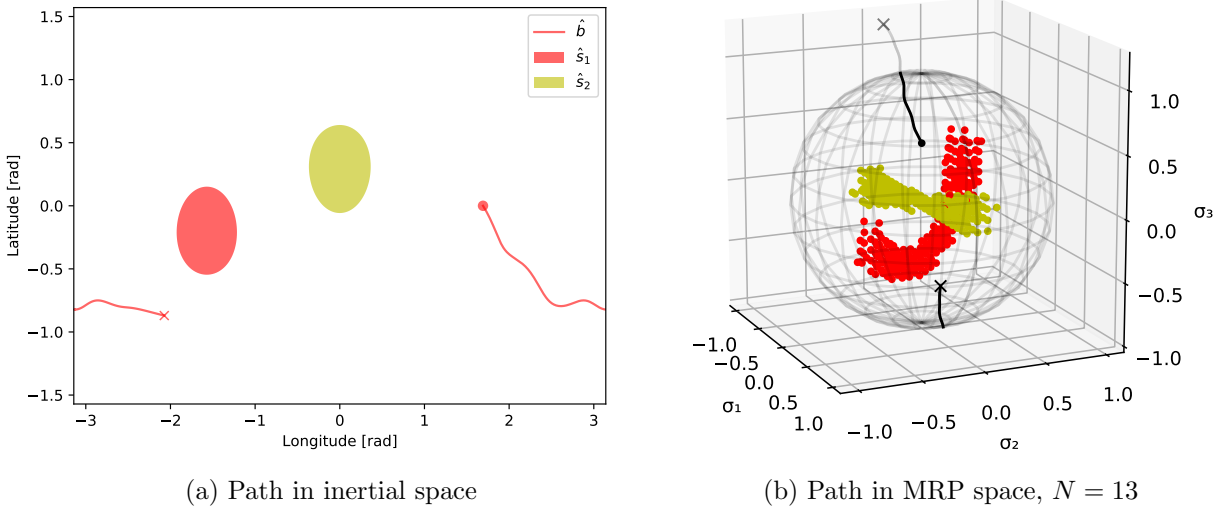


Figure 7.3: Interpolated metric-based A\* solution involving MRP switching

represents the interpolated path within the unit sphere, whereas the gray curve represents the interpolated curve using the shadow sets outside the unit sphere, where the gray ‘X’ represents the shadow set of the target attitude  $\sigma_G^S$ .

**7.3.3 NURBS Least Squares Fit**

This subsection outlines the procedure to obtain a LS approximating NURBS curve from a set of  $q + 1$  attitude waypoints. The idea that motivates the use of a LS fit of the waypoints, instead of a precise interpolation, is to reduce the dependence from the waypoints. The waypoints provide guidance in the attitude space to avoid the keep-out zones, but other than that, they are artificial constructs and, therefore, there is no real need to track them precisely.

The approach followed in this subsection aims to find a NURBS curve which provides a least-squares fit of the attitude waypoints by means of defining the  $n + 1$  control points. The first two control points,  $P_0$  and  $P_1$ , and the last two,  $P_{n-1}$  and  $P_n$ , are determined imposing the endpoint coordinates by solving the  $4 \times 4$  linear system outlined in Section 7.3.1. The remaining  $n - 3$  control points are obtained via a least-squares fit.

The following two subsections offer two different options on how to perform the LS fit: in the first case, the LS fit is only based on the intermediate attitude waypoints  $\boldsymbol{\sigma}_k$ . In the second case, the LS also accounts for the desired angular rate along the resulting continuous NURBS trajectory.

### 7.3.3.1 Least-Squares Fit with Attitude Waypoints

In this first approach, the remaining  $q - 1$  attitude waypoints are the only elements that contribute to the LS routine. The following quantities are defined first:

$$\boldsymbol{\rho}_k = \boldsymbol{\sigma}_k - N_{0,p}(\bar{u}_k)\mathbf{P}_0 - N_{1,p}(\bar{u}_k)\mathbf{P}_1 - N_{n-1,p}(\bar{u}_k)\mathbf{P}_{n-1} - N_{n,p}(\bar{u}_k)\mathbf{P}_n \quad \text{for } k = 1, \dots, q - 1. \quad (7.19)$$

The  $\boldsymbol{\rho}_k$  terms are used to remove the dependence of the waypoints from the four control points that have already been pre-computed. Defining with  $\boldsymbol{\sigma}_k$  the waypoints to be fitted, and with  $\boldsymbol{\sigma}(\bar{u}_k)$  the output of the NURBS curve according to Equation (7.15), it is possible to set up an error function that is the sum of the quadratic errors, and only depends on the remaining  $\mathbf{P}_i$  for  $i = 2, \dots, n - 2$ :

$$f = \sum_{k=1}^{q-1} \|\boldsymbol{\sigma}_k - \boldsymbol{\sigma}(\bar{u}_k)\|^2 = \sum_{k=1}^{q-1} \|\boldsymbol{\rho}_k - \sum_{i=2}^{n-2} N_{i,p}(\bar{u}_k)\mathbf{P}_i\|^2. \quad (7.20)$$

To minimize for  $f$ , its derivative with respect to the control points is equated to zero:  $\frac{df}{d\mathbf{P}_l} = 0$  for  $l = 2, \dots, n - 2$ . This gives:

$$\sum_{i=2}^{n-2} \left( \sum_{k=1}^{q-1} N_{l,p}(\bar{u}_k)N_{i,p}(\bar{u}_k) \right) \mathbf{P}_i = \sum_{k=1}^{q-1} N_{l,p}(\bar{u}_k)\boldsymbol{\rho}_k \quad \text{for } l = 2, \dots, n - 2. \quad (7.21)$$

For more details on this derivation the reader is referred to Reference 86. Defining  $[N]$  the  $(q - 1) \times (n - 3)$  matrix of scalars:

$$[N] = \begin{bmatrix} N_{2,p}(\bar{u}_1) & \dots & N_{n-2,p}(\bar{u}_1) \\ \vdots & \ddots & \vdots \\ N_{2,p}(\bar{u}_{q-1}) & \dots & N_{n-2,p}(\bar{u}_{q-1}) \end{bmatrix} \quad (7.22)$$

Equation (7.21) is rewritten in the matrix form:

$$([N]^T[N]) \mathbf{P} = [N]^T \boldsymbol{\rho} \quad \text{with} \quad \mathbf{P} = \begin{Bmatrix} \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-2} \end{Bmatrix} \quad \text{and} \quad \boldsymbol{\rho} = \begin{Bmatrix} \boldsymbol{\rho}_1 \\ \vdots \\ \boldsymbol{\rho}_{q-1} \end{Bmatrix}. \quad (7.23)$$

Equation (7.23) represents three underdetermined systems, because the control points  $\mathbf{P}_i$  and the terms  $\boldsymbol{\rho}_k$  are three-dimensional. The system can be solved choosing the minimum norm solution, which yields the vector  $\mathbf{P}$  for which the NURBS curve gives the Least Squares approximation of the  $q - 1$  intermediate waypoints:

$$\mathbf{P} = ([N]^T[W][N])^{-1} [W][N]^T \boldsymbol{\rho} \quad (7.24)$$

where  $[W]$  is a square, diagonal, positive semidefinite matrix of size  $(q - 1)$  containing the weights associated with each waypoint. If  $[W]$  is the identity, all the waypoints are weighed equally. Equation (7.24) only works if the matrix  $([N]^T[W][N])$  is full rank. It is important to note that the elements that compose the matrix  $[N]$  are strongly dependent from the knot vector  $U$  defined in Equation (7.2). DeBoor showed that when every knot span in  $U$  contains at least one  $\bar{u}_k$ , the matrix  $([N]^T[W][N])$  is positive definite and well-defined [89]. The following algorithm for choosing the internal knots ensures that this is true [86]:

$$\begin{aligned} d &= \frac{q+1}{n-p+1} & i &= \text{int}(jd) & \alpha &= jd - i \\ u_{p+j} &= (1-\alpha)\bar{u}_{i-1} + \alpha\bar{u}_i & \text{for} & & j &= 1, \dots, n-p. \end{aligned} \quad (7.25)$$

In this work, the algorithm presented in Equation (7.25) is slightly modified such that if  $u_{p+1} < \bar{u}_1$ , then  $u_{p+1}$  is set equal to  $\bar{u}_1$ , and if  $u_n > \bar{u}_{q-1}$ , then  $u_n$  is set equal to  $\bar{u}_{q-1}$ , to ensure that the first and last polynomial arcs covers at least the distance between the first two and last two waypoints, respectively. This has been observed to stabilize the NURBS function in those intervals, and avoiding overshoots.

Figure 7.4 shows the performance of the LS approximating NURBS for a set of constraint-compliant MRP waypoints, where an angular rate norm of  $\omega^* = 0.03$  rad/s is desired. The guidance

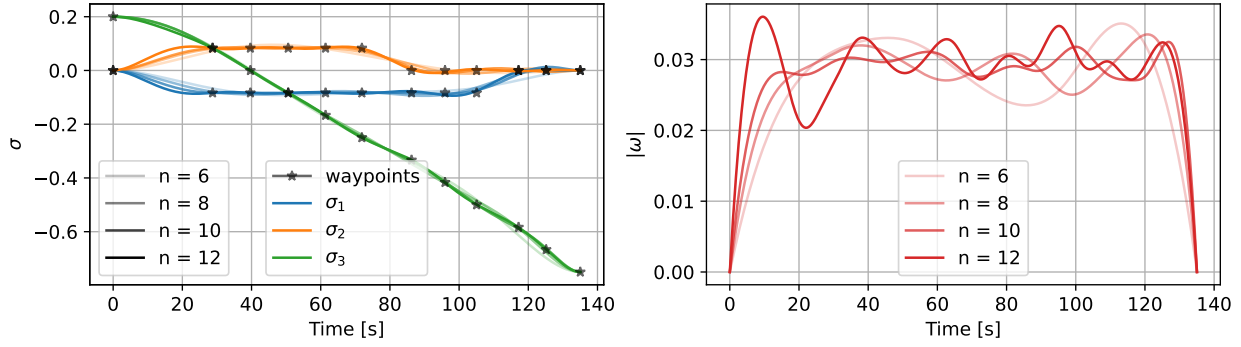


Figure 7.4: LS approximating NURBS, attitude-based, for varying numbers of control points  $n$

waypoints are obtained once again applying the elementary  $A^*$  algorithm presented in Chapter 6, with the same constraints as in Figure 6.5, but with boundary conditions  $\sigma_S = \{0.0, 0.0, 0.2\}$  and  $\sigma_G = \{0.0, 0.0, -0.75\}$ . The left plot shows different LS solutions obtained for varying numbers of control points  $n$ , for each MRP guidance waypoint  $\sigma_j$ . It can be observed that for a smaller number of control points the approximation fits the waypoints more loosely. Conversely, as the number of control points is increased, the approximation fits the waypoints almost perfectly. The right-hand-side plot shows the angular rate norm associated with the LS approximating solutions, also displayed for different numbers of control points. It is observed that, while the angular rate norm is on average close to the desired value, it still oscillates noticeably. This is not surprising, since the required angular rate norm is not accounted for in this LS approximation.

### 7.3.3.2 Least-Squares Fit with Attitude Waypoints and Angular Rate Norm

This subsection elaborates on the results of the previous one, enhancing the LS approximating NURBS curve to match the requirement on the angular rate norm. Because Equation (7.23) is an underdetermined system, it is possible to compute a minimum norm, LS solution that incorporates in its error function not only the error with respect to the waypoints, but also to the desired angular velocities at such waypoints. The requirement for the reference trajectory is to have a constant angular rate norm  $\|\omega\| = \omega^*$ . While this requirement cannot be enforced at the trajectory endpoints, it can be enforced in the intermediate waypoints.

Ideally, one should set up an error function for the MRP derivatives like:

$$g = \sum_{k=1}^{q-1} (\|\boldsymbol{\sigma}'_k\|^2 - \|\boldsymbol{\sigma}'(\bar{u}_k)\|^2)^2 = \sum_{k=1}^{q-1} \left[ \|\boldsymbol{\sigma}'_k\|^2 - \sum_{j=1}^3 \left( \sum_{i=2}^{n-2} N'_{i,p}(\bar{u}_k) \mathbf{P}_{i,j} \right)^2 \right]^2. \quad (7.26)$$

However, solving for the control points  $\mathbf{P}_i$  that minimize the error function in Equation (7.26) is nontrivial, because due to the quadratic dependence of the terms in square brackets from the  $\mathbf{P}_i$ , computing the solution would require nonlinear programming. A nonlinear program could be set up and solved. The desire, however, is to find a solution without involving iterative methods. Instead, an estimate of the MRP derivatives  $\boldsymbol{\sigma}'_k$  is provided as follows. As a first step, such derivatives are computed using central finite differences:

$$\hat{\boldsymbol{\sigma}}'_k = \frac{\bar{u}_{k+1} - \bar{u}_k}{\bar{u}_{k+1} + \bar{u}_{k-1}} \cdot \frac{\boldsymbol{\sigma}_k - \boldsymbol{\sigma}_{k-1}}{\bar{u}_k - \bar{u}_{k-1}} + \frac{\bar{u}_k - \bar{u}_{k-1}}{\bar{u}_{k+1} + \bar{u}_{k-1}} \cdot \frac{\boldsymbol{\sigma}_{k+1} - \boldsymbol{\sigma}_k}{\bar{u}_{k+1} - \bar{u}_k} \quad \text{for } k = 1, \dots, q-1. \quad (7.27)$$

Knowing the desired angular rate norm  $\omega^*$ , it is possible to leverage Equation (2.29) to scale the estimated MRP derivatives obtained via finite differences, to make them correspond to an angular rate vector with the desired magnitude:

$$\boldsymbol{\sigma}'_k = \frac{\hat{\boldsymbol{\sigma}}'_k}{\|\hat{\boldsymbol{\sigma}}'_k\|} \cdot \frac{1 + \|\boldsymbol{\sigma}_k\|^2}{4} \omega^* \cdot t_q \quad \text{for } k = 1, \dots, q-1. \quad (7.28)$$

Having estimated the desired MRP rates, it is possible to proceed setting up a LS problem that is analogous to that introduced in the previous subsection. Let's define:

$$\boldsymbol{\rho}'_k = \boldsymbol{\sigma}'_k - N'_{0,p}(\bar{u}_k) \mathbf{P}_0 - N'_{1,p}(\bar{u}_k) \mathbf{P}_1 - N'_{n-1,p}(\bar{u}_k) \mathbf{P}_{n-1} - N'_{n,p}(\bar{u}_k) \mathbf{P}_n \quad \text{for } k = 1, \dots, q-1 \quad (7.29)$$

where the four control points are computed as described in Section 7.3.1. The new error function is set up to incorporate the squared errors with respect to the MRP waypoints and associated MRP rates:

$$\begin{aligned} f &= \sum_{k=1}^{q-1} (\|\boldsymbol{\sigma}_k - \boldsymbol{\sigma}(\bar{u}_k)\|^2 + \|\boldsymbol{\sigma}'_k - \boldsymbol{\sigma}'(\bar{u}_k)\|^2) \\ &= \sum_{k=1}^{q-1} \left( \|\boldsymbol{\rho}_k - \sum_{i=2}^{n-2} N_{i,p}(\bar{u}_k) \mathbf{P}_i\|^2 + \|\boldsymbol{\rho}'_k - \sum_{i=2}^{n-2} N'_{i,p}(\bar{u}_k) \mathbf{P}_i\|^2 \right). \end{aligned} \quad (7.30)$$

Applying the same procedure to minimize  $f$  with respect to  $\mathbf{P}_i$  gives the same minimum norm solution as in Equation (7.24), only in this case the matrices that appear in the equation take the form:

$$[N] = \begin{bmatrix} N_{2,p}(\bar{u}_1) & \dots & N_{n-2,p}(\bar{u}_1) \\ \vdots & \ddots & \vdots \\ N_{2,p}(\bar{u}_{q-1}) & \dots & N_{n-2,p}(\bar{u}_{q-1}) \\ N'_{2,p}(\bar{u}_1) & \dots & N'_{n-2,p}(\bar{u}_1) \\ \vdots & \ddots & \vdots \\ N'_{2,p}(\bar{u}_{q-1}) & \dots & N'_{n-2,p}(\bar{u}_{q-1}) \end{bmatrix}, \quad \mathbf{P} = \begin{Bmatrix} \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{n-2} \end{Bmatrix}, \quad \boldsymbol{\rho} = \begin{Bmatrix} \rho_1 \\ \vdots \\ \rho_{q-1} \\ \rho'_1 \\ \vdots \\ \rho'_{q-1} \end{Bmatrix} \quad (7.31)$$

with  $[N]$  being a  $(2q-2) \times (n-3)$  matrix and  $\boldsymbol{\rho}$  being a  $(2q-2)$  vector, while  $\mathbf{P}$  remains a  $(n-3)$  vector. The weight matrix  $[W]$  has size  $(2q-2) \times (2q-2)$  and it can be used to give more weight, within the LS approximation, to the MRP waypoints or the MRP derivatives. It should be noted that, according to this formulation, the angular rate  $\boldsymbol{\omega}$  is a function of both  $\boldsymbol{\sigma}$  and  $\dot{\boldsymbol{\sigma}}$ : this means that if the error on  $\boldsymbol{\sigma}$  is large, the desired  $\boldsymbol{\omega}^*$  will not be achieved even if the error on  $\dot{\boldsymbol{\sigma}}$  is small. For this reason, it is conceptually wrong to attribute a higher weight to the MRP derivative elements than the MRP waypoints, since it is the latter that need to be approximated well enough for the rest of the approximation to hold.

Figure 7.5 shows the LS solutions, for varying numbers of control points, obtained using the formulation that accounts for waypoints and angular rates, with equal weight. In the left-hand-side plot, the approximation yields curves that approximate the waypoints less tightly, compared to Figure 7.4, but still recover the distribution of the waypoints faithfully. The reduced accuracy in tracking the waypoints is due to the fact that the minimum norm solution is trying, in this case, to optimize not only the waypoints, but also the angular rates. More interestingly, it can be observed from the right-hand-side plot that the angular rate norm, in this case, is much closer to the desired target  $\boldsymbol{\omega}^* = 0.03$  rad/s, and the approximation is better for a higher number of control points.

The Least Squares approach allows to also model the accelerations at each intermediate waypoint and add them to the error function analogously. However, in such case the resulting function

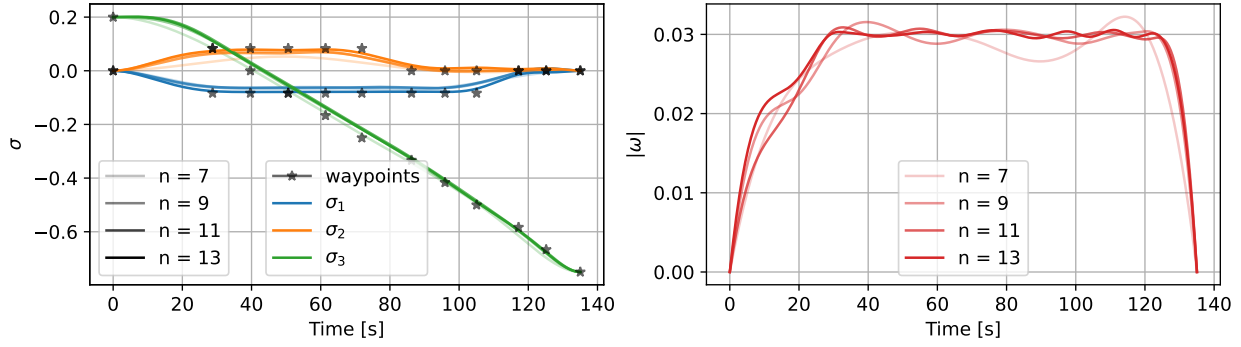


Figure 7.5: LS approximating NURBS, attitude- and angular rate-based, for varying numbers of control points  $n$

attempts at satisfying three different requirements, on attitude, angular rates and accelerations, therefore losing tracking accuracy on all three. If the desire is to reduce the angular accelerations, especially in the on-ramping and off-ramping phases, the best choice is to reduce the nominal angular rate  $\omega^*$ , which ultimately results in a longer total maneuver time.

#### 7.4 Performance Study of NURBS Interpolation vs LS Approximation

Now that the mathematical formulation for both the interpolating and the LS-approximating NURBS have been laid out, the next step is to compare their performance. The motivation for using the LS approximating NURBS is to relax the dependence of the trajectory from the waypoints, which serve as a guidance for the trajectory but are, ultimately, an ‘artificial’ constraint. Removing the constraint of precisely hitting the waypoints would allow the trajectory to have a smoother profile, which would ultimately lead to a smaller control effort. In fact, the interpolating NURBS cause the trajectory to wiggle more between the waypoints, which causes some ‘parasitic’ torques to appear and ultimately increase the required control effort. This section compares the performances of the two NURBS curves in three different scenarios with different sets of constraints. Moreover, the comparison is performed, for each scenario, for a varying level of grid density  $N$ . For the LS approximating NURBS,  $n + 1$  control control points are used, where  $n = q + 2$ , that is, one control point per waypoint plus two control points for the endpoint derivatives constraints. This is the



same number of control points required by the interpolating NURBS. The trajectories are based, for each scenario, on the sequence of waypoints obtained applying a basic implementation of the A\* algorithm outlined in Chapter 6, using the metric presented in Equation (7.6) as the algorithm's cost function: such trajectories are, therefore, not effort-optimal, but this is not relevant for the comparisons that this sections aims to make. Lastly, the computational costs of the two NURBS routines are also compared.

