



**PhD-FSTM-2025-023**

**Faculty of Science, Technology and Medicine**

## **DISSERTATION**

Defence held on 13 February 2025 in Luxembourg  
to obtain the degree of

**DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG  
EN SCIENCES DE L'INGÉNIEUR**

by

**Ahmed Mohamed Taha Emam Mahfouz**

Born on 18 December 1994 in Kalyobia (Egypt)

**AUTONOMOUS GUIDANCE, NAVIGATION AND CONTROL OF FORMATIONS  
OF UNDER-ACTUATED MICROSATELLITES**

### **Dissertation defence committee:**

Dr. Andreas Hein (Chairman)

Associate Professor, Université du Luxembourg

Dr. Jose Luis Sanchez-Lopez (Vice Chairman)

Research Scientist, Université du Luxembourg

Dr. Holger Voos (Dissertation Supervisor)

Full professor, Université du Luxembourg

Dr. Gabriella Gaias

Associate Professor, Politecnico di Milano

Dr. Mohamed Darouach

Full Professor, Université de Lorraine







PhD-FSTM-2025-023

Faculty of Science, Technology and Medicine

## DISSERTATION

Defence held on 13 February 2025 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG  
EN SCIENCES DE L'INGÉNIEUR

by

Ahmed Mohamed Taha Emam Mahfouz

Born on 18 December 1994 in Kalyobia (Egypt)

---

**AUTONOMOUS GUIDANCE, NAVIGATION AND  
CONTROL OF FORMATIONS OF UNDER-ACTUATED  
MICROSATELLITES**

---



## Declaration of Authorship

I, Ahmed Mohamed Taha Emam Mahfouz, declare that the Ph.D. thesis entitled "Autonomous Guidance, Navigation and Control of Formations of Under-actuated Microsatellites" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Ahmed Mohamed Taha Emam Mahfouz



Luxembourg, February 14, 2025

# Abstract

*Modern satellite missions increasingly rely on formations of smaller, cost-effective satellites working in coordination to achieve objectives traditionally handled by a single, complex spacecraft. This paradigm shift, driven by advancements in technology and the demand for robust and flexible systems, has underscored the need for innovative solutions to guidance, navigation, and control challenges specific to satellite formations. This thesis develops a comprehensive toolbox tailored to the Triton-X micro-satellite platform, enabling autonomous formation flying in low Earth orbit while addressing navigation and control challenges associated with under-actuated propulsion systems and limited navigation capabilities.*

*In the navigation domain, this research focuses on the problem of relative navigation for widely separated satellites equipped with single-frequency global navigation satellite system receivers. An extended Kalman filter algorithm is developed, incorporating a bi-linear ionospheric model to mitigate ionospheric errors that can significantly degrade measurement accuracy over large inter-satellite distances. Additionally, a carefully selected set of state variables and observables optimizes the filter's performance for the considered problem.*

*For the guidance layer, the satellite formation reconfiguration problem is formulated as a series of convex optimization problems, including quadratic programming, second-order cone programming, and linear programming formulations. These formulations balance fuel efficiency and computational feasibility while maintaining the same constraints. The minimum*

*thrust constraint, stemming from the hardware limitations is approximated by an affine relaxation, for which an acceleration pruning algorithm is developed to enhance the problem’s feasibility. Furthermore, centralized guidance strategies are extended to distributed frameworks, addressing scalability issues for large satellite formations by achieving linear computational growth as the number of satellites increases.*

*The control strategies introduced in this thesis integrate the most efficient guidance solutions into two model predictive control frameworks: the shrinking-horizon model predictive control and the fixed-horizon model predictive control. These strategies enable closed-loop real-time operation, ensuring continuous adjustments to control inputs based on sensor feedback. The fixed-horizon strategy offers computational stability and simplicity, while the shrinking-horizon strategy excels in adapting to disturbance-rich environments.*

*Finally, the toolbox is extended to include absolute orbit maintenance capabilities using time-optimal maneuvers using formation flying techniques. A nonlinear programming-based model predictive control framework addresses the coupled dynamics of attitude and orbital control, enabling efficient maneuvering with thrust applied during attitude slews.*

*This work provides a robust and versatile guidance, navigation, and control solution for underactuated satellite formations, enabling autonomous, fuel-optimal operations for future missions in low Earth orbit.*



*Pas de patience, pas de science.*

—Jean-Pierre Jarroux





## Acknowledgements

All the praises and thanks be to Allah, who has guided me to this, and never could I have found guidance, were it not that Allah had guided me!

As I reach the end of this PhD journey, I have come to realize that what truly matters, beyond the quality of the research, are the personal connections I was fortunate to make along the way. Being part of such a healthy and supportive environment within our research group has been an invaluable experience, and I am deeply grateful for the camaraderie and encouragement I have received throughout these years.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Dr. Holger Voos, whose unwavering guidance and support have been instrumental in shaping both this thesis and my academic growth. I am also profoundly thankful to my external advisor, Prof. Dr. Gabriella Gaias, from Politecnico di Milano (PloMi), whose contributions and guidance have left a lasting impact on both this research and my professional development. My two-month research stay at PoliMi provided an inspiring and enriching experience, and I am sincerely grateful to Prof. Gais as well as PoliMi for hosting me during this period.

I would also like to extend my thanks to the team at LuxSpace for their technical suggestions and constructive feedback along the way. Their practical expertise and collaborative spirit were invaluable to this work.

A special note of appreciation goes to Dr. Davide Menzio, the previous postdoctoral researcher on this project. His mentorship, both during and beyond our time working together, provided me with valuable perspectives and guidance that I will always cherish.

I am deeply thankful to my Egyptian and Pakistani friends, whose warmth and hospitality turned challenging times into moments of joy. Their get-togethers and delicious koshary,

kofta, and biryani dishes gave me the much-needed energy and comfort to finalize this thesis.

Lastly, but most importantly, I am forever grateful to my family. To my wife, for her endless support and patience; to my parents, for their unwavering encouragement and belief in me; and to my siblings, for their motivation and love. Your presence in my life has been the foundation of my success, and this journey would not have been possible without you.

This thesis is as much a result of personal connections and support as it is a product of academic effort. To everyone who walked this journey with me, I offer my heartfelt thanks.

Ahmed Mohamed Taha Emam Mahfouz



# Index

<b>I</b>	<b>Background</b>	<b>1</b>
<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Triton-X . . . . .	4
1.2	Aims and scope of the thesis . . . . .	5
1.3	Thesis outline . . . . .	7
1.4	Contributions to the state of the art . . . . .	9
<b>2</b>	<b>Mathematical models</b>	<b>13</b>
2.1	Reference frames . . . . .	13
2.1.1	Earth-Centered-Inertial frame . . . . .	13
2.1.2	Earth-Centered-Earth-Fixed frame . . . . .	14
2.1.3	Satellite-body-fixed frame . . . . .	14
2.1.4	Radial-Transversal-Normal frame . . . . .	15
2.2	Absolute orbital dynamics . . . . .	15
2.2.1	Dynamics of the Cartesian state vector . . . . .	17
2.2.2	Dynamics of the mean orbital elements . . . . .	21
2.3	Relative orbital dynamics . . . . .	23
2.3.1	Relative Cartesian elements formulation . . . . .	23
2.3.2	ROE formulation . . . . .	28

2.3.3	Transformation between the ROE and the RTN Cartesian state vector	36
<b>II</b>	<b>Contributions</b>	<b>39</b>
<b>3</b>	<b>GNSS-based relative navigation</b>	<b>40</b>
3.1	Introduction . . . . .	40
3.1.1	Problem statement . . . . .	41
3.1.2	State of the art . . . . .	42
3.1.3	Methodology and contributions . . . . .	43
3.1.4	Chapter outline . . . . .	44
3.2	GNSS observables . . . . .	44
3.2.1	Native GNSS measurements . . . . .	44
3.2.2	Differential measurements . . . . .	46
3.2.3	Measurements combinations . . . . .	49
3.3	Estimation strategy . . . . .	51
3.3.1	Measurement choice . . . . .	51
3.3.2	Ionospheric model . . . . .	53
3.3.3	State variables choice . . . . .	54
3.4	Dynamics of the EKF state vector . . . . .	55
3.4.1	Relative orbital dynamics . . . . .	56
3.4.2	Dynamics of the auxiliary variables . . . . .	57
3.4.3	State transition matrix . . . . .	57
3.4.4	Measurement model . . . . .	59
3.5	EKF initialization and operation . . . . .	61
3.6	Results and discussion . . . . .	64
3.7	Conclusion . . . . .	67
<b>4</b>	<b>Guidance strategies for multi-satellite formation reconfiguration</b>	<b>70</b>
4.1	Introduction . . . . .	70
4.1.1	Problem statement . . . . .	71

4.1.2	State of the art . . . . .	72
4.1.3	Methodology and contribution . . . . .	73
4.1.4	Chapter outline . . . . .	76
4.2	Guidance formulations . . . . .	77
4.3	Problem normalization . . . . .	92
4.3.1	Quadratic objective functions . . . . .	92
4.3.2	Linear objective functions . . . . .	93
4.3.3	Quadratic constraints . . . . .	93
4.3.4	Linear constraints . . . . .	93
4.3.5	Second-order cone constraints . . . . .	93
4.4	Parameters Sensitivity Analysis . . . . .	94
4.5	Results and discussion . . . . .	98
4.5.1	Case study . . . . .	98
4.5.2	Handling operational constraints . . . . .	105
4.5.3	Solvers benchmark . . . . .	107
4.6	Minimum thrust constraint . . . . .	119
4.7	Infeasibility handling . . . . .	128
4.8	Distributed guidance . . . . .	136
4.9	Conclusion . . . . .	143
<b>5</b>	<b>Control approaches for multi-satellite formation reconfiguration</b>	<b>146</b>
5.1	Introduction . . . . .	146
5.1.1	Problem Statement . . . . .	146
5.1.2	State of the art . . . . .	147
5.1.3	Methodology and contribution . . . . .	148
5.1.4	Chapter outline . . . . .	149
5.2	Model predictive control strategies . . . . .	149
5.2.1	Shrinking-horizon MPC . . . . .	149
5.2.2	Fixed-horizon MPC . . . . .	150

5.2.3	MPC operation . . . . .	151
5.3	Results and discussion . . . . .	153
5.4	Conclusion . . . . .	162
<b>6</b>	<b>Guidance and control strategies for absolute orbit keeping</b>	<b>164</b>
6.1	Introduction . . . . .	164
6.1.1	Problem statement . . . . .	164
6.1.2	State of the art . . . . .	165
6.1.3	Methodology and contribution . . . . .	166
6.1.4	Chapter outline . . . . .	167
6.2	Surrogate model of the attitude control system . . . . .	168
6.3	Model Predictive Control . . . . .	171
6.3.1	Guidance layer . . . . .	171
6.3.2	Parameter Tuning . . . . .	175
6.3.3	Closing the Control Loop . . . . .	183
6.4	MPC Validation . . . . .	185
6.5	Conclusions . . . . .	194
<b>7</b>	<b>Conclusion</b>	<b>196</b>
<b>III</b>	<b>Appendices</b>	<b>199</b>
<b>A</b>	<b>Extended Kalman Filter</b>	<b>200</b>
<b>B</b>	<b>Linearization of a general difference function</b>	<b>202</b>
<b>C</b>	<b>Estimation of the scaling factor used in the affine relaxation of the maximum acceleration constraints</b>	<b>203</b>
<b>D</b>	<b>Reconfiguration scenarios used throughout the thesis</b>	<b>207</b>
<b>E</b>	<b>A brief introduction to some relevant convex optimization solvers</b>	<b>214</b>

<b>F</b>	<b>Softening of the different guidance formulations</b>	<b>218</b>
F.1	Softening of the QCQP formulation . . . . .	218
F.2	Softening of the QP formulation . . . . .	220
F.3	Softening of the SOCP formulation . . . . .	221
F.4	Softening of the LP formulation . . . . .	223
<b>G</b>	<b>Surrogate models of navigation and pointing errors</b>	<b>225</b>
G.1	Model of the absolute navigation system . . . . .	226
G.2	Model of the relative navigation system . . . . .	227
G.3	Effect of pointing errors on the control acceleration vector . . . . .	229
<b>H</b>	<b>Stability of the surrogate model of the attitude control system</b>	<b>230</b>
	<b>Bibliography</b>	<b>232</b>



# List of Figures

1.1	Summary of the published works . . . . .	12
2.1	ECI frame definition . . . . .	14
2.2	ECEF frame definition . . . . .	15
2.3	Body-fixed frame definition for Triton-X . . . . .	16
2.4	RTN frame definition . . . . .	17
2.5	Snapshot of the shape of the relative orbit neglecting the orbit drift . . . . .	30
3.1	Illustration of the single difference concept . . . . .	47
3.2	Illustration of the double difference concept . . . . .	48
3.3	Estimation errors for 1 day of the Swarm mission data during the peak of SC24	66
3.4	Precision indicators . . . . .	67
3.5	Estimation errors for 1 day of the Swarm mission data during the nadir of SC24	68
4.1	Graphical representation of the low-thrust guidance strategy . . . . .	77
4.2	Graphical representation of the SCP scheme . . . . .	83
4.3	Feasibility region comparison between the convex QCQP formulation (Problem 4.2) and the QP formulation (Problem 4.3) . . . . .	86
4.4	Control input feasibility regions of Problem 4.4 (red) and Problem 4.5 (blue)	90
4.5	Formation sizes over the 100 randomly chosen formation scenarios . . . . .	95
4.6	Sensitivity of the QCQP formulation (Problem 4.2) to $T_f$ and $T_n$ variations .	95

4.7	Sensitivity of the QP formulation (Problem 4.3) to $T_f$ and $T_n$ variations . . .	96
4.8	Sensitivity of the SOCP formulation (Problem 4.4) to $T_f$ and $T_n$ variations .	97
4.9	Sensitivity of the LP formulation (Problem 4.6) to $T_f$ and $T_n$ variations . . .	98
4.10	Trajectories followed by each of the deputies throughout the coplanar-to-PCO maneuver . . . . .	99
4.11	ROE profile over the coplanar-to-PCO maneuver . . . . .	100
4.12	Control acceleration vector profile over the coplanar-to-PCO maneuver . . . .	100
4.13	Control acceleration vector and slack variables over the coplanar-to-PCO ma- neuver . . . . .	102
4.14	Control acceleration vector and slack variables over the coplanar-to-PCO ma- neuver, Problem 4.4 (SOCP formulation) . . . . .	103
4.15	inter-satellite distances over the coplanar-to-PCO maneuver . . . . .	104
4.16	Expected behavior of the thruster in the presence of the thruster off-periods in Table 4.2 . . . . .	106
4.17	Guidance dimensional ROE profile under operational constraints - Scenario 1	107
4.18	Guidance dimensional ROE profile under operational constraints - Scenario 2	107
4.19	Guidance acceleration profile under operational constraints - Scenario 1 . . .	108
4.20	Guidance acceleration profile under operational constraints - Scenario 2 . . .	108
4.21	Delta-V saving by adopting a different formulation from the QCQP . . . . .	113
4.22	Benchmark summary . . . . .	114
4.23	Summary of benchmark excluding the normalization step . . . . .	116
4.24	Delta-V saving by adopting a different formulation from the QCQP when nor- malization is not employed . . . . .	119
4.25	Feasibility region of the acceleration vector . . . . .	120
4.26	Graphical representation of the SCP scheme with minimum acceleration con- straint . . . . .	122
4.27	Evolution of the acceleration profiles for all deputies across Reconfiguration 3, illustrating the pruning process (Page 1) . . . . .	126

4.27	Evolution of the acceleration profiles for all deputies across Reconfiguration 3, illustrating the pruning process (Page 2)	127
4.28	Violation of the affine relaxation of the collision avoidance constraint	133
4.29	Graphical representation of the SCP scheme in the distributed setting	139
4.30	ROE profiles resulting from solving the distributed guidance problems over Reconfiguration 3	142
4.31	Solve time as the number of deputies increases	143
4.32	Number of required SCP iterations as the number of deputies increases	144
5.1	Evolution of the shrinking horizons	150
5.2	Evolution of the fixed horizons	152
5.3	State error profiles for Reconfiguration 2 in the centralized setting.	156
5.4	Control acceleration profiles for Reconfiguration 2 in the centralized setting.	157
5.5	Baseline distances between satellites for Reconfiguration 2 in the centralized setting.	158
5.6	State error profiles of Sat. A in Reconfiguration 3	160
5.7	Solve times of the guidance problem over Reconfiguration 3	161
6.1	Validation of the surrogate ADCS model through random initial and desired attitudes	170
6.2	Assumed expected values $\bar{\mathbf{y}}$ , $\bar{\mathbf{u}}$ , and $\bar{\delta\mathbf{q}}$ over one prediction horizon	177
6.3	Fitness of the four performance metrics	182
6.4	Overall fitness ( $\phi_{\text{tot}}$ )	183
6.5	Closed loop of the deputy spacecraft	184
6.6	Performance metrics over the Monte Carlo simulation	186
6.7	ROE profile of the selected Monte Carlo run	187
6.8	Trajectory followed by the satellite of the selected Monte Carlo run	188
6.9	Thrust output of the single thruster	189
6.10	Thrust vector in the RTN frame	190
6.11	Attitude profile of the selected Monte Carlo run	191

6.12 Thrust level for the benchmark simulation: <b>(a)</b> thrust vector in the RTN frame	
<b>(b)</b> comparison of the thrust in the normal direction . . . . .	191
6.13 ROE profile of the benchmark maneuver . . . . .	192
6.14 Attitude profile of the benchmark maneuver . . . . .	194
C.1 Monte Carlo results for $n_{\text{dir}} = 12$ . . . . .	206
D.0 Unforced relative orbits for Reconfiguration 0 . . . . .	208
D.1 Unforced relative orbits for Reconfiguration 1 . . . . .	209
D.2 Unforced relative orbits for Reconfiguration 2 . . . . .	210
D.3 Unforced relative orbits for Reconfiguration 3 . . . . .	211
D.4 Unforced relative orbits for Reconfiguration 4 . . . . .	212
D.5 Unforced relative orbits for Reconfiguration 5 . . . . .	213

# List of Tables

1.1	Specifications of Triton-X platform variants . . . . .	5
3.1	Assumed constant statistics used in the simulations . . . . .	65
4.1	Parameters used in Problem 4.6 validation simulation . . . . .	99
4.2	Adopted no-thrust periods for two simulation scenarios . . . . .	105
4.3	Solvers overview . . . . .	110
4.4	Benchmark results . . . . .	111
4.5	Benchmark results when the normalization step is excluded . . . . .	117
4.6	Parameters used in the comparison experiment between the soft- and hard- constrained guidance problems . . . . .	134
4.7	Comparison of the soft- and the hard-constrained problems . . . . .	135
4.8	Comparison of the soft- and the hard-constrained problems . . . . .	137
4.9	Comparison of the soft- and the hard-constrained problems in the distributed setting . . . . .	140
5.1	Results of the proposed control strategies over the two case studies . . . . .	154
5.2	Benchmark summary . . . . .	159
6.1	Monte-Carlo simulation and MPC parameters . . . . .	186
6.2	Performance metrics over 500 simulations . . . . .	187

6.3	Comparison between the proposed and the reference MPCs . . . . .	192
C.1	Monte Carlo results over different values of $n_{\text{dir}}$ . . . . .	205
D.0	Initial and final (required) states for each deputy . . . . .	208
D.1	Initial and final (required) states for each deputy in Reconfiguration 1 . . . .	209
D.2	Initial and final (required) states for each deputy in Reconfiguration 2 . . . .	210
D.3	Initial and final (required) states for each deputy in Reconfiguration 3 . . . .	211
D.4	Initial and final (required) states for each deputy in Reconfiguration 4 . . . .	212
D.5	Initial and final (required) states for each deputy in Reconfiguration 5 . . . .	213
E.1	Benchmark solvers information . . . . .	217

# Acronyms

**3D** 3-Dimensional. 42, 65, 89, 91, 148, 165, 203–205

**ADCS** Attitude Determination and Control System. xxi, 105, 147, 168–170, 178, 225

**AoL** Argument of Latitude. 16, 22

**AoP** Argument of Periapsis. 16

**API** Application Programming Interface. 112, 216, 217

**COP** Convex Optimization Problem. 82, 84

**COTS** Commercial-Off-The-Shelf. 119

**CW** Clohessy–Wiltshire. 26, 27, 29, 36, 58

**DCM** Direction Cosine Matrix. 19, 21, 168

**DDCP** Double Difference Carrier Phase. 48

**DGNSS** Differential Global Navigation Satellite System. 40

**DMB** Digital Mother Board. 198

**e/i** eccentricity/inclination. 29

**ECEF** Earth-Centered-Earth-Fixed. [xix](#), [14](#), [15](#), [17](#), [19](#), [20](#), [23–25](#), [45](#), [55](#), [56](#), [60](#), [65](#)

**ECI** Earth-Centered-Inertial. [xix](#), [13](#), [14](#), [16–21](#), [23–25](#), [226](#)

**EKF** Extended Kalman Filter. [3](#), [7](#), [9](#), [42–44](#), [51](#), [52](#), [54–59](#), [61–64](#), [67](#), [196](#), [200](#), [201](#)

**EP** Electric Propulsion. [70](#)

**ERA** Earth Rotation Angle. [20](#)

**ESA** European Space Agency. [7](#), [10](#), [16](#), [44](#), [64](#), [196](#)

**FF** Formation Flying. [164](#)

**FHMPC** Fixed-Horizon [MPC](#). [8](#), [10](#), [129](#), [148–151](#), [153–163](#), [166](#), [197](#), [198](#)

**FNR** Luxembourg National Research Fund. [4](#)

**GEO** Geostationary Orbit. [29](#)

**GF** Geometry-Free. [50](#)

**GNC** Guidance, Navigation, and Control. [3–7](#), [13](#), [196](#)

**GNSS** Global Navigation Satellite System. [3](#), [5](#), [7](#), [9](#), [40–46](#), [48](#), [50–53](#), [55](#), [62–64](#), [67](#), [69](#), [196](#)

**GPS** Global Positioning System. [3](#), [41](#), [43](#), [53](#)

**GRAPHIC** Group and Phase Ionospheric Correction. [xxviii](#), [49](#), [50](#), [52](#), [68](#)

**GVE** Gauss Variational Equations. [31](#)

**HPC** High-Performance Computer. [179](#), [185](#)

**IPP** Ionospheric Pierce Point. [54](#)

**KF** Kalman Filter. [43](#), [200](#)



**KOZ** Keep-Out Zone. 72, 80, 81, 130, 132, 136, 154, 156, 157

**LAMBDA** Least-squares AMBiguity Decorrelation Adjustment. xxvii, 49

**LEO** Low Earth Orbit. 3–6, 29, 40, 43, 136

**LP** Linear Programming. xx, 73, 75, 84, 88, 95, 96, 98, 99, 102, 107, 110–115, 117, 118, 125, 144, 145, 203, 223

**LTI** Linear Time-Invariant. 26, 27

**LTS** Linear Thin Shell. 53, 54

**LTV** Linear Time-Varying. 34, 119

**MLAMBDA** Modified LAMBDA. 49

**MPC** Model-Predictive Control. xxvi, xxviii, 8, 9, 11, 72–74, 128, 129, 145–149, 151, 153, 154, 157–163, 166, 167, 174–179, 183–186, 188–190, 192–195, 197, 198

**NLP** Non-Linear Programming. 9, 114, 166, 167, 197

**OBC** On-Board Computer. 41, 42

**OOP** Out-Of-Plane. 189, 190, 193

**PCO** Projected Circular Orbit. xx, 99, 100, 102–104, 109, 141, 207–210

**POD** Precise Orbit Determination. 65

**PRN** Pseudo-Random Noise. 59

**QCQP** Quadratically Constrained Quadratic Programming. xix, xx, 73, 75, 81, 86, 95, 96, 98, 107–115, 117–119, 125, 144, 218, 220, 221

**QP** Quadratic Programming. xix, xx, 73, 75, 84–86, 89, 95, 96, 98, 107–115, 117, 118, 125, 144, 220, 221

**RAAN** Right Ascension of the Ascending Node. 17, 22

**RMS** Root Mean Square. 42, 43, 65, 67, 69, 179, 196

**RNG** Random Number Generator. 179, 181

**ROE** Relative Orbital Elements. xx, xxi, 13, 23, 28, 29, 31, 36–38, 73, 77, 79, 80, 82, 84, 90, 94, 100, 102, 105–107, 119, 120, 132, 133, 141, 142, 156, 159, 173–177, 179, 181, 184, 185, 187, 190, 192, 193, 207, 221, 227, 228

**RP-DOP** Relative Position Dilution Of Precision. 66

**RTN** Radial-Transversal-Normal. xix, xxi, xxii, 15, 17, 20, 21, 23, 25, 26, 32, 36, 37, 58, 65, 79, 80, 151, 172, 176, 187, 188, 190, 191

**SAR** Synthetic Aperture Radar. 2, 109

**SC24** 24<sup>th</sup> Solar Cycle. 64, 69, 196

**SCP** Sequential Convex Programming. xix–xxi, 73, 81–83, 103, 104, 111–114, 117, 121, 122, 125–128, 137–140, 142, 143

**SDCP** Single Difference Carrier Phase. 47, 50–52, 64, 68

**SDGF** Single Difference Geometry-Free. 50–52, 54, 63, 64, 68

**SDGR** Single Difference GRAPHIC. 50–52, 54, 63, 64, 68

**SDPR** Single Difference Pseudo-Range. 46, 50, 54, 63, 64

**SHMPC** Shrinking-Horizon MPC. 8, 10, 148–151, 153–159, 161–163, 197

**SOC** Second-Order Cone. 87, 88, 110, 203, 221, 223

**SOCP** Second-Order Cone Programming. xx, 73, 75, 86–88, 95–98, 102, 103, 107–115, 117, 118, 123–125, 130, 133, 137, 144, 197, 221, 222, 224

**STEC** Slant Total Electron Content. 45, 53, 59

**STM** State Transition Matrix. [27](#), [34](#), [57–59](#)

**TBP** Two-Body Problem. [15](#), [18](#)

**TECU** Total Electron Content Unit. [53](#)

**TT** Terrestrial Time. [13](#)

**V&V** Verification and Validation. [198](#)

**VTEC** Vertical Total Electron Content. [53](#), [59](#)

## Part I

# Background

# Introduction

The emergence of the *new space* era has brought significant changes to satellite mission design, favoring smaller, more versatile, and economically viable satellites. Unlike traditional missions that centered around a single, complex, and costly spacecraft, contemporary missions now often deploy formations of lightweight, simpler, and more affordable satellites operating in tandem. This coordinated approach offers multiple benefits, including wider coverage areas, more frequent data collection, and enhanced system redundancy [1]. Together, these advantages make satellite formation flying a critical strategy in modern missions, supporting robust applications in Earth observation, environmental monitoring, and communication services [2, 3].