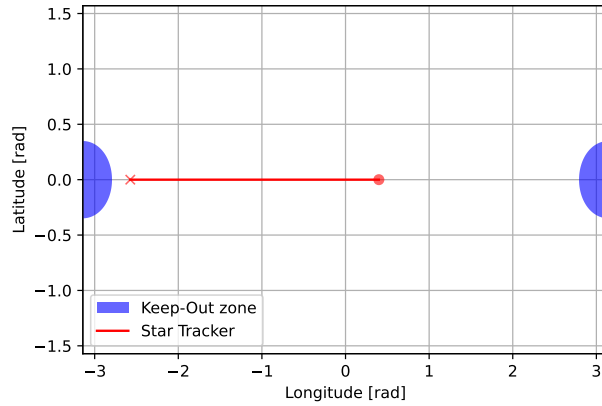
In order to define a common metric for the control effort required by the two approaches, the same inertia tensor for a 3-unit cubesat in Equation (7.18) is considered. The control effort is quantified as the integral over the trajectory of the norm of the instantaneous control torque:

$$C_0 = \int_0^T \|\mathbf{L}\| dt \quad (7.32)$$

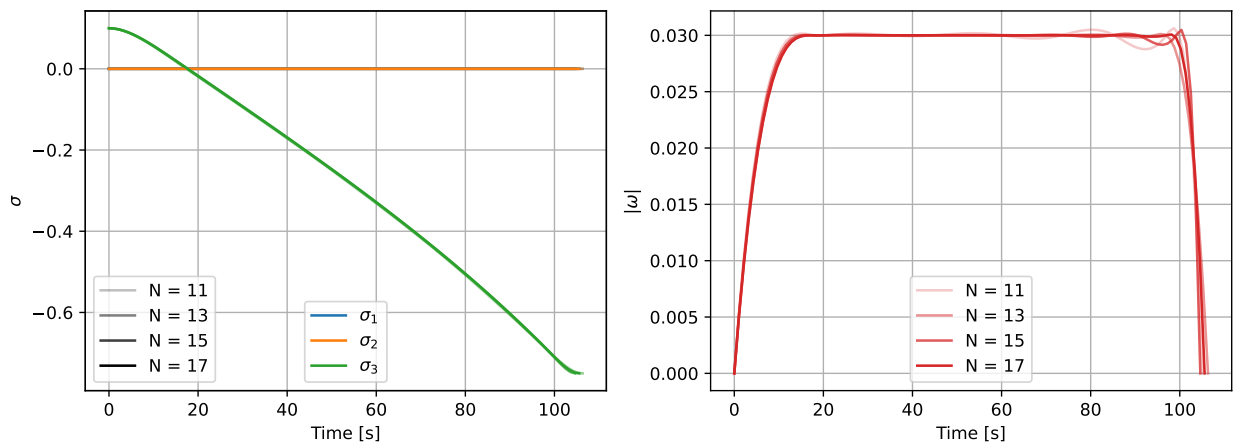
where the control torque  $\mathbf{L}$  is computed according to Euler's Equation (2.30) assuming a rigid body and no external perturbations. In reality, the reaction wheels that actuate the spacecraft contribute to the inertia and angular momentum of the system, and introduce gyroscopic effects that also contribute to the control effort. A more refined, although more computationally demanding, path planning algorithms is presented in Chapter 8, where the dynamics of the reaction wheels are factored into the cost function. All the scenarios presented in this section feature rest-to-rest maneuvers with a target angular rate norm  $\omega^* = 0.03$  rad/s.

#### 7.4.1 Scenario 1: Eigenaxis Rotation

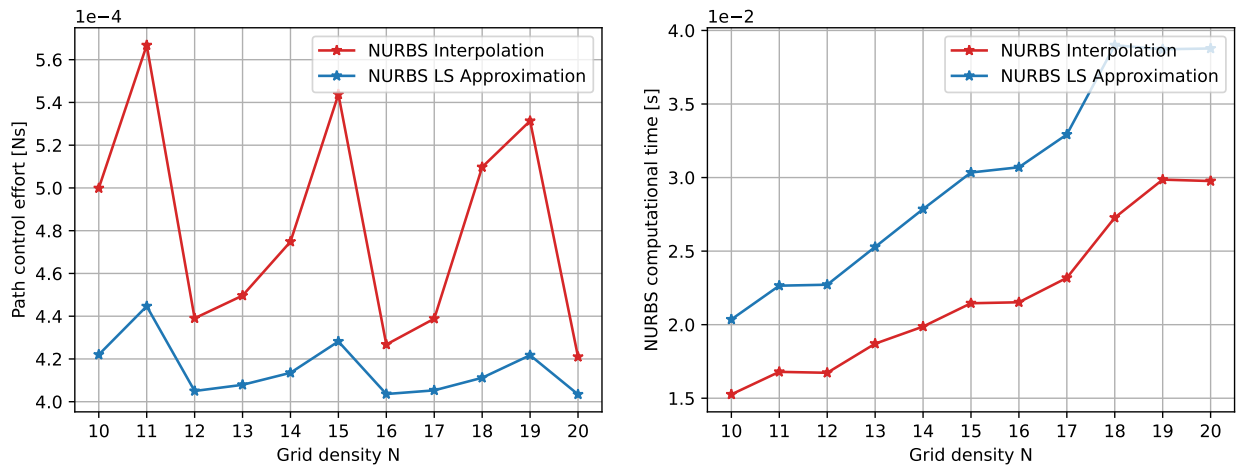
The first scenario features an eigenaxis rotation about the  $\hat{\mathbf{b}}_3$  axis. Only one sensitive instrument with a field of view of 20 deg is oriented along the  $\hat{\mathbf{b}}_1$  axis, and one keep-out zone is present along the inertial direction  ${}^{\mathcal{N}}\hat{\mathbf{s}} = \{-1, 0, 0\}^T$ . The initial and final attitude are, respectively,  $\boldsymbol{\sigma}_0 = \{0, 0, 0.1\}^T$  and  $\boldsymbol{\sigma}_q = \{0, 0, -0.75\}^T$ . Figure 7.6 shows all the plots related to this simulation. Figure 7.6a shows the trajectory of the sensitive boresight in inertial space, projected onto a 2D plane. All the trajectories obtained for different grid densities  $N$  overlap along the same projection on the 2D plot. More interesting are the results shown in Figure 7.6b, which also shows



(a) Boresight plot



(b) Attitude  $\sigma$  and angular rate  $\|\omega\|$  for varying grid densities  $N$



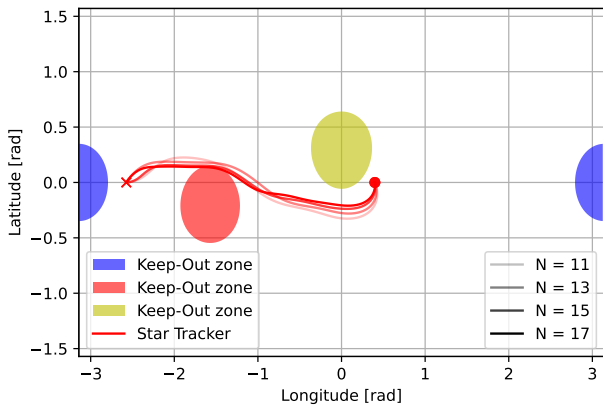
(c) Control effort and computational time of interpolating NURBS vs LS approximating NURBS, for varying grid densities  $N$

Figure 7.6: Scenario 1 - Eigenaxis Rotation

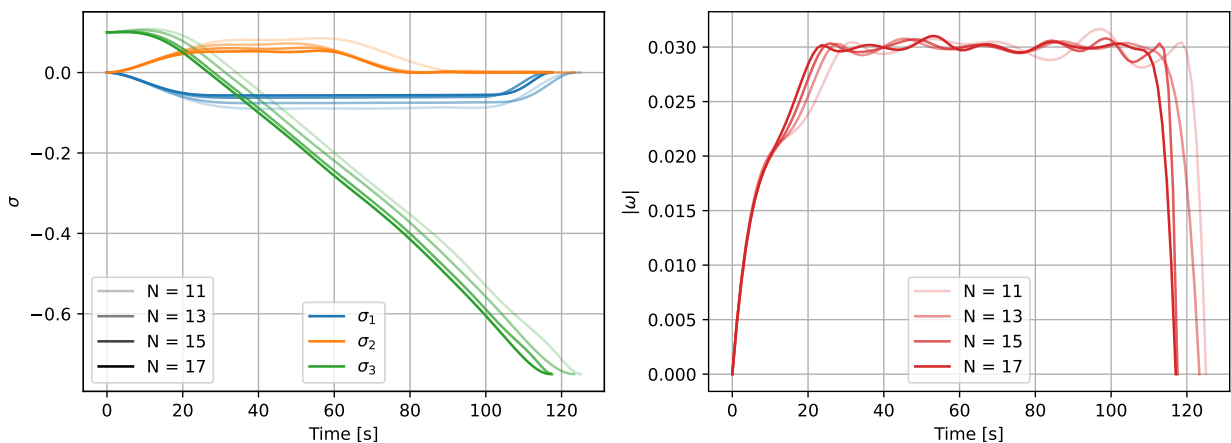
that the trajectories match. This happens because of the simple nature of the eigenaxis rotation, for which all the guidance waypoints lie along the  $\sigma_3$  axis. For all values of  $N$  the angular rate norm is approximated very well, as it stays very close to the target value. The on-ramping at the beginning of the maneuver is milder than the off-ramping. This can happen due to two main reasons: the first is that the sampling is uniform in MRP space, but not in attitude space, due to the nonlinearity in the MRP formulation. As a result, neighboring points farther away from the origin describe tighter maneuvers in terms of principal rotation angle. As a result, the NURBS curve at the end of the maneuver has less time to off-ramp from the average angular rate  $\omega^*$ . The second reason is that initial and final attitude point do not, in general, belong to the uniformly-spaced grid, and could therefore be located closer or farther to the next node than the average node distance. This can further emphasize the first phenomenon, if this happens when one of the endpoints is far away from the origin. As a result of these combined effects, the fast off-ramping can cause oscillations in the curve near the point where the velocity starts to decrease. Figure 7.6c shows the comparison between the two types of NURBS: as far as control effort, the LS approximating curve performs better across all grid densities, yielding in some cases a significant reduction. The computational cost of the LS approximating curve is, on the other hand, higher. Despite that, it scales comparably to the interpolating curve as the grid density (and therefore the number of waypoints  $q$ ) increases.

#### 7.4.2 Scenario 2: Multiple Keep-out Zones

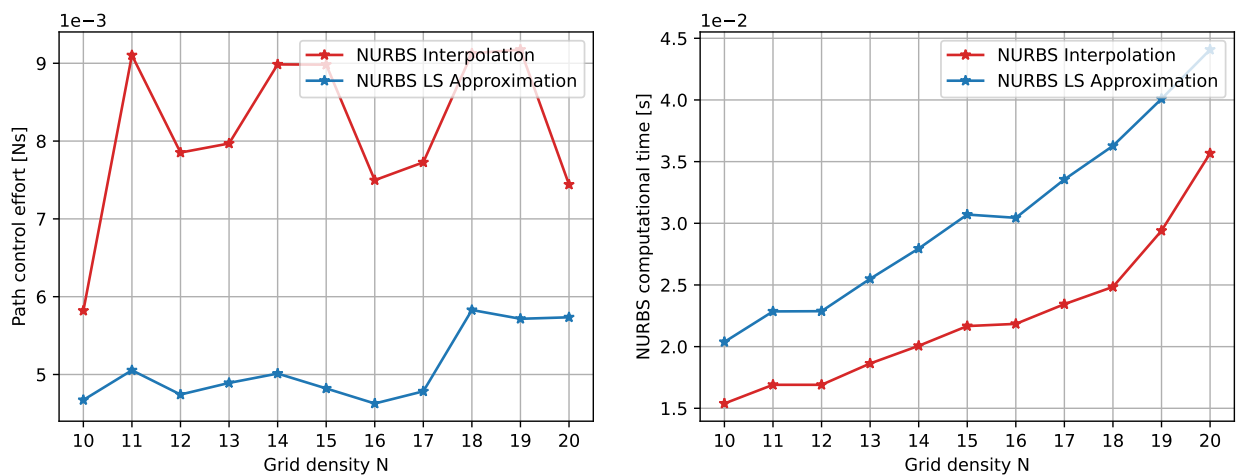
The second scenario is based off of the previous one, with the addition of two more keep-out zones along the inertial directions  ${}^{\mathcal{N}}\hat{\mathbf{s}}_2 = \{0, -0.981, -0.196\}^T$  and  ${}^{\mathcal{N}}\hat{\mathbf{s}}_3 = \{0.958, 0, 0.287\}$ . The relevant plots are reported in Figure 7.7. Figure 7.7a shows the trajectories of the sensitive instrument in inertial space, as it steers away from the keep-out zones. In this case, different trajectories are obtained for different grid densities  $N$ : this happens because, for different  $N$ , the attitude space is sampled with different nodes, and therefore the baseline path computed by  $A^*$  consists of different waypoints for each case. Regardless, the different trajectories remain fairly close to one another, although for high grid densities some brief constraint violations occur. Figure 7.7b



(a) Boresight plot



(b) Attitude  $\sigma$  and angular rate  $\|\omega\|$  for varying grid densities  $N$



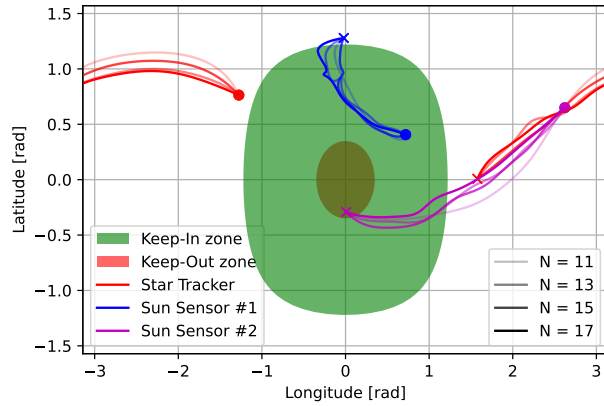
(c) Control effort and computational time of interpolating NURBS vs LS approximating NURBS, for varying grid densities  $N$

Figure 7.7: Scenario 2 - Multiple Keep-out Zones

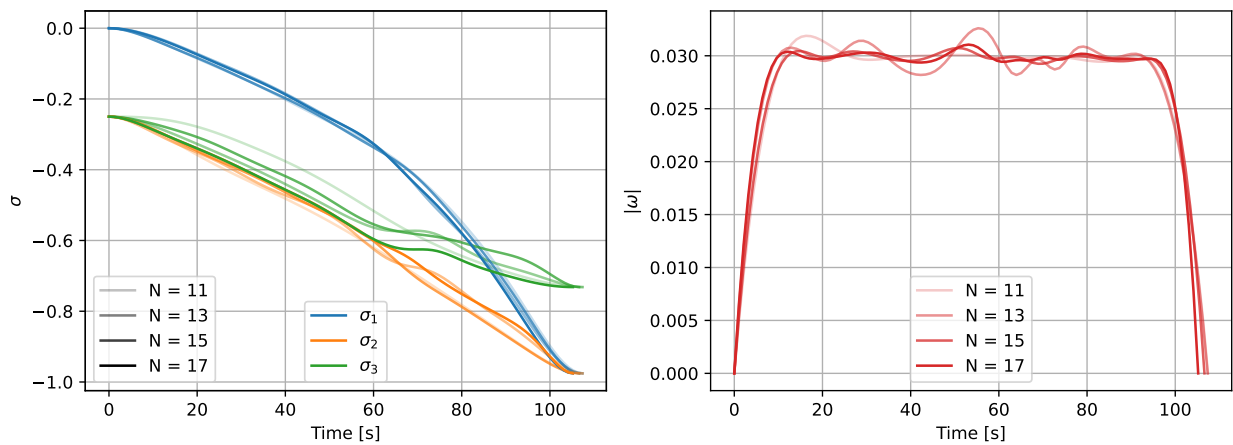
shows that, even in this case, the angular rate is approximated quite well, although the plateau around  $\omega^* = 0.03$  rad/s is not as flat as for the previous scenario. This happens because, for scenario 1, all the baseline waypoints are aligned along a straight line, for which the approximations intrinsic Equation (7.28) is much more accurate. It can also be observed that, for higher  $N$ , the angular rate norm is less oscillatory, thanks to a higher number of waypoints that provide more guidance for the output trajectory. Lastly, Figure 7.7c compares the performances of the two NURBS curves. In this scenario, the interpolating NURBS is associated with a control effort that is almost twice as high as the LS approximating NURBS. Moreover, the control effort for the first one has increased significantly with respect to scenario 1, whereas for the second, the control effort remains closer to the values computed for the first scenario. For the computational cost, the same considerations can be made: the LS approximating NURBS is consistently more costly, but comparable to the interpolating NURBS, and they display a similar trend with increasing levels of grid density.

#### 7.4.3 Scenario 3: Mixed Keep-in and Keep-out Zones

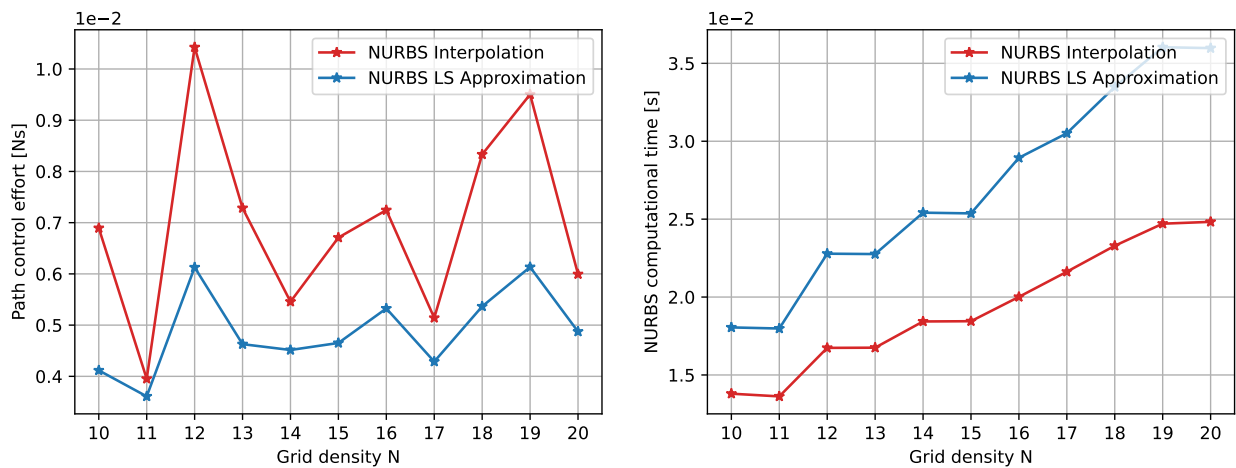
The third scenario is the most articulated, as it features multiple instruments. There is still a sensitive camera aligned along the  $\hat{\mathbf{b}}_1$  axis, and two sun sensors, along the  $\hat{\mathbf{b}}_2$  and  $\hat{\mathbf{b}}_3$  axes respectively, each with a field of view of 70 deg. The only celestial object is the Sun, located along the  ${}^{\mathcal{N}}\hat{\mathbf{s}}_3 = \{1, 0, 0\}^T$  inertial direction. The constraints for this problem consist of maneuvering such that the Sun remains out of the field of view of the camera, but within the field of view of at least one of the sun sensors. The initial and final attitude are, respectively,  $\boldsymbol{\sigma}_0 = \{0, -0.25, -0.25\}^T$  and  $\boldsymbol{\sigma}_q = \{0.4, 0.4, 0.3\}$ . The relevant plots are reported in Figure 7.8. Figure 7.8a shows the trajectories of the three boresights in inertial space: all three are compliant, in all their parts, for all grid densities  $N$ . The sun sensors switch, since at the initial attitude only sun sensor 1 sees the Sun, while sun sensor 2 sees the Sun once the final pose is reached. Similar considerations apply for Figure 7.8b as to the previous scenario: the target angular rate norm is approximated fairly well, and the more so when the grid density is higher. Relative to Figure 7.8c, the control effort displays an irregular behavior for both NURBS as  $N$  varies, but again the LS approximation outperforms



(a) Boresight plot



(b) Attitude  $\sigma$  and angular rate  $\|\omega\|$  for varying grid densities  $N$



(c) Control effort and computational time of interpolating NURBS vs LS approximating NURBS, for varying grid densities  $N$

Figure 7.8: Scenario 3 - Mixed Keep-in and Keep-out Zones

the interpolation for each grid density. Regarding computational time, the same considerations apply as in the previous two scenarios.

## 7.5 Closed-loop Performance of NURBS-based guidance for attitude path planning

This section involves some new Basilisk functionality that allows externally-generated attitude guidance solutions to be incorporated and their performance analyzed. The goal is to provide a framework where the only difference in the attitude maneuvering performance is the guidance solution. The use of this setup is illustrated by simulating the sample 3-unit cube satellite mentioned in the previous sections.

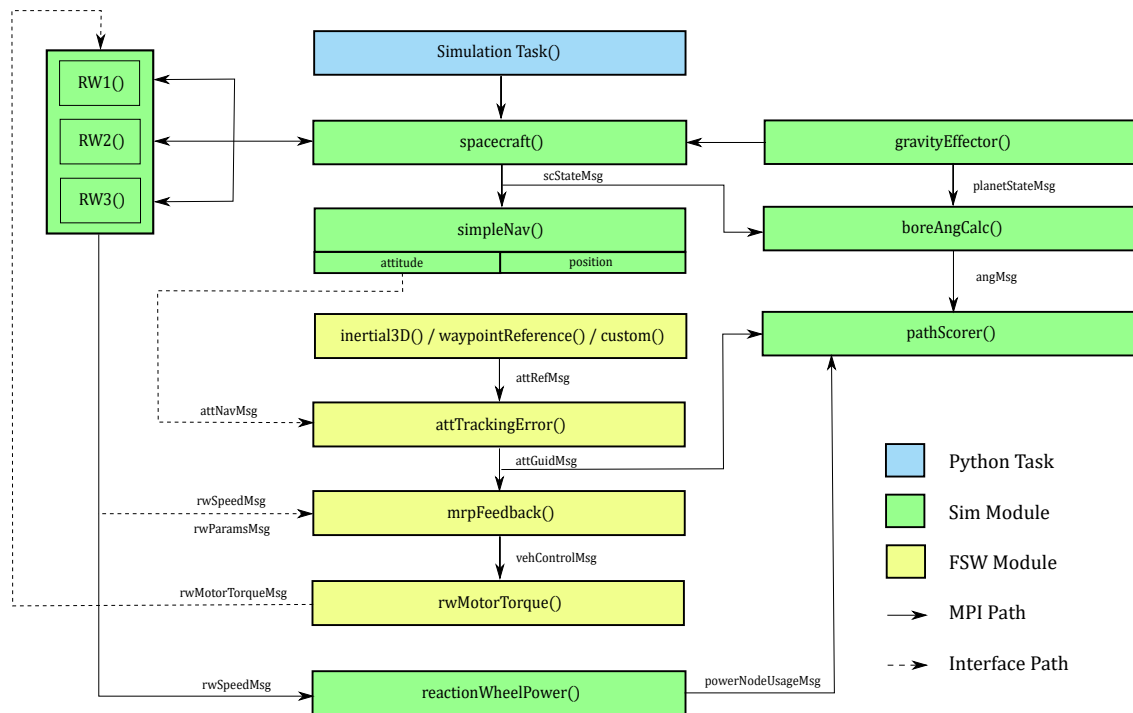


Figure 7.9: Basilisk modular structure for the current simulation

Figure 7.9 illustrates a sample attitude maneuver of a spacecraft using three RWs to control the orientation, and the translational motion is subject to both Earth and Sun gravity. This

scenario is substantially identical to the scenario presented in Figure 2.2. Two newly introduced components are here discussed in the details.