Satellites flying together to achieve a single mission objective are often referred to as a “formation” or a “constellation”. The two terms are sometimes used interchangeably, yet they represent distinct concepts with different operational requirements, leading to potential misunderstandings. A *formation* refers to a tightly controlled set of satellites maintaining precise relative positions with respect to one another to function as a coordinated system. Formations are often utilized for missions where spatial geometry is crucial, such as multi-static [Synthetic Aperture Radar \(SAR\)](#) imaging or scientific observations, where satellites need to sustain fixed distances relative to each other. A *constellation*, by contrast, describes

a group of satellites strategically distributed across various orbital planes to achieve extensive coverage, usually on a global or near-global scale. Each satellite in a constellation operates independently rather than maintaining close proximity to others. Constellations are common in communications and navigation systems, such as the Starlink and the [Global Positioning System \(GPS\)](#) constellations, where satellites in complementary orbits cover different areas to ensure seamless service availability without requiring the real-time precise coordination needed in formations.

A fundamental technology enabling satellite formations is the [Guidance, Navigation, and Control \(GNC\)](#) system. The [GNC](#) system serves as the backbone for coordinating multiple satellites, whether they operate in close proximity or are widely separated. It is responsible for reconfiguring the formation and maintaining the designated positions of individual satellites within the formation through constant feedback on the formation states such as the absolute and relative positions and velocities of satellites.

Effective formation flying relies heavily on accurate relative navigation, serving as an essential feedback element in the closed control loop. Satellites in [Low Earth Orbit \(LEO\)](#) often depend on [Global Navigation Satellite System \(GNSS\)](#) signals to obtain real-time positioning and velocity data. For single-satellite missions, absolute positioning, estimating the satellite's position relative to Earth's center, is sufficient. However, in multi-satellite missions, relative positioning becomes essential to facilitate the precise coordination needed for formation flying. For formations with medium-large inter-satellite distances, [GNSS](#) receivers are ideal, providing high accuracy and reliability in relative position estimation [4, 5]. This makes [GNSS](#)-based systems preferable to vision-based sensors, such as cameras or LiDAR, which may not function effectively over large baselines. [GNSS](#)-based [Extended Kalman Filter \(EKF\)](#) have demonstrated millimeter-level accuracy for formations with inter-satellite distances ranging from 1 – 10 km, making them highly effective for close formations. When dual-frequency receivers are used to correct for ionospheric delays, [GNSS](#)-based solutions can also be effective for wider-spaced formations, ensuring the accuracy needed for complex multi-satellite maneuvers.

Another vital element is the guidance system, which calculates the optimal trajectory for each satellite, ensuring they reach and maintain their target configurations within mission constraints. This involves trajectory planning to account for various factors, such as fuel efficiency, collision avoidance, and the impact of external forces like gravitational perturbations and atmospheric drag. Once the guidance system determines the trajectories, the control system implements them, using the propulsion subsystem to maintain accurate positioning within the formation. This process involves managing thrust levels to conduct precise maneuvers, especially during formation reconfigurations where satellites must change their relative positions.

This research is conducted as part of the AuFoSat project, a collaborative effort between the University of Luxembourg and LuxSpace, aiming to develop an autonomous [GNC](#) toolbox to enable formation flying missions, particularly those using LuxSpace’s Triton-X platform. The project’s primary goal is to create a comprehensive [GNC](#) library that supports all aspects of formation flying, including state estimation, trajectory planning, and control for autonomous operation. The AuFoSat project is funded by the [Luxembourg National Research Fund \(FNR\)](#) under the umbrella of the BRIDGES program with grant reference BRIDGES/19/MS/14302465.

## 1.1 Triton-X

The *Triton-X* platform, developed by LuxSpace, is a multi-mission microsatellite designed for a variety of applications in [LEO](#), including communication, Earth observation, scientific research, and space situational awareness. Triton-X features a modular design that allows users to tailor the satellite to specific mission requirements, accommodating different payloads, mission lifetimes, and operational needs. This flexibility supports both commercial and scientific missions, making Triton-X a versatile option for diverse mission types.

The Triton-X product line includes three variants: Light, Medium, and Heavy, as detailed in Table 1.1. For the purposes of this research, the Triton-X Heavy model is used and will be referred to simply as Triton-X.

Table 1.1: Specifications of Triton-X platform variants

	Light	Medium	Heavy
Payload mass	Up to 15 kg	Up to 45 kg	Up to 90 kg
Payload size	Up to 12 U	13 - 50 U	50 - 280 U
Spacecraft mass	40 - 65 kg	75 - 135 kg	130 - 250 kg

Source: Specifications from <https://luxspace.lu/smallsats/>, accessed on 12/11/2024.

A notable feature of Triton-X is its electric propulsion system, which extends mission lifetimes and supports advanced orbit maneuvers. Electric propulsion enables not only precise positioning but also orbit maintenance and reconfiguration for formation flying missions. By implementing responsible deorbiting strategies, Triton-X also aligns with sustainable space practices, supporting a cleaner orbital environment and reducing the risk of space debris.

## 1.2 Aims and scope of the thesis

The primary aim of this thesis is to develop a **GNC** toolbox tailored for formations of Triton-X or similar micro satellite platforms. While Triton-X provides significant potential for **LEO** missions, certain design features present specific challenges for formation flying, particularly in navigation and control.

A key limitation of Triton-X in terms of navigation is its reliance on a single-frequency **GNSS** receiver, without supplemental visual sensors like cameras or LiDAR. While single-frequency **GNSS** receivers are adequate for closely spaced formations [6], they present challenges for formations with larger inter-satellite distances, where ionospheric and environmental biases introduce significant measurement errors. This research addresses these limitations by developing filtering and estimation techniques designed to mitigate ionospheric errors, im-



proving the accuracy of relative navigation for formations with inter-satellite distances in the range of tens or hundreds of kilometers. The development of relative navigation filters for satellites flying in close formations is beyond the scope of this thesis, as it has been extensively studied in the literature [4, 7, 8].

Beyond navigation, the Triton-X platform presents unique guidance and control challenges due to its propulsion configuration: a single, un-gimbaled electric thruster. This underactuated setup requires an attitude maneuver before each thrust activation to direct the nozzle along the desired thrust direction. While efficient, this configuration complicates formation control, as each thrust arc must be meticulously managed to ensure alignment with formation requirements. Traditional guidance methods are not well-suited to such underactuated systems, especially when specific operational constraints must be respected. These constraints include:

- Thrust limits and operational windows, ensuring propulsion is applied only when suitable conditions are met;
- Collision avoidance within the formation, critical for safe reconfiguration maneuvers;
- Timing for attitude adjustments, as each realignment of the nozzle must be completed before subsequent thrust activations.

This thesis focuses on developing guidance and control strategies for Triton-X and similar platforms, aiming to achieve Delta-V optimality while integrating all necessary operational constraints. The resulting GNC toolbox is designed to enable precise, autonomous formation flying, supporting both current and future missions in LEO. Note that in the subsequent chapters, the terms fuel-optimal and Delta-V-optimal are used interchangeably since they do refer to the same thing for satellites which are equipped with a single thruster nozzle [9].

### 1.3 Thesis outline

The thesis is organized into two parts, each containing a series of chapters. The first part provides the foundational background necessary to understand the challenges addressed in the thesis, while the second part delves into the proposed solutions and contributions to these challenges.

The first part begins with an introductory chapter that outlines the primary domain of the thesis, offering context and motivation for the research conducted. Following the introduction, Chapter 2 presents the mathematical models that underpin the development of the GNC algorithms discussed in the second part. The first part concludes with these two chapters. It is worth noting that starting from the second part, each chapter includes a brief literature review relevant to the problem it addresses, which provides context for the specific contributions in that chapter.

The second part opens with Chapter 3, where a classical EKF algorithm is proposed to address GNSS-based relative navigation between two widely separated cooperative spacecraft, each equipped with only a single-frequency GNSS receiver. The primary challenge in this problem lies in resolving the ionospheric delay that affects the GNSS signals received by the spacecraft. Unlike traditional approaches, this chapter introduces a more precise ionospheric delay model that has not been previously applied to baseline vector estimation. Additionally, this chapter proposes a carefully chosen set of state variables and observables tailored to optimize filter performance for the problem considered. Validation of the proposed navigation filter is conducted using real sensor data from the European Space Agency's Swarm mission, collected during two distinct solar cycles representing favorable and unfavorable atmospheric conditions. Results indicate that, due to the adoption of the refined ionospheric model, the filter's performance remains consistent across different atmospheric conditions.

Chapter 4 addresses the optimization of relative trajectories for reconfiguring a satellite

formation consisting of an arbitrary number of satellites, initially using a centralized approach. At the beginning of the chapter, the guidance problem is formulated as a non-convex trajectory optimization, which is subsequently convexified and refined through various modifications aimed at minimizing both total Delta-V and solution time. Four distinct convex formulations are introduced and benchmarked across four different reconfiguration scenarios using 15 of the most common solvers to identify the most efficient formulation. These formulations share the same set of constraints.

In the same chapter, and due to the inherent non-convexity of the minimum thrust constraint imposed by the hardware of Triton-X, this constraint is convexified and supplemented by an *acceleration pruning* algorithm to facilitate solving the problem. Additionally, techniques for softening constraints are proposed to mitigate the risk of infeasibility by allowing small, controlled violations. These techniques are especially critical for enabling autonomy in the overall guidance and control system, particularly when the guidance strategies are employed as a prediction function within an [Model-Predictive Control \(MPC\)](#) framework, where avoiding infeasibility is crucial for repeated problem-solving.

While centralized guidance schemes typically offer better fuel efficiency than distributed ones, they face scalability challenges as the number of satellites increases. To address this limitation, the latter part of [Chapter 4](#) adapts the guidance strategies for distributed systems, which are more suitable for formations with larger numbers of satellites. Results show that, unlike the centralized approach, where computation time grows exponentially with the number of satellites, the distributed solution exhibits linear scalability.

The guidance strategies developed in [Chapter 4](#) function as open-loop control systems that require sensor feedback and navigation algorithms to close the control loop. [Chapter 5](#) introduces control strategies specifically designed for Triton-X satellite formations. This chapter incorporates the most efficient guidance strategy from [Chapter 4](#) within two [MPC](#) schemes: the [Shrinking-Horizon MPC \(SHMPC\)](#) and the [Fixed-Horizon MPC \(FHMPC\)](#). Performance

evaluations demonstrate that these proposed guidance and control strategies outperform classical [MPC](#) methods from the literature, offering superior fuel efficiency and more precise final formation configurations.

The final chapter, Chapter 6, extends the toolbox to include absolute/relative orbit maintenance capabilities using formation flying techniques. Unlike the previous chapters, which prioritize Delta-V optimization, this chapter focuses on time-optimal maneuvers. The proposed algorithm employs a [Non-Linear Programming \(NLP\)](#) optimization within an [MPC](#) framework, integrating attitude dynamics with relative orbital dynamics. Benchmarking results show that the proposed scheme achieves significant improvements over a reference controller from the literature, demonstrating enhanced performance for absolute orbit control.

## 1.4 Contributions to the state of the art

The contributions of each chapter to the state of the art are outlined at the beginning of their respective chapters. For completeness, they are summarized here for the chapters included in the second part of the thesis.

The main contributions of Chapter 3 are summarized as follows:

- A tailored set of [GNSS](#) observables is proposed for the problem of baseline vector estimation for two widely separated spacecraft equipped with single-frequency [GNSS](#) receivers. A detailed justification is provided for the relevance of each observable to the problem;
- A precise bi-linear ionospheric model is utilized in the context of [GNSS](#)-based relative navigation to mitigate the effects of ionospheric delay, which is significant for largely separated receivers;
- A specific set of state variables to be estimated by the [EKF](#) is selected based on the synthesized set of measurements and the adopted ionospheric model;

- The relative navigation scheme is validated using sensor data from the [European Space Agency \(ESA\)](#) Swarm mission [10] in a simulated real-time setting.

Chapter 4 is a core chapter of this thesis and provides the most significant contributions to the state of the art. These contributions are summarized as follows:

- Presentation of four different convex formulations of the formation reconfiguration guidance problem, which vary in terms of fuel efficiency and computational complexity.
- Execution of a benchmark experiment across more than a dozen solvers, comparing the performance of the four convex formulations;
- Development of guidelines for embedding the guidance algorithms in space-borne applications, specifically addressing typical reconfiguration challenges in remote sensing missions;
- Introduction of the minimum thrust constraint using an affine convexification method, along with a pruning algorithm to improve feasibility;
- Implementation of methods to soften hard-constrained problems, reducing the likelihood of infeasibility during closed-loop control system operations;
- Exploration of the guidance problem as a distributed trajectory optimization problem and the development of coordination algorithms to ensure collision safety throughout the reconfiguration maneuver.

The primary contribution of Chapter 5 lies in the tailored implementation of the [SHMPC](#) and [FHMPC](#) methods to address the specific challenges of multi-satellite formation reconfiguration, while the contributions of Chapter 6 are summarized as follows:

- Accounting for the coupled attitude and relative orbit control problem within the control loop;
- Introducing a surrogate model for the attitude control system, which can be tuned to emulate the behavior of various attitude control systems;

- Proposing an MPC scheme for satellites with unidirectional propulsion systems, allowing thrust to be applied during slew maneuvers.

The aforementioned contributions were distributed across five journal publications, which form the foundation of this thesis. A graphical summary of these published works, illustrating how the key contributions are distributed across the publications, is presented in Fig. 1.1.

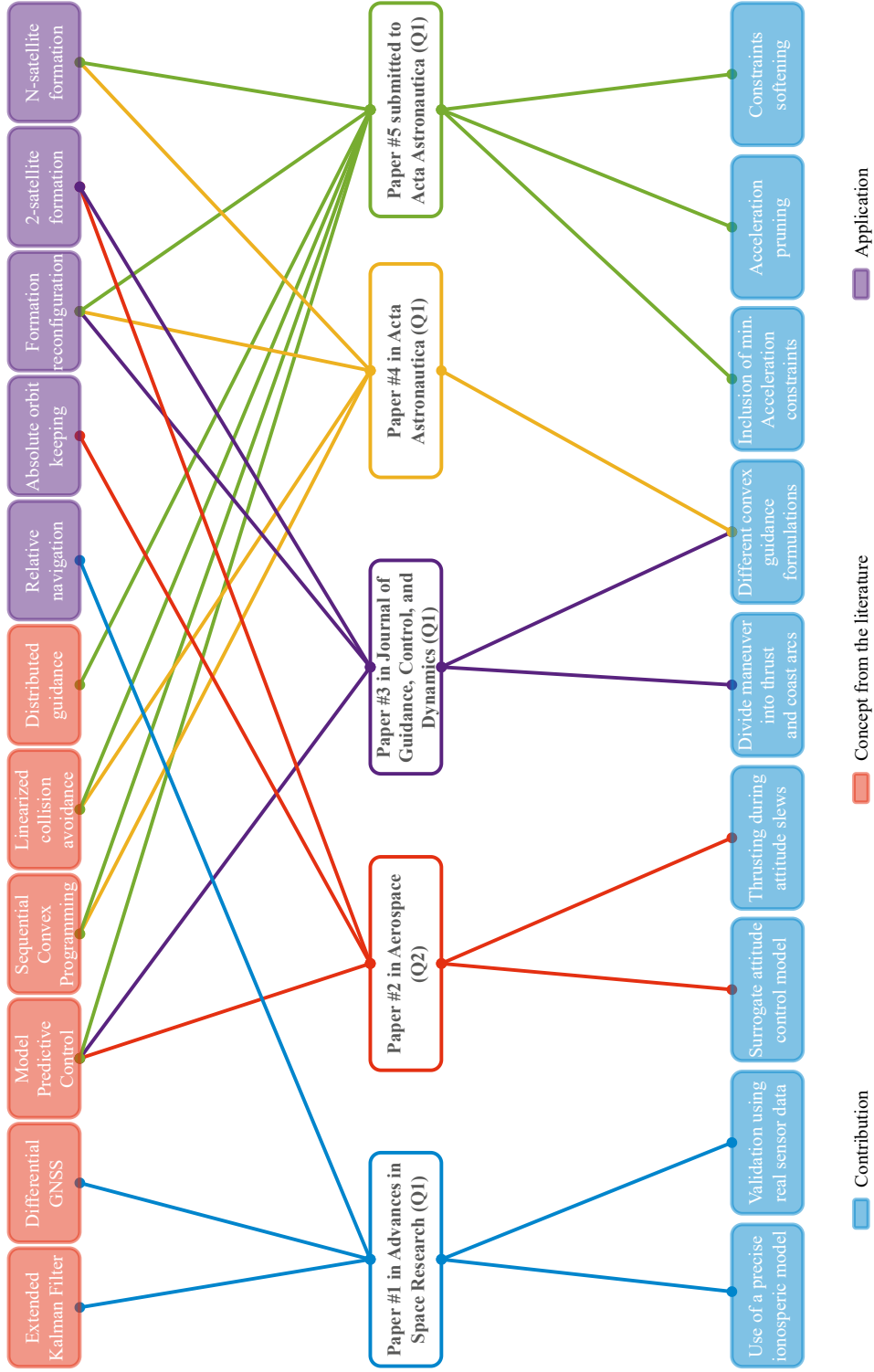


Figure 1.1: Summary of the published works

Paper #1 refers to [11], paper #2 refers to [12], paper #3 refers to [13], paper #4 refers to [14], and paper #5 refers to [15].

## Mathematical models

In this chapter, the foundational mathematical models for the design of a [Guidance, Navigation, and Control \(GNC\)](#) system for satellite formations are introduced. Initially, a definition of the relevant reference frames is provided. This is followed by a brief description of the absolute dynamics of a spacecraft, specifically in relation to a natural body, which in the context of this thesis is the Earth. Subsequently, the development of the relative dynamics, which describe the motion of one spacecraft relative to another, is presented. The relative dynamics are formulated using the relative Cartesian state vector, as well as the quasi-non-singular [Relative Orbital Elements \(ROE\)](#) parameterization.

### 2.1 Reference frames

#### 2.1.1 Earth-Centered-Inertial frame

The [Earth-Centered-Inertial \(ECI\)](#) reference frame is defined by having its center at the Earth's center, with its x-axis along the mean equinox on 01/01/2000 at 12:00 [Terrestrial Time \(TT\)](#), the z-axis is aligned to the mean north celestial pole at the same epoch, while the y-axis completes the right-handed triad (see Fig. 2.1). The [ECI](#) frame defined here is also referred to as the J2000, the J2K, or the EME2000 frame. The set which collates all the vectors which are expressed in the inertial frame is denoted by  $\mathbb{F}^i$ , where  $\mathbb{F}^i \subset \mathbb{R}^3$ . A vector



expressed in the **ECI** frame is signified by the superscript  $i$ , e.g.,  $\mathbf{v}^i \in \mathbb{F}^i$ .

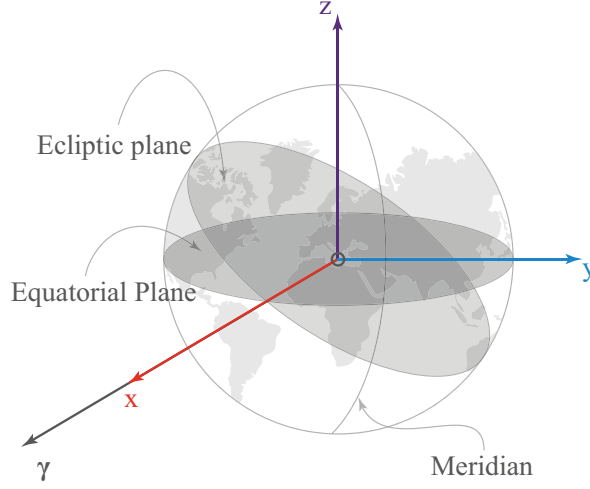


Figure 2.1: **ECI** frame definition

### 2.1.2 Earth-Centered-Earth-Fixed frame

The **Earth-Centered-Earth-Fixed (ECEF)** frame is also Earth-centered, however, it is not an inertial frame. It has its x-axis pointing to the Greenwich meridian, z-axis directed along the mean rotational axis of the Earth, and y-axis completing the right-handed system (refer to Fig. 2.2). The set  $\mathbb{F}^f \subset \mathbb{R}^3$  is that of all vectors which are expressed in the **ECEF** frame. A superscript  $f$  on top of a vector indicates that it is expressed in the **ECEF** frame, e.g.,  $\mathbf{v}^f \in \mathbb{F}^f$ .

### 2.1.3 Satellite-body-fixed frame

The body-fixed reference frame is a local reference frame to the spacecraft in question with the origin at the satellite's center of mass. One common choice of the directions of the three axes is along the three principal axes of inertia of the satellite. The body-fixed frame of Triton-X is depicted in Fig. 2.3, where the truster nozzle extends along its z-direction. The set of all vectors that are expressed in the body-fixed frame of a satellite is denoted by  $\mathbb{F}^b$ , where  $\mathbb{F}^b \subset \mathbb{R}^3$ . A vector expressed in the body-fixed frame of a satellite will have the superscript  $b$ , e.g.,  $\mathbf{v}^b \in \mathbb{F}^b$ .

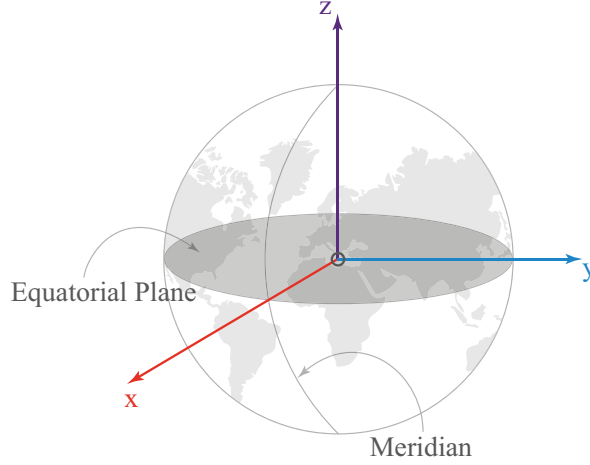


Figure 2.2: ECEF frame definition

#### 2.1.4 Radial-Transversal-Normal frame

The purpose of the **Radial-Transversal-Normal (RTN)** frame is to describe the motion of one spacecraft with respect to a reference one. The **RTN** frame is centered at the mass center of the chief satellite (also referred to as the reference satellite or the formation center), which can be either a real central satellite or a virtual point orbiting the Earth. The x-axis of the **RTN** frame is defined to be along the position vector of the chief satellite, pointing away from the Earth, the z-axis is directed along the chief's orbital angular momentum vector, and the y-axis completes the right-handed set (see Fig. 2.4). Similar to the previously defined reference frames, the set which collates all the vectors expressed in the **RTN** frame is referred to as  $\mathbb{F}^r$ , where  $\mathbb{F}^r \subset \mathbb{R}^3$ . In the following discussions, a vector expressed in the **RTN** frame is signified by the  $r$  superscript, e.g.,  $\mathbf{v}^r \in \mathbb{F}^r$ .

## 2.2 Absolute orbital dynamics

In this section, the orbital motion of an artificial object around the Earth is mathematically formulated as a **Two-Body Problem (TBP)**. Two different parameterization sets are used to describe the motion of the satellite in orbit, the Cartesian state vector and the orbital

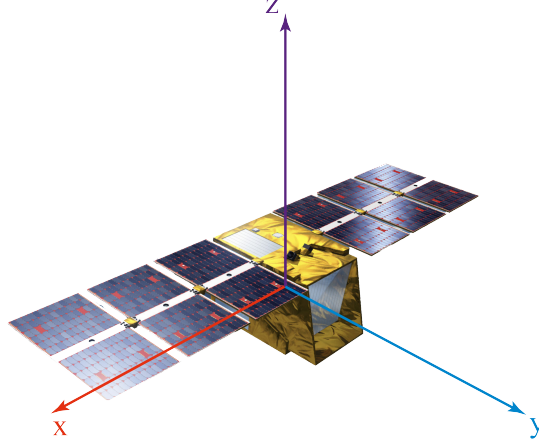


Figure 2.3: Body-fixed frame definition for Triton-X

Satellite image credits: [European Space Agency \(ESA\)](#)

elements. Concretely, the Cartesian elements are collated in a vector,  $\mathbf{x}^{(\cdot)}$ , such that,

$$\mathbf{x}^{(\cdot)} = \begin{bmatrix} \mathbf{r}^{(\cdot)} \\ \mathbf{v}^{(\cdot)} \end{bmatrix} \quad (2.1)$$

where  $\mathbf{r}^{(\cdot)} \in \mathbb{F}^{(\cdot)}$  is the instantaneous position vector of the satellite,  $\mathbf{v}^{(\cdot)} \in \mathbb{F}^{(\cdot)}$  is the instantaneous velocity vector as seen by an observer standing on frame  $(\cdot)$ , where the vectors are expressed on the same frame, with  $(\cdot)$  being a placeholder for the frame on which the states are expressed, which can be one of the frame defined in Section 2.1.