### 7.5.1 Incorporating Externally Generated Attitude Guidance

A new module called `waypointReference()` imports the reference trajectory of a reorientation maneuver from a text file. This enables Basilisk to be used as a benchmarking tool to test constrained attitude guidance solutions computed outside the Basilisk environment. The reference trajectory should be provided as an ordered list of time-tagged attitude waypoints  $\sigma_{\mathcal{R}/\mathcal{N}}$ , together with the associated reference angular rates  $\omega_{\mathcal{R}/\mathcal{N}}$  and accelerations  $\dot{\omega}_{\mathcal{R}/\mathcal{N}}$ .

The module outputs a reference message based on the reference trajectory described by the text file. When the simulation time is smaller than the first time tag, the output reference message contains the first reference attitude waypoint and zero angular rates and accelerations. When the simulation time exceeds the last time tag, the output reference message contains the last reference attitude waypoint with zero angular rates and accelerations. If the simulation time falls between two time tags  $t_i \leq t_{\text{sim}} < t_{i+1}$ , the reference message is obtained by means of linear interpolation of attitude, rates and accelerations between waypoints  $i$  and  $i + 1$ :

$$\hat{\sigma}_{\mathcal{R}/\mathcal{N}} = \sigma_{\mathcal{R}/\mathcal{N},i} + \frac{\sigma_{\mathcal{R}/\mathcal{N},i+1} - \sigma_{\mathcal{R}/\mathcal{N},i}}{t_{i+1} - t_i} (t_{\text{sim}} - t_i) \quad (7.33a)$$

$$\hat{\omega}_{\mathcal{R}/\mathcal{N}} = \omega_{\mathcal{R}/\mathcal{N},i} + \frac{\omega_{\mathcal{R}/\mathcal{N},i+1} - \omega_{\mathcal{R}/\mathcal{N},i}}{t_{i+1} - t_i} (t_{\text{sim}} - t_i) \quad (7.33b)$$

$$\hat{\dot{\omega}}_{\mathcal{R}/\mathcal{N}} = \dot{\omega}_{\mathcal{R}/\mathcal{N},i} + \frac{\dot{\omega}_{\mathcal{R}/\mathcal{N},i+1} - \dot{\omega}_{\mathcal{R}/\mathcal{N},i}}{t_{i+1} - t_i} (t_{\text{sim}} - t_i) \quad (7.33c)$$

where the hat indicates the quantities that are written to the output reference message. The linear interpolation between waypoints is an approximation, and it is therefore not correct from a kinematic standpoint. The three pieces of information contained in the reference message are, in fact, mutually interconnected by the relations in Equations (2.21) and (2.23). This discrepancy can cause the tracking error over time to appear irregular, because the reference trajectory that is being tracked is not kinematically consistent. However, this phenomenon becomes negligible as the



density of the waypoints increases and consequently the spacing between each pair of consecutive waypoints is reduced.

It should be noted that, when the attitude is specified using quaternions, the module always converts it to the equivalent MRP set, because a linear interpolation in MRP space always returns a set that represents a valid attitude description. The output of a linear attitude interpolation in quaternion space would produce a set that does not respect the unit-norm constraint on the quaternion at all times. Interpolation in quaternion space can be done via spherical linear interpolation (SLERP), which can however be convoluted and computationally demanding [90]. For these reasons, the choice of MRPs as the attitude representation set seems obvious in order to simplify the approach.

### 7.5.2 Evaluating the Constrained Attitude Maneuver Performance

A new module called `pathScorer()` provides more insight on the performances of different path planners. The main goal is obviously to ensure constraint compliance at all times during a slew maneuver. However, in the presence of multiple algorithms, it can be useful to compare other metrics in order to assess performances also with respect to other physical quantities. Five performance metrics are defined to test the difference planners.

- **Total keep-out violation time:**

The total keep-out violation time is defined as the cumulative time during which any of the keep-out constraints are violated by the sensitive instrument's boresight direction. This study focuses on hard positional constraints. However, this metric also becomes particularly interesting when integral constraints are considered, where the boresight is allowed to violate the keep-out zone for a certain amount of time.

$$T_{\text{KO}} = \int_0^T \delta_{\text{KO}} dt \quad \delta_{\text{KO}} = \begin{cases} 1 & \text{if keep-out is violated} \\ 0 & \text{otherwise} \end{cases} \quad (7.34)$$

- **Total keep-in violation time:**

In this keep-in constraints are considered satisfied when there is at least one constraint-compliant sensor at all times. This is to ensure that the keep-in object is visible at all times, which does not require for it to be visible by all the sensors at all times.

$$T_{KI} = \int_0^T \delta_{KI} dt \quad \delta_{KI} = \begin{cases} 1 & \text{if keep-in is violated} \\ 0 & \text{otherwise} \end{cases} \quad (7.35)$$

- **Attitude error integral:**

Defining  $\sigma_{B/R}$  as the MRP attitude error between the body frame  $\sigma_{B/N}$  and the target reference frame  $\sigma_{R/N}$ , this metric computes the integral of the principal rotation angle error  $\theta_{B/R}$  over the maneuver time, and is meant to provide an estimate of how accurately the reference can be tracked. Large tracking errors can lead to constraint violations even when the reference trajectory is constraint compliant, therefore this quantity must ideally remain small to ensure that the trajectory is tracked successfully.

$$\Theta_{B/R} = \int_0^T 4 \arctan (\|\sigma_{B/R}\|) dt = \int_0^T \theta_{B/R} dt \quad (7.36)$$

- **Commanded torque integral:**

The commanded torque  $L_r$  is the torque provided to the spacecraft to track the reference trajectory. Its expression is derived according to a nonlinear control law in Equation (2.49), with the integral term  $[K_I]$  set to zero. The commanded torque integral is defined as per Equation (7.32).

- **Total reaction wheel energy consumption:**

The total energy consumption is the integral over maneuver time of the power requirements of all the reaction wheels combined. For each reaction wheel, the power required is obtained as the product between the torque applied to the wheel  $u_s$  and the wheel speed relative to the spacecraft  $\Omega$ . The total required energy is:

$$E = \int_0^T \left( \sum_{i=1}^3 |u_{s_i} \Omega_i| \right) dt \quad (7.37)$$

### 7.5.3 Attitude Path Planners

To illustrate the use of these new capabilities, four different attitude guidance solutions are compared in this Basilisk simulation framework. The goal is to have an unconstrained attitude guidance solution as a baseline, and then compare this to different constrained attitude guidance solutions.

- **Planner #0:**

Planner #0 is, effectively, a constraint-naive planner. The slew maneuver is performed implementing nonlinear feedback control law in Equation (2.49) that drives the spacecraft from an initial attitude  $\sigma_{\mathcal{R}/\mathcal{N}_i}$  to a final attitude  $\sigma_{\mathcal{R}/\mathcal{N}_f}$  achieving a final rest state. In this case the reference message remains constant over time, with  $\hat{\sigma}_{\mathcal{R}/\mathcal{N}}$  being the final target attitude, and  $\hat{\omega}_{\mathcal{R}/\mathcal{N}} = \hat{\sigma}_{\mathcal{R}/\mathcal{N}} = 0$ . Since this planner is entirely based on the desired final attitude, constraint avoidance is not enforced. This planner is presented as an example of how constraints can easily be violated if not accounted for when performing a slew maneuver.

- **Planner #1:**

Planner #1 is based on a sequence of constraint-compliant reference attitude points  $\sigma_{\mathcal{R}/\mathcal{N}_j}$  with  $j = 0, \dots, N$ , with zero associated angular rates and accelerations  $\omega_{\mathcal{R}/\mathcal{N}_j} = \dot{\omega}_{\mathcal{R}/\mathcal{N}_j} = 0$ , where  $\sigma_{\mathcal{R}/\mathcal{N}_0} = \sigma_{\mathcal{R}/\mathcal{N}_i}$  and  $\sigma_{\mathcal{R}/\mathcal{N}_N} = \sigma_{\mathcal{R}/\mathcal{N}_f}$ . This approach is often found in literature involving attitude path planning [42, 36]: in the absence of any knowledge of the required angular rates and accelerations at the intermediate waypoints, these are set to zero. This choice yields a path planning algorithm that tries to perform rest-to-rest maneuvers between the intermediate waypoints. The sequence of waypoints is obtained applying the A\* algorithm to an undirected graph, whose nodes correspond to MRP sets that represent constraint compliance attitudes for the spacecraft. Such grid is obtained sampling the unit sphere centered at the origin in MRP space, which is a subset of the  $\mathbf{R}^3$  MRP space that, however, contains all the possible attitudes for a spacecraft that rotates in  $SO(3)$  [56]. The

unit sphere is sampled following the procedure outlined in Chapter 6. The cost function used by the A\* algorithm, for this planner, is the distance metric presented in Equation (6.2). The time-spacing between waypoints is proportional to such metric. For this planner, the time-tagged waypoints contained in the text file processed by the `waypointReference()` module are relatively sparse: with respect to Figure 6.1, the grid density corresponds to a level  $N = 13$ . In conclusion, this planner tries to target a sequence of constraint-compliant waypoints, in a rest-to-rest style, since the angular rates and accelerations contained in the reference message are always zeroed.

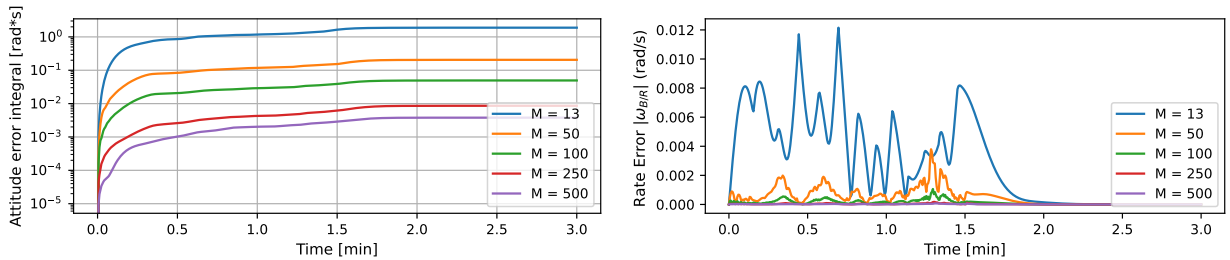
- **Planner #2:**

Planner #2 improves on the results of Planner #1. The same sequence of waypoints is used as by Planner #1. However, the constraint-compliant waypoints are interpolated in MRP space to obtain a smooth trajectory as a twice-differentiable  $C^2$  NURBS curve. Angular rates and accelerations are computed to ensure rest states (zero angular rates) at the endpoints and a constant angular rate norm of  $\|\boldsymbol{\omega}_{\mathcal{R}/\mathcal{N}}\| = 0.04$  rad/s along the trajectory. The angular rate norm is ramped up and down smoothly from the zero initial and final condition to the desired constant angular rate norm. The output text file, for this planner, contains a list of time tagged attitude waypoints, angular rates and accelerations that are sampled from the NURBS trajectory. Since such trajectory is a continuous function of time, the density of the waypoints in the output text file can be chosen arbitrarily, as the waypoints and associated rates and accelerations can be evaluated at any time. The denser the sampling along the trajectory, the smaller the approximation error introduced by the linear interpolation in Equation (7.33) introduced by the `waypointReference()` module. As validation, Figure 7.10 shows the attitude and rate tracking errors for different numbers of waypoints sampled from the interpolated trajectory, where the smallest,  $M = 13$ , corresponds to the same number of waypoints in the solution provided by Planner #0. It is clear how, for an increasing number of samples, the attitude error significantly diminishes

until becoming negligible. Therefore, with a time-continuous reference trajectory, it makes sense to always sample the waypoints with a high density. For Planner #2, 500 waypoints are sampled from the interpolated trajectory and processed by the `waypointReference()` module.

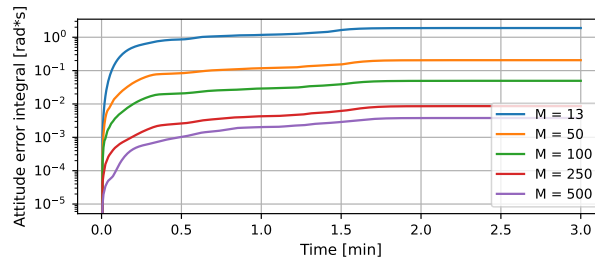
- **Planner #3:**

Planner #3 applies an effort-minimizing A\* algorithm to solve the path-planning query described in Chapter 6 using a cost function that yields the path of minimum integral torque according to Equation (7.32). Such algorithm, and other variants, will be better described in Chapter 8 of this dissertation. The expectation is, therefore, that Planner #3 outperforms Planner #2 with respect to the control torque integral performance metric. The output for Planner #3 is therefore, as for Planner #2, a smooth trajectory that has attitude, angular rates and accelerations as functions of time, with zero angular rates at the endpoints and a constant angular rate norm of  $\|\omega_{\mathcal{R}/\mathcal{N}}\| = 0.04$  rad/s along the trajectory. Once again,



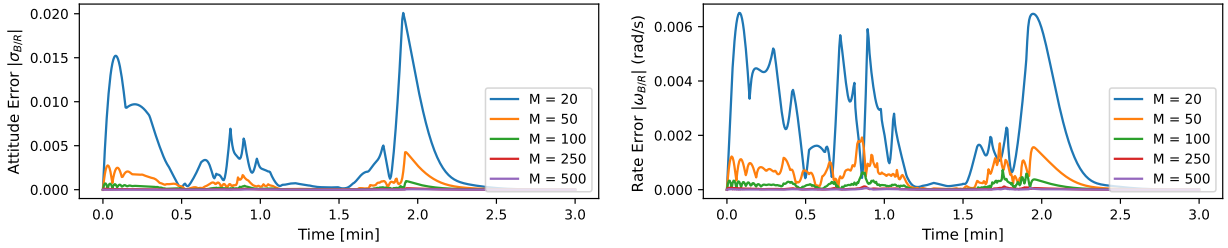
(a) Attitude Error Norm

(b) Angular Rate Error Norm



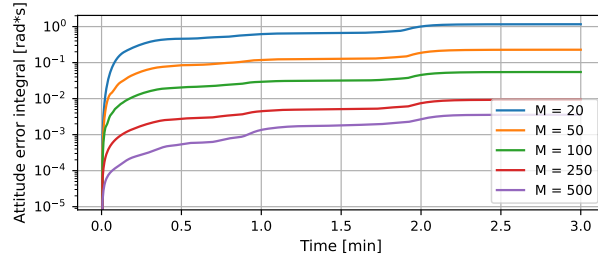
(c) Attitude Error Integral

Figure 7.10: Tracking error for Planner #2 for different density of points  $M$  sampled from the interpolated trajectory



(a) Attitude Error Norm

(b) Angular Rate Error Norm



(c) Attitude Error Integral

Figure 7.11: Tracking error for Planner #3 for different density of points  $M$  sampled from the interpolated trajectory

the text file produced by the planner and processed by the `waypointReference()` module contains 500 time-tagged attitude waypoints, angular rates and accelerations sampled from the effort-optimal interpolated trajectory. Figure 7.11 shows the same study, regarding waypoint sampling density from the interpolated trajectory, as for Planner #2, and the same considerations apply.

## 7.5.4 Benchmark Analysis

### 7.5.4.1 Planner Performance Comparison

This section shows the performance of the different planners based on a set of common evaluation criteria and the performance metrics described above. The scenario presented here features a slew maneuver between the attitudes  $\sigma_{\mathcal{R}/\mathcal{N}_i} = \{0.314, -0.251, 0.228\}^T$  and  $\sigma_{\mathcal{R}/\mathcal{N}_f} = \{0.326, -0.206, -0.823\}^T$ . The inertial position of the Sun is obtained from the SPICE database for the date January 15, 2021, at 00:30:30 UTC, which gives  $\mathcal{N}_s = \{0.419, -0.833, -0.361\}^T$ . The

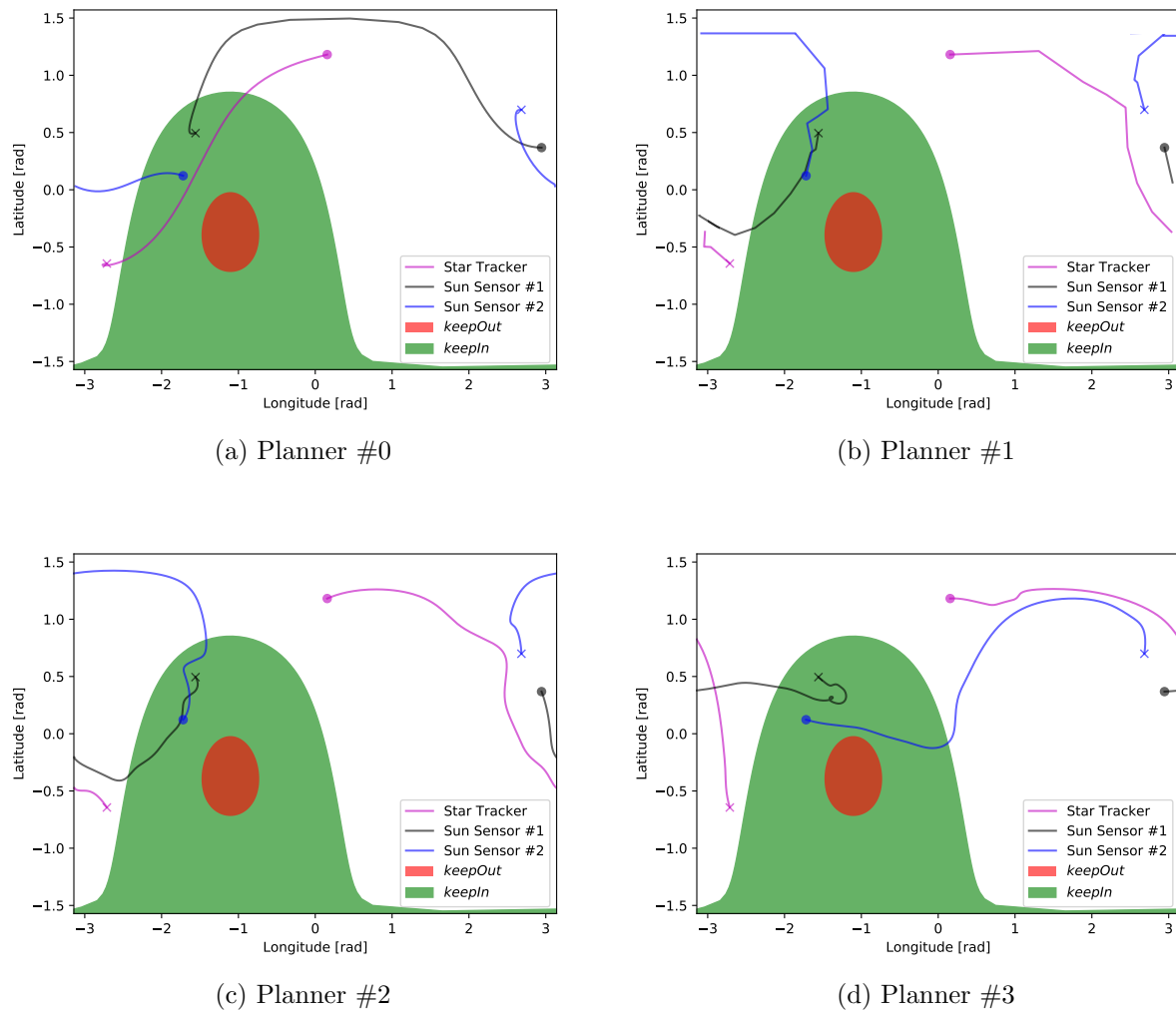


Figure 7.12: 2D plots of inertial boresight directions; ‘o’: starting point, ‘x’ endpoint.

spacecraft is assumed to be orbiting the Earth, in a position along its orbit where the Earth does not cause an eclipse.

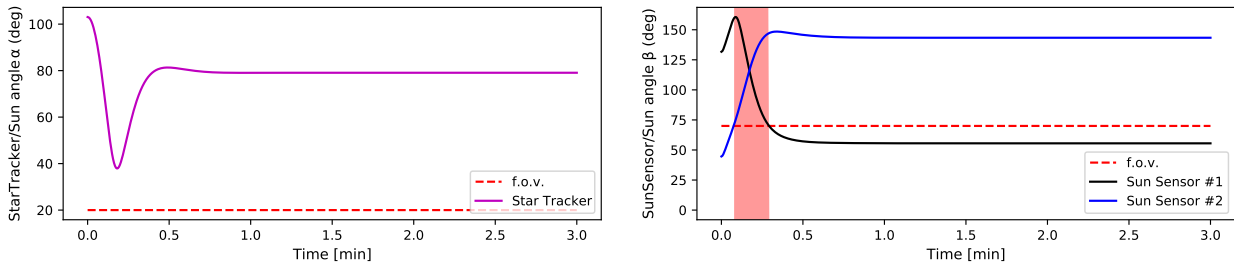
All the simulations run in this subsection use the gains  $K = 6 \cdot 10^{-3}$  N and  $P = 1.256 \cdot 10^{-2}$  Ns in Equation (2.49) to compute the commanded torque to the spacecraft. Such gains are chosen primarily for Planner #0, to ensure a near-to-critical response of the system which would converge to the desired target in about one minute. Figure 7.12 shows the projection on the 2D latitude-longitude plane of the boresight directions in inertial space, with respect to the keep-out

constraint (in red) and the keep-in constraint (in green). Figure 7.12a shows the actual trajectory of the boresights when the Planner #0 is used, whereas Figures 7.12b to 7.12d show the boresight directions for the reference waypoints  $\sigma_{\mathcal{R}/\mathcal{N}_j}$  provided to the `attTrackingError()` module. For Planner #0, this would correspond to just the initial and final reference attitudes. For Planner #1, the discrete reference waypoints produce a sequence of discrete target boresight inertial directions. Lastly, for Planners #2 and #3, full reference trajectories are obtained for the boresight directions as functions of time.

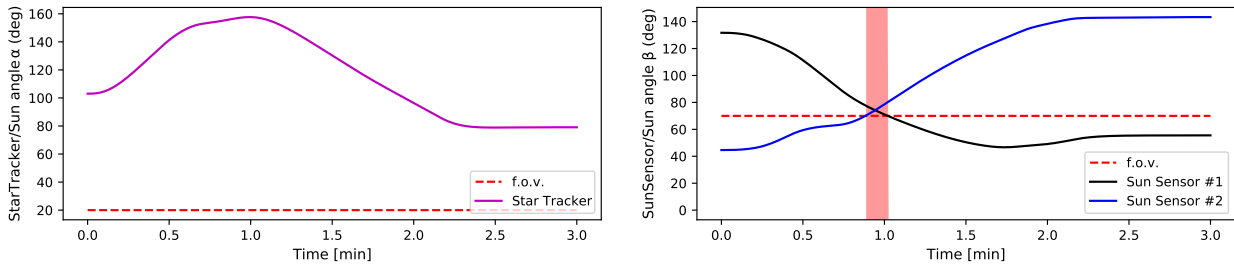
Figure 7.12 shows that the keep-out constraint is respected with all four planners. As far as the keep-in constraint, it is possible to see that Sensor #2 sees the Sun in the initial attitude, whereas Sensor #1 sees the Sun once the target attitude is reached. Figure 7.13 shows the angle between the Star Tracker and the Sun and the angles between the Sun Sensors and the Sun, together with the respective fields of view (f.o.v.) for each instrument. With planners #0, #1 and #2 the keep-in constraint is violated for a certain amount of time. This happens when the two Sun Sensors ‘exchange’ roles. Leaving aside Planner #0, which is constraint-naive, what happens for Planners #1 and #2 is more interesting. For Planner #1, a sequence of constraint-compliant waypoints is provided. However, the path that connects such waypoints is not constraint compliant in all its parts, and this is evident in the keep-in constraint violation. A similar phenomenon occurs with Planner #2, where the interpolated trajectory uses the same reference waypoints as Planner #1: although the interpolated trajectory violates the constraint for a shorter time, it still does, since the interpolating function used maintains the trajectory within the convex hull described by the interpolated attitude waypoints [86]. Planner #3, on the other hand, does not violate any of the constraints. This is not due to a refined sampling of the attitude space, but rather to the fact that the different nature of the cost function used by Planner #3 makes it converge to a trajectory that stays farther away from the boundary of the constraint-compliant space, thus avoiding the issue described for the previous two planners.

Figure 7.14 shows the performance metrics described above, and offers a direct comparison between the four planners. Figures 7.14a and 7.14b show the constraint violation times, where the

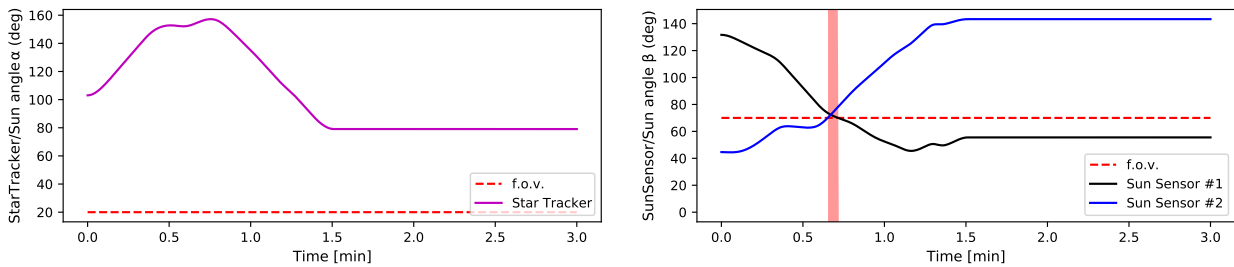




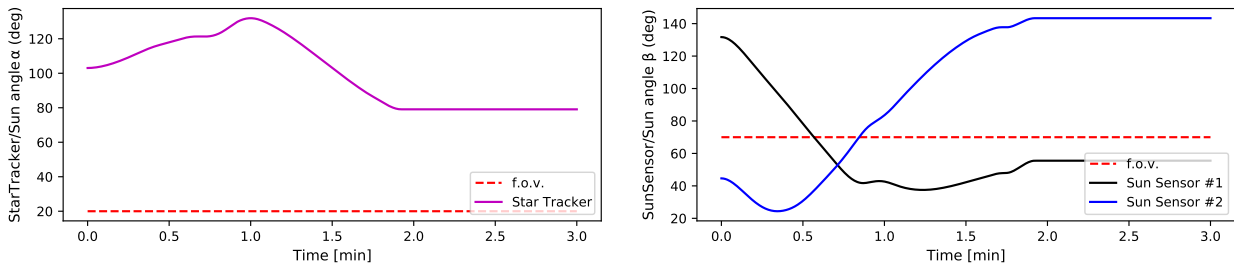
(a) Planner #0



(b) Planner #1



(c) Planner #2



(d) Planner #3

Figure 7.13: Angles between Star Tracker and Sun and between Sun Sensors and Sun

same information can be observed as in Figure 7.12, with more details on how long the constraint violations last for each planner. Figures 7.14c to 7.14e offer more insights on the performance of

the different planners other than just constraint compliance. Figure 7.14c shows that Planner #3 outperforms Planner #2 in terms of required commanded torque, as expected. Given the reaction wheel configuration, with one wheel along each principal body axis, this property transfers also to the Figure 7.14d, where Planner #3 is shown to outperform Planner #2 also in terms of energy consumption: this is due to the fact that each torque component is mapped directly to the corresponding reaction wheel. It is interesting to observe, however, how Planner #1 outperforms both Planners #2 and #3 in terms of total commanded torque and energy consumption. This was unexpected, because Planner #1 does not try to optimize for torque and/or power requirements. This unexpected behavior can be explained looking at Figure 7.14e, a gap of 3 orders of magnitude is observed between the attitude error integrals of Planners #0 and #1, and #2 and #3. As explained above, Planners #2 and #3 feed to the `attTrackingError()` module a time-dependent reference trajectory along with the required reference angular rates and accelerations: this allows the `mrpTracking()` module to accurately track the full desired state of the spacecraft along such reference trajectory. In contrast, Planner #1 only provides target attitude waypoints, therefore the `mrpTracking()` module tries to constantly steer the spacecraft towards the next attitude waypoint with zeroed final angular velocity. However, the target waypoint changes faster than the actuators can track, thus causing the spacecraft to be constantly chasing a moving target, until such target eventually settles at the final target attitude. This inefficient guidance strategy causes the attitude errors along the trajectory to be comparatively large, potentially resulting in constraint violations even when a constraint-compliant sequence of waypoints is used. On the other hand, Planners #2 and #3 are based on a nominal reference trajectory that interpolates the waypoints. In such cases, the interpolating spline can present wiggles between the waypoints. Such wiggles can appear as a consequence of having the spline curve pass through the waypoints precisely. This phenomenon becomes more significant as the waypoints are denser (denser grid). Since the reference trajectory is tracked accurately by the spacecraft, the torque integral is ultimately larger due to the effort required to track such undesired wiggles between the waypoints presented by the reference trajectory. In contrast, when Planner #1 is used, the simulation automatically smooths the path provided by

the waypoints due to its looser tracking capabilities, allowing for a smoother torque profile that ultimately results in a smaller torque integral and energy consumption.

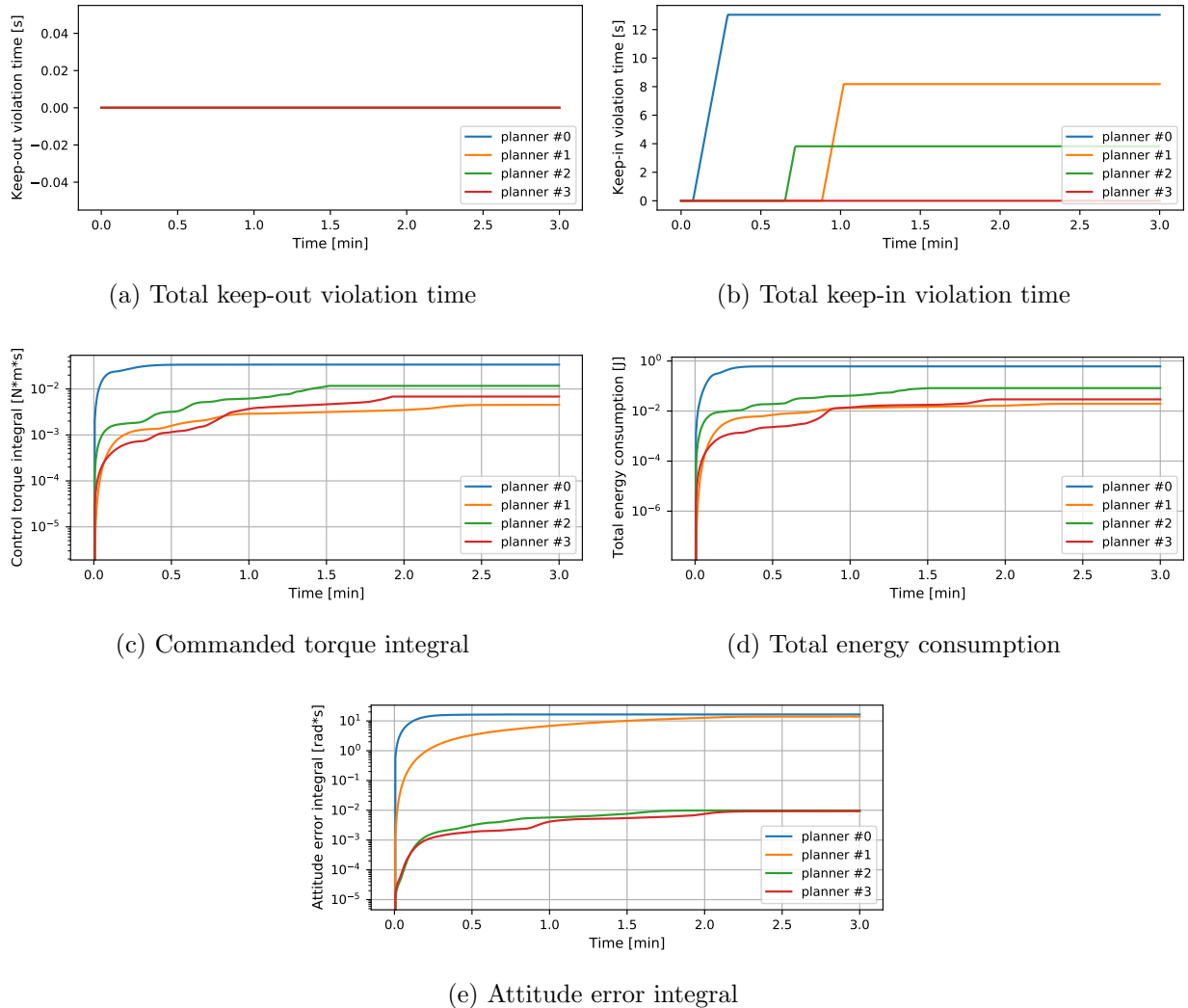


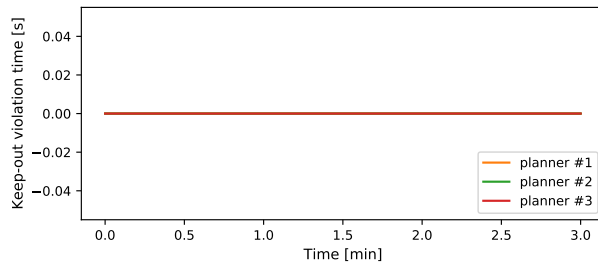
Figure 7.14: Performance metrics of the four planners compared

#### 7.5.4.2 Gain sensitivity

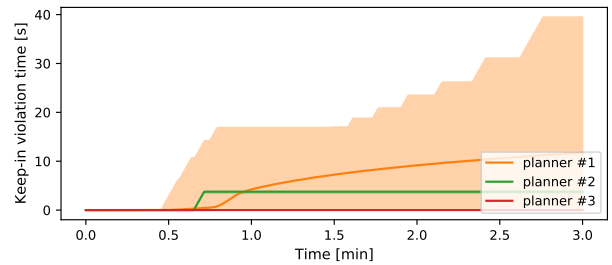
Not only do the planners have different performances based on the metrics above, they also have different levels of sensitivity to the gains  $K$  and  $P$  used in the nonlinear control law in the `mrpFeedback()` module. In the previous subsection the gains were chosen according to the performance of Planner #0. In the following simulations both gains are varied according to an

exponential distribution between  $K \in [6 \cdot 10^{-4}, 6 \cdot 10^{-2}]$  and  $P \in [1.256 \cdot 10^{-3}, 1.256 \cdot 10^{-1}]$ . This is done to highlight the performance of the different planners across a range of gains that spans from one order of magnitude lower to one order of magnitude higher than the values previously tested. With Planner #0 being constraint naive, the torque in Equation (2.49) is computed using the attitude error  $\sigma_{\mathcal{B}/\mathcal{R}}$  computed with respect to the target attitude. This, multiplied by the gain  $K$ , can result in a very high initial commanded torque when  $K$  is increased, which causes the spacecraft to overshoot the target. This, added to the fact that Planner #0 is different in nature to the other planners because constraint naive, makes it not interesting for the following gain sensitivity analysis, and it is therefore removed from it.

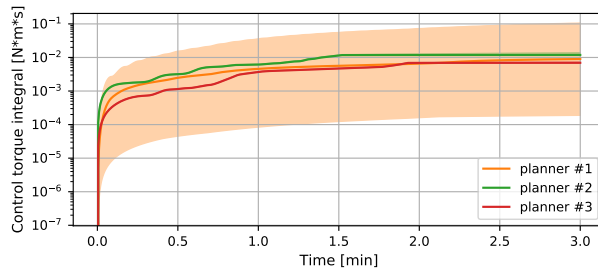
Figure 7.15 summarizes the results obtained with varying gains. Solid lines represent the averaged curves for each planner, whereas the colored shaded regions represent the bounds between best and worst case scenarios. First of all, Figure 7.15a shows that the keep-out constraint is never violated, whereas Figure 7.15b shows that Planner #1 can violate the keep-in constraint for very different intervals of time, and also not violate the constraint at all for the right choice of gains. More interestingly, Planners #2 and #3 show the same consistent behavior regardless of the gains: this is emphasized in Figures 7.15c and 7.15d where the upper and lower confidence bounds also coincide with the averaged curve. The same cannot be said about Planner #1, which is much more susceptible to gain changes. This analysis shows the robustness of the interpolated reference trajectories used by Planners #2 and #3 to gain tuning: having a well-defined reference makes the open-loop system track such reference well enough, to the point that the feedback terms  $K\sigma_{\mathcal{B}/\mathcal{R}}$  and  $P\omega_{\mathcal{B}/\mathcal{R}}$  in Equation (2.49) become irrelevant in the computation of the commanded torque. Lastly, Figure 7.15e shows the sensitivity of the integral tracking error to gains. In this case, Planners #2 and #3 do show appreciable variations due to gain selection. Nonetheless, within the gain bounds considered, the worst integral tracking error with Planners #2 and #3 is still one order of magnitude smaller than the best integral error with Planner #1.



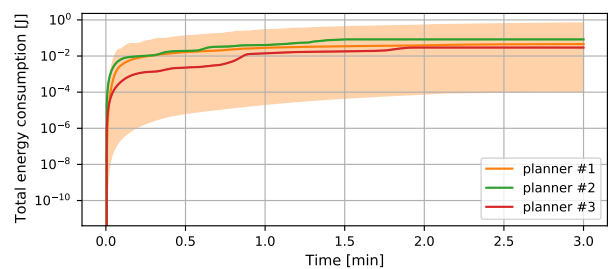
(a) Total keep-out violation time



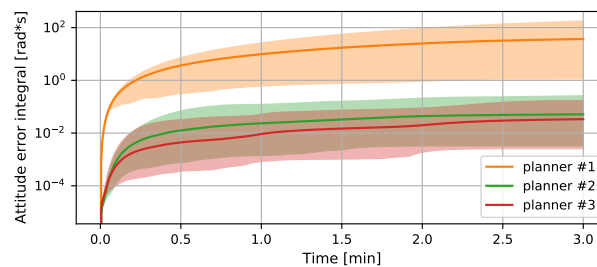
(b) Total keep-in violation time



(c) Commanded torque integral



(d) Total energy consumption



(e) Attitude error integral

Figure 7.15: Sensitivity of Planners #1 - #3 to gain variations

### 7.5.5 Monte Carlo Analysis

To conclude the discussion, the performance of the NURBS-based reference trajectory is tested in a batch of 500 Monte Carlo runs. The goal is to prove the robustness of the approach even in the presence of inaccurate initial conditions provided to the guidance algorithm.

The following analysis considers the scenario described in the previous subsections, where the reference provided by Planner #3 is used as a reference trajectory. The boundary conditions of the flight software remain unchanged, however the initial attitude and angular rates of the spacecraft

hub are dispersed uniformly with respect to the nominal initial values. The initial attitude is dispersed within a cube of side  $\Delta\sigma = 0.15$  centered at the initial nominal attitude. The initial angular rate is dispersed within a cube of side  $\Delta\omega = 0.015$  rad/s centered at the initial nominal angular rate. The gains in the control law are  $K = 6.7 \cdot 10^{-3}$  Nm and  $P = 2.5 \cdot 10^{-2}$  Nms.

The results of the MC runs are summarized in Figures 7.16 to 7.18. These figures prove the expected behavior, where the initial conditions are distributed according to the dispersions presented above. However, due to the asymptotically stabilizing nature of the control equation in Equation (2.49) and the kinematic-compliance of the NURBS-based reference trajectory, all trajectories are shown to converge to the reference eventually.

Performance depends on the control gains, which must be tuned in order to get a faster convergence to the reference trajectory. The problem here is that, with wrong initial conditions,

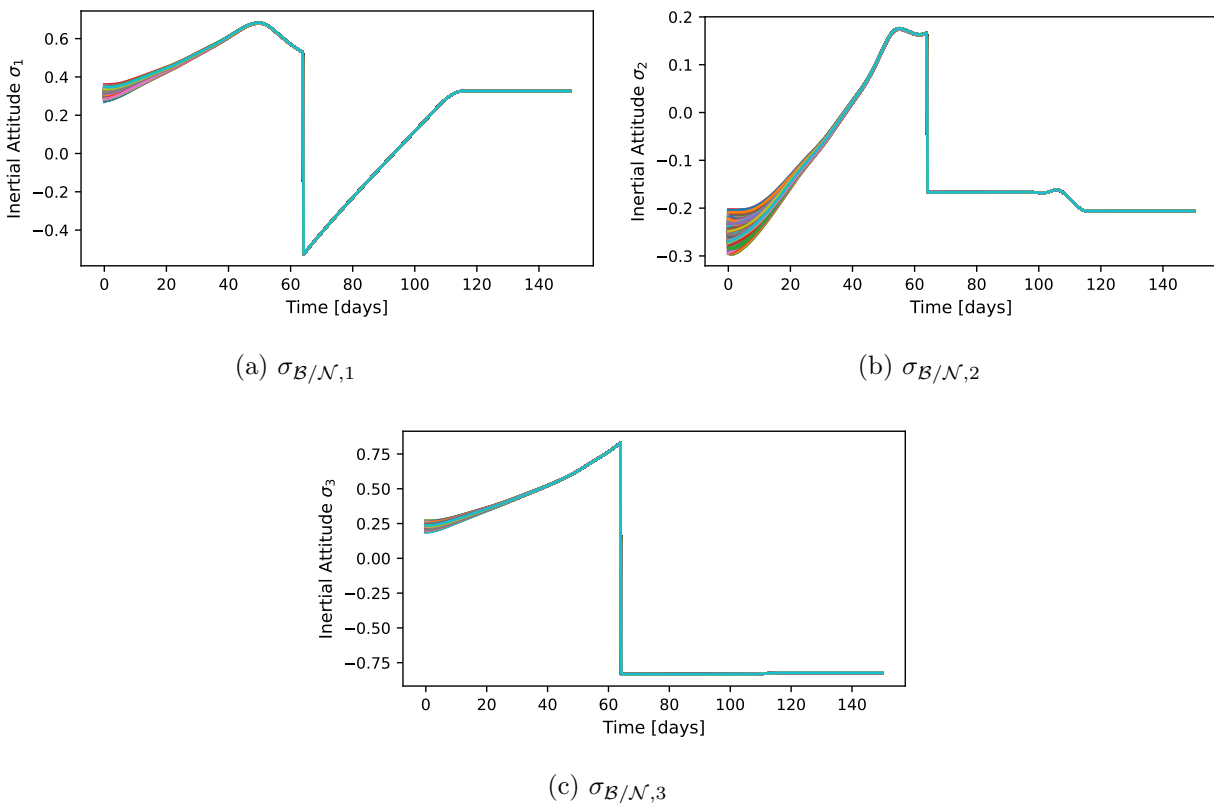


Figure 7.16: Inertial attitude in the MC runs

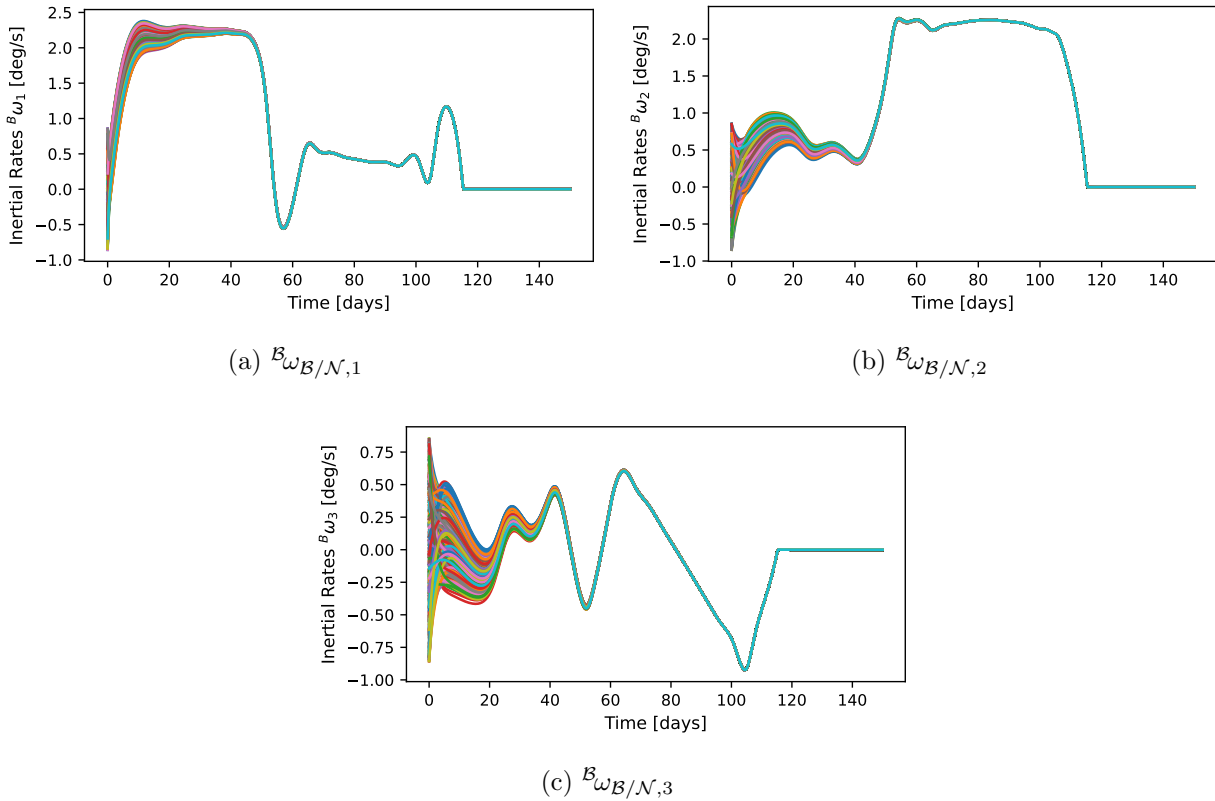


Figure 7.17: Angular rates in the MC runs

the actual spacecraft slew could violate a constraint during the transient phase before converging to the reference trajectory, even though such trajectory is itself constraint-compliant. Figure 7.18, however, shows a monotonic decrease in the PRA error with respect to the reference. Therefore, an estimate on the accuracy of the initial attitude can be used to inform the path planning algorithm in order to artificially enlarge keep-out zones. This would ensure that the sensitive boresight does not violate the keep-out zone even during a transient phase in the slew maneuver.