The following set of orbital elements can be used to describe the orbital motion of a satellite, which is under the gravitational influence of a large celestial body such as the Earth, in a planet centered reference frame. In contrast to the Cartesian states, the orbital elements are always described in this thesis with respect to the [ECI](#) frame, and are formally introduced as,

$$\boldsymbol{\alpha} = \begin{bmatrix} a & \theta & e_x & e_y & i & \Omega \end{bmatrix}^{\top}, \quad (2.2)$$

where  $a$  is the semi-major axis,  $\theta = M + \omega$  is the mean [Argument of Latitude \(AoL\)](#), with  $M$  being the mean anomaly and  $\omega$  being the [Argument of Periapsis \(AoP\)](#). Moreover,

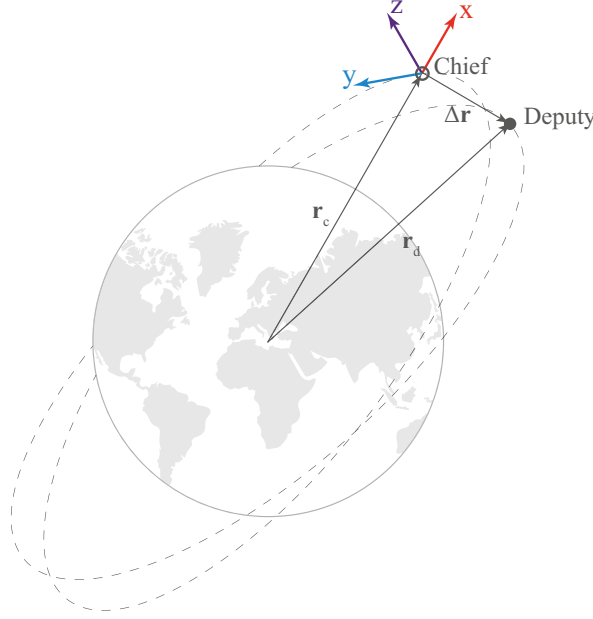


Figure 2.4: RTN frame definition

$\mathbf{e} = \begin{bmatrix} e_x & e_y \end{bmatrix}^T = \begin{bmatrix} e \cos \omega & e \sin \omega \end{bmatrix}^T$  is the eccentricity vector with  $e$  being the eccentricity of the orbit,  $i$  is the orbital inclination, and  $\Omega$  is the [Right Ascension of the Ascending Node \(RAAN\)](#). There exist nonlinear mappings which transform from a Cartesian state vector (expressed in the [ECI](#) frame) into an orbital elements vector and vice versa [16]. When the proper mapping is applied to the Cartesian state vector of the satellite at a certain epoch, the instantaneous osculating orbital elements are produced, which, in the discussions to follow, are denoted by  $\tilde{\boldsymbol{\alpha}}$ . Conversely, the mean orbital elements, denoted in the rest of the thesis by  $\boldsymbol{\alpha}$ , are the one-orbit averaged elements, where the short- and long-term oscillations generated by the  $J_2$  harmonic of the Earth gravitational potential are removed. The transformations in [17] are utilized to perform the mean/osculating elements mapping.

### 2.2.1 Dynamics of the Cartesian state vector

This section introduces the equations which describes the evolution of the Cartesian state vector as seen by observers attached to the [ECI](#), to the [ECEF](#), and to the [RTN](#) frames.

## ECI frame

The dynamics of a satellite that moves under the gravity field of the Earth can be formulated as a TBP [16] such that,

$$\ddot{\mathbf{r}}^i = -\frac{\mu_{\oplus}}{r^3}\mathbf{r}^i + \mathbf{u}^i + \mathbf{u}_{J_2}^i + \mathbf{w}_{\mathbf{v}}^i, \quad (2.3)$$

where  $\mathbf{r}^i$  is the position vector of the satellite in the ECI frame,  $r = \|\mathbf{r}^i\|_2$  is the magnitude of position vector,  $\mu_{\oplus}$  is the Earth's standard gravitational parameter,  $\mathbf{u}^i$  is the known control acceleration vector,  $\mathbf{u}_{J_2}^i$  is the  $J_2$  perturbing acceleration vector, and  $\mathbf{w}_{\mathbf{v}}^i$  collates all the unmodeled perturbing accelerations.

Following the definition of the Cartesian elements in the Eq. (2.1), Eq. (2.3) can be easily transformed to,

$$\dot{\mathbf{x}}^i = \begin{bmatrix} \mathbf{v}^i \\ -\frac{\mu_{\oplus}}{r^3}\mathbf{r}^i + \mathbf{u}^i + \mathbf{u}_{J_2}^i + \mathbf{w}_{\mathbf{v}}^i \end{bmatrix}, \quad (2.4)$$

where, again,  $\mathbf{v}^i$  is the velocity vector of the satellite expressed in, and as seen by an observer attached to, the ECI frame.

According to [18], the perturbing acceleration vector resulting from the effect of the  $J_2$  zonal harmonic can be modelled for a satellite with a position vector  $\mathbf{r}^i = [x^i \ y^i \ z^i]^T$  as,

$$\mathbf{u}_{J_2}^i = -\underbrace{\frac{3\mu_{\oplus}J_2R_{\oplus}^2}{2r^7}}_{\psi} \underbrace{\begin{bmatrix} r^2 - 5z^{i2} & 0 & 0 \\ 0 & r^2 - 5z^{i2} & 0 \\ 0 & 0 & 3r^2 - 5z^{i2} \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix}, \quad (2.5)$$

where  $J_2$  is the coefficient of the second zonal harmonic, and  $R_{\oplus}$  is the equatorial radius of the Earth.

## ECEF frame

The acceleration vector of the satellite as seen by an observer attached to the ECEF frame can be formulated using the transport theorem [19] in conjunction with Eq. (2.3) as,

$$\ddot{\mathbf{r}}^f = -\frac{\mu_{\oplus}}{r^3}\mathbf{r}^f - \boldsymbol{\omega}_{\oplus}^f \times \mathbf{v}^f - \boldsymbol{\omega}_{\oplus}^f \times \boldsymbol{\omega}_{\oplus}^f \times \mathbf{r}^f + \mathbf{u}^f + \mathbf{u}_{J_2}^f + \mathbf{w}_v^f, \quad (2.6)$$

which, in turn, can be rewritten as,

$$\dot{\mathbf{x}}^f = \begin{bmatrix} \mathbf{v}^f \\ -\frac{\mu_{\oplus}}{r^3}\mathbf{r}^f - 2\boldsymbol{\omega}_{\oplus}^f \times \mathbf{v}^f - \boldsymbol{\omega}_{\oplus}^f \times \boldsymbol{\omega}_{\oplus}^f \times \mathbf{r}^f + \mathbf{u}^f + \mathbf{u}_{J_2}^f + \mathbf{w}_v^f \end{bmatrix}, \quad (2.7)$$

where  $\mathbf{r}^f$  is the position vector of the satellite,  $r$  is its magnitude,  $\boldsymbol{\omega}_{\oplus}^f$  is the angular velocity vector of the ECEF frame with respect to the ECI frame,  $\mathbf{u}^f$  is the known input acceleration vector,  $\mathbf{u}_{J_2}^f$  is the  $J_2$  perturbing acceleration vector, and  $\mathbf{w}_v^f$  collates all the unmodeled disturbance accelerations, all expressed in the ECEF frame. It is to be emphasized that  $\mathbf{v}^f$  is the velocity vector as seen by an observer fixed to the ECEF frame, and unlike the other vectors in (2.7), cannot be simply obtained by rotating  $\mathbf{v}^i$  to  $\mathbb{F}^f$ . It rather can be obtained by applying the transport theorem as,

$$\mathbf{v}^f = \mathbf{A}^{fi}\mathbf{v}^i - \boldsymbol{\omega}_{\oplus}^f \times \mathbf{r}^f. \quad (2.8)$$

A simplified version of Eq. (2.6) can be obtained by ignoring the Earth's axial precession and approximating the rotation of the ECEF frame to be only around the z-axis with respect to the ECI frame, i.e.,  $\boldsymbol{\omega}_{\oplus}^f \approx \boldsymbol{\omega}_{\oplus}^i \approx \begin{bmatrix} 0 & 0 & \omega_{\oplus} \end{bmatrix}^T$ , with  $\omega_{\oplus} \approx 7.292115 \cdot 10^{-5}$  rad/sec being the Earth's spinning speed. Using this approximation, the Direction Cosine Matrix (DCM) that transforms a vector from the ECI frame to the ECEF frame,  $\mathbf{A}^{fi} = \mathbf{A}^{ifT}$  can be written as,

$$\mathbf{A}^{fi} \approx \begin{bmatrix} \cos(\theta_{\oplus}) & \sin(\theta_{\oplus}) & 0 \\ -\sin(\theta_{\oplus}) & \cos(\theta_{\oplus}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.9)$$

where  $\theta_\oplus$  is the [Earth Rotation Angle \(ERA\)](#).

Using (2.5) in conjunction with (2.9), and emphasizing that  $z^i \approx z^f$ , with  $\mathbf{r}^f = \begin{bmatrix} x^f & y^f & z^f \end{bmatrix}^\top$  being the position vector of the satellite in the [ECEF](#) frame, the  $J_2$  perturbing acceleration vector in the [ECEF](#) frame can be written as,

$$\mathbf{u}_{J_2}^f \approx -\psi \mathbf{A}^{fi} \mathbf{M} \mathbf{A}^{fi\top} \mathbf{r}^f. \quad (2.10)$$

Due to the structure of  $\mathbf{A}^{fi}$  and  $\mathbf{M}$ , the term  $\mathbf{A}^{fi} \mathbf{M} \mathbf{A}^{fi\top}$  reduces to  $\mathbf{M}$ , which results in,

$$\mathbf{u}_{J_2}^f \approx -\psi \mathbf{M} \mathbf{r}^f = -\frac{3\mu_\oplus J_2 R_\oplus^2}{2r^7} \begin{bmatrix} r^2 - 5z^{f2} & 0 & 0 \\ 0 & r^2 - 5z^{f2} & 0 \\ 0 & 0 & 3r^2 - 5z^{f2} \end{bmatrix} \begin{bmatrix} x^f \\ y^f \\ z^f \end{bmatrix}. \quad (2.11)$$

### RTN frame

For an observer attached to the [RTN](#) frame, and utilising the transport theorem [19], the acceleration vector of the satellite appears as,

$$\ddot{\mathbf{r}}^r = -\frac{\mu_\oplus}{r^3} \mathbf{r}^r - \boldsymbol{\omega}_r^r \times \mathbf{v}^r - \dot{\boldsymbol{\omega}}_r^r \times \mathbf{r}^r + \mathbf{u}^r + \mathbf{u}_{J_2}^r + \mathbf{w}_v^r, \quad (2.12)$$

which, in turn, can be rewritten as,

$$\dot{\mathbf{x}}^r = \begin{bmatrix} \mathbf{v}^r \\ -\frac{\mu_\oplus}{r^3} \mathbf{r}^r - 2\boldsymbol{\omega}_r^r \times \mathbf{v}^r - \dot{\boldsymbol{\omega}}_r^r \times \mathbf{r}^r + \mathbf{u}^r + \mathbf{u}_{J_2}^r + \mathbf{w}_v^r \end{bmatrix}, \quad (2.13)$$

where  $r = \|\mathbf{r}^r\|_2$  and  $\boldsymbol{\omega}_r^r = \begin{bmatrix} 0 & 0 & \dot{\nu}_c \end{bmatrix}^\top$  is the angular velocity vector of the [RTN](#) frame with respect to the [ECI](#) frame, expressed in the [RTN](#) frame. Moreover,  $\nu_c$  is the instantaneous true anomaly of the chief spacecraft, and  $\dot{\nu}_c$  is its instantaneous rate of change, which can be calculated [20] as,

$$\dot{\nu}_c = \frac{h_c}{r_c^2}, \quad (2.14)$$

where  $h_c = \|\mathbf{h}_c^i\|_2$  is the magnitude of the specific angular momentum vector of the chief's orbit, and  $r_c$  is the magnitude of the position vector of the chief. The specific angular momentum vector of the chief can be obtained through,

$$\mathbf{h}_c^i = \mathbf{r}_c^i \times \mathbf{v}_c^i. \quad (2.15)$$

with  $\mathbf{r}_c^i$  and  $\mathbf{v}_c^i$  being the position and velocity vectors of the chief, expressed, and as seen by an observer, in the [ECI](#) frame.

It is important to emphasize that  $\dot{\mathbf{x}}^r$  and  $\mathbf{v}^r$  in Eq. (2.13) are the change rate of the relative Cartesian elements vector and the relative velocity vector, respectively, as seen by an observer attached to the [RTN](#) frame, and expressed in the same frame. All the other vectors in Eq. (2.13), i.e.,  $\mathbf{r}^r$ ,  $\boldsymbol{\omega}_r^r$ ,  $\mathbf{u}^r$ ,  $\mathbf{u}_{J_2}^r$ , and  $\mathbf{w}_v^r$ , can be thought of as rotated versions of their  $\mathbb{F}^i$  counterparts, e.g.,

$$\mathbf{r}^r = \mathbf{A}^{ri} \mathbf{r}^i, \quad (2.16)$$

where  $\mathbf{A}^{ri} = \mathbf{A}^{ir\top}$  is the [DCM](#) that transforms a vector from  $\mathbb{F}^i$  to  $\mathbb{F}^r$ . The [DCM](#),  $\mathbf{A}^{ri}$ , is defined as,

$$\mathbf{A}^{ri} = \begin{bmatrix} \hat{\mathbf{r}}_c^i & \hat{\mathbf{h}}_c^i \times \hat{\mathbf{r}}_c^i & \hat{\mathbf{h}}_c^i \end{bmatrix}^\top, \quad (2.17)$$

with  $\hat{\mathbf{r}}_c^i$  being the unit vector in the direction of the position vector of the chief, and  $\hat{\mathbf{h}}_c^i$  being the unit vector in the direction of the orbital angular momentum vector.

### 2.2.2 Dynamics of the mean orbital elements

In this section, the dynamics of the mean orbital elements are introduced. These mean orbital elements can be transformed into their osculating counterparts at any given epoch using the transformation in [17]. An inverse transformation, i.e., from osculating to mean orbital elements, is also discussed in [17].

The unperturbed variational equations for the mean orbital elements of a satellite flying



under the gravitational field of the Earth, can be written [21] as,

$$\dot{\mathbf{a}} = \begin{bmatrix} 0 & n & 0 & 0 & 0 & 0 \end{bmatrix}^T, \quad (2.18)$$

where  $n = \sqrt{\frac{\mu_{\oplus}}{a^3}}$  is the mean motion of the satellite.

While it does not affect the semi-major axis, the inclination, or the eccentricity of the orbit, the second zonal harmonic,  $J_2$ , introduces a secular drift on the mean RAAN (nodal precession), the mean AoL (apsidal precession), and on the mean mean anomaly, such that the dynamics of the classical mean Keplerian elements including the effect  $J_2$  could be written [22] as,

$$\begin{aligned} \dot{a} &= 0 \\ \dot{e} &= 0 \\ \dot{i} &= 0 \\ \dot{\Omega} &= -3n\gamma \cos(i), \\ \dot{\omega} &= \frac{3}{2}n\gamma (5 \cos^2(i) - 1), \\ \dot{M} &= n + \frac{3}{2}n\gamma\eta (3 \cos^2(i) - 1), \end{aligned} \quad (2.19)$$

where,

$$\gamma = \frac{1}{2}J_2 \left( \frac{R_{\oplus}}{p} \right)^2, \quad (2.20)$$

where  $p = a\eta^2$  is the semi-latus rectum of the elliptic orbit, and  $\eta = \sqrt{1 - e^2}$  is the eccentricity factor.

The dynamics of the orbital elements vector can be deduced by taking the time derivative

of both sides of Eq. (2.2) while leveraging Eq. (2.19),

$$\dot{\boldsymbol{\alpha}} = \begin{bmatrix} \dot{a} \\ \dot{\theta} \\ \dot{e}_x \\ \dot{e}_y \\ \dot{i} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \dot{a} \\ \dot{M} + \dot{\omega} \\ -\dot{\omega}e_y \\ \dot{\omega}e_x \\ \dot{i} \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} 0 \\ n + \frac{3}{2}n\gamma\eta(3\cos^2(i) - 1) + \frac{3}{2}n\gamma(5\cos^2(i) - 1) \\ -\frac{3}{2}n\gamma(5\cos^2(i) - 1)e_y \\ \frac{3}{2}n\gamma(5\cos^2(i) - 1)e_x \\ 0 \\ -3n\gamma\cos(i) \end{bmatrix}. \quad (2.21)$$

## 2.3 Relative orbital dynamics

In this section, the relative motion of one artificial object with respect to a reference point orbiting the Earth is presented. The reference orbit can indeed be a real central satellite, or a virtual point of certain significance (e.g., the formation center). The relative dynamics can be either formulated using the Cartesian elements (in the [ECI](#), the [ECEF](#), or the [RTN](#) frame) or using the quasi-non-singular [Relative Orbital Elements \(ROE\)](#) vector, which is a nonlinear mapping of the orbital elements vector. Both formulations are presented in this section. In the rest of the thesis, and since the context is of multi-satellites, any quantity with the subscript  $(\cdot)_c$  signifies a quantity that relates to the chief satellite/virtual satellite, while the subscript  $(\cdot)_i$  relates to the  $i^{\text{th}}$  deputy quantities. The subscript becomes simply  $(\cdot)_d$  if only two satellites, chief and deputy, are considered. To this end, the  $\Delta$  operator is introduced to indicate the arithmetic difference between quantities that relate to a deputy and the chief. Concretely,

$$\Delta(\cdot)_i = (\cdot)_i - (\cdot)_c, \quad (2.22)$$

### 2.3.1 Relative Cartesian elements formulation

The relative Cartesian elements are defined to simply be the arithmetic difference between the Cartesian elements of the deputy spacecraft and that of the chief satellite, regardless of

the reference frame, such that,

$$\Delta \mathbf{x}^{(\cdot)} = \mathbf{x}_d^{(\cdot)} - \mathbf{x}_c^{(\cdot)}, \quad (2.23)$$

where  $\Delta \mathbf{x}^{(\cdot)}$  is the relative Cartesian elements in the proper frame  $\mathbb{F}^{(\cdot)}$ .

### ECI frame

Using Eq. (2.4) in conjunction with Eq. (2.23), the relative dynamics as observed from the ECI frame can be written as,

$$\Delta \dot{\mathbf{x}}^i = \left[ -\frac{\mu_{\oplus}}{r_c^3} \left( \left( \frac{r_c}{\sqrt{r_c^2 + 2\mathbf{r}_c^i \cdot \Delta \mathbf{r}^i + \Delta r^2}} \right)^3 (\mathbf{r}_c^i + \Delta \mathbf{r}^i) - \mathbf{r}_c^i \right) + \Delta \mathbf{u}^i + \Delta \mathbf{u}_{J_2}^i + \mathbf{w}_{\Delta \mathbf{v}}^i \right], \quad (2.24)$$

### ECEF frame

The relative motion in terms of the Cartesian elements in the ECEF frame can be derived in a similar manner to that of the ECI frame. Combining equations (2.7) and (2.23), one can write,

$$\Delta \dot{\mathbf{x}}^f = \left[ -\frac{\mu_{\oplus}}{r_c^3} \left( \left( \frac{r_c}{\sqrt{r_c^2 + 2\mathbf{r}_c^f \cdot \Delta \mathbf{r}^f + \Delta r^2}} \right)^3 (\mathbf{r}_c^f + \Delta \mathbf{r}^f) - \mathbf{r}_c^f \right) - 2\boldsymbol{\omega}_{\oplus}^f \times \Delta \mathbf{v}^f \right. \\ \left. - \boldsymbol{\omega}_{\oplus}^f \times \boldsymbol{\omega}_{\oplus}^f \times \Delta \mathbf{r}^f + \Delta \mathbf{u}^f + \Delta \mathbf{u}_{J_2}^f + \mathbf{w}_{\Delta \mathbf{v}}^f \right], \quad (2.25)$$

where  $\Delta \mathbf{r}^f$  and  $\Delta \mathbf{v}^f$  are the relative position and velocity vectors of the deputy spacecraft with respect to the chief satellite expressed in the ECEF frame, with magnitudes of  $\Delta r$  and  $\Delta v^f$ .

### RTN frame

Similar to the dynamics of the relative Cartesian state vector in the [ECI](#) and in the [ECEF](#) frames, the nonlinear dynamics of the relative Cartesian elements in the [RTN](#) frame can be derived by using Eq. (2.13) in conjunction with Eq. (2.23), and the resulting dynamical equation can be written as,

$$\Delta \dot{\mathbf{x}}^r = \begin{bmatrix} \Delta \mathbf{v}^r \\ -\frac{\mu_{\oplus}}{r_c^3} \left( \left( \frac{r_c}{\sqrt{r_c^2 + 2\mathbf{r}_c^r \cdot \Delta \mathbf{r}^r + \Delta r^2}} \right)^3 (\mathbf{r}_c^r + \Delta \mathbf{r}^r) - \mathbf{r}_c^r \right) - 2\boldsymbol{\omega}_r^r \times \Delta \mathbf{v}^r \\ -\dot{\boldsymbol{\omega}}_r^r \times \Delta \mathbf{r}^r + \boldsymbol{\omega}_r^r \times \boldsymbol{\omega}_r^r \times \Delta \mathbf{r}^r + \Delta \mathbf{u}^r + \Delta \mathbf{u}_{J_2}^r + \mathbf{w}_{\Delta \mathbf{v}}^r \end{bmatrix}. \quad (2.26)$$

Taking into account that  $\Delta \mathbf{v}^r = [\Delta v_x^r \ \Delta v_y^r \ \Delta v_z^r]^\top$ ,  $\Delta \mathbf{r}^r = [\Delta x^r \ \Delta y^r \ \Delta z^r]^\top$ ,  $\mathbf{r}_c^r = [r_c \ 0 \ 0]^\top$ ,  $\boldsymbol{\omega}_r^r = [0 \ 0 \ \dot{\nu}_c]^\top$ , and  $\dot{\boldsymbol{\omega}}_r^r = [0 \ 0 \ \ddot{\nu}_c]^\top$ , and ignoring the unmodeled disturbances, equation (2.26) can be expanded and rewritten as,

$$\Delta \dot{\mathbf{x}}^r = \begin{bmatrix} \Delta v_x^r \\ \Delta v_y^r \\ \Delta v_z^r \\ \left( \dot{\nu}_c^2 - \frac{\mu_{\oplus}}{\beta} \right) \Delta x^r + \ddot{\nu}_c \Delta y^r + 2\dot{\nu}_c \Delta v_y^r + \frac{\mu_{\oplus}}{r_c^2} \left( 1 - \frac{r_c^3}{\beta} \right) \\ -\ddot{\nu}_c \Delta x^r + \left( \dot{\nu}_c^2 - \frac{\mu_{\oplus}}{\beta} \right) \Delta y^r - 2\dot{\nu}_c \Delta v_x^r \\ -\frac{\mu_{\oplus}}{\beta} \Delta z^r \end{bmatrix} + \mathbf{B}_{\Delta \mathbf{x}} (\Delta \mathbf{u}^r + \Delta \mathbf{u}_{J_2}^r), \quad (2.27)$$

where,

$$\beta = \sqrt{\left( (r_c + \Delta x^r)^2 + \Delta y^{r^2} + \Delta z^{r^2} \right)^3}, \quad (2.28)$$

and,

$$\mathbf{B}_{\Delta\mathbf{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^\top. \quad (2.29)$$

Equation (2.27) can be further simplified to be a [Linear Time-Invariant \(LTI\)](#) system, to what is widely known as the [Clohessy–Wiltshire \(CW\)](#) form [23], under the following assumptions:

1. The orbit of the chief is circular, which renders  $\dot{\nu}_c = n_c$ ,  $\ddot{\nu}_c = 0$ .
2. The separation between the two spacecraft is very small in comparison to the radius of the chief, i.e.,  $\Delta r \ll r_c$ .
3. The motion of the chief as well as the deputy is unperturbed, i.e., in equation (2.27)  $\Delta \mathbf{u}_{J_2}^r = \mathbf{0}$ .
4. Only the deputy satellite is controllable, i.e.,  $\Delta \mathbf{u}^r = \mathbf{u}_d^r = \begin{bmatrix} u_{d,x}^r & u_{d,y}^r & u_{d,z}^r \end{bmatrix}^\top$  (refer to (2.27)), with  $u_{d,x}^r, u_{d,y}^r, u_{d,z}^r$  being the control acceleration components exerted by the deputy in the  $x$ ,  $y$ , and  $z$  directions of the [RTN](#) frame.

Under the aforementioned assumptions, equation (2.27) is transformed to the [CW](#) equations,

$$\begin{bmatrix} \Delta \dot{x}^r \\ \Delta \dot{y}^r \\ \Delta \dot{z}^r \\ \Delta \dot{v}_x^r \\ \Delta \dot{v}_y^r \\ \Delta \dot{v}_z^r \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n_c^2 & 0 & 0 & 0 & 2n_c & 0 \\ 0 & 0 & 0 & -2n_c & 0 & 0 \\ 0 & 0 & -n_c^2 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_{CW}} \begin{bmatrix} \Delta x^r \\ \Delta y^r \\ \Delta z^r \\ \Delta v_x^r \\ \Delta v_y^r \\ \Delta v_z^r \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{B}_{\Delta\mathbf{x}}} \begin{bmatrix} u_{d,x}^r \\ u_{d,y}^r \\ u_{d,z}^r \end{bmatrix}, \quad (2.30)$$

where  $n_c$  is the mean motion of the chief satellite.

The LTI system (2.30) is integrated, and its solution is obtained in the form,

$$\Delta \mathbf{x}^r(t) = \Phi_{CW}(t_0, t) \Delta \mathbf{x}^r(t_0) + \Psi_{CW}(t_0, t) \mathbf{u}_d^r(t), \quad (2.31)$$

where  $\Phi_{CW}(t_0, t)$  is the State Transition Matrix (STM) of the CW system between the two time instants  $t_0$  and  $t$ , and  $\Psi_{CW}(t_0, t)$  is the convolution matrix of the CW system between the two time instants  $t_0$  and  $t$ .

The STM of the CW equations can be easily obtained by solving the homogeneous part of Eq. (2.30), which yields,

$$\begin{aligned} \Phi_{CW}(t_0, t) &= \exp(\mathbf{A}_{CW} \Delta t) \\ &= \frac{1}{n_c} \begin{bmatrix} n_c(4 - 3c_{n_c \Delta t}) & 0 & 0 & s_{n_c \Delta t} & 2(1 - c_{n_c \Delta t}) & 0 \\ 6n_c(s_{n_c \Delta t} - n_c t) & n_c & 0 & 2(c_{n_c \Delta t} - 1) & 4s_{n_c \Delta t} - 3n_c \Delta t & 0 \\ 0 & 0 & n_c c_{n_c \Delta t} & 0 & 0 & s_{n_c \Delta t} \\ 3n_c^2 s_{n_c \Delta t} & 0 & 0 & n_c c_{n_c \Delta t} & 2n_c s_{n_c \Delta t} & 0 \\ -6n_c^2(1 - c_{n_c \Delta t}) & 0 & 0 & -2n_c s_{n_c \Delta t} & n_c(4c_{n_c \Delta t} - 3) & 0 \\ 0 & 0 & -n_c^2 s_{n_c \Delta t} & 0 & 0 & n_c c_{n_c \Delta t} \end{bmatrix}, \end{aligned} \quad (2.32)$$

where  $\Delta t = t - t_0$ ,  $s_{n_c t} = \sin(n_c t)$  and  $c_{n_c t} = \cos(n_c t)$ .