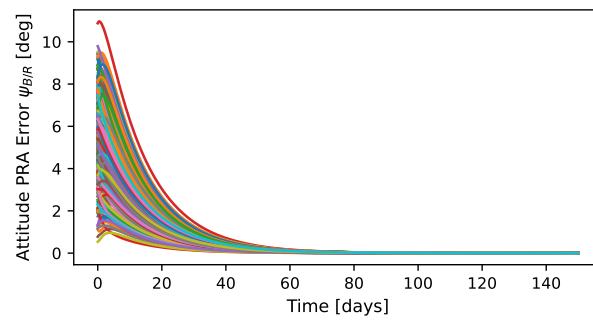


Figure 7.18: Attitude error w.r.t. reference trajectory in MC runs



## Effort-based Constrained Attitude Maneuvering

This final chapter combines the results presented in the previous two chapters to deliver a more comprehensive solution to the constrained attitude maneuvering problem. This chapter relies on the same attitude space discretization outlined in Chapter 6 to build a graph, but a variation of the A\* algorithm is applied to search such graph. The path smoothing techniques outlined in Chapter 7 are here incorporated into an effort-based A\* algorithm to produce a graph-search method that computes a continuous, constraint-compliant, kinematically feasible reference trajectory which is also “optimal” with respect to a cost function that describes the overall effort required to track that trajectory. Note that the word “optimal” is intended in a loose sense: no optimal control is performed in this chapter. However, the solutions computed are shown to be very close to optimal within the context of the assumptions made in this and the previous two chapters, when compared to other trajectories computed under the same constraints.

The purpose of this chapter is, therefore, to combine the elements of novelty outlined in the previous two chapters into an attitude path planning algorithm that is capable of computing a robust, kinematically feasible reference trajectory while respecting pointing constraints and kinematic/dynamic constraints, and while also remaining scalable and fast to execute.

Enhancing the pathfinding algorithm with more refined dynamical models comes at the expenses of computational speed, and stretches certain assumptions that are required to ensure its efficiency and effectiveness. Nonetheless, this contribution is not tied to one specific path planning algorithm such as A\*, considered here. The nonsingular MRP space discretization and path smooth-

ing via twice-differentiable, malleable NURBS curves, can be applied to other search methods that prove to be more suitable.

### 8.1 Effort-based A\* graph search

In Chapter 6, the constraint-compliant paths are found by the A\* algorithm, using a cost function based on the PRA distance metric expressed by Equation (6.2). Given the optimal guarantees of the A\* algorithm, the resulting path is also the shortest in terms of total angular displacement. From a dynamic perspective, however, this approach is not very meaningful, as it does not account for the dynamic state of the spacecraft, nor for its inertia distribution. For example, if the initial angular velocity of the spacecraft were directed in the opposite direction with respect to the minimum displacement path, tracking that trajectory would potentially require a large initial torque to steer the spacecraft. Moreover, more than one path can exist with the same length  $S$ , but they might not be equally feasible to track, depending on the required control torque, which itself depends on the inertia tensor. For all the above, this chapter aims to implement a modified version of A\* that searches the graph for a path that is optimal in terms of required open-loop control effort. This presents some challenges, since the “optimality” of such path, or cost, must be computed in terms of a scalar quantity that can be compared with the cost of other paths. Such cost can be defined as the integral of the torque magnitude over the slew maneuver:

$$C_0(p) = \int_0^T \|\mathbf{L}(t)\| dt \quad (8.1)$$

where  $\mathbf{L}(t)$  is the torque required to actuate a rigid body as it evolves along the trajectory with in  $t$ . The trajectory is assumed to be the fitting curve for a series of constraint-compliant attitude waypoints, either in the form of perfect interpolation or LS fit. The cost function described in Equation (8.1) is meant to be reasonable for a general application, but is not tied to any specific set of actuators. In a more refined analysis, the actuators’ equations of motion can be integrated based on the interpolated trajectory to yield the required reaction wheel control torques as functions of time. The cost function can therefore be modelled to minimize the reaction wheel torque or,

alternatively, the power required to actuate them.

As described in Chapter 6, the A\* algorithm chooses which nodes to expand based on the sum of the traveled path to current node ( $g(n)$ ) and a heuristic cost from current node to goal ( $h(n)$ ). Moreover, A\* is efficient at finding the best path when the heuristic is optimistic, meaning that  $h(n)$  should be less or equal than the actual cost of the path from node  $n$  to goal.

Using the cost function described in Equation (8.1) makes it challenging to compute  $g(n)$  and  $h(n)$ . One option could be to interpolate two paths, one going from the start node to  $n$ :  $p_{[0,n]} = [\sigma_0, \dots, \sigma_n]$ , and the other one from node  $n$  to the goal node:  $p_{[n,q]} = [\sigma_n, \sigma_q]$ . However, the required  $\dot{\sigma}_n$  at the intermediate node  $n$  is not known, therefore the two interpolated trajectories would have mismatching derivatives at  $\sigma_n$ . This phenomenon could give an incorrect representation of the total path cost that would otherwise be obtained interpolating a full path from  $\sigma_0$  to  $\sigma_q$  with a single, smooth trajectory. To overcome this problem, the total cost of a node  $f(n) = g(n) + h(n)$  is computed in one single step, applying Equation (8.1) to the path:

$$p_{[0,n,q]} = [\sigma_0, \dots, \sigma_n, \sigma_q]. \quad (8.2)$$

For such path, the integral between  $\sigma_0$  and  $\sigma_n$  corresponds to  $g(n)$ , whereas the integral between  $\sigma_n$  and  $\sigma_q$  corresponds to the heuristic  $h(n)$ :

$$g(n) = \int_{t(\sigma_0)}^{t(\sigma_n)} \|\mathbf{L}(t)\| dt \quad (8.3)$$

$$h(n) = \int_{t(\sigma_n)}^{t(\sigma_q)} \|\mathbf{L}(t)\| dt \quad (8.4)$$

$$C_0(p_{[0,n,q]}) = f(n) = g(n) + h(n) = \int_{t(\sigma_0)}^{t(\sigma_q)} \|\mathbf{L}(t)\| dt \quad (8.5)$$

Such heuristic is optimistic (or admissible [41]) in the sense that it assumes that it is possible to go directly from node  $n$  to  $n_q$ , neglecting the presence of obstacles along the way. There are two arguments can be provided to support that the heuristic is admissible. The first one is that, in the presence of an obstacle, the spacecraft would have to steer around it, which would require additional torquing effort. Secondly, it should be noted that the grid is an artificial construct. Therefore, forcing the trajectory to hit precisely all the waypoints of a path also requires additional

torquing effort. The  $h(n)$  defined above is based on the idea of “taking a shortcut” between the current node and goal node, thus avoiding unnecessary steering. Section 7.5 shows how, when the requirement of hitting all the waypoints precisely is lifted, the resulting control torque integral is reduced. Lastly, Equation (8.2) ensures that the cost is computed from a trajectory that matches the requisites of continuity and differentiability described in the previous subsections.

When computing the cost of such path, unless node  $n$  and goal node are neighbors, the distance  $\hat{d}(\sigma_n, \sigma_q)$  can be higher than the average distance between neighboring nodes. When this mismatch in the spacing between nodes is significant, it can generate instabilities in the interpolating function, causing the spacecraft to overshoot the target. To circumvent this problem, when  $\hat{d}(\sigma_n, \sigma_{\text{goal}}) > \frac{\sqrt{3}}{(N-1)}$ , a number of equally spaced “guidance nodes” with mutual distance  $\approx \frac{1}{(N-1)}$  are artificially added between nodes  $n$  and  $n_q$  to stabilize the interpolation.

The effort-based A\* algorithm is run for the same scenario described in Section 6.3.2, with zero endpoint angular rates. The resulting interpolated path is reported in Figures 8.1 and 8.2. As intuition suggests, the path computed by the regular A\* in Figure 6.5 is not the preferred one in terms of smallest effort. The effort-optimal path steers the spacecraft around the yellow

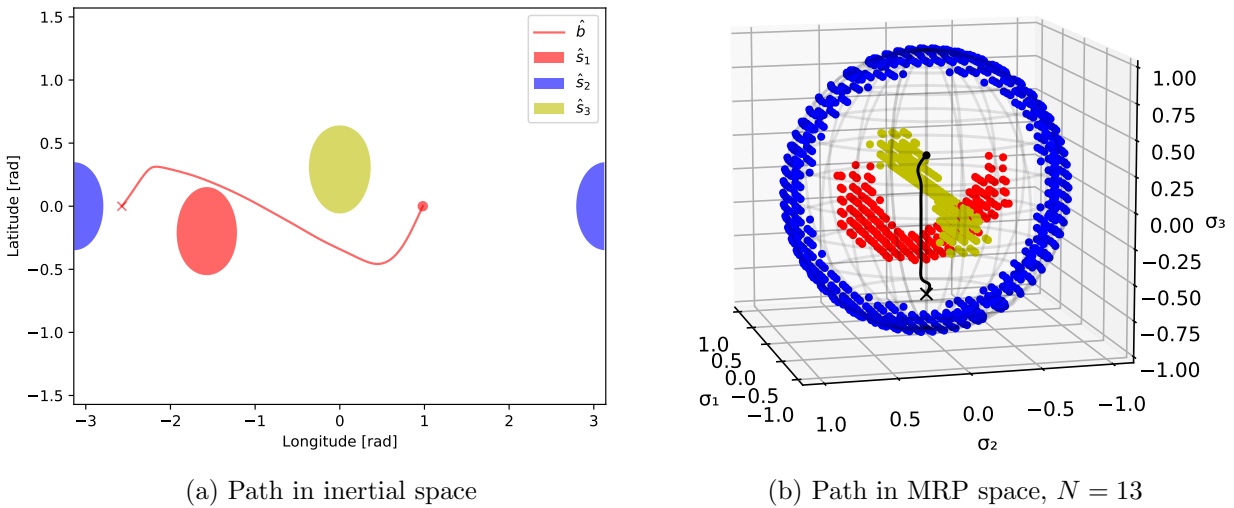


Figure 8.1: Interpolated effort-based A\* solution

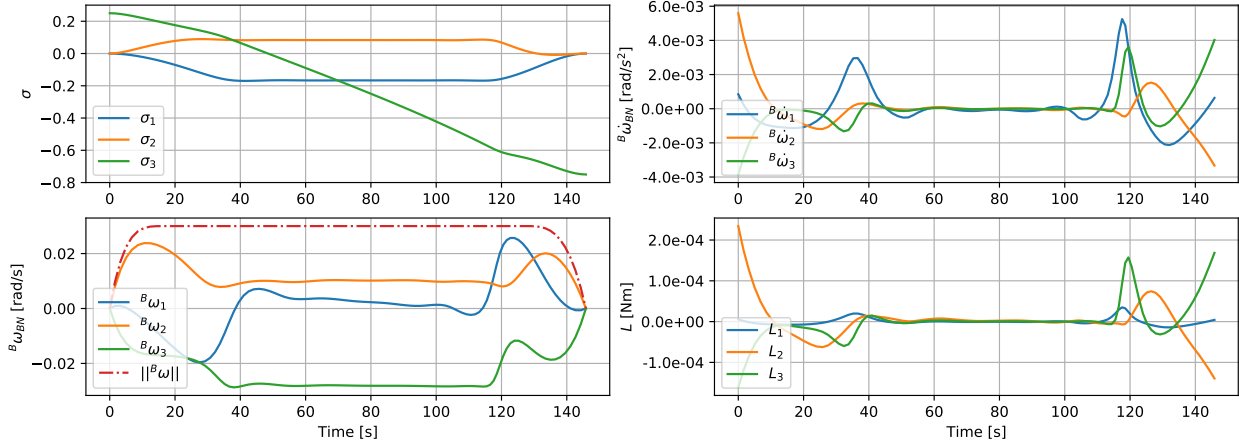


Figure 8.2: Attitude, rates, acceleration and torque over time for the trajectory in Figure 8.1

obstacle in the first part of the trajectory, to put the spacecraft onto a coasting arc between  $t \approx 50$  sec and  $t \approx 110$  sec, where the torques remain close to zero. Finally, after the quasi-coasting arc, the spacecraft is steered towards the target node. The total cost of the effort-optimal path is  $C_0(p_{\text{effort-optimal}}) = 5.53 \cdot 10^{-3}$  Nms, as opposed to the PRA-optimal path  $C_0(p_{\text{PRA-optimal}}) = 8.72 \cdot 10^{-3}$  Nms.

Note that the trajectory and plots in Figures 8.1 and 8.2 are obtained after additional manipulation, in which the resulting interpolating NURBS curve is time-warped in order to obtain a smooth, perfectly constant  $\|\omega\| = 0.03$  rad/s. The routine to perform this time warping is presented in Appendix B.

## 8.2 NURBS Comparison in Effort-based Graph Search Algorithm

This section offers a comparison of the interpolating and LS fitting NURBS curves presented in Chapter 7 when applied to the effort-based A\* algorithm outlined in the previous section. As this version of the A\* algorithm searches the graph, it uses intermediate paths consisting of the previously explored nodes  $[\sigma_0, \dots, \sigma_{n-1}]$ , the current open node  $\sigma_n$  and the goal node  $\sigma_q$  as a baseline for an intermediate trajectory computed via NURBS curves. The control effort integral in Equation (8.1), evaluated along these intermediate trajectories, is used as the the priority function

$p(n)$  to explore the graph, with the objective of finding the sequence of waypoints that yields the optimal trajectory in terms of required control effort.

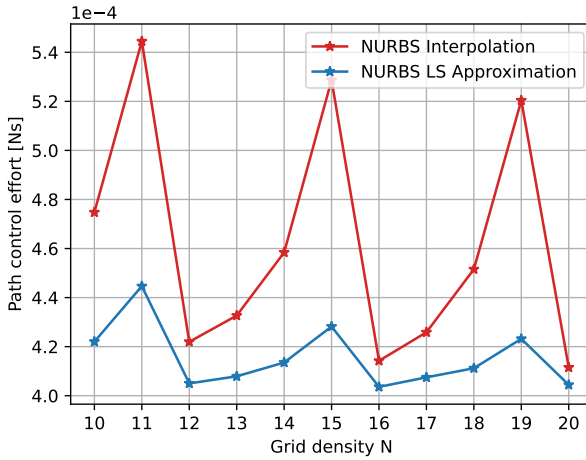
The effort-based A\* is run for the three scenarios presented in Section 7.4, using both the interpolating NURBS and the LS approximating NURBS.

Figure 8.3a shows that the control effort of the effort-optimal solutions, for both types of NURBS, match the results shown in Figure 7.6c. This is expected, because for the eigenaxis rotation in Scenario 1, the control strategy defaults to a bang-bang type with an effortless coasting arc in the middle. Any other trajectory that deviates from that would likely be more costly in terms of control effort, due to the insurgence of gyroscopic terms and torque components along the other two axes.

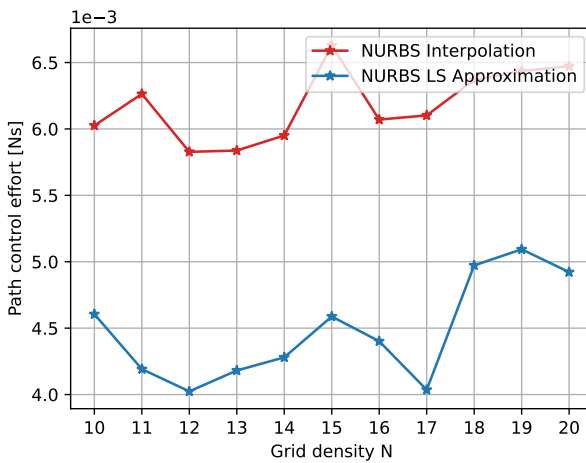
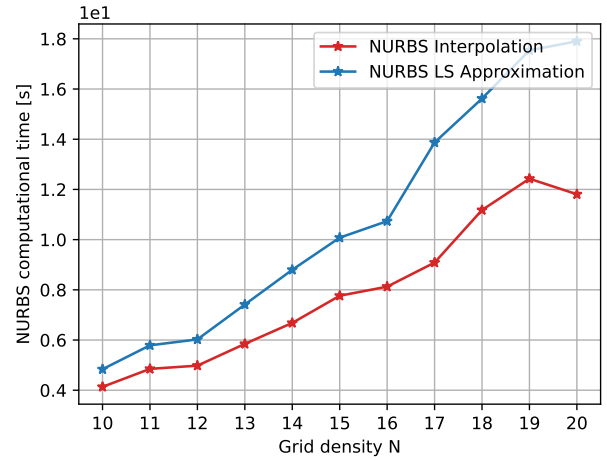
Figures 8.3b and 8.3c show that the effort-based solutions, for both NURBS curves, provide better trajectory in terms of control effort than those in the previous section, as expected. While the results improve for both types of NURBS, the LS approximating curve consistently outperforms the interpolating curve, always yielding less costly solutions.

The effort-based A\* algorithm is implemented in Python and run on a Macbook Pro with an M1 Pro chip. These plots show how computationally demanding the effort-based algorithm is, where for high grid densities it can take up to 15-20 minutes to compute the optimal solution. However, the code used in this work is not implemented to optimize for runtime. Interestingly, Figures 8.3b and 8.3c show that the computational times for the two different NURBS are comparable. This is not intuitive, if looking at the results shown in Figures 7.6c, 7.7c and 7.8c where, when the path is pre-computed by a metric-based version of A\*, the LS approximating NURBS seems to consistently require a larger computational time than the interpolating NURBS. The reason why the effort-based A\* with LS approximation is not consistently more time consuming is that, in the effort-based implementation, the priority function  $C_0$  outputs a smaller, more “optimistic” estimate of the total path cost. This allows the effort-based A\* algorithm combined with the LS approach to converge to the final solution more efficiently [41], i.e., exploring a smaller number of nodes.

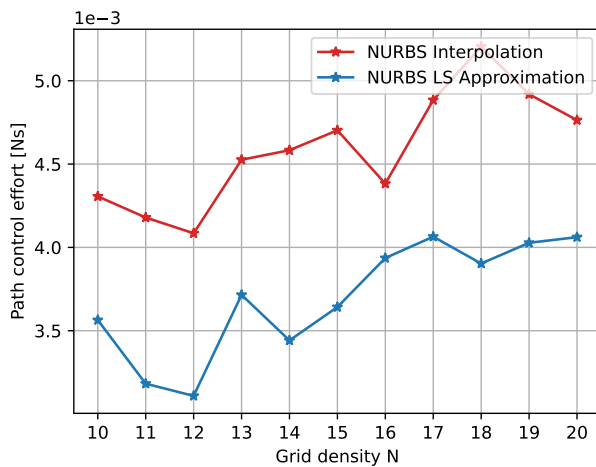
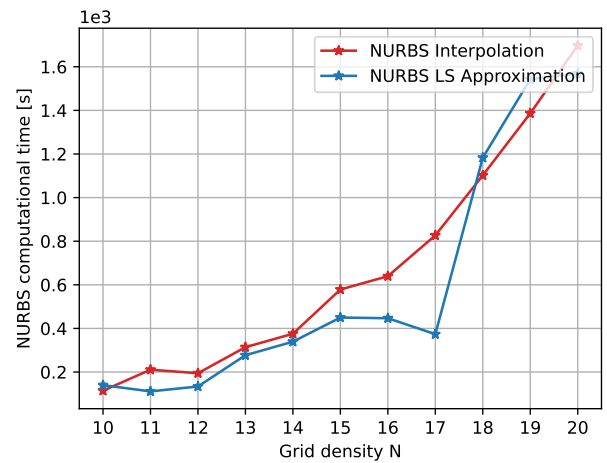
This benchmark analysis also showed, as it was expected, that increasing the grid density  $N$



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

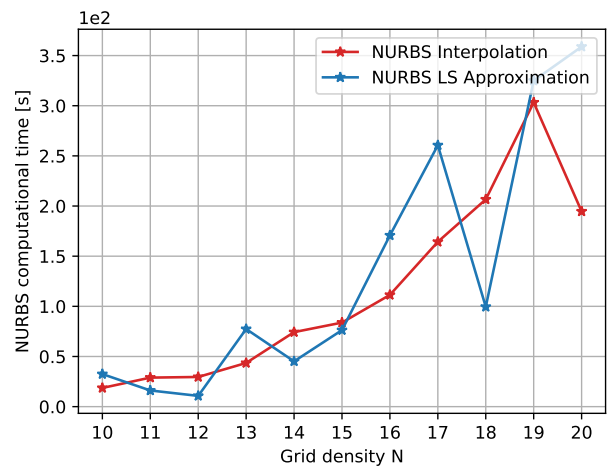


Figure 8.3: Control effort and computational time of effort-based A\*, NURBS comparison

yields longer computational times for the effort-based A\*. However, in the range  $10 \leq N \leq 15$ , such increase is still quite acceptable. The grid density  $N$  is, in this problem, a tradeoff parameter: increasing  $N$  gives a better representation of the attitude space and reduces the chances of violating the constraints. On the other hand, the computational time increases significantly and the computed trajectories, even the effort-optimal ones, are more costly in terms of control torque. It is important to state that these considerations are case specific. In a different scenario, with the same grid density, but where the obstacles occupy a very large portion of the attitude space, convergence is accelerated by the reduced number of compliant sampled waypoints. Ultimately, the appropriate grid size can be further characterized by exploring node density in correlation with the size of the constraint-incompliant attitude space.

### 8.3 Actuator Dynamics Incorporation

The previous sections assumed a rigid body when computing the effort required to track a given trajectory. This section expands on this assumption, incorporating reaction wheel dynamics into the cost function used by the effort-based A\* algorithm.

For a spacecraft with  $M$  RWs, the equations of motion are given by Equation (2.39). The assumption is made that all wheels have the same inertia  $I_{w_s}$  about the spin axis. The  $M$ -dimensional array containing the motor torques required to actuate each wheel is obtained combining Equations (2.39) and (2.42):

$$\mathbf{u}_w = -[G_s]^T ([G_s][G_s]^T)^{-1} ([I_{\text{tot}}]\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([I_{\text{tot}}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [G_s]\mathbf{h}_s)) \quad (8.6)$$

where the minimum-norm solution is computed, using the pseudo-inverse of  $[G_s]$ , in the presence of a redundant ( $M > 3$ ) reaction wheel set. With exactly three reaction wheels, the matrix  $[G_s]$  is invertible, assuming that the spin axes of the wheels are linearly independent. The motor torques obey the dynamics in Equation (2.41). Equation (8.6) allows to compute the wheel torques required to track the reference trajectory, for which the angular rates and accelerations are known at every time step through NURBS fit. Subsequently, inverting Equation (2.41) returns the array



of derivatives of the wheel speeds  $\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}$ . Knowing the wheel speeds at the initial time  $\boldsymbol{\Omega}(t_0)$ , it is possible to use the relative wheel angular accelerations to integrate the wheel speeds over time. A simple forward Euler integration method is:

$$\boldsymbol{\Omega}(t_{n+1}) = \boldsymbol{\Omega}(t_n) + (t_{n+1} - t_n)\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}}(t_n) \quad (8.7)$$

however, such simple approach can often yield numerically imprecise results. Specifically, when tracking a rest-to-rest maneuver with wheels initially at rest, it is observed that the Euler integration results in nonzero final wheel speeds, due to numerical errors. The final wheel speeds being nonzero means that the total angular momentum of the spacecraft system is not conserved, in the absence of an external torque. A more robust integration technique is obtained by means of a 4th-order Runge-Kutta (rk4) algorithm. For this method, let's define the function  $\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} = f(t, \boldsymbol{\Omega})$  as a combination of Equations (2.41) and (8.6):

$$f(t, \boldsymbol{\Omega}) = -\frac{[G_s]^T ([G_s][G_s]^T)^{-1}}{I_{w_s}} ([I_{\text{tot}}]\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([I_{\text{tot}}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [G_s]\mathbf{h}_s)) - [G_s]^T \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} \quad (8.8)$$

where the contribution of the wheel speeds  $\boldsymbol{\Omega}$  is included in the  $\mathbf{h}_s$  term, as per Equation (2.40). Knowing the wheel speeds at the instant  $\boldsymbol{\Omega}(t_n) = \boldsymbol{\Omega}_n$ , the integration proceeds calculating the four coefficients:

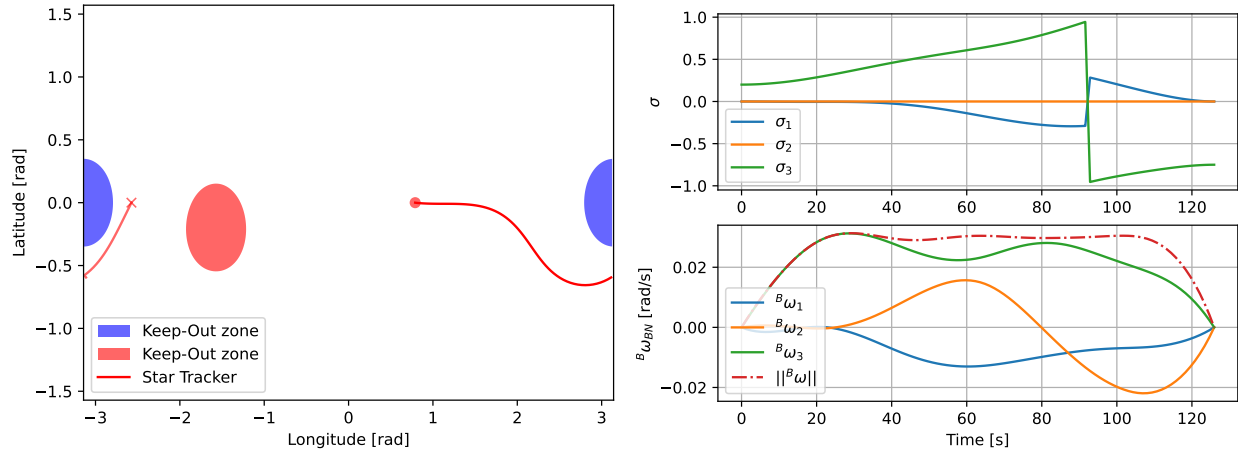
$$\begin{aligned} \mathbf{k}_1 &= f(t_n, \boldsymbol{\Omega}_n) \\ \mathbf{k}_2 &= f\left(t_n + \frac{t_{n+1} - t_n}{2}, \boldsymbol{\Omega}_n + \mathbf{k}_1 \frac{t_{n+1} - t_n}{2}\right) \\ \mathbf{k}_3 &= f\left(t_n + \frac{t_{n+1} - t_n}{2}, \boldsymbol{\Omega}_n + \mathbf{k}_2 \frac{t_{n+1} - t_n}{2}\right) \\ \mathbf{k}_4 &= f(t_{n+1}, \boldsymbol{\Omega}_n + \mathbf{k}_3(t_{n+1} - t_n)) \end{aligned} \quad (8.9)$$

from which the integrated wheels speeds at instant  $t_{n+1}$  are obtained as:

$$\boldsymbol{\Omega}(t_{n+1}) = \boldsymbol{\Omega}(t_n) + (t_{n+1} - t_n) \frac{\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4}{6}. \quad (8.10)$$

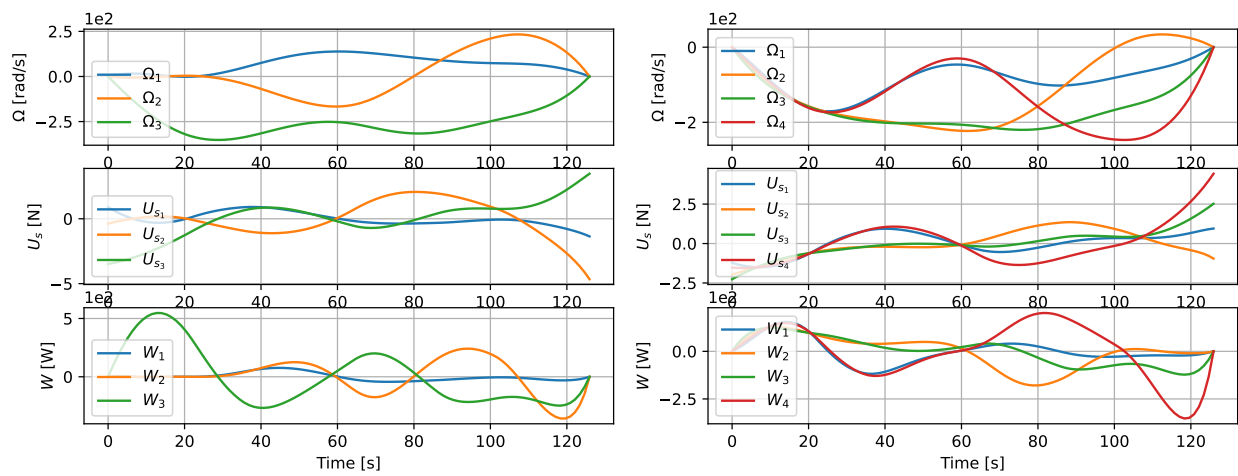
Once the equations of motion of the reaction wheels are integrated, it is possible to define the instantaneous power required by the motor torque to actuate the wheels:

$$W_i = u_{w_i} \Omega_i. \quad (8.11)$$



(a) Inertial trajectory of the Star Tracker and keep-out zones (b) Attitude and angular rates vs. time

Figure 8.4: Example trajectory computed with metric-based  $A^*$ ,  $N = 7$ .



(a)  $M = 3$  wheels (b)  $M = 4$  wheels

Figure 8.5: Wheel speeds, wheel control torques and power required to track the trajectory in Figure 8.4

In general,  $W_i(t)$  can be either positive or negative, where  $W_i > 0$  indicates that the wheel is acting as a power load, thus absorbing power from the spacecraft, while  $W_i < 0$  that the wheel is acting as a power source. This is only realistic for systems with regenerative wheels, where the power used to brake the wheel's speed can be harvested and transferred back to the spacecraft [91].

Figure 8.4 shows an example of a constraint-compliant reference trajectory computed using the PRA-based  $A^*$  outlined in Chapter 6 and subsequently applying a LS fit to obtain a continuous reference trajectory. Such trajectory is a rest-to-rest maneuver with an almost-constant angular rate norm of  $\|\boldsymbol{\omega}\| = 0.03$  rad/s, performed with a sensitive Star Tracker with a field of view of 20 degrees along the  $\hat{\mathbf{b}}_x$  body axis, and two inertially-fixed bright celestial objects. The scenario and the initial condition are the same as in Section 6.3.2, after removing the yellow keep-out zone. Figure 8.5 shows the dynamics of the reaction wheels used to actuate the spacecraft. In particular, Figure 8.5a is obtained using three RWs aligned with the principal inertia axes of the spacecraft, whereas Figure 8.5b is obtained assuming a redundant set of four reaction wheels, along the positive and negative  $\hat{\mathbf{b}}_x$  and  $\hat{\mathbf{b}}_y$  semiaxes, and with a tilt angle of 30 degrees towards the positive  $\hat{\mathbf{b}}_z$  semiaxis. In both cases, all RWs are actuated from an initial rest condition. As a proof of concept, it is possible to observe that all reaction wheel speeds converge to  $\boldsymbol{\Omega} = 0$  at the end of the maneuver: this is because the total angular momentum of the spacecraft and wheels system is zero. Since the final angular rate of the hub is zero, the wheel speeds also go to zero to ensure that the angular momentum is conserved. This is strictly true for the case with three reaction wheels. In the case with four reaction wheels, it could be possible to obtain nonzero final angular rates, for which the total net angular momentum of the system is still zero: this can happen using a different torque mapping than the one presented in Equation (8.6), due to the existence of a one-dimensional null-space for the reaction wheel set [58].

The final wheel speeds are still affected by integration errors; however, such integration errors are as small as  $10^{-4}$  rad/s.

### 8.3.1 Effort-based graph search

This subsection aims to develop new cost functions that can be incorporated into the effort-based  $A^*$  algorithm. The cost function in Equation (8.1) is, in fact, inaccurate for a spacecraft actuated by reaction wheels, because it is derived under the rigid body assumption, without factoring in the actuators' inertias and angular momentum contributions.

The control-based cost function is enhanced using the result in Equations (2.39) and (2.42):

$$\begin{aligned} C_1 &= \int_0^T \|\mathbf{L}\| dt = \int_0^T \|[G_s]\mathbf{u}_w\| dt \\ &= \int_0^T \|[I_{\text{tot}}]\dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} + \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \times ([I_{\text{tot}}]\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} + [G_s]\mathbf{h}_s)\| dt \end{aligned} \quad (8.12)$$

where the reaction wheels, their inertias, and the momentum build up on them are accounted for in the control torque  $\mathbf{L}$ .

The cost function can also be modeled after the amount of energy required to actuate the wheels, obtained as the integral of the motor power in Equation (8.11). In this case, it is important to define what the motor power describes in terms of “cost” to the spacecraft’s power system. With non-regenerative wheels, power is consumed by the motor torque to accelerate the wheels ( $W_i > 0$ ); when the wheels brake ( $W_i < 0$ ), the energy dissipated in the braking is lost. In this case, therefore, it makes sense to estimate the cost by integrating only over the time intervals in which the wheel powers are positive. For a set of  $M$  wheels, this can be done as [92]:

$$E = \sum_{i=1}^M \left[ \int_0^T \frac{1}{2} (W_i + |W_i|) dt \right]. \quad (8.13)$$

In a case with regenerative wheels, the energy released by the wheels during the braking phase can be harvested and given back to the spacecraft. This means that the cost function should account for positive contributions due to the energy used to accelerate the wheels, minus the energy that is given back when decelerating the wheels. Assuming to have a regeneration efficiency factor  $0 \leq \eta \leq 1$ , the energy-based cost function for a general regenerative wheel is computed as:

$$C_2 = \sum_{i=1}^M \left[ \int_0^T \left( \frac{1+\eta}{2} W_i + \frac{1-\eta}{2} |W_i| \right) dt \right], \quad (8.14)$$

where for  $\eta = 0$  the case with zero regenerative capability is obtained. Thanks to some energy principle considerations, it is possible to observe that the energy-based cost function  $C_2$  with perfect power regeneration is not suitable for all implementations. Specifically, with perfect power

regeneration, it is:

$$\begin{aligned}
C_2 &= \int_0^T I_{w_s} (\mathbf{u}_w \cdot \boldsymbol{\Omega}) dt \\
&= \int_0^T I_{w_s} \left( \dot{\boldsymbol{\Omega}} + [G_s]^T \dot{\boldsymbol{\omega}}_{\mathcal{B}/\mathcal{N}} \right) \cdot \boldsymbol{\Omega} dt = I_{w_s} \int_0^T \boldsymbol{\Omega} \cdot d\boldsymbol{\Omega} + I_{w_s} \int_0^T \boldsymbol{\Omega} \cdot [G_s]^T d\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}} \\
&= \frac{1}{2} I_{w_s} \boldsymbol{\Omega}(T) \cdot \boldsymbol{\Omega}(T) - \frac{1}{2} I_{w_s} \boldsymbol{\Omega}(0) \cdot \boldsymbol{\Omega}(0) + \boldsymbol{\Omega}(T) \cdot [G_s]^T \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}(T) - \boldsymbol{\Omega}(0) \cdot [G_s]^T \boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}(0).
\end{aligned} \tag{8.15}$$

For a rest-to-rest maneuver, the total kinetic energy of the reaction wheels is conserved for a non-redundant set of wheels, while it is also likely to be conserved with a redundant set, as shown in Figure 8.5, depending on how the torque is mapped onto the wheels. This means that, for a case with perfect power regeneration ( $\eta = 1$ ), it is also  $C_2 = 0$ . This means that using the  $C_2$  cost function would be inconclusive, since all the trajectories would have the same zero cost, and the algorithm would not be able to prioritize any path over the others.

### 8.3.2 Numerical results

This last section implements the algorithms highlighted so far, and aims to test the validity of the method as well as to provide a comparison between results obtained using the different cost functions. The spacecraft properties used in the simulations are the following:

$$\mathcal{B}[I] = \begin{bmatrix} 1700 & 0 & 0 \\ 0 & 1700 & 0 \\ 0 & 0 & 1800 \end{bmatrix} \text{ kg m}^2 \qquad I_{w_s} = I_{w_t} = 0.16 \text{ kg m}^2. \tag{8.16}$$

Simulations are set up for the spacecraft performing a non-rest-to-rest maneuver. In the first two cases, (a) and (b), the spacecraft is equipped with three reaction wheels aligned with the principal body axes; in the latter two, (c) and (d), the spacecraft is equipped with four reaction wheels with the same configuration as described in the previous section for Figure 8.5. For all four cases, the effort-based A\* graph search is run with the three cost functions defined above:  $C_0$ ,  $C_1$  and  $C_2$ . The results obtained with the cost function  $C_0$  are provided only to outline how the newer approach is superior: in fact, in all four cases, the  $C_0$ -optimal solution is always associated with a higher  $C_1$  cost than the  $C_1$ -optimal solution. Because the latter gives an accurate description of the

full dynamics, it should always be preferred. Table 8.1 summarizes, for all four scenarios, the results obtained using each cost function as the driver for the effort-based  $A^*$  solution, where zero power regeneration is assumed ( $\eta = 0$ ). For each optimal solution, all three cost functions are evaluated and reported in the table for comparison. The different trajectories of the sensitive boresight are shown in Figure 8.6. For case (a) only, the  $C_1$ -optimal and the  $C_2$ -optimal trajectories coincide. In cases (a), (b) and (c), the inertia of the spacecraft plays the major role in determining the initial

Table 8.1: Effort-based Path Planners; Simulation Result Data

(a)	$\omega_0 = [0, 0, 0.03]$ rad/s	$\Omega_0 = [0, 0, 0]$ rad/s		
$C_0$ -optimal:	$C_0 = 168.26$ Nms	$C_1 = 210.51$ Nms	$C_2 = 20297.77$ J	
$C_1$ -optimal:	$C_0 = 233.43$ Nms	$C_1 = 195.68$ Nms	$C_2 = 18857.63$ J	
$C_2$ -optimal:	$C_0 = 233.43$ Nms	$C_1 = 195.68$ Nms	$C_2 = 18857.63$ J	
(b)	$\omega_0 = [0, 0, 0.03]$ rad/s	$\Omega_0 = [500, 0, 0]$ rad/s		
$C_0$ -optimal:	$C_0 = 168.26$ Nms	$C_1 = 387.88$ Nms	$C_2 = 75831.74$ J	
$C_1$ -optimal:	$C_0 = 239.39$ Nms	$C_1 = 260.73$ Nms	$C_2 = 51244.46$ J	
$C_2$ -optimal:	$C_0 = 256.60$ Nms	$C_1 = 333.95$ Nms	$C_2 = 49957.61$ J	
(c)	$\omega_0 = [0, 0, 0.03]$ rad/s	$\Omega_0 = [100, -200, 0, 400]$ rad/s		
$C_0$ -optimal:	$C_0 = 168.26$ Nms	$C_1 = 207.82$ Nms	$C_2 = 27733.80$ J	
$C_1$ -optimal:	$C_0 = 168.26$ Nms	$C_1 = 194.27$ Nms	$C_2 = 27116.57$ J	
$C_2$ -optimal:	$C_0 = 233.43$ Nms	$C_1 = 245.10$ Nms	$C_2 = 27458.55$ J	
(d)	$\omega_0 = [0.03, 0, 0]$ rad/s	$\Omega_0 = [500, -200, -300, 400]$ rad/s		
$C_0$ -optimal:	$C_0 = 179.31$ Nms	$C_1 = 692.16$ Nms	$C_2 = 188392.80$ J	
$C_1$ -optimal:	$C_0 = 274.19$ Nms	$C_1 = 497.47$ Nms	$C_2 = 132997.65$ J	
$C_2$ -optimal:	$C_0 = 275.68$ Nms	$C_1 = 503.94$ Nms	$C_2 = 121433.82$ J	

direction of the trajectory: the component along the  $\hat{\mathbf{b}}_z$  axis of the initial angular velocity makes such that all the effort-optimal trajectories follow, at least initially, that direction. Figure 8.6d, on the contrary, shows a new, interesting behavior: the  $C_0$ -optimal trajectory around  $t = 0$  moves in a direction that is opposite to that of the  $C_1$ - and  $C_2$ -optimal trajectories. This highlights how the dynamics of the actuator can contribute significantly to the momentum and energy of the system, to the point where the effort computed based only on the properties of the spacecraft and the reference trajectory ( $C_0$  function) can be a deceiving metric. For case (c), it is possible to observe

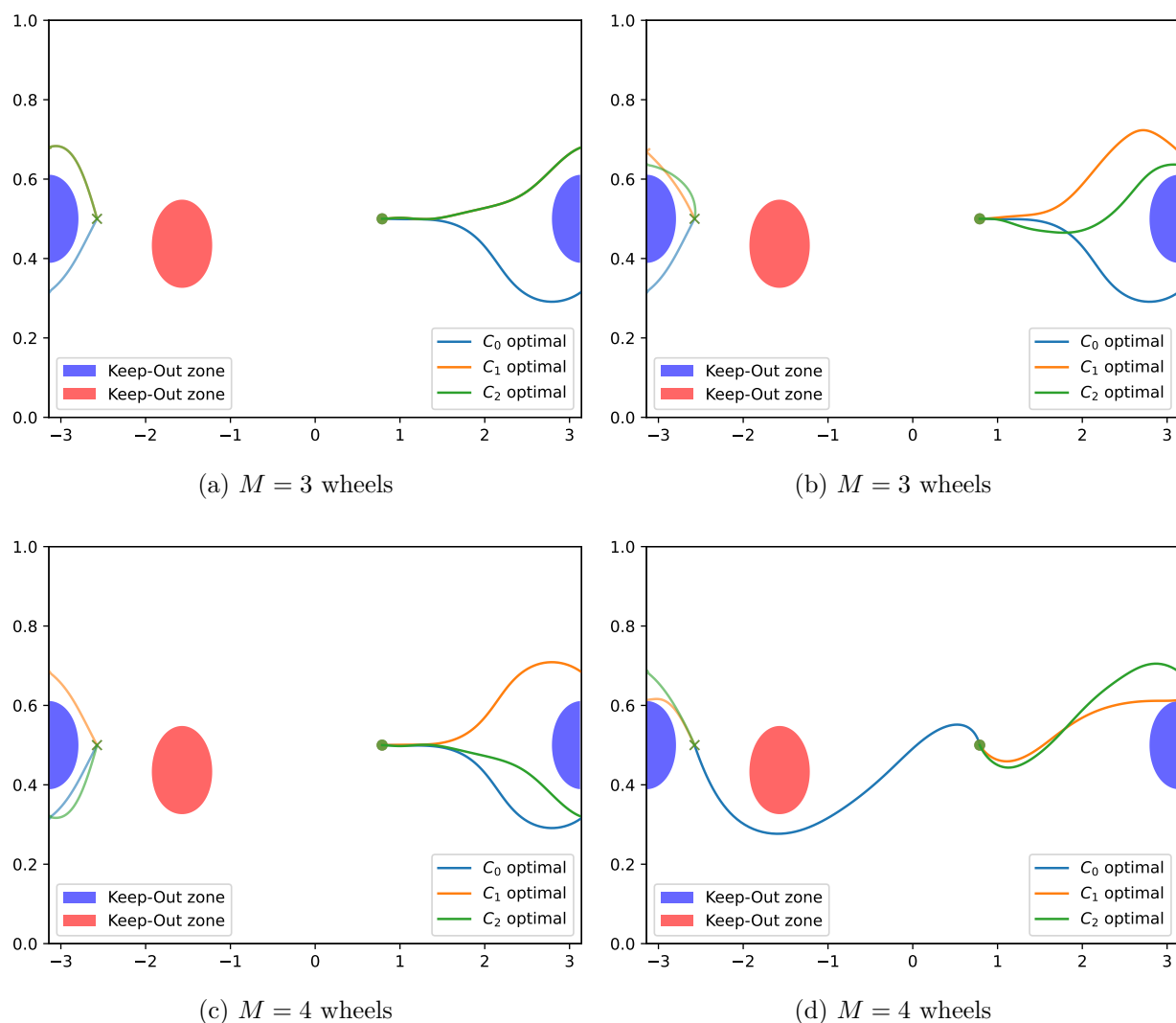


Figure 8.6: Inertial trajectories of the sensitive boresight in different simulation scenarios

that the  $C_2$ -optimal solution is, in fact, not optimal with respect to  $C_2$ , as the  $C_2$  value computed for the  $C_1$ -optimal solution is better. It can occasionally happen that these algorithms compute a sub-optimal solution: this is due to the way that NURBS curves affect the the update of the node priority inside the A\* algorithm. It can happen that an intermediate trajectory, which would ultimately lead to the cost-optimal path, is associated with an intermediate cost that is higher than other suboptimal trajectories, therefore forcing the algorithm to explore other areas of the graph. This phenomenon, however, happens around solutions that are relatively close in cost to one-another. In this case, for example, the difference between the two is less than 1.5%.