The convolution matrix,  $\Psi_{CW}(t_0, t)$ , can be obtained for  $\mathbf{u}_d^r(\tau) = \overline{\mathbf{u}}_d^r \forall \tau \in [t_0, t]$ , i.e., for

a constant acceleration over the period  $[t_0, t[$ , by solving the following convolution integral,

$$\begin{aligned}\Psi_{CW}(t_0, t) &= \int_{t_0}^t \Phi_{CW}(\tau, t) \mathbf{B}_{\Delta \mathbf{x}} d\tau \\ &= \frac{1}{n_c^2} \begin{bmatrix} 1 - c_{n_c \Delta t} & 2(n_c \Delta t - s_{n_c \Delta t}) & 0 \\ 2(s_{n_c \Delta t} - n_c \Delta t) & 4(1 - c_{n_c \Delta t}) - \frac{3}{2}(n_c \Delta t)^2 & 0 \\ 0 & 0 & 1 - c_{n_c \Delta t} \\ n_c s_{n_c \Delta t} & 2n_c(1 - c_{n_c \Delta t}) & 0 \\ -2n_c(1 - c_{n_c \Delta t}) & -n_c(3n_c \Delta t - 4s_{n_c \Delta t}) & 0 \\ 0 & 0 & n_c s_{n_c \Delta t} \end{bmatrix}. \end{aligned} \quad (2.33)$$

### 2.3.2 ROE formulation

To describe the relative motion between two spacecraft, a deputy and a chief, the quasi-non-singular **ROE**, also referred to in the rest of the thesis simply as **ROE**, can be utilized. The **ROE** vector is formally defined as,

$$\delta \boldsymbol{\alpha} = \begin{bmatrix} \delta a \\ \delta \lambda \\ \delta e_x \\ \delta e_y \\ \delta i_x \\ \delta i_y \end{bmatrix} = \begin{bmatrix} \frac{\Delta a}{a_c} \\ \Delta \theta + \Delta \Omega \cos i_c \\ \Delta e_x \\ \Delta e_y \\ \Delta i \\ \Delta \Omega \sin i_c \end{bmatrix}, \quad (2.34)$$

where  $\delta \boldsymbol{\alpha}$  is the dimensionless **ROE** vector,  $\delta a$  is the relative semi-major axis,  $\delta \lambda$  is the relative mean longitude,  $\delta \mathbf{e} = [\delta e_x \ \delta e_y]^\top$  is the relative eccentricity vector, and  $\delta \mathbf{i} = [\delta i_x \ \delta i_y]^\top$  is the relative inclination vector. It is to be noted that  $\delta(\cdot)$  signifies a relative quantity between the deputy and the chief which is not necessarily the arithmetic difference between that of the deputy and that of the chief, while  $\Delta(\cdot)$  signifies the arithmetic difference as stipulated in Eq. (2.22). As is the case for absolute orbital elements, the osculating **ROE** vector is distinguished from the mean one by the over-tilde, i.e., the osculating **ROE** vector is denoted

by  $\delta\tilde{\alpha}$ , whereas the mean ROE vector is referred to as  $\delta\alpha$ . A dimensional ROE vector is obtained by multiplying the dimensionless ROE vector by the semi-major axis of the chief,

$$\mathbf{y} = a_c \delta\alpha, \quad (2.35)$$

where  $\mathbf{y}$  is the dimensional mean ROE vector with units of length.

Different ROE sets have been in use in the literature to parameterize the relative motion between two satellites [24]. A list of such formulations include singular, quasi-non-singular and non-singular ROE vectors. The adopted ROE formulation, i.e., the quasi-non-singular one, offer various advantages, among which is the facilitation of designing passively safe relative orbits through the concept of eccentricity/inclination (e/i) vector separation [25]. Although the development of the e/i vector separation concept was originally meant for the safe collocation of Geostationary Orbit (GEO) satellites [25], it was extended to passively avoid collisions in Low Earth Orbit (LEO) satellite formations [26]. Moreover, the quasi-non-singular ROE do match the integration constants of the CW equations [27], which allows for a direct transformation from the ROE space into the Cartesian state vector and vice versa. One other advantage of the quasi-non-singular ROE formulation, and of all other ROE formulations for that matter, is their ability to provide a direct insight into the shape of the relative orbit, unlike the relative position and velocity vectors which need to be integrated over time in order to visualize the shape of the relative orbit. Given the set of ROE in Eq. (2.34), the shape of the relative orbit can be constructed as an ellipse, in both, the Transversal-Radial and the Normal-Radial planes as depicted in Fig. 2.5.

The dynamics of the dimensionless mean ROE can be obtained by differentiating both sides of Eq. (2.34), which can be easily expanded with the help of Eq. (2.19). Differentiating



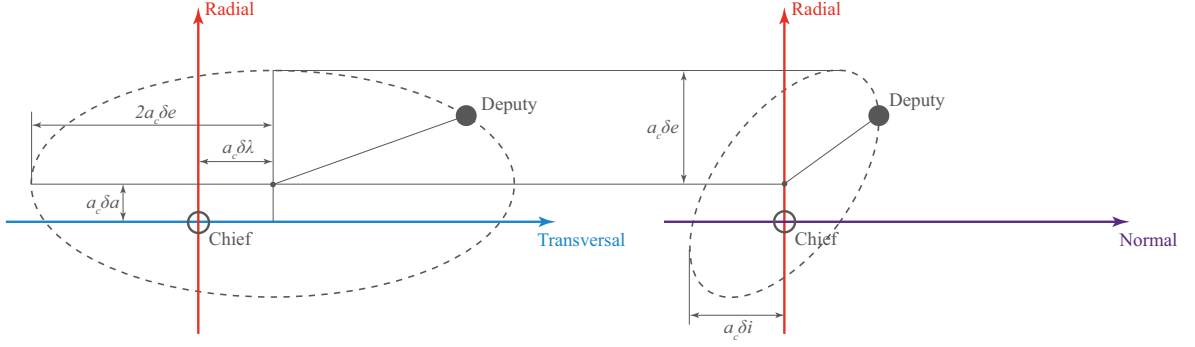


Figure 2.5: Snapshot of the shape of the relative orbit neglecting the orbit drift

both sides of Eq. (2.34) yields,

$$\begin{aligned} \delta \dot{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_c, \boldsymbol{\alpha}_d) &= \begin{bmatrix} 0 \\ \Delta \dot{\theta} + \Delta \dot{\Omega} \cos i_c \\ \Delta(-e\dot{\omega} \sin(\omega)) \\ \Delta(e\dot{\omega} \cos(\omega)) \\ 0 \\ \Delta \dot{\Omega} \sin i_c \end{bmatrix}, \\ &= \begin{bmatrix} 0 \\ \Delta n + \frac{3}{2} \Delta [n\gamma(\eta + 1)(3 \cos^2(i) - 1)] + 3n_d \gamma_d \cos(i_d) \Delta \cos(i) \\ \frac{3}{2} \Delta [n\gamma(1 - 5 \cos^2(i)) e_y] \\ \frac{3}{2} \Delta [n\gamma(1 - 5 \cos^2(i)) e_x] \\ 0 \\ -3 \sin(i_c) \Delta(n\gamma \cos(i)) \end{bmatrix} \end{aligned} \quad (2.36)$$

Using a first-order Taylor expansion of  $\delta \dot{\boldsymbol{\alpha}}$  around the linearization point  $\boldsymbol{\alpha}_c^*$  and  $\boldsymbol{\alpha}_d^*$ , Eq. (2.36) can be approximated by,

$$\delta \dot{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_c, \boldsymbol{\alpha}_d) \approx \delta \dot{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_c^*, \boldsymbol{\alpha}_d^*) + \left. \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}_c} \right|_{\substack{\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_c^* \\ \boldsymbol{\alpha}_d = \boldsymbol{\alpha}_d^*}} (\boldsymbol{\alpha}_c - \boldsymbol{\alpha}_c^*) + \left. \frac{\partial \delta \dot{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}_d} \right|_{\substack{\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_c^* \\ \boldsymbol{\alpha}_d = \boldsymbol{\alpha}_d^*}} (\boldsymbol{\alpha}_d - \boldsymbol{\alpha}_d^*). \quad (2.37)$$

Taking the linearization point to be  $\boldsymbol{\alpha}_c^* = \boldsymbol{\alpha}_c$  and  $\boldsymbol{\alpha}_d^* = \boldsymbol{\alpha}_c$ , the approximated dynamics of

the mean ROE can be rewritten as,

$$\begin{aligned} \delta\dot{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_c, \boldsymbol{\alpha}_d) &\approx \cancel{\delta\dot{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_c, \boldsymbol{\alpha}_c)} + \overset{0}{\frac{\partial \delta\dot{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}_e} \bigg|_{\substack{\boldsymbol{\alpha}_c=\boldsymbol{\alpha}_c \\ \boldsymbol{\alpha}_d=\boldsymbol{\alpha}_c}}} (\boldsymbol{\alpha}_c - \boldsymbol{\alpha}_c) + \overset{0}{\frac{\partial \delta\dot{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}_d} \bigg|_{\substack{\boldsymbol{\alpha}_c=\boldsymbol{\alpha}_c \\ \boldsymbol{\alpha}_d=\boldsymbol{\alpha}_c}}} (\boldsymbol{\alpha}_d - \boldsymbol{\alpha}_c), \\ &\approx - \frac{\partial \delta\dot{\boldsymbol{\alpha}}}{\partial \boldsymbol{\alpha}_d} \bigg|_{\substack{\boldsymbol{\alpha}_c=\boldsymbol{\alpha}_c \\ \boldsymbol{\alpha}_d=\boldsymbol{\alpha}_c}} \Delta\boldsymbol{\alpha}, \end{aligned} \quad (2.38)$$

which, after calculating the Jacobian matrix, and assuming a near-circular orbit of the chief, i.e.,  $e_{x,c} \approx 0$  and  $e_{y,c} \approx 0$ , can be expressed [28] as,

$$\delta\dot{\boldsymbol{\alpha}} \approx \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -L_c & 0 & 0 & 0 & -K_c(4 + 3\eta_c)\sin(2i_c) & 0 \\ 0 & 0 & 0 & -\dot{\omega}_c & 0 & 0 \\ 0 & 0 & \dot{\omega}_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{7}{2}K_c\sin(2i_c) & 0 & 0 & 0 & 2K_c\sin^2(i_c) & 0 \end{bmatrix}}_{\mathbf{A}_{\delta\boldsymbol{\alpha}}} \delta\boldsymbol{\alpha}, \quad (2.39)$$

where  $\dot{\omega}_c$  is the apsidal precession rate of the chief's orbit, induced by the  $J_2$  zonal harmonic and defined in Eq. (2.19), and,

$$\begin{aligned} L_c &= \frac{3}{2}n_c + \frac{21}{3}n_c\gamma_c(1 + \eta_c)(3\cos^2(i_c) - 1), \\ K_c &= \frac{3}{2}n_c\gamma_c. \end{aligned} \quad (2.40)$$

It is important to note that Eq. (2.39) describes the natural dynamics of the ROE vector without including the effect of providing a control acceleration to the deputy spacecraft. Leveraging the Gauss Variational Equations (GVE) [29], and assuming  $\tilde{\boldsymbol{\alpha}}_d \approx \boldsymbol{\alpha}_d$ , which is a valid assumption especially for low-thrust applications [29], the evolution of the quasi-non-singular ROE can be described by,

$$\delta\dot{\boldsymbol{\alpha}} \approx \mathbf{A}_{\delta\boldsymbol{\alpha}}\delta\boldsymbol{\alpha} + \overline{\mathbf{B}}_{\delta\boldsymbol{\alpha}}\mathbf{u}_d^T, \quad (2.41)$$

where  $\mathbf{u}_d^r$  is the input acceleration vector imposed on the deputy spacecraft, expressed in the RTN frame, and where  $\bar{\mathbf{B}}_{\delta\alpha}$  is a nonlinear control matrix that can be expressed [28] as,

$$\bar{\mathbf{B}}_{\delta\alpha} = \begin{bmatrix} \frac{2e_d \sin(\nu_d)}{a_c n_d \eta_d} & \frac{2\kappa_d}{a_c n_d \eta_d} & 0 \\ -\frac{\eta_d e_d \cos(\nu_d)}{a_d n_d (1 + \eta_d)} - \frac{2\eta_d^2}{a_d n_d \kappa_d} & -\frac{\eta_d e_d (\kappa_d + 1) \sin(\nu_d)}{a_d n_d \kappa_d (1 + \eta_d)} & \frac{\eta_d \sin(\theta_d) \Delta \cos(i)}{a_d n_d \kappa_d \sin(i_d)} \\ \frac{\eta_d \sin(\theta_d)}{a_d n_d} & \frac{\eta_d (\kappa_d + 1) \cos(\theta_d) + \eta_d e_{x,d}}{a_d n_d \kappa_d} & \frac{\eta_d e_{y,d} \sin(\theta_d) \cot(i_d)}{a_d n_d \kappa_d} \\ -\frac{\eta_d \cos(\theta_d)}{a_d n_d} & \frac{\eta_d (\kappa_d + 1) \sin(\theta_d) + \eta_d e_{y,d}}{a_d n_d \kappa_d} & -\frac{\eta_d e_{x,d} \sin(\theta_d) \cot(i_d)}{a_d n_d \kappa_d} \\ 0 & 0 & \frac{\eta_d \sin(\theta_d)}{a_d n_d \kappa_d} \\ 0 & 0 & \frac{\eta_d \cos(\theta_d) \sin(i_c)}{a_d n_d \kappa_d \sin(i_d)} \end{bmatrix}, \quad (2.42)$$

where,

$$\kappa_d = 1 + e_d \cos(\nu_d). \quad (2.43)$$

The term  $\bar{\mathbf{B}}_{\delta\alpha} \mathbf{u}_d^r$ , or more explicitly  $\bar{\mathbf{B}}_{\delta\alpha}(\alpha_c, \alpha_d) \mathbf{u}_d^r$ , in Eq. (2.41) can be approximated using a first order Taylor expansion around the linearization point  $\alpha_c^*$ ,  $\alpha_d^*$ , and  $\mathbf{u}_d^{r*}$ , such that,

$$\begin{aligned} \bar{\mathbf{B}}_{\delta\alpha}(\alpha_c, \alpha_d) \mathbf{u}_d^r &\approx \bar{\mathbf{B}}_{\delta\alpha}(\alpha_c^*, \alpha_d^*) \mathbf{u}_d^{r*} + \frac{\partial \bar{\mathbf{B}}_{\delta\alpha} \mathbf{u}_d^r}{\partial \alpha_c} \bigg|_{\substack{\alpha_c = \alpha_c^* \\ \alpha_d = \alpha_d^* \\ \mathbf{u}_d^r = \mathbf{u}_d^{r*}} (\alpha_c - \alpha_c^*) \\ &+ \frac{\partial \bar{\mathbf{B}}_{\delta\alpha} \mathbf{u}_d^r}{\partial \alpha_d} \bigg|_{\substack{\alpha_c = \alpha_c^* \\ \alpha_d = \alpha_d^* \\ \mathbf{u}_d^r = \mathbf{u}_d^{r*}} (\alpha_d - \alpha_d^*) + \frac{\partial \bar{\mathbf{B}}_{\delta\alpha} \mathbf{u}_d^r}{\partial \mathbf{u}_d^r} \bigg|_{\substack{\alpha_c = \alpha_c^* \\ \alpha_d = \alpha_d^* \\ \mathbf{u}_d^r = \mathbf{u}_d^{r*}} (\mathbf{u}_d^r - \mathbf{u}_d^{r*}), \end{aligned} \quad (2.44)$$

which, taking  $\alpha_c^* = \alpha_c$ ,  $\alpha_d^* = \alpha_c$ , and  $\mathbf{u}_d^{r*} = \mathbf{0}$ , can be rewritten as,

$$\begin{aligned} \bar{\mathbf{B}}_{\delta\alpha}(\alpha_c, \alpha_d) \mathbf{u}_d^r &\approx \bar{\mathbf{B}}_{\delta\alpha}(\alpha_c, \alpha_c) \cdot \mathbf{0} + \frac{\partial \bar{\mathbf{B}}_{\delta\alpha} \mathbf{u}_d^r}{\partial \alpha_c} \bigg|_{\substack{\alpha_c = \alpha_c \\ \alpha_d = \alpha_c \\ \mathbf{u}_d^r = \mathbf{0}}} (\alpha_c - \alpha_c) \\ &+ \frac{\partial \bar{\mathbf{B}}_{\delta\alpha} \mathbf{u}_d^r}{\partial \alpha_d} \bigg|_{\substack{\alpha_c = \alpha_c \\ \alpha_d = \alpha_c \\ \mathbf{u}_d^r = \mathbf{0}}} (\alpha_d - \alpha_c) + \frac{\partial \bar{\mathbf{B}}_{\delta\alpha} \mathbf{u}_d^r}{\partial \mathbf{u}_d^r} \bigg|_{\substack{\alpha_c = \alpha_c \\ \alpha_d = \alpha_c \\ \mathbf{u}_d^r = \mathbf{0}}} \mathbf{u}_d^r, \end{aligned} \quad (2.45)$$

which, in-turn, can be rewritten as,

$$\bar{\mathbf{B}}_{\delta\boldsymbol{\alpha}} \approx \mathbf{B}_{\delta\boldsymbol{\alpha}} = \left. \frac{\partial \bar{\mathbf{B}}_{\delta\boldsymbol{\alpha}} \mathbf{u}_d^r}{\partial \mathbf{u}_d^r} \right|_{\substack{\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_c \\ \boldsymbol{\alpha}_d = \boldsymbol{\alpha}_c \\ \mathbf{u}_d^r = \mathbf{0}}} = \bar{\mathbf{B}}_{\delta\boldsymbol{\alpha}} \Big|_{\substack{\boldsymbol{\alpha}_c = \boldsymbol{\alpha}_c \\ \boldsymbol{\alpha}_d = \boldsymbol{\alpha}_c \\ \mathbf{u}_d^r = \mathbf{0}}}. \quad (2.46)$$

Under the assumption of a near-circular orbit of the chief, i.e.,  $e_c \approx e_{x,c} \approx e_{y,c} \approx 0$  and  $\eta_c \approx \kappa_c \approx 1$ , as well as the assumption of closely inclined chief and deputy orbits, i.e.,  $i_c \approx i_d$ , the control matrix of the linearized system can be written as,

$$\mathbf{B}_{\delta\boldsymbol{\alpha}} \approx \frac{1}{a_c n_c} \begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ \sin(\theta_c) & 2 \cos(\theta_c) & 0 \\ -\cos(\theta_c) & 2 \sin(\theta_c) & 0 \\ 0 & 0 & \cos(\theta_c) \\ 0 & 0 & \sin(\theta_c) \end{bmatrix}. \quad (2.47)$$

Combining equations (2.47) with (2.39), Eq. (2.41) can be rewritten as,

$$\dot{\delta\boldsymbol{\alpha}} \approx \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -L_c & 0 & 0 & 0 & -K_c(4 + 3\eta_c) \sin(2i_c) & 0 \\ 0 & 0 & 0 & -\dot{\omega}_c & 0 & 0 \\ 0 & 0 & \dot{\omega}_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{7}{2} K_c \sin(2i_c) & 0 & 0 & 0 & 2K_c \sin^2(i_c) & 0 \end{bmatrix}}_{\mathbf{A}_{\delta\boldsymbol{\alpha}}} \delta\boldsymbol{\alpha}$$

$$+ \frac{1}{a_c n_c} \underbrace{\begin{bmatrix} 0 & 2 & 0 \\ -2 & 0 & 0 \\ \sin(\theta_c) & 2 \cos(\theta_c) & 0 \\ -\cos(\theta_c) & 2 \sin(\theta_c) & 0 \\ 0 & 0 & \cos(\theta_c) \\ 0 & 0 & \sin(\theta_c) \end{bmatrix}}_{\mathbf{B}_{\delta\alpha}} \mathbf{u}_d^r, \quad (2.48)$$

where, again,

$$\begin{aligned} n_c &= \sqrt{\frac{\mu_\oplus}{a_c^3}} \\ \eta_c &= \sqrt{1 - e_c^2} \\ p_c &= a_c \eta_c^2 \\ \gamma_c &= \frac{1}{2} J_2 \left( \frac{R_\oplus}{p_c} \right)^2, \\ \dot{\omega}_c &= \frac{3}{2} n_c \gamma_c (5 \cos^2(i_c) - 1), \\ L_c &= \frac{3}{2} n_c + \frac{21}{3} n_c \gamma_c (1 + \eta_c) (3 \cos^2(i_c) - 1), \\ K_c &= \frac{3}{2} n_c \gamma_c. \end{aligned} \quad (2.49)$$

The solution to the [Linear Time-Varying \(LTV\)](#) system described by Eq. (2.48) can be expressed as,

$$\delta\alpha(t) = \Phi_{\delta\alpha}(t_0, t) \delta\alpha(t_0) + \Psi_{\delta\alpha}(t_0, t) \mathbf{u}_d^r(t), \quad (2.50)$$

The calculation of the [STM](#) of an [LTV](#) system is generally an elaborate process, however, due to the fact that the Jacobian matrix,  $\mathbf{A}_{\delta\alpha}$ , is constant over time, as it only depends on the time-invariant Keplerian elements of the chief, i.e.,  $a_c$ ,  $e_c$ , and  $i_c$ , the [STM](#),  $\Phi_{\delta\alpha}(t_0, t)$  can

be calculated by solving the homogeneous part of Eq. (2.48) simply as,

$$\begin{aligned} \Phi_{\delta\alpha}(t_0, t) &= \exp(\mathbf{A}_{\delta\alpha}\Delta t) \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -L_c\Delta t & 1 & 0 & 0 & -K_c(4+3\eta_c)\sin(2i_c)\Delta t & 0 \\ 0 & 0 & \cos(\dot{\omega}_c\Delta t) & -\sin(\dot{\omega}_c\Delta t) & 0 & 0 \\ 0 & 0 & \sin(\dot{\omega}_c\Delta t) & \cos(\dot{\omega}_c\Delta t) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{7}{2}K_c\sin(2i_c)\Delta t & 0 & 0 & 0 & 2K_c\sin^2(i_c)\Delta t & 1 \end{bmatrix}, \end{aligned} \quad (2.51)$$

where  $\Delta t = t - t_0$ . The convolution matrix, however, can be obtained in a similar fashion to Eq. (2.33) by assuming a constant acceleration,  $\mathbf{u}_d^r$ , over the period  $[t_0, t]$ , as,

$$\begin{aligned} \Psi_{\delta\alpha}(t_0, t) &= \int_{t_0}^t \Phi_{\delta\alpha}(\tau, t) \mathbf{B}_{\delta\alpha} d\tau \\ &= \frac{1}{a_c n_c \dot{\theta}_c} \begin{bmatrix} 0 & 2D_c & 0 \\ -2D_c & -\frac{L_c D_c^2}{\dot{\theta}_c} & \psi_{23} \\ \psi_{31} & -2\psi_{41} & 0 \\ \psi_{41} & 2\psi_{31} & 0 \\ 0 & 0 & \sin(\theta_c) - \sin(\theta_{c,0}) \\ 0 & \frac{7K_c \sin(2i_c) D_c^2}{2\dot{\theta}_c} & \psi_{63} \end{bmatrix}, \end{aligned} \quad (2.52)$$

where,

$$\begin{aligned}
D_c &= \theta_c - \theta_{c,0}, \\
\dot{\theta}_c &= \dot{M}_c + \dot{\omega}_c = n_c + \frac{3}{2}n_c\gamma_c\eta_c(3\cos^2(i_c) - 1) + \frac{3}{2}n_c\gamma_c(5\cos^2(i_c) - 1), \\
C_c &= \frac{\dot{\omega}_c}{\dot{\theta}_c} = \frac{\gamma_c(5\cos^2(i_c) - 1)}{\frac{2}{3} + \gamma_c\eta_c(3\cos^2(i_c) - 1) + \gamma_c(5\cos^2(i_c) - 1)}, \\
\psi_{23} &= \frac{K_c(4 + 3\eta_c)\sin(2i_c)[\cos(\theta_c) - \cos(\theta_{c,0}) + \sin(\theta_{c,0})D_c]}{\dot{\theta}_c}, \\
\psi_{31} &= -\frac{\cos(\theta_c) - \cos(\theta_{c,0} + C_c D_c)}{1 - C_c}, \\
\psi_{41} &= -\frac{\sin(\theta_c) - \sin(\theta_{c,0} + C_c D_c)}{1 - C_c}, \\
\psi_{63} &= -\frac{\left[\dot{\theta}_c + 2K_c\sin^2(i_c)\right][\cos(\theta_c) - \cos(\theta_{c,0})] + 2K_c\sin^2(i_c)\sin(\theta_{c,0})D_c}{\dot{\theta}_c},
\end{aligned} \tag{2.53}$$

with  $\theta_c \equiv \theta_c(t)$  being the mean argument of latitude of the chief at the current epoch,  $t$ , and  $\theta_{c,0} \equiv \theta_c(t_0)$  being the mean argument of latitude of the chief at the initial time,  $t_0$ . The two are related through the following formula,

$$\theta_c = \theta_{c,0} + \dot{\theta}_c \Delta t. \tag{2.54}$$

Note that, in Eq. (2.52), only the mean argument of latitude is specified whether it has been evaluated at  $t$  or at  $t_0$ , since it is the only variable that changes with time. All the other variables, e.g.,  $n_c$ ,  $\gamma_c$ ,  $\eta_c$ ,  $i_c$ , are constant over time (refer to Eq. (2.19)).

### 2.3.3 Transformation between the ROE and the RTN Cartesian state vector

The long discussion in [27] concluded that the ROE match the integration constants of the CW equations, which, under the same assumption of the CW equations, allows for the deduction of a direct transformation between the ROE and the relative Cartesian state vector in the RTN frame [30], such that,

$$\Delta \mathbf{x}^r = a_c \bar{\mathbf{T}}(\tau) \delta \boldsymbol{\alpha} = \bar{\mathbf{T}}(\tau) \mathbf{y} \tag{2.55}$$

where  $\tau$  is the time elapsed time since a predefined reference time (the time at which the initial state is given), and,

$$\overline{\mathbf{T}}(\tau) = \begin{bmatrix} 1 & 0 & -\cos(n_c\tau) & -\sin(n_c\tau) & 0 & 0 \\ 0 & 1 & 2\sin(n_c\tau) & -2\cos(n_c\tau) & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin(n_c\tau) & -\cos(n_c\tau) \\ 0 & 0 & n_c\sin(n_c\tau) & -n_c\cos(n_c\tau) & 0 & 0 \\ -\frac{3}{2}n_c & 0 & 2n_c\cos(n_c\tau) & 2n_c\sin(n_c\tau) & 0 & 0 \\ 0 & 0 & 0 & 0 & n_c\cos(n_c\tau) & n_c\sin(n_c\tau) \end{bmatrix}. \quad (2.56)$$

The **ROE** vector can be transformed into its corresponding **RTN** relative position vector through,

$$\begin{aligned} \Delta \mathbf{r}^r &= a_c \mathbf{T}(\tau) \delta \boldsymbol{\alpha}, \\ &= a_c \begin{bmatrix} 1 & 0 & -\cos(n_c\tau) & -\sin(n_c\tau) & 0 & 0 \\ 0 & 1 & 2\sin(n_c\tau) & -2\cos(n_c\tau) & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin(n_c\tau) & -\cos(n_c\tau) \end{bmatrix} \delta \boldsymbol{\alpha}. \end{aligned} \quad (2.57)$$

The transformation matrix  $\overline{\mathbf{T}}$  is non-singular for all the values of  $\tau$ , and hence can be inverted. The inverse transformation matrix,  $\overline{\mathbf{T}}^{-1}$ , is able to transform the relative Cartesian state vector into the **ROE** vector, such that,

$$a_c \delta \boldsymbol{\alpha} = \overline{\mathbf{T}}^{-1}(\tau) \Delta \mathbf{x}^r \quad (2.58)$$



and,

$$\overline{\mathbf{T}}^{-1}(\tau) = \frac{1}{n_c} \begin{bmatrix} 4n_c & 0 & 0 & 0 & 2 & 0 \\ 0 & n_c & 0 & -2 & 0 & 0 \\ 3n_c \cos(n_c\tau) & 0 & 0 & \sin(n_c\tau) & 2\cos(n_c\tau) & 0 \\ 3n_c \sin(n_c\tau) & 0 & 0 & -\cos(n_c\tau) & 2\sin(n_c\tau) & 0 \\ 0 & 0 & n_c \sin(n_c\tau) & 0 & 0 & \cos(n_c\tau) \\ 0 & 0 & -n_c \cos(n_c\tau) & 0 & 0 & \sin(n_c\tau) \end{bmatrix}. \quad (2.59)$$

It is important to emphasize that the transformations presented in this section are valid only for close-range near-circular formation flying. For situations which violate these assumptions, there exist several other non-linear mappings that relate the ROE to the relative Cartesian state vector [31, 32].

# Part II

## Contributions

# GNSS-based relative navigation

## 3.1 Introduction

A reliable state estimation subsystem is essential to close the control loop for any control system. Satellites in [Low Earth Orbit \(LEO\)](#) have long relied on the signals of the [Global Navigation Satellite System \(GNSS\)](#) to estimate their position and velocity vectors in real-time which enabled precise orbit maneuvers. Two distinct [GNSS](#)-based positioning schemes are extricated, absolute positioning and relative positioning. Absolute positioning aims at estimating the position vector of the receiver with respect to the center of the Earth, either by solely relying on the measurements from that receiver (standalone positioning) or by combining the measurements from the main receiver and another nearby stationary base receiver with a precisely known position. The latter leverages [Differential Global Navigation Satellite System \(DGNSS\)](#) techniques and is not suitable for space applications since the existence of a base in space with a precisely known position and with a sustained communication link to the main receiver is an elaborate task. Relative positioning on the other hand is after estimating the baseline vector of one receiver with respect to another, conceivably using the [GNSS](#) signals collected by these receivers and also sometimes, leveraging [DGNSS](#) techniques. The accuracy of the absolute positioning schemes can vary from few centimeters to tens of meters [33, 34] depending on many factors such as whether [DGNSS](#) is incorporated, atmo-

spheric conditions, receiver quality and design features, and signal blockage. While absolute positioning is a natural choice for one-satellite missions, relative positioning can be essential for the multi-satellite ones.

### 3.1.1 Problem statement

In formation flying missions, estimating the relative position and velocity of a spacecraft with respect to another is of huge interest. In this chapter, the relative state estimation is considered, for a formation consisting of two spacecraft, a chief and a deputy. The setup of the problem as well as the system requirements are summarized as follows:

- The baseline vector from the chief to the deputy is estimated onboard the deputy;
- The two spacecraft are separated by a large distance (10 – 500 km). In this setting, ionospheric delay is, in most cases, the largest bias that needs to be filtered out;
- The two spacecraft are each equipped with single-frequency GNSS receivers, each of them can provide an estimate of the absolute position and velocity of its antenna phase center, with a noise level of 10 m for the position and of 25 cm/s for the velocity ( $1\sigma$ ). The exact make of the receivers is hidden for commercial reasons. The consideration of a single-frequency receiver is especially interesting for Triton-X as it uses a 12-channel L1 GPS receiver. Although Triton-X onboard GNSS receiver leverages only Global Positioning System (GPS) signals, the proposed navigation scheme can be used with other GNSS receivers;
- An inter-satellite link from the chief to the deputy is assumed to be constructed, which allows the transfer of the necessary data to estimate the position and velocity of the deputy with respect to the chief. The treatment of latency and synchronization is out of the scope of this thesis;
- The complexity of the relative navigation schemes has to be bearable by the Triton-X On-Board Computer (OBC). This constrains the tweaks that could be done to the already existing algorithms in the literature. For instance, double-difference integer

ambiguity resolution routines are excluded since it would be a computational burden on the OBC, in spite of the fact that these techniques are expected to improve the estimation accuracy while compromising the convergence time;

- As Triton-X is a multi-mission platform, the mission to which the relative navigation algorithm is to be applied is tentative and the relative navigation requirements are not yet concrete. However, the estimation scheme is developed for missions which do not require stringent onboard estimation accuracy such as inspection and gravimetry missions [35]. Concretely, it is assumed that an estimation error of less than 0.5 m (3-Dimensional (3D) Root Mean Square (RMS)) is required, which is similar to the real-time orbit determination requirements of the PRISMA technology demonstration mission [36].

### 3.1.2 State of the art

Sensor sets other than GNSS receivers could be used to carry out relative navigation tasks for different mission scenarios. Schemes for relative navigation between two closely flying objects, either cooperative or non-cooperative, have been developed based on line of sight measurements [37, 38], LiDAR [39], GNSS receivers [40, 41, 42], and both, cameras as well as GNSS receivers [43]. However, GNSS sensors stand as the perfect choice for cooperative spacecraft, especially when the baseline between the flying satellites is large, where vision-based sensors could no longer capture the features of the target spacecraft.

GNSS-based Extended Kalman Filter (EKF) have shown to achieve superb estimates with millimeter-level accuracy of the baseline vector for formations with small inter-satellite distances (1 – 10 km) in Low Earth Orbits [4, 7, 8]. Baseline estimation has also been tackled for longer inter-satellite distances [5, 44] and remarkable accuracy could be obtained using dual-frequency receivers. Dual-frequency receivers in the case of long baselines (10 – 500 km) are very important not only to mitigate the huge difference in ionospheric delay between the two receivers, but also to help fixing the double difference integer ambiguities. Nevertheless, the construction of precise relative orbit determination algorithms using single-frequency receivers

shall be an enabling technology for low-cost satellites formations in LEO. In [45], a hybrid Extended/Unscented Kalman Filter (KF) is proposed for baseline determination for widely spaced formations equipped with single-frequency GNSS receivers. The algorithm which performs double-difference integer ambiguity resolution could achieve excellent estimations of the baseline vector (2 cm RMS error) using GPS data that was generated by a GNSS receiver emulator for two spacecraft 500 km apart. Although fairly precise for large baselines, this algorithm suffers from slow convergence time (around 1 hour).

### 3.1.3 Methodology and contributions

A bi-linear ionospheric model is used in conjunction with ionospheric-free combinations to achieve relative state estimates without the need to perform integer ambiguity resolution. Single difference quantities are fed to an EKF to produce estimates for the relative position and velocity vectors of the deputy with respect to the chief. Single difference observables are specifically chosen for their advantage of cancelling out common biases such as the instrumental delays of the commonly tracked satellites. The relative position and velocity between two spacecraft are estimated directly by the filter rather than estimating the absolute states then subtracting them from each other to obtain the baseline position and velocity, hence, nonlinear relative dynamics between the deputy and the chief spacecraft are used.

Indeed, for large inter-satellite distances with loose relative navigation requirements, differencing the onboard available position solution is shown in [46] to be a good alternative to differencing the GNSS observables, although with reference to dual frequency receivers. Differencing the standalone onboard solutions for the adopted single-frequency receiver is not an option in the context of our problem since the adopted receiver provides a position accuracy ( $1\sigma$ ) of around 10 m [47], which shall provide estimates that are not compliant with the requirements in Section 3.1.1.

The main contributions of this chapter are summaries as follows:

- A set of GNSS observables are tailored for the problem stated in Section 3.1.1, where a

detailed justification is given for why each observable is relevant to our problem;

- A precise bi-linear ionospheric model is utilized in the context of GNSS-based relative navigation to assist in reducing the effect ionospheric delay, which prevails in the context of largely separated receivers;
- Based on the synthesized set of measurements and on the adopted ionospheric model, a set of state variables to be estimated by the [EKF](#) is selected;
- The relative navigation scheme is validated using sensor data from the [European Space Agency \(ESA\)](#) Swarm mission [10] in a simulated real-time setting.

### 3.1.4 Chapter outline

This chapter is organized such that Section 3.2 discusses the different [GNSS](#) observables as well as their models, while Section 3.3 presents the relative navigation methodology in details, where the [EKF](#) measurement and state vectors are synthesized. Moreover, Section 3.4 introduces the dynamics of the state variables as well as the Jacobian matrices necessary for the operation of the [EKF](#), and Section 3.5 outlines the initialization as well as the details of operation of the [EKF](#). The second to last section, Section 3.6, presents the results of running the relative navigation scheme using the Swarm mission data in the best as well as the worst ionospheric conditions of the previous solar cycle. The last section concludes the chapter.

## 3.2 GNSS observables

To get a clearer understanding of the challenges that face achieving precise baseline vector estimates for two satellites flying in a formation, a summary of the [GNSS](#) observables is briefly introduced in the following discussion.

### 3.2.1 Native GNSS measurements

The ultimate goal of [GNSS](#) is to measure the distance between the phase center of the [GNSS](#) satellite antenna and the phase center of the receiver's antenna. Two main native measure-

ments, which represent this range, are obtained by any GNSS receiver; the pseudo-range (code) and the carrier phase. The pseudo-range measurement,  $\rho_{pr}$ , is simply the measured travel time of the signal from the GNSS satellite to the receiver, multiplied by the speed of light. The pseudo-range measurement is a coarse measurement of the geometric range that can be modeled as follows:

$$\rho_{pr} = \rho + c(\delta t - \delta t^s) + \alpha_f S + k_{pr} - k_{pr}^s + \rho_{\oplus} + \nu_{pr}, \quad (3.1)$$

where  $\rho$  is the geometric range,  $c$  is the speed of light,  $\delta t$  and  $\delta t^s$  are the GNSS receiver and satellite clock bias from the GNSS time,  $\alpha_f S$  is the frequency-dependent ionospheric delay with  $\alpha_f$  being the frequency dependent mapping function from the Slant Total Electron Content (STEC),  $S$ , to the ionospheric delay. Moreover, the signal delay,  $k_{pr}$  and  $k_{pr}^s$  are the receiver's and satellite's instrumental delays corresponding to the pseudo-range,  $\rho_{\oplus}$  is the additional distance between satellite and receiver antennas that the signal travels due to the rotation of the earth, and  $\nu_{pr}$  is the pseudo-range noise. It is important to emphasize that all the necessary parameters to calculate the satellite's clock bias  $\delta t^s$  are transmitted in the broadcast navigation message. A general signal that travels from the satellite to the receiver experiences a non-dispersive tropospheric delay as well as a dispersive multipath delay. The two are omitted here, as the receivers in the context of this thesis fly above the troposphere and are far away from any reflecting surface. Letting  $\boldsymbol{\omega}_{\oplus}^f \approx \begin{bmatrix} 0 & 0 & \omega_{\oplus} \end{bmatrix}^T$  be the rotational velocity vector of the Earth in the ECEF frame,  $\mathbf{r}_r^f(t_r) = \begin{bmatrix} x_r^f & y_r^f & z_r^f \end{bmatrix}^T$  be the position vector of the receiver of interest at the time of receiving the signal,  $t_r$ , in the ECEF frame, and  $\mathbf{r}_i^f(t_t) = \begin{bmatrix} x_i^f & y_i^f & z_i^f \end{bmatrix}^T$  be the position vector of the  $i^{\text{th}}$  GNSS satellite at the time of transmitting the signal,  $t_t$ , also represented in the ECEF frame, the correction distance  $\rho_{\oplus}$  can be written as follows [48],

$$\rho_{\oplus} = \frac{1}{c} \left( \boldsymbol{\omega}_{\oplus}^f \times \mathbf{r}_i^f \right)^T \mathbf{r}_r^f = \frac{\omega_{\oplus}}{c} \left( x_i^f y_r^f - x_r^f y_i^f \right), \quad (3.2)$$

where  $\times$  is the cross product operator.



Besides the pseudo-range, the phase of the carrier on which the information are modulated can be utilized to measure the geometric range between the GNSS satellite and the receiver. The carrier phase measurement is much smoother than the pseudo-range, however, it includes an ambiguous term that results from the unknown integer number of wavelengths, and also from the wind-up effect. It can be modeled as,

$$\rho_{cp} = \rho + c(\delta t - \delta t^s) + \lambda_f N_{cp} - \alpha_f S + k_{cp} - k_{cp}^s + \rho_{\oplus} + \nu_{cp}, \quad (3.3)$$

where  $\lambda_f$  is the wavelength corresponding to frequency  $f$ ,  $N_{cp}$  is the floating point ambiguity in cycles,  $k_{cp}$  and  $k_{cp}^s$  are the receiver's and satellite's instrumental delays corresponding to the carrier phase, respectively, and  $\nu_{cp}$  is the carrier phase noise. Typically, the carrier phase is two orders of magnitude more precise than the pseudo-range (i.e.,  $|\nu_{cp}| \approx 0.01 |\nu_{pr}|$ ).

From this point on, and in order to simplify the models of the native measurements, the receiver's instrumental delays,  $k_{pr}$  and  $k_{cp}$ , are assumed to be assimilated by the receiver's clock bias  $c\delta t$  which is to be estimated by the filter that is developed later in Section 3.3, and the satellite's instrumental delays,  $k_{pr}^s$  and  $k_{cp}^s$ , are assimilated by the signal noise. Equations (3.1) and (3.3) can then be rewritten as,

$$\rho_{pr} = \rho + c(\delta t - \delta t^s) + \alpha_f S + \rho_{\oplus} + \nu_{pr}, \quad (3.4)$$

$$\rho_{cp} = \rho + c(\delta t - \delta t^s) + \lambda_f N_{cp} - \alpha_f S + \rho_{\oplus} + \nu_{cp}, \quad (3.5)$$

where  $\nu_{pr}$  and  $\nu_{cp}$  are assumed to be uncorrelated normally distributed random variables with zero means and with standard deviations of  $\sigma_{pr}$  and  $\sigma_{cp}$  respectively.

### 3.2.2 Differential measurements

In the context of relative estimation of the states of the deputy satellite with respect to the chief, the well known single and double differences can be used. Concretely, the [Single Difference Pseudo-Range \(SDPR\)](#) can be obtained by subtracting the code measurement of

the chief from that of the deputy at the same instant and for signals received from a commonly tracked satellite.

$$\rho_{sdpr} = \rho_{pr}|_d - \rho_{pr}|_c = \Delta\rho + c\Delta\delta t + \alpha_f\Delta S + \Delta\rho_{\oplus} + \nu_{sdpr}, \quad (3.6)$$

where  $(\cdot)_c$  is a chief related quantity,  $(\cdot)_d$  is a deputy related quantity,  $\Delta(\cdot) = (\cdot)_d - (\cdot)_c$ , and  $\nu_{sdpr} \sim \mathcal{N}(0, \sigma_{sdpr}^2)$ , with  $\mathcal{N}(\mu, \sigma^2)$  being the normal distribution with a mean  $\mu$  and a variance  $\sigma^2$ , and  $\sigma_{sdpr} \approx \sqrt{2}\sigma_{pr}$ .

The Single Difference Carrier Phase (SDCP) [4] can similarly be produced,

$$\rho_{sdcp} = \rho_{cp}|_d - \rho_{cp}|_c = \Delta\rho + c\Delta\delta t + \lambda_f\Delta N_{cp} - \alpha_f\Delta S + \Delta\rho_{\oplus} + \nu_{sdcp}, \quad (3.7)$$

where  $\nu_{sdcp} \sim \mathcal{N}(0, \sigma_{sdcp}^2)$  with  $\sigma_{sdcp} \approx \sqrt{2}\sigma_{cp}$ .

The single difference concept is graphically illustrated in Figure 3.1.

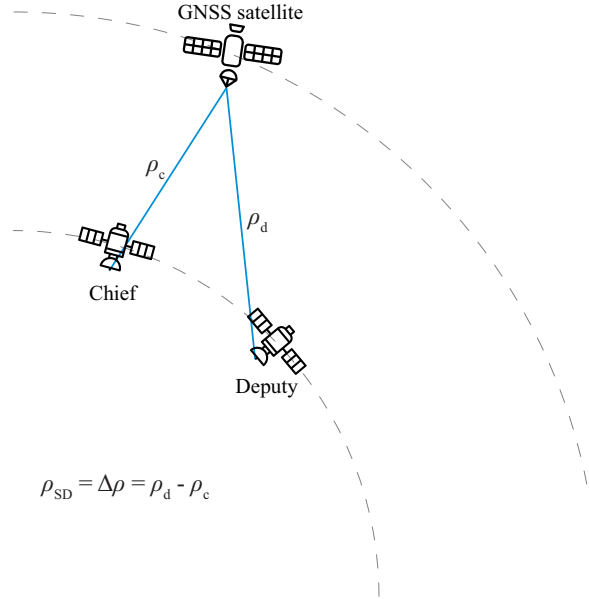


Figure 3.1: Illustration of the single difference concept

Double difference quantities can be obtained by subtracting single difference quantities that were collected from two different satellites. Namely, the **Double Difference Carrier Phase (DDCP)** is constructed by subtracting the single difference carrier phase collected from satellite  $q$  from that which is collected from satellite  $p$  as follows:

$$\rho_{ddcp} = \rho_{sdc p}^p - \rho_{sdc p}^q = \Delta^{p,q} \rho + \lambda_f \Delta^{p,q} N_{cp} - \alpha_f \Delta^{p,q} S + \Delta^{p,q} \rho_{\oplus} + \nu_{ddcp}, \quad (3.8)$$

where  $\Delta^{p,q}(\cdot) = \Delta^p(\cdot) - \Delta^q(\cdot)$ , and  $\Delta^p(\cdot)$  and  $\Delta^q(\cdot)$  are single difference quantities related to the GNSS satellites  $p$  and  $q$  respectively. If we assume  $\nu_{ddcp} \sim \mathcal{N}(0, \sigma_{ddcp}^2)$ , it can be deduced that  $\sigma_{ddcp} \approx 2\sigma_{cp}$ .

The concept of double differential measurements is illustrated in Figure 3.2.

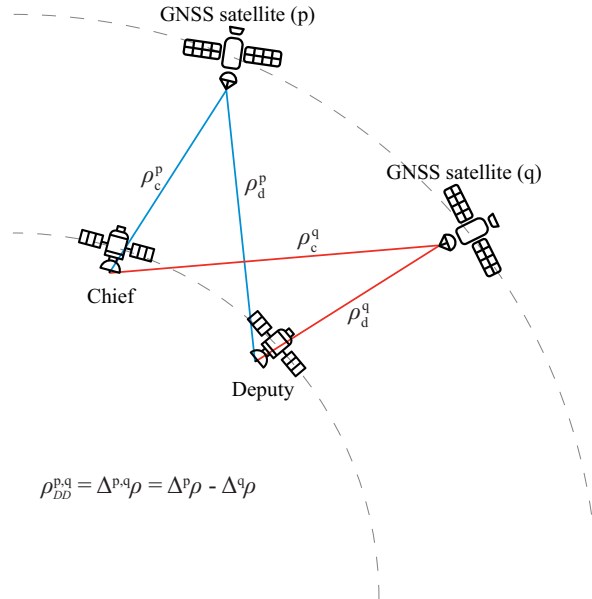


Figure 3.2: Illustration of the double difference concept

It can be seen that using differential measurements not only allows the direct estimation of the baseline vector between two receivers, but also features some interesting properties such as canceled common disturbances. The most important property of the **DDCP** is that

it is not affected by the wind-up [49] which renders the double difference ambiguity  $\Delta^{p,q}N_{cp}$  integer. This property is usually leveraged in resolving and fixing ambiguities mostly by using the [Least-squares AMBiguity Decorrelation Adjustment \(LAMBDA\)](#) [50] or the [Modified LAMBDA \(MLAMBDA\)](#) [51] algorithms. Resolving double difference integer ambiguities is a key factor in obtaining precise relative position and velocity estimates [5, 45, 52].

It is to be stressed that there are various types of differential measurements with different advantages yet, this discussion is out of the scope of this thesis. Interested reader is referred to [48]. It is also important to note that the single and double difference concepts are not limited to the native measurements, as they are also applicable to the measurement combinations, which will be discussed in the next section.

### 3.2.3 Measurements combinations

When the two receivers are close to each other, the differential ionospheric delay  $\alpha_f \Delta S$ , together with the other disturbances, become negligible, that is why differential measurements perform adequately even without integer ambiguity resolution [4, 53]. However, when the two spacecraft are separated by a large baseline, the ionospheric effect becomes the main source of perturbation, and rigorous ionospheric models need to be used. One way to mitigate this problem is to use ionospheric-free combinations. There are many ionospheric-free combinations that require dual-frequency receivers in order to be realized, nevertheless, for single-frequency receivers, the well-known [Group and Phase Ionospheric Correction \(GRAPHIC\)](#) combination [54] can be used. The [GRAPHIC](#) combination is simply the arithmetic mean of the pseudo-range and the carrier phase measurements. It is modeled as,

$$\rho_{gr} = \frac{\rho_{pr} + \rho_{cp}}{2} = \rho + c(\delta t - \delta t^s) + \frac{\lambda_f N_{cp}}{2} + \rho_{\oplus} + \nu_{gr}, \quad (3.9)$$

where  $\nu_{gr} \sim \mathcal{N}(0, \sigma_{gr}^2)$  with  $\sigma_{gr} \approx 0.5\sigma_{pr}$ . The advantages of the [GRAPHIC](#) combination are not limited to being an ionospheric-free combination, as it also experiences lower levels of noise in comparison to the native pseudo-range measurement ( $|\nu_{gr}| \approx 0.5|\nu_{pr}|$ ), however, the noise

level in the GRAPHIC combination is still not as low as that of the carrier phase measurement.

Similar to the SDPR and the SDCP, the Single Difference GRAPHIC (SDGR) [54] can be obtained by subtracting the GRAPHIC value for the chief spacecraft from that of the deputy spacecraft,

$$\rho_{sdgr} = \rho_{gr}|_d - \rho_{gr}|_c = \Delta\rho + c\Delta\delta t + \frac{\lambda_f \Delta N_{cp}}{2} + \Delta\rho_{\oplus} + \nu_{sdgr}, \quad (3.10)$$

where  $\nu_{sdgr} \sim \mathcal{N}(0, \sigma_{sdgr}^2)$  with  $\sigma_{sdgr} \approx \sigma_{pr}/\sqrt{2}$ .

One other interesting linear combination of the GNSS native measurements is the Geometry-Free (GF) ionospheric combination, which can be obtained by subtracting the carrier phase from the pseudo-range. It can be modeled according to (3.4) and (3.5) as,

$$\rho_{gf} = \rho_{pr} - \rho_{cp} = 2\alpha_f S - \lambda_f N_{cp} + \nu_{gf}. \quad (3.11)$$

Again, the Single Difference Geometry-Free (SDGF) [55] combination can be obtained in the same way as the SDPR, the SDCP, and the SDGR,

$$\rho_{sdgf} = \rho_{gf}|_d - \rho_{gf}|_c = 2\alpha_f \Delta S - \lambda_f \Delta N_{cp} + \nu_{sdgf}, \quad (3.12)$$

where  $\nu_{sdgf} \sim \mathcal{N}(0, \sigma_{sdgf}^2)$  with  $\sigma_{sdgf} \approx \sqrt{2}\sigma_{pr}$ .

The linear combinations of measurements discussed in this section can be used individually or concurrently to estimate the relative position of the receivers as well as the auxiliary quantities such as the differential clock bias  $\Delta\delta t$ , the differential ionospheric delay  $\alpha_f \Delta S$ , or the differential float ambiguity  $\Delta N_{cp}$ . It is to be emphasized once again that the real challenge for far-range formations is the elimination of the ionospheric effect as the differential ionospheric delay cannot be approximated by zero.

### 3.3 Estimation strategy

The relative state estimation is considered between two largely separated spacecraft flying in a formation. Although only two spacecraft are considered, the proposed state estimation scheme is applicable to any two spacecraft (typically the chief and any deputy) in a large formation. An [EKF](#) [56] is proposed to process the [GNSS](#) data in order to provide relative position and velocity estimates that are inline with the mission requirements discussed in Section 3.1.1. A brief discussion of how the [EKF](#) works is given in Appendix A. Indeed, using an [EKF](#) for relative navigation is a common practice in the literature, however, a consensus has not been reached on the choice of the states to be estimated as well as the choice of the measurements to be processed. Different settings of the filter can lead to drastically different results even if the same data is processed [57]. In this chapter, the setup of the filter is sought to be optimized for the problem in hand.

#### 3.3.1 Measurement choice

To achieve the relative navigation goal, the native [GNSS](#) measurements (i.e., the pseudo-range and the carrier phase) can be processed, while different advantages can arise from processing different linear combinations of these measurements. The use of the single, or double, difference of the native measurements or their combinations could also be useful [48]. For the problem in hand, the possible measurements and their combinations are briefly discussed in Section 3.2. Following our preliminary study [57], it has been concluded that the [SDCP](#), the [SDGR](#), as well as the [SDGF](#) have the potential to overcome the ionospheric effect, to the level that makes the resulting relative position and velocity estimates inline with the requirements.