## Conclusions

### 9.1 Summary of Contributions

First, this work has provided a general and comprehensive analysis of the problem of defining the attitude of a spacecraft with multiple pointing constraints, presented by a gimbaled electric thruster, rotating solar arrays, and other softer, keep-out constraints presented by a thermally-sensitive surface. The spacecraft and its articulated components have been simplified into body-frame vectors that can move with the spacecraft hub and with respect to the spacecraft hub. The celestial objects surrounding the spacecraft, which constitute the directions that the spacecraft and its components should or should not point at, have been reduced to inertial unit direction vectors as would result from onboard sensor measurements. This has allowed to analyze the problem of defining the optimal reference attitude for the spacecraft in a general form, delivering clear and analytical solutions. These solutions are defined depending on an ordered list of pointing priorities dictated by the pointing mode of the spacecraft and the power needs at that state. The analysis has shown under which circumstances multiple pointing constraints can be satisfied simultaneously, and under which circumstances they cannot. This analytical derivation is powerful because, on top of informing the user beforehand on the ability to meet the pointing constraints of the spacecraft, it also lends itself well to the development of onboard flight software that can compute the optimal reference attitude at all times with minimal computational effort, provided that a pointing priority list is agreed on in case of an overdetermined problem.

Secondly, this work has leveraged the gimbaled electric thruster and rotating solar arrays

spacecraft design to perform continuous momentum management on the reaction wheels that ensure that pointing is maintained accurately throughout the mission. A control law has been derived for the gimbaleed thruster platform, assuming small deviations from the system center of mass, to ensure that the resulting thruster torque counters the momentum buildup due to external perturbations. Asymptotical stability of such control law has been shown along two out of three axes, as expected from such an underactuated control strategies. To achieve full controllability, a new control law is derived to actuate the solar arrays differentially, in order to leverage solar radiation pressure torque on the arrays. This is done to further drive down the net momentum of the bus along the component that cannot be controlled by the thruster, while also ensuring that the power loss due to offpointing the solar arrays is minimal. This momentum management technique, based on the solar arrays, provides even more limited control than the thruster, in that it is highly underactuated and contingent upon significant enough solar radiation pressure. Nevertheless, this method is shown to be rather effective when combined with the thruster action, where the momentum about two axes are managed by the thruster, and the momentum about the third axis is managed by the arrays in whatever capacity is possible. These momentum management strategies are paired with a recursive least-squares algorithm that continuously estimates the location of the system center of mass. This algorithm leverages the steady-state control torques imparted to the spacecraft to hold the optimal attitude to triangulate the location of the center of mass. Results have shown that this algorithm, which relies entirely on measurements and is not modeling the dynamics of the center of mass, is subject to biases due to unmodeled external torques and poor steady-state observability. Nevertheless, simulations have proved that the biased center of mass estimate still provided a valid pointing reference for the gimbaleed thruster, ensuring the convergence to a steady state where external torques are balanced.

Lastly, this work has provided a solution to the problem of maneuvering a spacecraft that is subject to multiple keep-in and keep-out type of constraints. The solution proposed in this work is based on a roadmap that is the result of evenly sampling the attitude space. Modified Rodrigues Parameters are a particularly suitable attitude representation for this problem, because

they provide a compact and bounded domain within which the attitude space can be sampled uniformly and compactly without loss of generality and without introducing singularities in the formulation. This comes at the expenses of a discontinuity in the MRP representation, which is dealt with leveraging the properties of MRPs and building the undirected graph with the sampled nodes intelligently. This undirected graph can be searched for a path that connects the initial to the target attitude while respecting the pointing constraint. The granularity of the graph can be defined ad hoc, depending on the application, the memory available for onboard computation, and without any changes to the algorithm. Furthermore, this work explored fitting functions to transform sequences of discrete attitude waypoints into kinematically feasible, twice differentiable, time dependant reference trajectories that can be easily tracked in a closed-loop control scheme with negligible tracking errors. NURBS curves have been shown to be a valid choice of fitting curves due to their malleability in fitting the attitude waypoints while also accounting for other desired kinematic properties of the resulting reference trajectory. This work finally blends these fitting functions into a variation of the A\* algorithm that searches the graph for the optimal path. The purpose is to search the graph for a path that has global optimal properties in terms of different metrics of effort required to perform the slew maneuver. While incorporating the fitting functions into A\* required sacrificing some assumptions of the A\* algorithm itself, simulations still show good global optimal behavior with respect to other analogously-generated slew maneuvers.

## 9.2 Future Work

Despite the novel and exciting contributions that resulted from this work, more questions were raised during the process which have not yet found an answer. This section highlights some of these questions, suggesting possible directions for future work.

### 9.2.1 Momentum Management via SRP Torque on Solar Arrays

Section 4.2 has showed how to rotate the solar arrays in order to maximize the torque along the opposite direction to the net RW momentum. This technique makes a fundamental assumption,

which is that the spacecraft attitude is held steady throughout the process. This limits the ability to generate SRP torque along the desired direction, thus preventing the momentum to be zeroed completely. It would be very interest to expand the problem presented in Section 4.2 to factor in not only the rotation angle of the array(s), but also the attitude of the spacecraft. Ideally, it should be possible to compute the optimal reference attitude and array configuration that yield the maximum torque along the opposite direction to the total, inertially-fixed system momentum. This way, the underactuated nature of the solar-array-based momentum management method would be compensated controlling the attitude instead. This strategy would be particularly effective in the inner regions of the Solar System, where SRP is strong and where optimal array pointing can be sacrificed without significant power losses. In return, it would enable free, full actuator desaturation.

### 9.2.2 Dynamics Models in the CM Estimator

Section 4.3 has shown that the center of mass estimate was significantly affected by poor steady state observability and the bias due to unmodeled external perturbations. The major of such perturbations is the SRP torque, for which models exist. However, accurately modeling SRP for a spacecraft with a complex geometry is nontrivial, due to multiple second-order effects like reflected light. In the presence of an accurate SRP model, the SRP torque as a function of the attitude could be added to the model to eliminate the bias.

Additionally, a future refinement should incorporate dynamics models of the fuel tanks. This would allow to provide a prediction step to the LS algorithm, thus upgrading it to a full Kalman filter. Sloshing models, if accurate, could provide better observability along the poorly observable direction, and blowdown models could inform the algorithm on how the CM location evolves over time due to mass depletion.

### 9.2.3 Benchmarking the Effort-based Path Planning Algorithm

Due to time constraints, it has not been possible to benchmark the performance of the effort-based A\* algorithm. Future work would see this algorithm compared to the performance of

other, well established and near-optimal path planning algorithms such as SST. The first term of comparison would involve the computational time of the effort-based A\*. Because SST is near-optimal, its solution will approach optimality as the algorithm is left to iterate. Therefore, it would be interesting to test “how much more optimal” the solution of SST when run for the same amount of time. Another interesting term of comparison would be the closed-loop performance. SST builds kinematically feasible reference trajectories propagating the dynamics forward, but relies on randomly sampled, piecewise-constant controls. Consequently, it would be interesting to verify the adherence of the closed-loop trajectory to the reference in the proximity of the discontinuities in the control, when tracked with actuators with own dynamics such as reaction wheels.

## Bibliography

- [1] K. Holste, P. Dietz, S. Scharmann, K. Keil, T. Henning, D. Zschätzsch, M. Reitemeyer, B. Nauschütt, F. Kiefer, F. Kunze, et al., “Ion thrusters for electric propulsion: Scientific issues developing a niche technology into a game changer,” Review of Scientific Instruments, vol. 91, no. 6, p. 061101, 2020.
- [2] E. M. Petro and R. J. Sedwick, “Survey of moderate-power electric propulsion systems,” Journal of Spacecraft and Rockets, vol. 54, no. 3, pp. 529–541, 2017.
- [3] M. D. Rayman and D. H. Lehman, “Deep space one: Nasa’s first deep-space technology validation mission,” Acta Astronautica, vol. 41, no. 4-10, pp. 289–299, 1997.
- [4] M. Yoshikawa, J. Kawaguchi, A. Fujiwara, and A. Tsuchiyama, “The hayabusa mission,” in Sample return missions, pp. 123–146, Elsevier, 2021.
- [5] Y. Tsuda, M. Yoshikawa, M. Abe, H. Minamino, and S. Nakazawa, “System design of the hayabusa 2—asteroid sample return mission to 1999 ju3,” Acta Astronautica, vol. 91, pp. 356–362, 2013.
- [6] G. Racca, A. Marini, L. Stagnaro, J. Van Dooren, L. Di Napoli, B. Foing, R. Lumb, J. Volp, J. Brinkmann, R. Grünagel, et al., “SMART-1 mission description and development status,” Planetary and space science, vol. 50, no. 14-15, pp. 1323–1337, 2002.
- [7] M. D. Rayman, T. C. Fraschetti, C. A. Raymond, and C. T. Russell, “Dawn: A mission in development for exploration of main belt asteroids vesta and ceres,” Acta Astronautica, vol. 58, no. 11, pp. 605–616, 2006.
- [8] A. Loehberg, C. Gruenwald, J. Birkel, A. Brandl, N. Kuebler, H. Perrin, T. Andreev, U. Schuhmacher, and S. Taylor, “The BepiColombo Mercury planetary orbiter (MPO) solar array design, major developments and qualification,” in E3S Web of Conferences, vol. 16, p. 04006, EDP Sciences, 2017.
- [9] D. Y. Oh, S. Collins, D. Goebel, B. Hart, G. Lantoine, S. Snyder, G. Whiffen, L. Elkins-Tanton, P. Lord, Z. Pirkl, et al., “Development of the Psyche mission for NASA’s discovery program,” in 35th International Electric Propulsion Conference, (Atlanta, GA), October 8–12, 2017.
- [10] R. Calaon, C. Allard, and H. Schaub, “Solar electric propulsion GN&C pointing state overview for the Emirates Mission to the Asteroid Belt,” in AAS Guidance and Control Conference, Feb. 1–7 2024. Paper No. AAS-24-056.

- [11] A. Boca, C. MacFarland, and R. Kowalczyk, "Solar power for deep-space applications: state of art and development," in AIAA Propulsion and Energy 2019 Forum, p. 4236, 2019.
- [12] T. Bondo, R. Walker, A. Willig, A. Rathke, D. Izzo, and M. Ayre, "Preliminary design of an advanced mission to Pluto," in 24th International Symposium on Space Technology and Science, Miyazaki, Japan, 2004.
- [13] C. Casaregola, K. Geurts, P. Pergola, and M. Andrenucci, "Radioisotope low-power electric propulsion missions to the outer planets," in 43rd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, p. 5234, 2007.
- [14] R. Calaon, E. Fantino, R. Flores, and J. Peláez, "Design of the trajectory of a tether mission to Saturn," in 8th European Conference for Aeronautics and Aerospace Sciences (EUCASS), 2019.
- [15] C. R. Mercer, "Chapter 12 - Solar array designs for deep space science missions," in Photovoltaics for Space, pp. 349–378, Elsevier, 2023.
- [16] B. Heinrich, J. Zemann, and F. Rottmeier, "Development of the BepiColombo MPO solar array drive assembly (SADA)," in European Space Mechanisms and Tribology Symposium (ESMATS), vol. 87, pp. 1–21, 2011.
- [17] M. Ziebart, "Generalized analytical solar radiation pressure modeling algorithm for spacecraft of complex shape," Journal of spacecraft and rockets, vol. 41, no. 5, pp. 840–848, 2004.
- [18] C. Früh, T. M. Kelecy, and M. K. Jah, "Coupled orbit-attitude dynamics of high area-to-mass ratio (hamr) objects: influence of solar radiation pressure, earth's shadow and the visibility in light curves," Celestial Mechanics and Dynamical Astronomy, vol. 117, pp. 385–404, 2013.
- [19] M. J. Sidi, Spacecraft dynamics and control: a practical engineering approach, vol. 7. Cambridge university press, 1997.
- [20] A. K. Banerjee, N. Pedreiro, and W. E. Singhose, "Vibration reduction for flexible spacecraft following momentum dumping with/without slewing," Journal of Guidance, Control, and Dynamics, vol. 24, no. 3, pp. 417–427, 2001.
- [21] N. Pedreiro, "Spacecraft architecture for disturbance-free payload," Journal of guidance, Control, and Dynamics, vol. 26, no. 5, pp. 794–804, 2003.
- [22] M. A. Karami and F. Sassani, "Spacecraft momentum dumping using fewer than three external control torques," Journal of guidance, control, and dynamics, vol. 32, no. 1, pp. 242–247, 2009.
- [23] D. Tong, "Spacecraft momentum dumping using gravity gradient," Journal of spacecraft and rockets, vol. 35, no. 5, pp. 714–717, 1998.
- [24] E. A. Hogan and H. Schaub, "Three-axis attitude control using redundant reaction wheels with continuous momentum dumping," Journal of Guidance, Control, and Dynamics, vol. 38, no. 10, pp. 1865–1871, 2015.
- [25] G. Racca, G. Whitcomb, and B. Foing, "The SMART-1 mission," ESA bulletin, vol. 95, pp. 72–81, 1998.

- [26] D. Y. Oh, S. Collins, T. M. Randolph, C. Vanelli, and S. Tilley, "Feasibility of all-electric three axis momentum management for deep space small body rendezvous," in 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, (Cleveland, OH), July 28–30, 2014.
- [27] E. Bergmann and J. Dzielski, "Spacecraft mass property identification with torque-generating control," Journal of Guidance, Control, and Dynamics, vol. 13, no. 1, pp. 99–103, 1990.
- [28] E. Bergmann, B. K. Walker, and D. R. Levy, "Mass property estimation for control of asymmetrical satellites," Journal of Guidance, Control, and Dynamics, vol. 10, no. 5, pp. 483–491, 1987.
- [29] D.-H. Kim, D.-G. Choi, and H.-S. Oh, "Inertia estimation of spacecraft based on modified law of conservation of angular momentum," Journal of Astronomy and Space Sciences, vol. 27, no. 4, pp. 353–357, 2010.
- [30] S. Tanygin and T. Williams, "Mass property estimation using coasting maneuvers," Journal of Guidance, Control, and Dynamics, vol. 20, no. 4, pp. 625–632, 1997.
- [31] O. Ma, H. Dang, and K. Pham, "On-orbit identification of inertia properties of spacecraft using a robotic arm," Journal of guidance, control, and dynamics, vol. 31, no. 6, pp. 1761–1771, 2008.
- [32] C. R. McInnes, "Large angle slew maneuvers with autonomous sun vector avoidance," Journal of guidance, control, and dynamics, vol. 17, no. 4, pp. 875–877, 1994.
- [33] M. Diaz Ramos and H. Schaub, "Kinematic steering law for conically constrained torque-limited spacecraft attitude control," Journal of Guidance, Control, and Dynamics, vol. 41, no. 9, pp. 1990–2001, 2018.
- [34] U. Lee and M. Mesbahi, "Spacecraft reorientation in presence of attitude constraints via logarithmic barrier potentials," in Proceedings of the 2011 American Control Conference, pp. 450–455, IEEE, 2011.
- [35] U. Lee and M. Mesbahi, "Feedback control for spacecraft reorientation under attitude constraints via convex potentials," IEEE Transactions on Aerospace and Electronic Systems, vol. 50, no. 4, pp. 2578–2592, 2014.
- [36] H. C. Kjellberg and E. G. Lightsey, "Discretized constrained attitude pathfinding and control for satellites," Journal of Guidance, Control, and Dynamics, vol. 36, no. 5, pp. 1301–1309, 2013.
- [37] H. C. Kjellberg and E. G. Lightsey, "Discretized quaternion constrained attitude pathfinding," Journal of Guidance, Control, and Dynamics, vol. 39, no. 3, pp. 713–718, 2015.
- [38] S. Tanygin, "Fast autonomous three-axis constrained attitude pathfinding and visualization for boresight alignment," Journal of Guidance, Control, and Dynamics, vol. 40, no. 2, pp. 358–370, 2017.
- [39] R. Tarjan, "Depth-first search and linear graph algorithms," SIAM journal on computing, vol. 1, no. 2, pp. 146–160, 1972.
- [40] C. Y. Lee, "An algorithm for path connections and its applications," IRE transactions on electronic computers, vol. EC-10, no. 3, pp. 346–365, 1961.



- [41] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968.
- [42] E. Frazzoli, M. Dahleh, E. Feron, and R. Kornfeld, A randomized attitude slew planning algorithm for autonomous spacecraft, vol. 4155. AIAA Reston, VA, 2001.
- [43] X. Tan, S. Berkane, and D. V. Dimarogonas, "Constrained attitude maneuvers on  $SO(3)$ : Rotation space sampling, planning and low-level control," Automatica, vol. 112, p. 108659, 2020.
- [44] C. Xiaojun, C. Hutao, C. Pingyuan, and X. Rui, "Large angular autonomous attitude maneuver of deep spacecraft using pseudospectral method," in 3rd international symposium on systems and control in aeronautics and astronautics, pp. 1510–1514, IEEE, 2010.
- [45] R. G. Melton, "Maximum-likelihood estimation optimizer for constrained, time-optimal satellite reorientation," Acta Astronautica, vol. 103, pp. 185–192, 2014.
- [46] R. G. Melton, "Hybrid methods for determining time-optimal, constrained spacecraft reorientation maneuvers," Acta Astronautica, vol. 94, no. 1, pp. 294–301, 2014.
- [47] D. Spiller, L. Ansalone, and F. Curti, "Particle swarm optimization for time-optimal spacecraft reorientation with keep-out cones," Journal of Guidance, Control, and Dynamics, vol. 39, no. 2, pp. 312–325, 2016.
- [48] D. Spiller, R. G. Melton, and F. Curti, "Inverse dynamics particle swarm optimization applied to constrained minimum-time maneuvers using reaction wheels," Aerospace Science and Technology, vol. 75, pp. 1–12, 2018.
- [49] T. L. Dearing, J. Hauser, X. Chen, M. M. Nicotra, and C. Petersen, "Efficient trajectory optimization for constrained spacecraft attitude maneuvers," Journal of Guidance, Control, and Dynamics, vol. 45, no. 4, pp. 638–650, 2022.
- [50] H. Schaub and J. L. Junkins, Analytical Mechanics of Space Systems. AIAA, 4 ed., 2018.
- [51] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," SIAM journal on Matrix Analysis and Applications, vol. 20, no. 2, pp. 303–353, 1998.
- [52] M. D. Shuster et al., "A survey of attitude representations," Navigation, vol. 8, no. 9, pp. 439–517, 1993.
- [53] S. W. Shepperd, "Quaternion from rotation matrix," Journal of guidance and control, vol. 1, no. 3, pp. 223–224, 1978.
- [54] O. Rodrigues, "Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire," Journal de mathématiques pures et appliquées, vol. 5, pp. 380–440, 1840.
- [55] T. F. Wiener, Theoretical analysis of gimballess inertial reference equipment using delta-modulated instruments. PhD thesis, Massachusetts Institute of Technology, 1962.