Having chosen the observables to be processed by the [EKF](#), the measurement vector  $\mathbf{z}$  of

the [EKF](#) is constructed as,

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{\rho}_{sdcp} \\ \boldsymbol{\rho}_{sdgr} \\ \boldsymbol{\rho}_{sdgf} \end{bmatrix}. \quad (3.13)$$

It is to be noted that  $\boldsymbol{\rho}_{sdcp}$ ,  $\boldsymbol{\rho}_{sdgr}$ , and  $\boldsymbol{\rho}_{sdgf}$  are not scalar values that correspond to one [GNSS](#) satellite, but rather vectors that comprise the measured combinations from all the commonly tracked satellites. In this setting, the length of the measurement vector is  $3n$ , where  $n$  is the number of channels of the [GNSS](#) receiver (12 channels for the Triton-X onboard receiver).

The choice for the measurements to be staged for processing in our context is justified in the following points.

1. The relative position and velocity of one spacecraft with respect to another are estimated directly by the filter, instead of estimating the absolute states of the two spacecraft then subtract the state vector of one spacecraft from that of the other. In this setting, differential measurements have to be used, hence the single difference combinations are chosen. Moreover, another advantage of differential measurements is that they allow some of the common biases (i.e., satellite's clock bias and instrumental delays) to cancel out.
2. As the ionospheric delay is the most significant bias to be accounted for, the use of ionospheric-free combinations comes as no surprise. That is why the [SDGR](#) is used, as [GRAPHIC](#) is the only known single-frequency ionospheric-free combination.
3. Although the [SDGR](#) is an ionospheric-free combination, it is still a noisy measurement, that is why more precise combinations, like the [SDCP](#), need to be included to augment the overall accuracy of the filter. It has to be noted that the inclusion of the [SDCP](#) comes with its own challenges, like having to estimate the ionospheric effect as well as the floating point ambiguities.
4. It is believed that the inclusion of the single difference ionospheric geometry-free combination, the [SDGF](#), shall assist in estimating the ionospheric delay as well as the

ambiguities however being coarse. This hypothesis was shown to be true in [57].

5. Despite the availability of range measurements directly from the inter-satellite link, it is a coarse imprecise measurement [58], which is not expected to improve the overall estimation precision.

### 3.3.2 Ionospheric model

The ionospheric delay  $\alpha_f S$  which appears in the measurement models (3.7) and (3.12) is modeled as the integral of the linear electron density along the ray path between the satellite and receiver [59]. The mapping function,  $\alpha_f$ , is modeled in [60] as,

$$\alpha_f = \frac{40.3 \cdot 10^{16}}{f^2} \text{ m/TECU}, \quad (3.14)$$

where  $f$  is the carrier frequency in Hz, which for the GPS L1 carrier is equal to 1575.42 MHz, and TECU is the Total Electron Content Unit ( $\text{TECU} = 10^{16} \text{ e}^-/\text{m}^2$ ). Furthermore, the STEC,  $S$ , is itself modeled by the Linear Thin Shell (LTS) model [61] as,

$$\begin{aligned} S &= M_{ts} V, \\ M_{ts} &= \frac{R_{\oplus} + h_{ts}}{r_r \sqrt{\left(\frac{R_{\oplus} + h_{ts}}{r_r}\right)^2 - \cos^2(E_{s,r})}}, \\ V &= \begin{bmatrix} 1 & \delta\phi_{ipp} & \delta\lambda_{ipp} \end{bmatrix} \cdot \mathbf{q} \end{aligned} \quad (3.15)$$

where  $M_{ts}$  is the thin shell mapping function from the Vertical Total Electron Content (VTEC) to the STEC through the path of the ray,  $R_{\oplus}$  is the mean radius of the Earth,  $h_{ts}$  is the altitude of the thin shell,  $r_r$  is the magnitude of the position vector, in an Earth centered reference frame, of user  $r$  (i.e., receiver  $r$ ),  $E_{s,r}$  is the elevation angle of the GNSS satellite  $s$  with respect to the user  $r$ , measured up from the local horizon, and  $V$  is the VTEC. The bi-linear approximation of the VTEC [62] is dependant on the coefficient vector  $\mathbf{q} = \begin{bmatrix} q_0 & q_1 & q_2 \end{bmatrix}^T$



which is to be estimated by the filter, as well as  $\delta\phi_{ipp}$  and  $\delta\lambda_{ipp}$  which are defined as follows:

$$\begin{aligned}\delta\phi_{ipp} &= \phi_{ipp} - \phi_r, \\ \delta\lambda_{ipp} &= \lambda_{ipp} - \lambda_r,\end{aligned}\tag{3.16}$$

where  $\phi_{ipp}$  and  $\lambda_{ipp}$  are the latitude and longitude, respectively, of the [Ionospheric Pierce Point \(IPP\)](#) [63], while  $\phi_r$  and  $\lambda_r$  are those of the receiver.

A common arbitrary choice for ground-based receivers is setting  $h_{ts}$  as the altitude of the F2 peak [64], nonetheless, in this thesis, it is set to a free variable that is estimated by the [EKF](#).

Indeed, the ionospheric effect is the largest bias in the available measurements that needs to be accounted for, that is why the [LTS](#) model is chosen, as it is preferred to the other empirical ionospheric models such as the commonly used Lear mapping [65]. The [LTS](#) model provides better estimates of the ionospheric delay, although at the cost of a slightly higher computational demands [61].

### 3.3.3 State variables choice

This section is dedicated to introducing the state variables to be estimated by the [EKF](#). Clearly, the main variables that need to be estimated are the baseline vector and the relative velocity between two spacecraft, however, some auxiliary variables (e.g., receiver's clock bias, carrier phase float ambiguities,...) need to be estimated in order to increase the precision of the filter. In fact, if an [EKF](#) filter is run without considering these auxiliary variables, especially the receiver's clock bias, it is susceptible to generate unusable estimates.

As discussed in Section 3.3.1, a unique set of state variables does not exist, and the choice of the state vector is customary. In fact, the choice of the state variables to be estimated is heavily dependant on the choice of the measurements combinations to be fed to the filter, which itself is customary to choose. In this chapter, the advantages of the [SDPR](#), the [SDGR](#), and the [SDGF](#) combinations, defined in equations (3.7), (3.10), and (3.12) respectively, are

leveraged.

The **EKF** state vector  $\mathbf{x}$  is constructed based on the chosen measurements. The choice of the state variables is presented below.

$$\mathbf{x} = \begin{bmatrix} \Delta \mathbf{x}^f \\ c\Delta\delta t \\ c\Delta\dot{\delta}t \\ h_{ts} \\ \mathbf{q}^\top \\ \Delta \mathbf{N}_{cp}^\top \end{bmatrix}, \quad (3.17)$$

where  $\Delta \mathbf{x}^f \in \mathbb{R}^6$  is the vector that contains the relative position and velocity coordinates of the deputy with respect to the chief, expressed in the **ECEF** frame,  $\Delta\delta t$  is the differential receiver's clock bias with  $\Delta\dot{\delta}t$  being its rate of change,  $h_{ts}$  and  $\mathbf{q} \in \mathbb{R}^3$  are the parameters related to the ionospheric model, and  $\Delta \mathbf{N}_{cp} \in \mathbb{R}^n$  is the differential carrier phase float ambiguity vector for all the commonly tracked satellites, with  $n$  being the number of channels of the **GNSS** receiver (12 channels for Triton-X onboard receiver). In this setting, the length of the state vector in this setting is  $12 + n$ .

### 3.4 Dynamics of the EKF state vector

In this section, the nonlinear dynamical models of the state variables together with the Jacobian matrices necessary for the operation of the **EKF** are presented. The dynamics of the state vector variables are separated into the relative orbital dynamics, concerning the  $\Delta \mathbf{x}^f$  vector, and the dynamics of the auxiliary variables which concern the rest of the state variables in (3.17).

### 3.4.1 Relative orbital dynamics

The relative motion of a deputy satellite with respect to a chief spacecraft as seen by an observer lying on the [ECEF](#) frame is described by Eq. (2.25), and is mentioned here once again for completeness.

$$\Delta \dot{\mathbf{x}}^f = \begin{bmatrix} \Delta \mathbf{v}^f \\ -\frac{\mu_{\oplus}}{r_c^3} \left( \left( \frac{r_c}{\sqrt{r_c^2 + 2\mathbf{r}_c^f \cdot \Delta \mathbf{r}^f + \Delta r^2}} \right)^3 (\mathbf{r}_c^f + \Delta \mathbf{r}^f) - \mathbf{r}_c^f \right) - 2\boldsymbol{\omega}_{\oplus}^f \times \Delta \mathbf{v}^f \\ -\boldsymbol{\omega}_{\oplus}^f \times \boldsymbol{\omega}_{\oplus}^f \times \Delta \mathbf{r}^f + \Delta \mathbf{u}^f + \Delta \mathbf{u}_{J_2}^f + \mathbf{w}_{\Delta \mathbf{v}}^f \end{bmatrix}, \quad (3.18)$$

where  $\mathbf{r}_c^f$  is the position vector of the chief satellite with a magnitude of  $r_c$ ,  $\Delta \mathbf{r}^f = \mathbf{r}_d^f - \mathbf{r}_c^f$  and  $\Delta \mathbf{v}^f = \mathbf{v}_d^f - \mathbf{v}_c^f$  are the relative position and velocity vectors respectively from the chief to the deputy spacecraft with magnitudes of  $\Delta r$  and  $\Delta v^f$ , all expressed, and as seen by an observed, in the [ECEF](#) frame. Moreover,  $\Delta \mathbf{u}_{J_2}^f$  is the differential  $J_2$  perturbing acceleration vector, and  $\mathbf{w}_{\Delta \mathbf{v}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\Delta \mathbf{v}})$  collates all the relative acceleration noises, with  $\mathbf{Q}_{\Delta \mathbf{v}}$  being defined as,

$$\mathbf{Q}_{\Delta \mathbf{v}} = \text{diag} \left( \sigma_{\Delta v_x}^2, \sigma_{\Delta v_y}^2, \sigma_{\Delta v_z}^2 \right), \quad (3.19)$$

where  $\text{diag}(\cdot, \cdot, \dots)$  creates a diagonal matrix with the specified inputs as its diagonal elements.

In the prediction phase of the [EKF](#), equation (3.18) is numerically integrated after omitting the disturbance, to obtain a prediction of the relative position and velocity states at each time step. It is important to note that an estimate of the absolute position and velocity vectors of the chief spacecraft is essential for the numerical integration of (3.18) and for a proper operation of the [EKF](#). These are directly acquired from the receiver's noisy onboard solution at each prediction step. The same approach was used in [44].

### 3.4.2 Dynamics of the auxiliary variables

The state variables  $c\Delta\dot{\delta}t$  and  $\mathbf{q}$  are modeled as Gaussian random walk processes while the rest of the auxiliary variables are modeled as constants,

$$\begin{aligned} c\Delta\ddot{\delta}t &= w_{c\Delta\dot{\delta}t}, \\ \dot{h}_{ts} &= 0, \\ \dot{\mathbf{q}} &= \mathbf{w}_{\mathbf{q}}, \\ \Delta\dot{\mathbf{N}}_{cp} &= \mathbf{0}, \end{aligned} \tag{3.20}$$

where  $w_{c\Delta\dot{\delta}t} \sim \mathcal{N}(0, \sigma_{c\Delta\dot{\delta}t}^2)$  and  $\mathbf{w}_{\mathbf{q}} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\mathbf{q}})$ . The noise in the  $c\Delta\ddot{\delta}t$  signal is set to account for the sudden clock jumps that manufacturers embed into their receivers in order to control the magnitude of the clock biases [66]. Moreover,  $\mathbf{Q}_{\mathbf{q}}$  is defined as,

$$\mathbf{Q}_{\mathbf{q}} = \text{diag}(\sigma_{q_0}^2, \sigma_{q_1}^2, \sigma_{q_2}^2). \tag{3.21}$$

### 3.4.3 State transition matrix

With the definition of the state variables (3.17) in mind, together with their models (3.18) and (3.20), the **State Transition Matrix (STM)** which is essential for the operation of the **EKF** can be divided into several sub-STMs as,

$$\mathbf{F} = \frac{\partial \mathbf{x}_k}{\partial \mathbf{x}_{k-1}} = \begin{bmatrix} \mathbf{F}_{\Delta\mathbf{x}^f} & \mathbf{0}_{6 \times 1} & \mathbf{0}_{6 \times 1} & \mathbf{0}_{6 \times 1} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times n} \\ \mathbf{0}_{1 \times 6} & 1 & t_k - t_{k-1} & 0 & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times n} \\ \mathbf{0}_{1 \times 6} & 0 & F_{c\Delta\dot{\delta}t} & 0 & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times n} \\ \mathbf{0}_{1 \times 6} & 0 & 0 & F_{h_{ts}} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times n} \\ \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{F}_{\mathbf{q}} & \mathbf{0}_{3 \times n} \\ \mathbf{0}_{n \times 6} & \mathbf{0}_{n \times 1} & \mathbf{0}_{n \times 1} & \mathbf{0}_{n \times 1} & \mathbf{0}_{n \times 3} & \mathbf{F}_{\Delta\mathbf{N}_{cp}} \end{bmatrix}, \tag{3.22}$$

where  $\mathbf{F}_{\Delta\mathbf{x}^f} \in \mathbb{R}^{6 \times 6}$ ,  $F_{c\Delta\dot{\delta}t}$ ,  $F_{h_{ts}} \in \mathbb{R}$ ,  $\mathbf{F}_{\mathbf{q}} \in \mathbb{R}^{3 \times 3}$ , and  $\mathbf{F}_{\Delta\mathbf{N}_{cp}} \in \mathbb{R}^{n \times n}$  are the state transition matrices for  $\Delta\mathbf{x}$ ,  $c\Delta\dot{\delta}t$ ,  $h_{ts}$ ,  $\mathbf{q}$ , and  $\Delta\mathbf{N}_{cp}$ , respectively, while  $t_k$  and  $t_{k-1}$  are two consecutive

time instants to which the STMs correspond. Furthermore,  $\mathbf{0}$  represents a matrix of zeros with its dimensions indicated as a subscript.

Various linearization techniques can be adopted in order to obtain the STM  $\mathbf{F}_{\Delta \mathbf{x}^f}$  [67], and in any case, it is not used to propagate the states, but rather to propagate the estimated states covariance matrix (refer to Eq. (A.3) representing the prediction phase of the EKF), hence the constraint of having a very precise STM becomes much looser. The simple STM in Eq. (2.32) which is obtained from the closed form solution to the Clohessy–Wiltshire (CW) equations can also be utilized, however, one needs to keep an eye on the fact that CW equations provide the solution in the RTN frame, and the obtained STM has to be rotated to the ECEF. Here, an even simpler approach is adopted by linearizing the equations of motion (3.18) taking the chief’s orbit as a reference for the linearization. In this case, the STM of the relative states  $\mathbf{F}_{\Delta \mathbf{x}^f}$  is only dependent on the chief’s states as suggested by Eq. (B.3). Appendix B includes the details of why this assumption is plausible. Approximating the Jacobian of the relative states by the Jacobian of the chief’s states yields,

$$\begin{aligned} \mathbf{J}_{\Delta \mathbf{x}^f} &= \frac{\partial \Delta \dot{\mathbf{x}}^f}{\partial \Delta \mathbf{x}^f} \approx \left. \frac{\partial \dot{\mathbf{x}}^f}{\partial \mathbf{x}^f} \right|_{\mathbf{x}^f = \mathbf{x}_c^f} \\ \mathbf{F}_{\Delta \mathbf{x}^f} &= \frac{\partial \Delta \mathbf{x}_k^f}{\partial \Delta \mathbf{x}_{k-1}^f} \approx \exp \left( (t_k - t_{k-1}) \mathbf{J}_{\Delta \mathbf{x}^f} |_{\mathbf{x}_{c,k-1}^f} \right), \end{aligned} \quad (3.23)$$

where  $\mathbf{J}_{\Delta \mathbf{x}^f}$  is the state-dependant Jacobian matrix that can be calculated by partial differentiation of (2.7), while  $\mathbf{J}_{\Delta \mathbf{x}^f} |_{\mathbf{x}_{c,k-1}^f}$  is the same matrix evaluated at the chief’s state at time  $t_{k-1}$ .

The state transition matrices of the auxiliary variables can be directly derived from the

linear system of equations (3.20) after omitting the noise as follows:

$$\begin{aligned}
F_{c\Delta\delta t} &= 1, \\
F_{h_{ts}} &= 1, \\
\mathbf{F}_q &= \mathbf{I}_3. \\
F_{\Delta\mathbf{N}_{cp}|_{i,j}} &= \begin{cases} 1, & P_i \in \mathcal{P}_{k-1} \text{ And } i = C_i|_k \text{ And } j = C_i|_{k-1}, \\ 0, & \text{Otherwise} \end{cases}
\end{aligned} \tag{3.24}$$

where  $\mathbf{I}_{(\cdot)}$  is an identity matrix of size  $(\cdot)$ ,  $F_{\Delta\mathbf{N}_{cp}|_{i,j}}$  is the entry  $(i, j)$  of the  $\mathbf{F}_{\Delta\mathbf{N}_{cp}}$  matrix,  $P_i$  is the Pseudo-Random Noise (PRN) code of the  $i^{\text{th}}$  commonly tracked satellite (at time  $t_k$ ),  $\mathcal{P}_{k-1}$  is the set of PRNs of the commonly visible satellites at time instant  $t_{k-1}$ , and  $C_i|_k$  and  $C_i|_{k-1}$  are indices to the chief's receiver channels that captured the measurement from the  $i^{\text{th}}$  commonly visible satellite at times  $t_k$  and  $t_{k-1}$  respectively. Indeed, the differential ambiguity for the newly tracked satellites are set to zero, however, the process noise covariance matrix has to be adapted to reflect the uncertainty of this initial guess. Equation (3.24) can be used to construct the full STM (3.22), and also to propagate the auxiliary states in the prediction phase of the EKF.

#### 3.4.4 Measurement model

Although the measurement vector in (3.13) is modeled by equations (3.7), (3.10), and (3.12), the Jacobian matrix  $\mathbf{H}$  of these nonlinear functions needs to be constructed as a requirement for the update phase of the EKF. Concretely, the nonlinear measurement model is presented once again as follows:

$$\mathbf{h} = \begin{bmatrix} \Delta\boldsymbol{\rho} + c\Delta\delta t + \lambda_f\Delta\mathbf{N}_{cp} - \alpha_f\Delta\mathbf{S} + \Delta\boldsymbol{\rho}_{\oplus} \\ \Delta\boldsymbol{\rho} + c\Delta\delta t + \frac{1}{2}\lambda_f\Delta\mathbf{N}_{cp} + \Delta\boldsymbol{\rho}_{\oplus} \\ 2\alpha_f\Delta\mathbf{S} - \lambda_f\Delta\mathbf{N}_{cp} \end{bmatrix}. \tag{3.25}$$

where the differential STEC vector,  $\Delta\mathbf{S}$ , is obtained from the VTEC vector according to the mapping (3.15).

The state dependant Jacobian matrix  $\mathbf{H}$  can then be obtained,

$$\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{H}_{\Delta \mathbf{r}^f} & \mathbf{0}_{m \times 3} & \mathbf{1}_{m \times 1} & \mathbf{0}_{m \times 1} & -\mathbf{H}_{h_{ts}} & -\mathbf{H}_{\mathbf{q}} & \lambda_f \mathbf{H}_{\Delta \mathbf{N}_{cp}} \\ \mathbf{H}_{\Delta \mathbf{r}^f} & \mathbf{0}_{m \times 3} & \mathbf{1}_{m \times 1} & \mathbf{0}_{m \times 1} & \mathbf{0}_{m \times 1} & \mathbf{0}_{m \times 3} & \frac{1}{2} \lambda_f \mathbf{H}_{\Delta \mathbf{N}_{cp}} \\ \mathbf{0}_{m \times 3} & \mathbf{0}_{m \times 3} & \mathbf{0}_{m \times 1} & \mathbf{0}_{m \times 1} & 2\mathbf{H}_{h_{ts}} & 2\mathbf{H}_{\mathbf{q}} & -\lambda_f \mathbf{H}_{\Delta \mathbf{N}_{cp}} \end{bmatrix}, \quad (3.26)$$

where  $\mathbf{H}_{\Delta \mathbf{r}^f} \in \mathbb{R}^{m \times 3}$ ,  $\mathbf{H}_{h_{ts}} \in \mathbb{R}^{m \times 1}$ ,  $\mathbf{H}_{\mathbf{q}} \in \mathbb{R}^{m \times 3}$ , and  $\mathbf{H}_{\Delta \mathbf{N}_{cp}} \in \mathbb{R}^{m \times n}$  are partial derivative matrices, with  $m \leq n$  being the number of the commonly visible satellites and  $n$  is the number of channels of the receiver. Moreover,  $\mathbf{1}$  represents a matrix of ones with its dimensions indicated in the subscript.

Matrix  $\mathbf{H}_{\Delta \mathbf{r}^f}$  is a Jacobian matrix that can be obtained by partial differentiation of the  $(\Delta \boldsymbol{\rho} + \Delta \boldsymbol{\rho}_{\oplus})$  part of the measurement model (3.25) using the states of the chief (refer to Appendix B),

$$\mathbf{H}_{\Delta \mathbf{r}^f} = \frac{\partial}{\partial \Delta \mathbf{r}} (\Delta \boldsymbol{\rho} + \Delta \boldsymbol{\rho}_{\oplus}) \approx \frac{\partial}{\partial \mathbf{r}} (\boldsymbol{\rho} + \boldsymbol{\rho}_{\oplus}) \Big|_c = \begin{bmatrix} \mathbf{e}_{1,c}^f & \mathbf{e}_{1,c}^f & \dots & \mathbf{e}_{m,c}^f & \mathbf{0}_{3 \times 1} & \dots & \mathbf{0}_{3 \times 1} \end{bmatrix}^T, \quad (3.27)$$

where the zero vectors at the end account for the chief's receiver channels that did not track any satellite or that tracked a satellite that is not visible by the deputy,  $\mathbf{e}_{i,c}^f$  comprises the unit baseline vector from the  $i^{\text{th}}$  commonly tracked satellite to the chief in the ECEF frame in addition to a correction term for the Earth's rotation,

$$\mathbf{e}_{i,c}^f = \frac{\mathbf{r}_c^f - \mathbf{r}_i^f}{\|\mathbf{r}_c^f - \mathbf{r}_i^f\|} + \frac{\omega_{\oplus}}{c} \begin{bmatrix} -y_i^f & x_i^f & 0 \end{bmatrix}^T. \quad (3.28)$$

Using Equations (3.14) and (3.15), the  $i^{\text{th}}$  entry in the  $\mathbf{H}_{h_{ts}}$  vector, which corresponds to the measurement from the  $i^{\text{th}}$  commonly visible satellite by both the deputy and the chief,

can be written as,

$$\begin{aligned} H_{h_{ts}}|_i &= \frac{40.3 \cdot 10^{16}}{f^2} \left[ \left( \frac{\partial M_{ts}}{\partial h_{ts}} V \right)_{i,d} - \left( \frac{\partial M_{ts}}{\partial h_{ts}} V \right)_{i,c} \right], \\ \frac{\partial M_{ts}}{\partial h_{ts}} &= - \frac{\cos^2(E_{i,r})}{r_r \sqrt{\left[ \left( \frac{R_{\oplus} + h_{ts}}{r_r} \right)^2 - \cos^2(E_{i,r}) \right]^3}}, \end{aligned} \quad (3.29)$$

while the  $i^{\text{th}}$  row in the  $\mathbf{H}_{\mathbf{q}}$  can be written as,

$$\begin{aligned} H_{\mathbf{q}}|_i &= \frac{40.3 \cdot 10^{16}}{f^2} \left[ \left( M_{ts} \frac{\partial V}{\partial \mathbf{q}} \right)_{i,d} - \left( M_{ts} \frac{\partial V}{\partial \mathbf{q}} \right)_{i,c} \right], \\ \frac{\partial V}{\partial \mathbf{q}} &= \begin{bmatrix} 1 & \delta\phi_{ipp} & \delta\lambda_{ipp} \end{bmatrix}. \end{aligned} \quad (3.30)$$

Matrix  $\mathbf{H}_{\Delta \mathbf{N}_{cp}}$  is, in fact, a boolean rearrangement matrix with entries,

$$H_{\Delta \mathbf{N}_{cp}}|_{i,j} = \begin{cases} 1, & P_i \in \mathcal{P}_{k-1} \text{ And } j = C_i \\ 0, & \text{Otherwise} \end{cases}. \quad (3.31)$$

### 3.5 EKF initialization and operation

In order for the EKF to operate properly, it needs an initial guess for the state vector,  $\mathbf{x}$ , as well as the estimated states covariance matrix,  $\mathbf{P}$ . Indeed, many ways such as the available onboard solution of each satellite or weighted least-square schemes can be incorporated to calculate a fairly close initial relative position and velocity vectors, yet, in order to demonstrate that the proposed filter converges even for uncertain initial conditions, the state variables are initiated with randomly selected values within the true range of each variable. The estimation covariance matrix, on the other hand, is inaugurated as a diagonal matrix with large variances to reflect the uncertainties of the initial guess of the state vector. During operation, it is of a huge importance to re-initiate the values of  $\Delta \mathbf{N}_{cp}$  to account for newly tracked satellites (see equation (3.24)). Once a value is initialized, the corresponding process noise variance needs to be set to a large value in the process noise covariance matrix  $\mathbf{Q}$ . Moreover, a proper



rearrangement of the estimated state covariance matrix,  $\mathbf{P}$ , needs to be carried out in order to account for the satellites that went out of sight.

Besides the initialization of the state vector and the estimated states covariance matrix, two other covariance matrices need to be defined for the [EKF](#) to operate efficiently, the process noise covariance matrix  $\mathbf{Q}$  and the observation covariance matrix  $\mathbf{R}$ .

The process noise covariance matrix  $\mathbf{Q}$  is set to the following semi-definite time-varying matrix,

$$\mathbf{Q} = \text{diag} \left( \mathbf{0}_{6 \times 6}, \mathbf{Q}_{\Delta \mathbf{v}}, 0, \sigma_{c\Delta\delta t}^2, 0, \mathbf{Q}_{\mathbf{q}}, \mathbf{Q}_{\Delta \mathbf{N}_{cp}} \right), \quad (3.32)$$

where  $\mathbf{Q}_{\Delta \mathbf{N}_{cp}}$  is generally a matrix of zeros, except when a newly tracked satellite is introduced, then the corresponding diagonal entry is set to a large value to reflect the unreliability of the corresponding initiated  $\Delta N$  value. The state variables' noises are assumed uncorrelated, thus all the covariance elements of the  $\mathbf{Q}$  matrix are zeros. Furthermore, the process noise covariance matrix is in general a time-varying matrix, as  $\mathbf{Q}_{\Delta \mathbf{N}_{cp}}$  needs constant rearrangement as new satellites are being tracked. The diagonal entries for  $\mathbf{Q}_{\Delta \mathbf{N}_{cp}}$  are defined as,

$$Q_{\Delta \mathbf{N}_{cp}}|_{i,i} = \begin{cases} 10^2, & C_i \text{ tracks a new satellite} \\ 0, & \text{Otherwise} \end{cases}, \quad (3.33)$$

Although the multipath effect was not incorporated in the measurements model [\(3.25\)](#), it is not entirely true that a receiver in a low Earth orbit will not experience any multipath interference. It has been shown in [\[68\]](#) that the Earth's surface reflections could indeed disturb the [GNSS](#) signals especially those which have to travel from a low elevation satellite with respect to the receiver. Nonetheless, this type of interference lasts only for short periods and can be avoided by choosing a suitable cut-off elevation angle, beyond which the signal is simply discarded. Instead of hard coding a cut-off elevation angle, which also means that parts of the observations have to be discarded, the measurement covariance matrix is set to vary with the elevation angle of the transmitting satellite. In this setting, the lower the elevation angle,

the higher the variance of the observation noise is set. Assuming uncorrelated measurement noises, the observation noise covariance matrix is set to the following time-varying matrix,

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{sdcp} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{sdgr} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{sdgf} \end{bmatrix}, \quad (3.34)$$

where  $\mathbf{R}_{sdcp}$ ,  $\mathbf{R}_{sdgr}$ , and  $\mathbf{R}_{sdgf}$  are the noise covariance matrices corresponding to the SDPR, SDGR, SDGF measurements respectively. The off-diagonal elements of the  $\mathbf{R}_{sdcp}$ ,  $\mathbf{R}_{sdgr}$ , and  $\mathbf{R}_{sdgf}$  matrices are all zeros, while the diagonal elements are defined as,

$$R_{(\cdot)}|_{i,i} = \frac{\sigma_{(\cdot)}^2}{\sin^2(E_{min}^s) + \epsilon}, \quad (3.35)$$

where  $\sigma_{(\cdot)}$  is a pre-defined constant standard deviation for the observation (either the SDPR, the SDGR, or the SDGF),  $E_{min}^s = \min(|E_d^s|, |E_c^s|)$  is the minimum magnitude elevation angle between either the deputy or the chief when both track the GNSS satellite,  $s$ , and  $\epsilon$  is a small constant that acts as a safeguard from dividing by zero. Clearly, the size of each of the observation noise covariance matrices is time-varying as it depends on the number of commonly tracked satellites.

During the operation of the EKF, a measurement vector at any time step is said to have at least one outlier in case the collective squared Mahalanobis distance [69] is found to be greater than the inverse of the  $\chi^2$  (chi-squared) cumulative distribution function of a significance level of 5% and with the length of the measurement vector as the degrees of freedom. Once a batch of measurements is detected to have outliers, those are detected and rejected using the individual squared Mahalanobis distance test [70].

### 3.6 Results and discussion

In order to validate the algorithm, data from the [ESA Swarm mission \[10\]](#) are collected and simulated in a real-time setting. Swarm is an Earth observation mission that consists of three identical satellites, Alpha, Bravo, and Charlie; which were launched in November 2013.

The algorithm was tested on two different full days worth of the Swarm mission [GNSS](#) data, namely on 16-Jul-2014 and on 08-Dec-2020. These two days are chosen keeping in mind that the algorithm needs to be tested in the best and in the worst ionospheric conditions. The year 2020 witnessed the lowest solar activity of the [24<sup>th</sup> Solar Cycle \(SC24\) \[71\]](#), hence, the best ionospheric conditions for [GNSS](#) navigation are expected. On the other hand, the peak of the [SC24](#) solar cycle happened in 2014 [71] and the worst ionospheric conditions are present around this time. Indeed, [SC24](#) is not the most intense cycle in terms of the solar radiation, and consequently in the total electron content of the ionosphere. However, the author did not manage to get access to other [GNSS](#) data that were recorded during a stronger solar cycle, for two receivers flying in low earth orbit with a large inter-satellite distance.

The relative navigation is performed between Alpha (deputy) and Charlie (chief) spacecraft which share the same orbit (150 km apart in 2014 and 100 km apart in 2020 on the dates of estimation).

In order for the [EKF](#) to converge, precise statistics of the process noises as well as the measurement noises need to be provided. A particular issue that needs to be addressed here is that, although the [SDCP](#) is commonly considered a smooth signal with a noise level of order of millimeters, it cannot be considered very smooth in the context of largely separated receivers since the differential ionospheric delay is not negligible, and also as the ionospheric model is not accurate to the millimeters level. The statistics that were adopted in the simulations are presented in the [Table 3.1](#). It is to be noted that the standard deviations of the noises of [SDPR](#), [SDGR](#), and [SDGF](#) can be deduced from the standard deviations of the carrier phase

and the pseudo-range noises (refer to the discussions in Section 3.2). Moreover, the standard deviation of the carrier phase noise for the 2014 data is set to a higher value than that of the 2020 data since the quality of the latter is better than that of the former.

Table 3.1: Assumed constant statistics used in the simulations

$\sigma_{\Delta v_x} \equiv \sigma_{\Delta v_y} \equiv \sigma_{\Delta v_z}$	$\sigma_{c\Delta\dot{t}}$	$\sigma_{q_0} \equiv \sigma_{q_1} \equiv \sigma_{q_2}$	$\sigma_{pr}$	$\sigma_{cp _{2014}}$	$\sigma_{cp _{2020}}$
0.01 m/s <sup>2</sup>	1 m/s	$1/\sqrt{2}$ TECU/s	1.5 m	0.2 m	0.1 m

After processing the 2014 data, estimates for the relative position and velocity vectors became available. Figure 3.3 depicts the relative position and velocity errors between the two spacecraft for the 2014 data. It is important to note that the relative position and velocity vectors are both estimated in the ECEF frame, then the error vectors as well as the estimation covariance matrix are rotated to the RTN frame of the chief, by making use of the knowledge of the receiver’s onboard solution for the chief absolute position. In order to obtain the error signals, the output of the post-processing Precise Orbit Determination (POD) algorithms is used as the ground truth. The POD output is accurate to the level of  $\pm 1$  cm and is provided as part of the Swarm mission data.

Results in Fig. 3.3 show that more than 99.7% of the errors lie below the  $3\sigma$  threshold, with  $\sigma$  being the standard deviation of the estimation error, which suggests that our assumptions for the model and measurement statistics (see Table 3.1) are conservative.

For the 2014 data, 3D RMS errors of 26 cm and 4 cm/s could be obtained for relative distance and relative speed respectively, while the filter converged after only 1 iteration. The convergence point in this setting is defined as the first occurrence of three consecutive filter estimation errors that are less than or equal to twice the RMS. Results with such convergence rate and accuracy, especially without implementing integer ambiguity resolution and fixing techniques, could only be obtained thanks to the incorporation of the presented measurement setting as well as the incorporation of a precise ionospheric model. This claim is supported by the preliminary trials to run the filter with different measurement and ionospheric model settings [57].

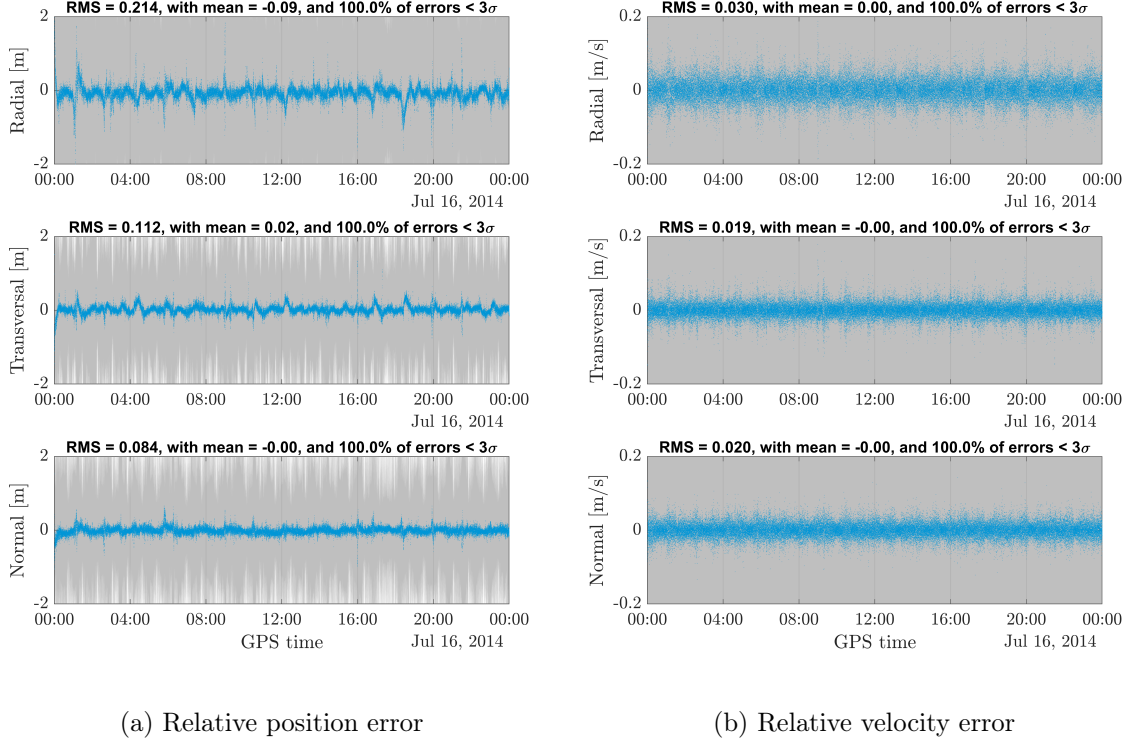


Figure 3.3: Estimation errors for 1 day of the Swarm mission data during the peak of SC24

The number of the commonly tracked satellites as well as the [Relative Position Dilution Of Precision \(RP-DOP\)](#) for the 2014 data are depicted in Fig. 3.4, where the RP-DOP is defined as,

$$\text{RP-DOP} = \sqrt{\sigma_{\Delta x}^2 + \sigma_{\Delta y}^2 + \sigma_{\Delta z}^2}, \quad (3.36)$$

with  $\sigma_{\Delta x}^2$ ,  $\sigma_{\Delta y}^2$ , and  $\sigma_{\Delta z}^2$  being the estimation variances of the relative position components, extracted from the estimated covariance matrix.

It is clear from Fig. 3.4 that the RP-DOP is likely to have spikes when the number of the commonly tracked satellites is low (e.g., 3 satellites). It is believed that these spikes appear not only due to the low number of the commonly visible satellites, but also due to the geometry of these tracked satellites as well as the elevation angles of the satellites with respect to the receivers (see for example the spike around 16:00 in Fig. 3.4 which appear when the

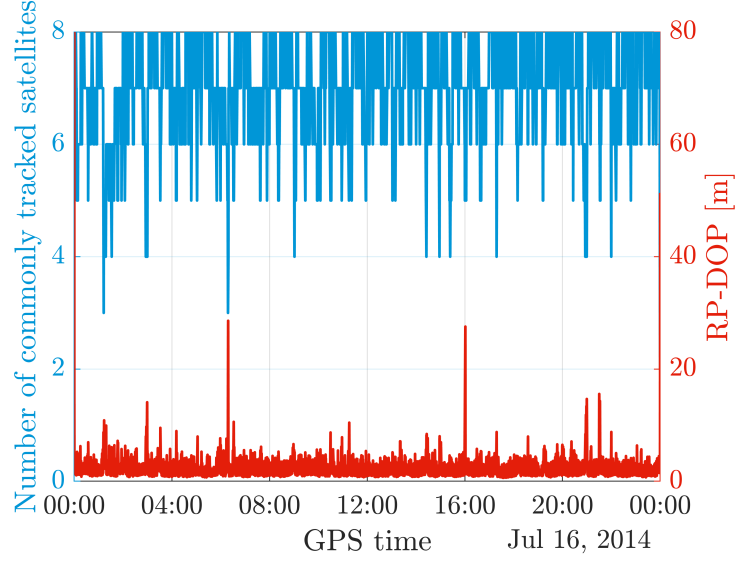


Figure 3.4: Precision indicators

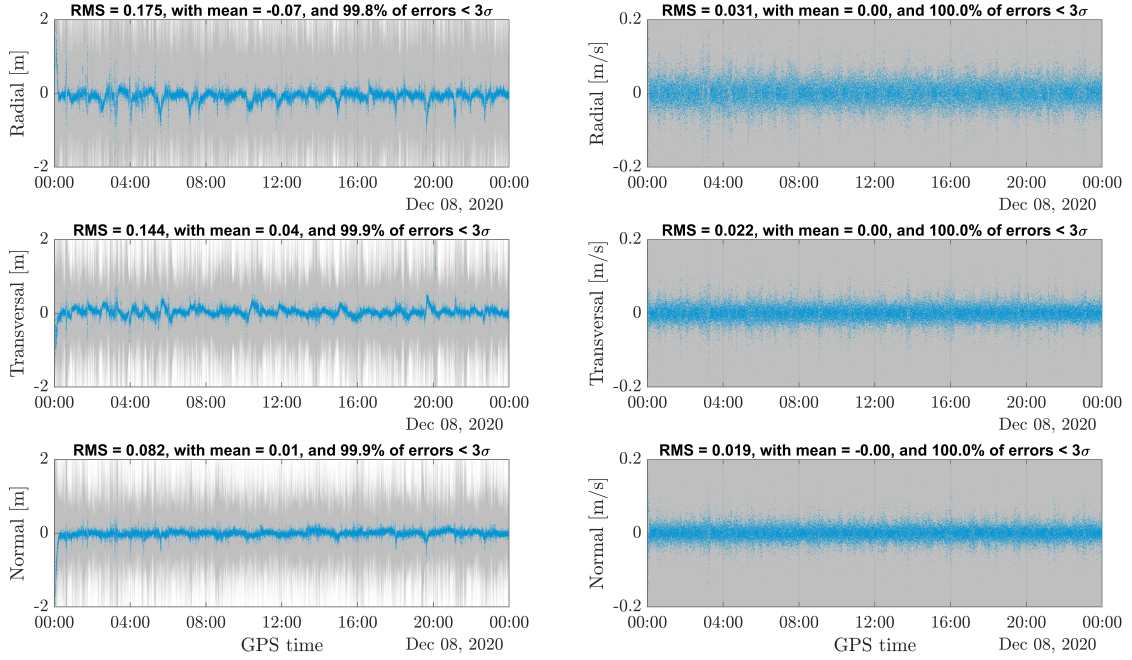
number of commonly tracked satellite is fairly large, 5 satellites).

The 2020 data were processed as well, and the relative position and velocity are estimated. Figure 3.5 illustrates the relative position and velocity estimation errors respectively for the 2020 data.

An RMS error of 24 cm could be obtained for the relative distance and of 4 cm/s for the relative speed. Indeed, the estimates of the 2020 data are of better quality than those of the 2014 data since the ionospheric conditions are better, however, the obtained estimation errors from both data are not substantially different. It is believed that thanks to the adoption of a fairly precise ionospheric model, almost indistinguishable performances of the filter could be obtained, even for different GNSS data streams with different ionospheric conditions.

### 3.7 Conclusion

In this chapter, the problem of GNSS-based relative navigation between two spacecraft with large inter-satellite distance and with single-frequency receivers is treated through adopting the classical EKF. Although using an EKF for relative navigation is a common practice in the



(a) Relative position error

(b) Relative velocity error

Figure 3.5: Estimation errors for 1 day of the Swarm mission data during the nadir of SC24

literature, there exists endless variants of filter settings. The contribution of this research is the optimization of the filter setting for the specific problem in hand, by fixing the measurements to be fed to the filter, the handling of the ionospheric delay, the choice of the auxiliary state variables, and the definition of the process and measurement noises covariance matrices.

A measurement setting comprising the [Single Difference Carrier Phase \(SDCP\)](#), the [Single Difference GRAPHIC \(SDGR\)](#), and the [Single Difference Geometry-Free \(SDGF\)](#) combination is proposed. While the incorporation of the [SDCP](#) and the [SDGR](#) comes as no surprise, augmenting the measurement vector with the [SDGF](#) combination could help estimating the differential ionospheric delay as well as the differential floating point ambiguities, improving the overall estimation accuracy as well as the convergence time.

The fairly precise, however simple, thin shell ionospheric model was adopted as it provides

superior estimates of the ionospheric delay than those of the commonly used empirical models.

As the quality of the GNSS signal improves proportionally with the elevation angle of the tracked satellite, the measurement noise covariance matrix was made dependent on this elevation angle, which improved the reliability of the statistics of the measurement noises.

In order to test the sensitivity of the proposed filter setting to the change in ionospheric conditions, the algorithm was tested on a simulated real-time setting using two different data streams that were collected by the Swarm mission during the peak and the nadir of the 24<sup>th</sup> Solar Cycle. Thanks to the adopted ionospheric model, the filter could produce almost indistinguishable relative position and velocity estimates for both cases. For the extreme ionospheric conditions during the peak of the solar cycle (in 2014), the filter could achieve 26 cm and 4 cm/s RMS errors in the estimated relative distance and relative speed respectively, while it could achieve 24 cm and 4 cm/s RMS errors during the relaxed ionospheric conditions (in 2020).

It is important to emphasize that the obtained results could be achieved without implementing integer ambiguity resolution, which was excluded to relax the computational demand from the satellites onboard computer. Indeed, excluding the integer ambiguity resolution together with the adoption of an adequate ionospheric model contributed to the very fast convergence of the filter which typically converges after one iteration.



# Guidance strategies for multi-satellite formation reconfiguration

## 4.1 Introduction

As the satellites gradually became smaller in size since the beginning of the new space paradigm, it was no wonder that [Electric Propulsion \(EP\)](#) systems were increasingly used to support their maneuverability in orbit. At the cost of lower thrust levels, electric thrusters typically offer higher specific impulse, and hence better fuel efficiency than their chemical counterparts [72]. The low fuel and propellant storage requirements is mainly what makes [EP](#) an attractive option for small satellites [73]. Often motivated by the need to reduce weight, complexity, and hence cost, some satellite manufacturers started opting to incorporate a single electric thruster onboard their spacecraft. Examples of such satellites include the PLATiNO platform [74] and Triton-X [75]. A satellite that employs a single electric thruster is, in nature, under-actuated, since a desired thrust vector is only achievable after an attitude maneuver which redirects the thruster nozzle into the desired direction. In this chapter, we develop guidance strategies for the problem of reconfiguring multiple of such under-actuated satellites flying in a formation.

#### 4.1.1 Problem statement

This chapter introduces Delta-V-optimal/fuel-optimal, safe, relative orbit trajectory optimization strategies for multi-satellite formation reconfiguration in centralized and distributed settings. The following points are the main considerations for guidance schemes developed in this chapter:

- The formations under consideration comprise an uncontrolled chief spacecraft flying with an arbitrary number,  $N$ , of deputy satellites;
- Each deputy is equipped with a single un-gimbaled electric thrust nozzle;
- In the centralized setting, the chief spacecraft is the satellite onboard which the formation trajectory optimization calculations take place. The optimized trajectories are then transmitted to each of the deputies via an inter-satellite link. An exception to this rule is when the formation is composed of only one deputy flying with a real/virtual chief. In this case, the calculations are performed onboard the deputy, and the precise orbit of the chief is constantly transmitted to the deputy at each ground contact. No inter-satellite link is required in such scenario;
- In the distributed approach, the computation of optimized relative trajectories is shared among all deputies, with each deputy independently optimizing its own trajectory. To ensure coordination, an inter-satellite link is established between all the neighboring deputies. Additionally, the chief spacecraft is treated as a virtual point in this approach, as the requirement for a central processing unit is eliminated;
- The main purpose of the trajectory optimization scheme is to drive the deputies from an initial configuration at the initial time to a final desired configuration at the final time;
- To enhance the predictability of the maneuver, the initial time, the final time, as well as the time steps at which the thrust magnitude changes, are all treated as user inputs;

- During the formation reconfiguration maneuver, the maximum and minimum thrust bounds of each onboard thruster must never be violated;
- Inter-deputy collision must be avoided through keeping each satellite outside the [Keep-Out Zone \(KOZ\)](#) of all its neighboring satellites at each time step the trajectory optimizer considers;
- Deputy-chief collision must be avoided in cases when the chief is treated as a physical spacecraft, namely when centralized schemes are used to optimize the reconfiguration of a multi-satellite formation.

#### 4.1.2 State of the art

The problem of formation reconfiguration has been a subject of extensive research. While guidance and control strategies have been proposed for formations that leverage impulsive thrusters [28, 76, 77, 78], they are not applicable to formations that employ electric ones. Existing low-thrust centralized and distributed guidance schemes assume omnidirectional thrusting [79, 80, 81, 82], rendering them inapplicable to single-thruster satellite formations. Guidance policies were developed for the AVANTI mission [77, 83, 84], which incorporates a satellite equipped with one impulsive thruster flying in a formation with a non-cooperative object, yet again, these guidance methods are not suitable for low-thrust formations. On a related note, a satellite incorporating a single impulsive thruster cannot be considered under-actuated, unlike a spacecraft with a sole low-thruster. An impulsive thruster is idle for the most part of the maneuver due to the fact that it is able to provide large Delta-V changes, typically in a matter of seconds. A thruster-idle period is ample for an attitude redirection maneuver to take place before the following thruster firing. To address the problem of relative orbit corrections for mono-electric-thruster satellites, trajectory optimization methods which are used within [Model-Predictive Control \(MPC\)](#) schemes have been proposed in [12, 85], yet operational constraints, such as the necessary thruster-off-periods during ground contact, during payload operations, or during eclipse, were not considered in these works.

### 4.1.3 Methodology and contribution

In this chapter, the Delta-V-optimal, collision-free, formation reconfiguration problem is considered for an arbitrary number,  $N$ , of deputy satellites flying in a formation with an uncontrolled chief. To allow for such reconfiguration to take place, each deputy is equipped with one electric thruster. The proposed trajectory optimization procedures are open-loop control strategies which can be incorporated in the closed feedback loop in many ways, such as the shrinking-horizon MPC and the fixed-horizon MPC that will be discussed later in Chapter 5.

The first attempt to formulate the guidance problem is presented as a non-convex Quadratically Constrained Quadratic Programming (QCQP) problem, which is later relaxed to a convex QCQP that can be solved using Sequential Convex Programming (SCP). A relaxation of the quadratic constraints of the QCQP problem could cast it into a Quadratic Programming (QP) problem, which is a simpler form of QCQPs. Due to the fact that the optimal solutions to the QCQP formulation, as well as that of the QP problem, require unnecessarily large Delta-V changes, the convex QCQP problem is reformulated into a Second-Order Cone Programming (SOCP) problem through adopting a new cost function which makes better use of the available fuel. A final relaxation is applied to the SOCP formulation to transform it into the simplest form of convex programming problems; Linear Programming (LP). The dimensional Relative Orbital Elements (ROE) formulation, which is introduced in Chapter 2, is utilized to develop all the guidance schemes. It is to be emphasized that while these guidance formulations were developed specifically for the case of deputies with unidirectional low-thrust capabilities, it can be also adapted, with minor changes, for formations with satellites that are able to provide omnidirectional low-thrust.

The four convex formulations of the guidance strategy were initially developed without the minimum thrust constraint, which is a limitation imposed by the onboard thruster's specifications. This constraint was omitted in the initial formulations due to its non-convex nature, which presents significant challenges for convexification. In this chapter, an affine convexification approach is introduced to integrate the minimum thrust constraint into the set of

constraints for the four guidance schemes. Notably, the convexified problem becomes inherently more restrictive than it would be if the minimum thrust constraint were not convexified. This restrictiveness can lead to situations where the convexified problem is deemed infeasible for certain reconfiguration scenarios, even though the original non-convex problem remains feasible. To increase the likelihood of obtaining a feasible solution from the convexified problem, a *pruning* algorithm is introduced. This algorithm first solves the convex problem without enforcing the minimum thrust constraint. It then examines the control profiles for each deputy satellite, identifying instances where the required thrust is minimal. These instances are subsequently constrained to zero, and the problem is re-solved with these pruned thrust instances fixed at zero. Through this iterative approach, the thrust effort is redistributed more effectively, thereby improving compliance with the minimum thrust requirement and enhancing the feasibility of the solution. Finally, the problem is solved once more time to incorporate the minimum thrust constraint, with the pruned steps remaining constrained to zero during this final stage. This ensures that the solution aligns with the minimum thrust requirement while leveraging the improvements achieved through the pruning algorithm.

Since the guidance algorithms developed in this chapter are optimization problems that incorporate hard constraints, they are susceptible to infeasibility, especially when later incorporated into MPC schemes where the guidance problem is solved recurrently under varying initial and final reconfiguration conditions. To mitigate the risk of infeasibility during the operation of the autonomous closed control loop, constraints that are likely to lead to infeasibility are identified and softened. This softening process allows specific constraints to be violated within small, controlled margins, thereby enhancing the feasibility of the solution. The softening approach involves introducing slack variables to the problem, effectively increasing the number of variables but enhancing the reliability of the overall control loop by addressing potential infeasibility directly within the guidance layer. A full section is dedicated to exploring how constraints can be systematically softened to maintain feasibility while keeping deviations minimal.

Finally, the softened guidance problem is formulated as a distributed trajectory optimization problem to enhance the scalability of the proposed methods to formations involving large numbers of deputy satellites. In the distributed setting, each deputy calculates its own guidance profile onboard, while constant communication among deputies ensures inter-deputy collision safety. Although the problem is typically solved simultaneously for all satellites, certain situations require serial calculations to ensure proper coordination and prevent potential collisions.

The main contributions of this chapter are as follows:

- Presentation of the [QP](#) problem formulation, which serves as a foundation for a simpler and more efficient formulation: the [LP](#) formulation;
- Development of the mathematical formulations for the [SOCP](#) and [LP](#) problems, which yield more Delta-V efficient solutions compared to the classical [QCQP](#) problems, while also being easier to implement and faster to solve with most solvers;
- Execution of a benchmark experiment across over a dozen solvers, comparing the performance of the convex [QCQP](#), convex [QP](#), [SOCP](#), and [LP](#) formulations;
- Development of guidelines for embedding the guidance algorithms in space-borne applications, tailored to address typical reconfiguration challenges in remote sensing missions;
- Introduction of the minimum thrust constraint via an affine convexification method, along with the associated pruning algorithm to improve feasibility;
- Implementation of methods for softening hard-constrained problems to decrease the likelihood of encountering infeasibility during closed-loop control system operations;
- Investigation of formulating the guidance problem as a distributed trajectory optimization problem, along with the development of coordination algorithms necessary to guarantee collision safety throughout the reconfiguration maneuver.