- [56] H. Schaub and J. L. Junkins, “Stereographic orientation parameters for attitude dynamics: A generalization of the rodrigues parameters,” Journal of the Astronautical Sciences, vol. 44, no. 1, pp. 1–19, 1996.
- [57] H. Goldstein, C. Poole, and J. Safko, Classical Mechanics. Addison-Wesley, 3 ed., 2001.
- [58] H. Schaub and V. J. Lappas, “Redundant reaction wheel torque distribution yielding instantaneous L2 power-optimal spacecraft attitude control,” Journal of guidance, control, and dynamics, vol. 32, no. 4, pp. 1269–1276, 2009.
- [59] H. K. Khalil, Nonlinear Systems. Prentice Hall, 3 ed., 2002.
- [60] R. Mukherjee and D. Chen, “Asymptotic stability theorem for autonomous systems,” Journal of Guidance, Control, and Dynamics, vol. 16, no. 5, pp. 961–963, 1993.
- [61] S. Krishnan and S. R. Vadali, “An inverse-free technique for attitude control of spacecraft using CMGs,” Acta Astronautica, vol. 39, no. 6, pp. 431–438, 1996.
- [62] P. W. Kenneally, S. Piggott, and H. Schaub, “Basilisk: A flexible, scalable and modular astrodynamics simulation framework,” Journal of Aerospace Information Systems, vol. 17, no. 9, pp. 496–507, 2020.
- [63] S. Carnahan, S. Piggott, and H. Schaub, “A new messaging system for Basilisk,” in AAS Guidance and Control Conference, (Breckenridge, CO), Jan. 30 – Feb. 5 2020. AAS 20-134.
- [64] J. Alcorn, C. Allard, and H. Schaub, “Fully coupled reaction wheel static and dynamic imbalance for spacecraft jitter modeling,” Journal of Guidance, Control, and Dynamics, vol. 41, no. 6, pp. 1380–1388, 2018.
- [65] C. Allard, M. Diaz Ramos, and H. Schaub, “Computational performance of complex spacecraft simulations using back-substitution,” Journal of Aerospace Information Systems, vol. 16, no. 10, pp. 427–436, 2019.
- [66] C. Allard, M. D. Ramos, H. Schaub, P. Kenneally, and S. Piggott, “Modular software architecture for fully coupled spacecraft simulations,” Journal of Aerospace Information Systems, vol. 15, no. 12, pp. 670–683, 2018.
- [67] J. Vaz Carneiro, C. Allard, and H. Schaub, “General dynamics for single-and dual-axis rotating rigid spacecraft components,” Journal of Spacecraft and Rockets, pp. 1–16, 2024.
- [68] M. Cols-Margenet, H. Schaub, and S. Piggott, “Modular attitude guidance: Generating rotational reference motions for distinct mission profiles,” Journal of Aerospace Information Systems, vol. 15, no. 6, pp. 335–352, 2018.
- [69] A. Madni, N. Bradley, D. Cervantes, D. Eldred, D. Oh, D. Mathews, and P. C. Lai, “Pointing error budget development and methodology on the Psyche project,” in 2021 IEEE Aerospace Conference, 2021.
- [70] C. Vanelli, E. Swenka, and B. Smith, “Verification of pointing constraints for the Dawn spacecraft,” in AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 2008. Paper No. AIAA-2008-7-85.

- [71] M. D. Shuster and S. D. Oh, “Three-axis attitude determination from vector observations,” Journal of guidance and Control, vol. 4, no. 1, pp. 70–77, 1981.
- [72] C. Rodriguez-Solano, U. Hugentobler, and P. Steigenberger, “Adjustable box-wing model for solar radiation pressure impacting gps satellites,” Advances in Space Research, vol. 49, no. 7, pp. 1113–1128, 2012.
- [73] D. Simon, Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons, 2006.
- [74] E. D. Sontag, Mathematical control theory: deterministic finite dimensional systems, vol. 6. Springer Science & Business Media, 2013.
- [75] J. L. Geeraert, J. T. Fischetti, D. R. Stanbridge, C. D. Jackman, J. V. McAdams, K. Berry, J. M. Leonard, D. Wibben, J. Y. Pelgrift, M. J. Salinas, et al., “Lucy orbit determination performance from launch through EGA-1,” in AAS/AIAA Astrodynamics Specialist Conference, (Big Sky, MT), August 13–17, 2023. Paper No. AAS 23-247.
- [76] J. S. Snyder, V. H. Chaplin, D. M. Goebel, R. R. Hofer, A. L. Ortega, I. G. Mikellides, T. Kerl, G. Lenguito, F. Aghazadeh, and I. Johnson, “Electric propulsion for the Psyche mission: Development activities and status,” in AIAA Propulsion and Energy 2020 Forum, (Virtual Event), August 24–28, 2020.
- [77] P. Tsiotras, J. L. Junkins, and H. Schaub, “Higher-order Cayley transforms with applications to attitude representations,” Journal of Guidance, Control, and Dynamics, vol. 20, no. 3, pp. 528–534, 1997.
- [78] S. Tanygin, “Attitude parameterizations as higher-dimensional map projections,” Journal of Guidance, Control, and Dynamics, vol. 35, no. 1, pp. 13–24, 2012.
- [79] S. Tanygin, “Projective geometry of attitude parameterizations with applications to control,” Journal of Guidance, Control, and Dynamics, vol. 36, no. 3, pp. 656–666, 2013.
- [80] A. Yershova, S. Jain, S. M. Lavalle, and J. C. Mitchell, “Generating uniform incremental grids on  $so(3)$  using the hopf fibration,” The International journal of robotics research, vol. 29, no. 7, pp. 801–812, 2010.
- [81] H. M. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, and R. C. Arkin, Principles of robot motion: theory, algorithms, and implementation. MIT press, 2005.
- [82] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to algorithms. MIT press, 2009.
- [83] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, “Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges,” Sensors, vol. 18, no. 9, p. 3170, 2018.
- [84] H. Schaub and S. Piggott, “Speed-constrained three-axes attitude control using kinematic steering,” Acta Astronautica, vol. 147, pp. 1–8, 2018.

- [85] B. Fornberg and J. Zuev, “The Runge phenomenon and spatially variable shape parameters in RBF interpolation,” Computers & Mathematics with Applications, vol. 54, no. 3, pp. 379–398, 2007.
- [86] L. Piegl and W. Tiller, The NURBS book. Springer Science & Business Media, 1996.
- [87] C. De Boor, “On calculating with b-splines,” Journal of Approximation theory, vol. 6, no. 1, pp. 50–62, 1972.
- [88] G. Farin, Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann, 2002.
- [89] C. De Boor and C. De Boor, A practical guide to splines, vol. 27. Springer-Verlag New York, 1978.
- [90] M. Leeney, “Fast quaternion slerp,” International Journal of Computer Mathematics, vol. 86, no. 1, pp. 79–84, 2009.
- [91] R. Blendin and H. Schaub, “Regenerative power-optimal reaction wheel attitude control,” Journal of guidance, control, and dynamics, vol. 35, no. 4, pp. 1208–1217, 2012.
- [92] H. C. Marsh, M. Karpenko, and Q. Gong, “Relationships between maneuver time and energy for reaction wheel attitude control,” Journal of Guidance, Control, and Dynamics, vol. 41, no. 2, pp. 335–348, 2018.

## Appendix A

### Pseudocode for Algorithms of Chapter 3

Pseudocode is provided in this appendix for the main algorithms discussed in Section 3.2.

Algorithm 2 computes the optimal PRA  $\psi$  given the coefficients  $A$ ,  $B$ , and  $C$  as defined in Equation (3.31), for the maximum power generation problem presented in Section 3.2.1. The code implements all the necessary logic steps to circumvent any singularities that arise due to the presence at the denominator of the coefficient  $A$  in Equation (3.35) or  $B$  in Equation (3.37), as well as the singularity in the CRP formulation. The output of the algorithm is the optimal PRA, or PRAs.

Algorithm 3 solves the problem of determining the range of admissible PRAs in the lower-bounded power generation problem presented in Section 3.2.2.1. The inputs are the coefficients  $A$ ,  $B$ , and  $C$  as defined in Equation (3.31), and the parameter  $K = \sin \gamma$  which is the maximum admissible incidence angle in order to generate the lower power bound. The output of this algorithm is a range of compliant PRAs  $\Psi$ .

Algorithm 4 computes the solution space of compliant PRAs for the keep-out constraint in Section 3.2.2.2. Its inputs are the coefficients  $D$ ,  $E$ , and  $F$  as defined in Equation (3.39), and the parameter  $Q = \cos \theta$ , where  $\theta$  is the half-angle of the conical keep-out constraint. The output of this algorithm is a range of PRAs  $\Theta$  compliant with the keep-out constraint.

---

**Algorithm 2**  $\psi = \text{optimalPRA}(A, B, C)$ 


---

```

 $\Delta = B^2 - 4AC$ 
if  $A = 0$  then
  if  $B = 0$  then
     $\psi = \pi$ 
  else
     $\psi = -2 \arctan(C/B) \cup \psi = \pi$ 
else
  if  $\Delta < 0$  then
    if  $B = 0$  then
      if  $|A| > |C|$  then
         $\psi = 0$ 
      else
         $\psi = \pi$ 
    else
       $q = \frac{A-C}{B}$ 
       $t_1 = q + \sqrt{1 + q^2}$ 
       $t_2 = q - \sqrt{1 + q^2}$ 
       $f_1 = \frac{|At_1^2 + Bt_1 + C|}{1 + t_1^2}$ 
       $f_2 = \frac{|At_2^2 + Bt_2 + C|}{1 + t_2^2}$ 
      if  $f_1 < f_2$  then
         $\psi = 2 \arctan(t_1)$ 
      else
         $\psi = 2 \arctan(t_2)$ 
    else
       $\psi = 2 \arctan\left(\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}\right)$ 
return  $\psi$ 

```

---

---

**Algorithm 3**  $\Psi = \text{solSpaceLB}(A, B, C, K)$ 


---

```

if  $A < 0$  then
   $A = -A$ 
   $B = -B$ 
   $C = -C$ 
 $\Delta = B^2 - 4AC$ 
 $\Delta^+ = B^2 - 4(A + K)(C + K)$ 
 $\Delta^- = B^2 - 4(A - K)(C - K)$ 
 $\psi_1^+ = 2 \arctan\left(\frac{-B - \sqrt{\Delta^+}}{2(A + K)}\right)$ 
 $\psi_2^+ = 2 \arctan\left(\frac{-B + \sqrt{\Delta^+}}{2(A + K)}\right)$ 
 $\psi_1^- = 2 \arctan\left(\frac{-B - \sqrt{\Delta^-}}{2(A - K)}\right)$ 
 $\psi_2^- = 2 \arctan\left(\frac{-B + \sqrt{\Delta^-}}{2(A - K)}\right)$ 
 $\psi^- = 2 \arctan\left(\frac{A - C}{B}\right)$ 
if  $\Delta > 0$  then
  if  $K > A$  then
    if  $\Delta^+ \geq 0$  and  $\Delta^- \geq 0$  then
       $\Psi = [\psi_1^-, \psi_1^+] \cup [\psi_2^+, \psi_2^-]$ 
    else if  $\Delta^+ < 0$  and  $\Delta^- \geq 0$  then
       $\Psi = [\psi_1^-, \psi_2^-]$ 
    else
       $\Psi = \emptyset$ 
    else if  $K = A$  then
      if  $\Delta^+ > 0$  then
         $\Psi = [-\pi, \psi_1^+] \cup [\psi_2^+, \psi^-]$ 
      else
         $\Psi = [-\pi, \psi^-]$ 
    else
      if  $\Delta^+ \leq 0$  and  $\Delta^- \geq 0$  then
        if  $B < 0$  then
           $\Psi = [-\pi, \psi_2^-] \cup [\psi_1^-, \psi_1^+] \cup [\psi_2^+, \pi]$ 
        else
           $\Psi = [-\pi, \psi_1^+] \cup [\psi_2^+, \psi_2^-] \cup [\psi_1^-, \pi]$ 
        else if  $\Delta^+ \leq 0$  and  $\Delta^- \geq 0$  then
           $\Psi = [-\pi, \psi_2^-] \cup [\psi_1^-, \pi]$ 
        else if  $\Delta^+ > 0$  and  $\Delta^- < 0$  then
           $\Psi = [-\pi, \psi_1^+] \cup [\psi_2^+, \pi]$ 
        else
           $\Psi = [-\pi, \pi]$ 
    else
       $\psi_{\min} = \text{optimalPRA}(A, B, C)$ 
      if  $K < A$  then
        if  $\Delta^- < 0$  then
           $\Psi = [\psi_{\min}, \psi_{\min}]$ 
        else
           $\Psi = [\psi_1^-, \psi_2^-]$ 
      else if  $K = A$  then
         $\Psi = [-\pi, \psi^-]$ 
      else
         $\Psi = [-\pi, \psi_2^-] \cup [\psi_1^-, \pi]$ 
  return  $\Psi$ 

```

---

---

**Algorithm 4**  $\Theta = \text{solSpaceKO}(D, E, F, Q)$

---

```

 $\theta_{\min} = \text{optimalPRA}(D, E, F)$ 
 $\Delta_{\text{KO}} = D^2 - 4(D - Q)(F - Q)$ 
if  $D = Q$  then
  if  $E = 0$  and  $F \geq Q$  then
     $\Theta = [-\pi, \pi]$ 
  else if  $E = 0$  and  $F < Q$  then
     $\Theta = [\theta_{\min}, \theta_{\min}]$ 
  else if  $E > 0$  then
     $\Theta = \left[ 2 \arctan\left(\frac{Q-F}{E}\right), \pi \right]$ 
  else
     $\Theta = \left[ -\pi, 2 \arctan\left(\frac{Q-F}{E}\right) \right]$ 
else
  if  $\Delta_{\text{KO}} < 0$  and  $D > Q$  then
     $\Theta = [-\pi, \pi]$ 
  else if  $\Delta_{\text{KO}} < 0$  and  $D < Q$  then
     $\Theta = [\theta_{\min}, \theta_{\min}]$ 
  else
     $\theta_1 = 2 \arctan\left(\frac{-E - \sqrt{\Delta_{\text{KO}}}}{2(D-Q)}\right)$ 
     $\theta_2 = 2 \arctan\left(\frac{-E + \sqrt{\Delta_{\text{KO}}}}{2(D-Q)}\right)$ 
    if  $D > Q$  then
       $\Theta = [-\pi, \theta_1] \cup [\theta_2, \pi]$ 
    else
       $\Theta = [\theta_2, \theta_1]$ 
return  $\Theta$ 

```

---



## Appendix B

### Time-warping Routine for Constant Angular Rate Maneuvering

This appendix is provided to describe how to enforce a constant angular rate magnitude  $\omega^*$  along an trajectory. This is done by means of a change in the parameter of the curve, from the original time  $t$  to a warped time  $\theta$ . Fixing  $\omega^*$  to a value that is below the speed limit, automatically bounds all the components  $\omega_i$  to be within such limit, thus ensuring that none of the angular rate component exceed that bound. While this procedure can be applied to any trajectory that has a continuous support in MRP space, this procedure is intended for a trajectory whose angular rate norm is already fairly close to a constant value, except for the initial and final segments where the angular rates ramp on and/or off to meet the endpoint derivative constraints.

The aim of this subsection is to obtain an angular rate  $\boldsymbol{\omega}(\theta)$  such that:

$$\boldsymbol{\omega}(\theta) = \boldsymbol{\omega}(t) \frac{dt}{d\theta} \tag{B.1}$$

where  $\theta(t)$  is a “warped” time that allows to navigate the trajectory  $\boldsymbol{\sigma}(\theta)$  at the desired angular rate. The first condition that must be met to ensure that the same trajectory is tracked, is that the first integral of the trajectory remain unaltered:

$$\Omega = \int_0^T \|\boldsymbol{\omega}(t)\| dt = \int_0^\Theta \|\boldsymbol{\omega}(\theta)\| d\theta. \tag{B.2}$$

where  $\Omega$  is computed by numerical integration of  $\|\boldsymbol{\omega}(t)\|$  coming from the original trajectory.

Clearly, a constant  $\|\boldsymbol{\omega}(\theta)\|$  over the entire domain  $\theta \in [0, \Theta]$  cannot match the required conditions at the endpoints. For this reason, the profile of  $\|\boldsymbol{\omega}(\theta)\|$  is defined as a piecewise function

that is a 4th-order polynomial around the endpoints and constant in the central part (80%) of the trajectory. Such piecewise function is defined as:

$$\|\boldsymbol{\omega}(\theta)\| = \begin{cases} a_0 + a_1\theta + a_2\theta^2 + a_3\theta^3 + a_4\theta^4 & \text{for } 0 \leq \theta < \theta_a \\ \omega^* & \text{for } \theta_a \leq \theta \leq \theta_b \\ b_0 + b_1(\theta - \Theta) + b_2(\theta - \Theta)^2 + b_3(\theta - \Theta)^3 + b_4(\theta - \Theta)^4 & \text{for } \theta_b < \theta \leq \Theta \end{cases} \quad (\text{B.3})$$

with:

$$\int_0^{\theta_a} \|\boldsymbol{\omega}(\theta)\| d\theta = \frac{1}{10}\Omega \quad \int_{\theta_a}^{\theta_b} \|\boldsymbol{\omega}(\theta)\| d\theta = \frac{4}{5}\Omega \quad \int_{\theta_b}^{\Theta} \|\boldsymbol{\omega}(\theta)\| d\theta = \frac{1}{10}\Omega. \quad (\text{B.4})$$

The problem now is to determine the coefficients of the polynomials  $a_i$  and  $b_i$ . The first four are obtained from the endpoint values and first endpoint derivatives of  $\|\boldsymbol{\omega}(t)\|$ , in order to ensure the same behavior around the endpoints:

$$a_0 = \|\boldsymbol{\omega}(0)\| \quad a_1 = \frac{d\|\boldsymbol{\omega}(0)\|}{dt} \quad (\text{B.5})$$

$$b_0 = \|\boldsymbol{\omega}(T)\| \quad b_1 = \frac{d\|\boldsymbol{\omega}(T)\|}{dt}. \quad (\text{B.6})$$

The other six coefficients are obtained imposing continuity ( $\|\boldsymbol{\omega}(\theta_a)\| = \|\boldsymbol{\omega}(\theta_b)\| = \omega^*$ ), together with first and second order derivatives equal to 0 for  $\theta = \theta_a$  and  $\theta = \theta_b$ . The fourth-order polynomials, together with these constraints on first and second order derivatives in  $\theta_a$  and  $\theta_b$ , ensure the continuity and differentiability of both  $\|\boldsymbol{\omega}(\theta)\|$  and  $\|\dot{\boldsymbol{\omega}}(\theta)\|$ . After some cumbersome algebra, this yields:

$$a_2 = \frac{6(\omega^* - a_0)}{\theta_a^2} - \frac{3a_1}{\theta_a} \quad b_2 = \frac{6(\omega^* - b_0)}{\theta_b^2} - \frac{3b_1}{\theta_b} \quad (\text{B.7})$$

$$a_3 = \frac{8(a_0 - \omega^*)}{\theta_a^3} + \frac{3a_1}{\theta_a^2} \quad b_3 = \frac{8(b_0 - \omega^*)}{\theta_b^3} + \frac{3b_1}{\theta_b^2} \quad (\text{B.8})$$

$$a_4 = \frac{3(\omega^* - a_0)}{\theta_a^4} - \frac{a_1}{\theta_a^3} \quad b_4 = \frac{3(\omega^* - b_0)}{\theta_b^4} - \frac{b_1}{\theta_b^3}. \quad (\text{B.9})$$

At this point, knowing the expressions for all the coefficients, it is possible to analytically integrate Equation (B.4), which yields a system of three equations in the variables  $\theta_a$ ,  $\theta_b$  and  $\Theta$ , whose

solution is:

$$\theta_a = \frac{-6\omega^* - 4a_0 + \sqrt{(4a_0 + 6\omega^*)^2 + 2a_1\Omega}}{a_1} \quad (\text{B.10})$$

$$\theta_b = \theta_a + \frac{4\Omega}{5\omega^*} \quad (\text{B.11})$$

$$\Theta = \theta_b + \frac{6\omega^* + 4b_0 - \sqrt{(4b_0 + 6\omega^*)^2 - 2b_1\Omega}}{b_1}. \quad (\text{B.12})$$

Now that the expression for  $\|\boldsymbol{\omega}(\theta)\|$  is complete, the relation  $\theta(t)$  is found solving the nonlinear equation:

$$\int_0^\theta \|\boldsymbol{\omega}(\theta)\| d\theta = \int_0^t \|\boldsymbol{\omega}(t)\| dt \quad (\text{B.13})$$

for every  $\theta \in [0, \Theta]$  and  $t \in [0, T]$ . The integral on the left-hand side of Equation (B.13) is computed analytically, since the expression in the integral is a polynomial. The integral on the right hand side, however, is computed numerically from the  $\|\boldsymbol{\omega}(t)\|$  that results from the original, continuous trajectory, which is obtained, for example, applying a NURBS fit. Carrying out both integrals leads, for  $0 \leq \theta < \theta_a$  and  $\theta_b < \theta \leq \Theta$ , to a 5th-order equation in the variable  $\theta$ , which can be solved numerically with Newton-Rhapson's method. The constant angular rate constraint is applied to the basic-A\* trajectory described in the previous subsections. Since the support of the trajectory in MRP space is unaltered, Figure 7.1 remains a valid representation of the trajectory, whereas the evolution of rates, accelerations and torque with respect to the new time variable  $\theta$  are reported

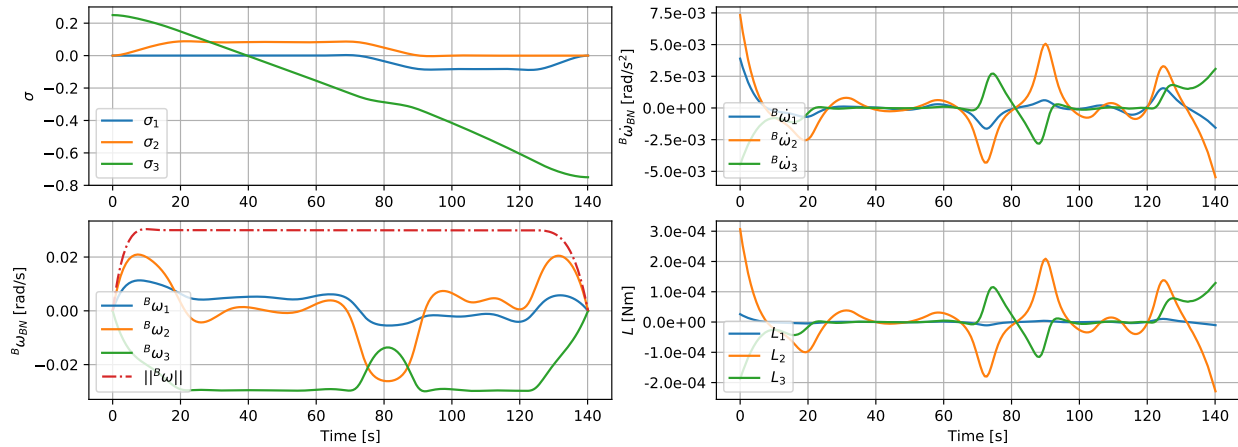


Figure B.1: Attitude, rates, acceleration and torque with constant  $\|\boldsymbol{\omega}\|$  for the trajectory in Figures 7.1 and 7.2

in Figure B.1. The magnitude of the angular rate vector ramps up and down from zero at the endpoints, and remains constant at  $\|\boldsymbol{\omega}\| = 0.03$  rad/s throughout most of the maneuver. With this correction, the total maneuver time is reduced to 140 s, instead of 150 s as shown in Figure 7.2.