#### 4.1.4 Chapter outline

The organization of this chapter is as follows: the next section introduces the four foundational guidance formulations that underpin all subsequent developments. In Section 4.3, strategies to mitigate numerical instability during the solution process are discussed, emphasizing the use of classical normalization techniques prior to solving the guidance problems. Following this, Section 4.4 presents a sensitivity analysis of the guidance strategies, focusing on two key user-defined parameters. This sensitivity analysis serves as the basis for selecting the parameters involved for the remainder of the thesis.

Section 4.5 presents and discusses the results of applying the four proposed trajectory optimization techniques across four reconfiguration scenarios. These scenarios involve typical configurations in multi-satellite remote sensing missions. This section benchmarks the four techniques, comparing their performance in terms of Delta-V and the computational time required by each method, using 15 different solvers. The results from this benchmark experiment serve as the basis for practical recommendations on which formulation to use under various conditions.

The convex relaxation of the minimum thrust constraint and its associated pruning algorithm are introduced in Section 4.6. The softening strategies and their impact are then examined in detail in Section 4.7, addressing how they enhance the feasibility of solutions under stringent conditions.

The potential to distribute the computational effort of the guidance problem among the deputies is explored in Section 4.8, where the corresponding coordination algorithms required for the distributed formulation are also discussed. The chapter concludes with a summary in Section 4.9.

## 4.2 Guidance formulations

In this section, a guidance scheme is developed for a formation that comprises an arbitrary number,  $N$ , of deputy satellites together with a chief spacecraft. The ultimate goal of the guidance module is to drive the state of each deputy from its initial values, parameterized by the dimensional ROE vectors,  $\mathbf{y}_{i,0}$  at the initial time,  $t_0$ , to the user-defined reference state,  $\bar{\mathbf{y}}_{i,f}$  at the final time,  $t_f$ . Again, the  $N$  deputies are assumed to be each equipped with a single throttleable electric thruster dedicated for (relative) orbit reconfiguration, while the chief is uncontrolled. In this under-actuated setting, each deputy has to undergo repeating attitude maneuvers in order to redirect its thruster in the desired direction before every thruster firing. The time required for these redirection maneuvers was taken into consideration in the design phase of the guidance procedure, and the thruster of each deputy is thus dictated to operate on an alternating on/off mode, where the slew maneuvers are allocated in the thruster off-periods. On top of the natural relative orbit changes during the thrusters off-periods, further forced ROE corrections are simultaneously induced for all the deputies through  $\frac{m+1}{2}$  finite burns, with  $m$  being an odd number. Figure 4.1 illustrates the alternation between the on and off states of the electric thruster onboard one of the deputies which shall drive its state from the initial to the reference value.

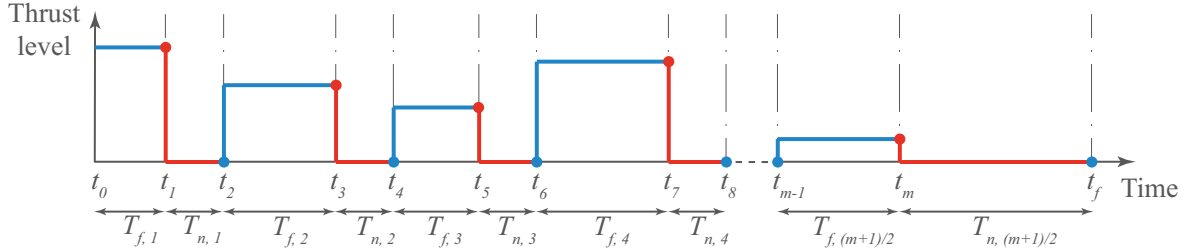


Figure 4.1: Graphical representation of the low-thrust guidance strategy

It is important to emphasize that the time instances at which the thruster alternates between the on and off states,  $\mathbf{t} = [t_0 \ t_1 \ \dots \ t_f]$  (see Fig. 4.1), are assumed to be identical for all the deputies. It is, in fact, a common practice to unify the discretization steps for all the deputies in centralized trajectory optimization schemes [79, 81]. The common time vector is left for the mission operator to determine, in order to enhance the mission predictability,



which is a very important factor in the operation of real missions. It appears that leaving  $\mathbf{t}$  as a user-input not only helps predict the behavior of the formation ahead of time, but also helps accommodate any operational constraints within the thrusters idle intervals. Operational constraints in our context might include not being able to use the thruster or the attitude control system for relative orbit correction during specific times, e.g., during ground contact, payload operations, or during times when the satellite is in the shadow. The time vector,  $\mathbf{t}$ , can be defined either through the time instances at which the thrusters switch between the on and off states, or through defining the initial time,  $t_0$ , and the periods during which the thrusters are turned on or off. As a result, by letting  $\mathcal{L} = \{1, 2, \dots, (m+1)/2\}$ , these periods are divided into two categories;

- The forced motion time periods,  $T_{f,l}$ ,  $l \in \mathcal{L}$ ;
- The natural motion periods (coast arcs),  $T_{n,l}$ ,  $l \in \mathcal{L}$ .

Since the attitude redirection maneuvers are allocated during the natural motion periods, a lower bound for these coast arcs has to be imposed which is related to the maximum allowable angular speed, i.e., angular velocity's Euclidean norm, of the deputies, such that,

$$T_{n,l} \geq \frac{\pi}{\omega_{\max}} + T_{\text{safety}} \quad \forall l \in \mathcal{L}, \quad (4.1)$$

where  $\omega_{\max} = \min(\omega_{i,\max})$  is the maximum angular speed of the deputy with slowest angular rate, with  $\omega_{i,\max}$  being the maximum angular speed of the  $i^{\text{th}}$  deputy, and  $T_{\text{safety}} \geq 0$  is added to ensure that the coast arc can accommodate the most stringent attitude maneuver. The rationale behind Eq. (4.1) is that, since the maximum possible slew angle is  $\pi$ , the longest time period it takes the satellite to perform any slew, using the maximum angular speed, is  $\frac{\pi}{\omega_{\max}}$ . Indeed, the satellite never uses the maximum angular speed throughout the whole slew, that is why the  $T_{\text{safety}}$  term is added to make sure that the allocated time for the slew maneuver,  $T_{n,l}$ , is always more than enough.

Let,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{1,0} & \mathbf{y}_{2,0} & \dots & \mathbf{y}_{N,0} \\ \mathbf{y}_{1,1} & \mathbf{y}_{2,1} & \dots & \mathbf{y}_{N,1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_{1,m+1} & \mathbf{y}_{2,m+1} & \dots & \mathbf{y}_{N,m+1} \end{bmatrix}, \quad (4.2)$$

$$\bar{\mathbf{U}} = \begin{bmatrix} \bar{\mathbf{u}}_{1,0} & \bar{\mathbf{u}}_{2,0} & \dots & \bar{\mathbf{u}}_{N,0} \\ \bar{\mathbf{u}}_{1,1} & \bar{\mathbf{u}}_{2,1} & \dots & \bar{\mathbf{u}}_{N,1} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{u}}_{1,m} & \bar{\mathbf{u}}_{2,m} & \dots & \bar{\mathbf{u}}_{N,m} \end{bmatrix}, \quad (4.3)$$

where  $\mathbf{y}_{i,k}$  is the dimensional ROE vector of the  $i^{\text{th}}$  deputy at time  $t_k$ , and  $\bar{\mathbf{u}}_{i,k} = a_c \mathbf{u}_{i,k}$  is the scaled control vector, with  $a_c$  being the semi-major axis of the chief's orbit, and  $\mathbf{u}_{i,k}$  being the control acceleration vector, expressed in the RTN frame, provided by the  $i^{\text{th}}$  deputy thruster between the two time instances  $t_k$  and  $t_{k+1}$ . Note that in this chapter, and in order to simplify the notation, the superscript indicating the frame on which a vector is expressed is dropped. The guidance problem can thus be formally written as an optimization problem with a quadratic objective function, as is classically done [86], as follows:

**Problem 4.1** (Non-convex formulation).

$$\min_{\mathbf{Y}, \bar{\mathbf{U}}} \quad \frac{1}{a_c^2} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f} \left( \Delta t_k^2 \bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \right)$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.4)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.5)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_n, \quad (4.6)$$

$$\bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \leq a_c^2 u_{i,max}^2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.7)$$

$$(\mathbf{y}_{i,k} - \mathbf{y}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA}^2 \quad \forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K}, \quad (4.8)$$

$$\mathbf{y}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA}^2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.9)$$

where  $\Delta t_k = t_{k+1} - t_k$ ,  $\Phi_k \equiv \Phi_{\delta\alpha}(t_k, t_{k+1})$  (refer to Eq. (2.51)), and  $\Psi_k \equiv \Psi_{\delta\alpha}(t_k, t_{k+1})$  (refer to Eq. (2.52)). Moreover,  $\mathcal{I} = \{1, 2, \dots, N\}$  is the set of deputies' indices,  $\mathcal{K} = \mathcal{K}_f \cup \mathcal{K}_n$ , with  $\mathcal{K}_f = \{0, 2, 4, \dots, m-1\}$  being the set of time indices where the thruster of each deputy is turned on, i.e., forced motion periods, and  $\mathcal{K}_n = \{1, 3, 5, \dots, m\}$  being the set of time indices where the thruster of each deputy is turned off, i.e., natural motion periods,  $u_{i,\max}$  is the maximum allowable acceleration by the onboard thruster of the  $i^{\text{th}}$  deputy. Generally, it is the maximum allowable thrust that is provided in the data sheets of electric thrusters, and not the maximum acceleration. However, the latter can be calculated from the former according to:  $u_{i,\max} = f_{i,\max}/M_i$ , with  $f_{i,\max}$  being the maximum thrust of the  $i^{\text{th}}$  deputy, and  $M_i$  being its mass, which is assumed constant throughout the maneuver, and is set to the mass of the satellite at  $t_0$ . Furthermore, in Problem 4.1,  $R_{\text{CA}} \geq 0$  is the radius of the collision avoidance sphere, i.e., **KOZ**, and  $\mathbf{T}_k \equiv \mathbf{T}(t_k - t_{\text{ref}})$  is the matrix that transforms a dimensional **ROE** vector into its corresponding position vector in the **RTN** frame, introduced in Eq. (2.57), with  $t_{\text{ref}}$  being the reference time, typically taken as the start time of the simulation. It is important to note that  $\bar{\mathbf{u}}_{i,k}$  are chosen to be included as optimization variables instead of  $\mathbf{u}_{i,k}$  in order to ensure that all the decision variables, i.e.,  $\mathbf{Y}$  and  $\bar{\mathbf{U}}$ , are of comparable orders of magnitude, which makes it less probable that a solver will run into numerical issues. Poorly-scaled problems typically require longer times to solve, if a solution can be found in the first place.

The constraints imposed on Problem 4.1 are summarized as follows:

- Equation (4.4) represents the boundary constraints which dictates the guidance profile to have the final state exactly equal to the set points defined by the user, while respecting the initial state of each deputy;
- The relative orbital dynamics are imposed on the trajectory optimizer through Eq. (4.5);
- The acceleration constraints are forced through equations (4.6) and (4.7). Note that Eq. (4.6) is a hard constraint to ensure that the input acceleration onboard each deputy provided during attitude redirection maneuvers is exactly zero;

- Inter-deputy collision is avoided by imposing Eq. (4.8), which guarantees that no deputy enters the KOZ of another, while deputy-chief collision is prohibited at each time step through Eq. (4.9). Since the time steps of our application are relatively small when considering how slow the relative orbital dynamics can be, inter-step collision avoidance is ignored in this study.

It is clear that the cost function of Problem 4.1 is the sum of the squared L2 norms of all the control Delta-V across all deputies and across all time instances during the reconfiguration maneuver. Minimizing this cost function not only results in a Delta-V-optimal solution, but also produces a fuel-optimal solution. Indeed, the terms "Delta-V-optimal" and "fuel-optimal" can be used interchangeably in our case, since they really do refer to the same thing when the controlled spacecraft are each equipped with a single thruster [9].

Problem 4.1 is a non-convex optimization problem which requires a series of elaborate processes in order to find its globally optimal solution. The non-convexity of Problem 4.1 arises solely from the non-convexity of the two last constraints. In fact, if it were not for the constraints in Eqs. (4.8) and (4.9), Problem 4.1 would have been a convex optimization problem that is guaranteed to produce a globally optimal solution. One way to convexify the problem is to approximate the non-convex constraints by affine ones, then solve the problem through Sequential Convex Programming (SCP). Using the relaxation proposed in [81] for the collision avoidance constraints, Problem 4.1 can be rewritten in its relaxed convex Quadratically Constrained Quadratic Programming (QCQP) form as follows where the modifications to the previous problem are marked in red:

**Problem 4.2** (Convex QCQP formulation).

$$\min_{\mathbf{Y}, \bar{\mathbf{U}}} \quad \frac{1}{a_c^2} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f} \left( \Delta t_k^2 \bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \right)$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.10)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{K}, \quad (4.11)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_n, \quad (4.12)$$

$$\bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \leq a_c^2 u_{i,max}^2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.13)$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA} \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 \quad \forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K}, \quad (4.14)$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA} \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.15)$$

where  $\check{\mathbf{y}}_{i,k}$  is a predicted value for the dimensional ROE vector of the  $i^{\text{th}}$  deputy at time  $t_k$  and  $\|\cdot\|_2$  is the  $L_2$  norm of a vector. The solution to Problem 4.2 can be obtained sequentially through SCP, where  $\check{\mathbf{y}}_{i,k}$  for the current iteration is set to  $\mathbf{y}_{i,k}$  from the previous iteration. For the first iteration,  $\check{\mathbf{y}}_{i,k}$  can be obtained in a variety of ways, e.g., imposing the dynamics solution, Eq. (2.50), from the initial to the final times with no control inputs, or alternatively solving the problem without the collision avoidance constraints first and extracting  $\mathbf{y}_{i,k}$  from the solution, then setting  $\check{\mathbf{y}}_{i,k} = \mathbf{y}_{i,k}$ . The SCP scheme is set to terminate when one of the following criteria is satisfied,

- $\|\check{\mathbf{y}}_{i,k} - \mathbf{y}_{i,k}\|_2 \leq \epsilon$  at the current iteration for all satellites at every time step, with  $\epsilon > 0$  being a user-defined threshold;
- The guidance profile of the current iteration is collision-free;
- The user-defined maximum number of iteration is reached, in which case, the solution trajectory is not guaranteed to be collision-free.

A schematic illustrating the adopted flow of the SCP scheme in this chapter is presented in Fig. 4.2.

Indeed, Problem 4.2 stands as a Convex Optimization Problem (COP) since it fulfills all the necessary conditions for a problem to be one [87], which namely are,

- Cost function must be convex (for minimization problems);
- Inequality constraints must be convex;
- Equality constraints must be affine.

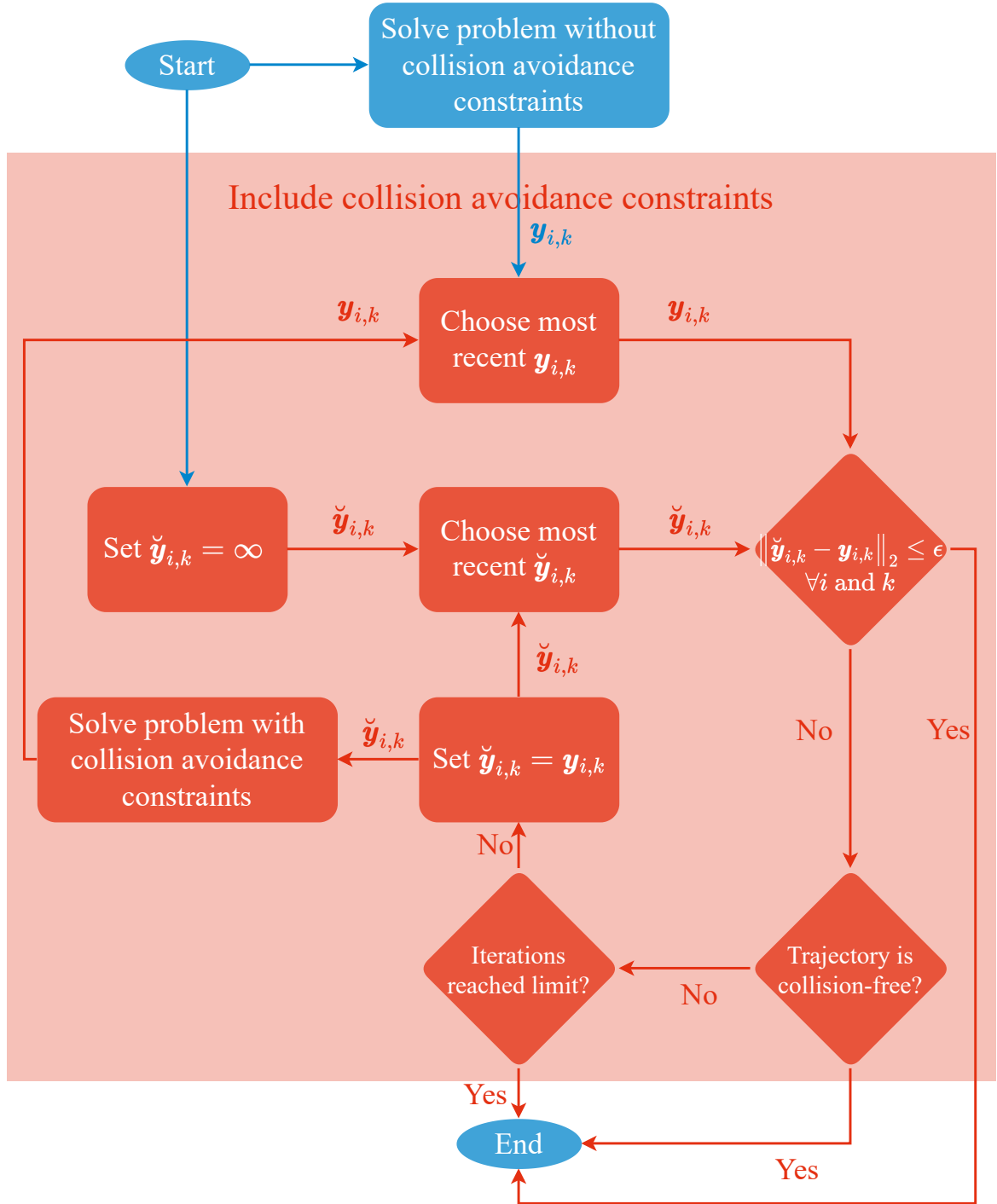


Figure 4.2: Graphical representation of the SCP scheme

Being a COP, Problem 4.2 is guaranteed to have a globally-optimal solution, however, it would be much more efficient to solve if it could be put in one of the standard classes of convex optimization problems, e.g. Quadratic Programming (QP) and LP, since dedicated solvers with their dedicated algorithms for these classes have matured over the past decades. It is clear that the only thing which prevents Problem 4.2 from being put in the convex QP canonical form is Eq. (4.13) since it is a quadratic constraint and not an affine one. It can be, nonetheless, transformed into multiple affine constraints using the methodology proposed in [88] which suggests that,

$$\|\mathbf{b}\|_2 \leq c, \quad \mathbf{b} \in \mathbb{R}^2$$

can be relaxed by: (4.16)

$$\begin{bmatrix} \cos(\gamma_d) & \sin(\gamma_d) \end{bmatrix} \mathbf{b} \leq c \cos(\gamma_{\max}) \quad \forall d \in \mathcal{D},$$

where  $\mathcal{D} = \{1, 2, \dots, n_{\text{dir}}\}$ , with  $n_{\text{dir}} \geq 3$  being the number of affine inequality constraints that approximate the Euclidean norm constraint,  $\gamma_d = \frac{2(d-1)\pi}{n_{\text{dir}}} + \gamma_{\text{first}}$ , with  $\gamma_{\text{first}}$  being the angle corresponding to the first direction, and  $\gamma_{\max} = \frac{\pi}{n_{\text{dir}}}$ . It is important to emphasize that the constraint relaxation, Eq. (4.16), is only applicable to vectors in  $\mathbb{R}^2$ , while the norm constraint in Problem 4.2, Eq. (4.13), is imposed on a vector in  $\mathbb{R}^3$ ,  $\bar{\mathbf{u}}_{i,k}$ . The relaxation is, therefore, applied to the projection of the constraining sphere on each plane individually. The ROE dynamics (refer to Chapter 2) possess unique characteristics that necessitates an accurate approximation of the constraining circle lying in the Transversal-Normal (T-N) plane, while coarse approximations of the constraints on the two other planes are acceptable. Concretely, the optimal solution to Problem 4.2 is expected to rarely incorporate radial thrust [77, 85] since it is known to be more expensive, from the Delta-V point of view, than relying solely on transversal thrust, especially when the reconfiguration can afford long maneuver times for large in-plane maneuvers. It is for this reason that the relaxation in Eq. (4.16) is applied in the T-N plane with a larger number of directions,  $n_{\text{dir}}$ , than the number of directions used in the Radial-Transversal (R-T) and the Radial-Normal (R-N) planes. Namely, we introduce  $\bar{n}_{\text{dir}} = 4 < n_{\text{dir}}$  as the number of directions that approximate the constraining circles lying

in the R-T and the R-N planes, with  $\bar{\gamma}_d = \frac{2(d-1)\pi}{\bar{n}_{\text{dir}}} + \bar{\gamma}_{\text{first}} \forall j \in \bar{\mathcal{D}} = \{1, \dots, \bar{n}_{\text{dir}}\}$ , and with  $\bar{\gamma}_{\text{max}} = \bar{\gamma}_{\text{first}} = \frac{\pi}{4}$ . In this setting, the constraints in the R-T and the R-N planes are approximated by two rhombuses which cover around 64% of the original constraining circles.

Having introduced the constraint relaxations which transform the quadratic constraints into multiple affine ones, the reformulation of Problem 4.2 as a QP problem can be written as,

**Problem 4.3** (QP formulation).

$$\min_{\mathbf{Y}, \bar{\mathbf{U}}} \frac{1}{a_c^2} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f} \left( \Delta t_k^2 \bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \right)$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.17)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.18)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_n, \quad (4.19)$$

$$\begin{bmatrix} 0 & \cos(\gamma_d) & \sin(\gamma_d) \end{bmatrix} \bar{\mathbf{u}}_{i,k} \leq a_c u_{i,\text{max}} \cos(\gamma_{\text{max}}) \quad \forall d \in \mathcal{D}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.20)$$

$$\begin{bmatrix} \cos(\bar{\gamma}_d) & \sin(\bar{\gamma}_d) & 0 \\ \cos(\bar{\gamma}_d) & 0 & \sin(\bar{\gamma}_d) \end{bmatrix} \bar{\mathbf{u}}_{i,k} \leq a_c u_{i,\text{max}} \cos(\bar{\gamma}_{\text{max}}) \quad \forall d \in \bar{\mathcal{D}}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.21)$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA} \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 \quad \forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K}, \quad (4.22)$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA} \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.23)$$

A graphical representation of the feasibility regions of the control acceleration,  $\mathbf{u}_{i,k} = [u_{i,R}, u_{i,T}, u_{i,N}]^\top$ , is given in Fig. 4.3 for both problems, Problem 4.2 and Problem 4.3. In Fig. 4.3, the constraint relaxation in Eq. (4.16) is depicted for  $n_{\text{dir}} = 12$  and  $\gamma_{\text{first}} = \frac{\pi}{n_{\text{dir}}}$  which covers approximately 95.5% of the original constraining circle in the T-N plane. It is important to note that while the relaxed Eq. (4.21) does not explicitly constrain the radial thrust component any better than the original circular constraint, it does affect the choice of



the radial-transversal and normal-radial combinations.

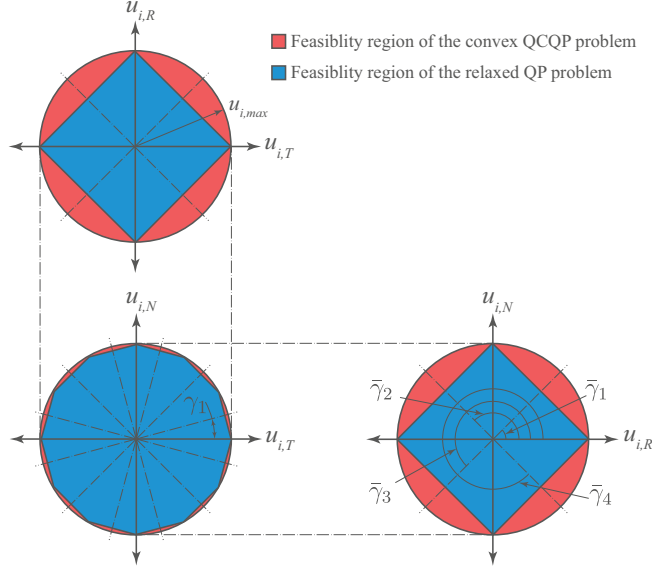


Figure 4.3: Feasibility region comparison between the convex QCQP formulation (Problem 4.2) and the QP formulation (Problem 4.3)

Since the objective function of Problem 4.3, and that of Problems 4.1 and 4.2 for that matter, aggregates the squared L2 norms of the Delta-V vectors, across deputies and across time instances, it comes as no surprise that the solver gives a stronger emphasis on larger values of Delta-V. While a quadratic objective function suppresses the control peaks, and hence enhances the smoothness of the optimal state trajectory [89], it makes Problems 4.1, 4.2, and 4.3 close to being minimization problems for the maximum Delta-V instance. An alternative approach is to set the cost function to the sum of the second-norms of the Delta-V vectors, rather than the sum of the squared norms. Adopting this approach, then transforming the problem into its separable epigraph form [90], which is a very close form to the epigraph problem form [87], the new problem can be written as a Second-Order Cone Programming (SOCP) problem as follows:

**Problem 4.4** (SOCP formulation).

$$\min_{\mathbf{Y}, \mathbf{U}, \mathbf{\Gamma}} \quad \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f} (\Delta t_k \Gamma_{i,k})$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.24)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.25)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_n, \quad (4.26)$$

$$\|\bar{\mathbf{u}}_{i,k}\|_2 \leq \Gamma_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.27)$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.28)$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA} \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 \quad \forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K}, \quad (4.29)$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA} \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.30)$$

where  $\mathbf{\Gamma}$  is a matrix collates all the auxiliary variables that had to be introduced in order to put the problem into the separable epigraph form. The matrix  $\mathbf{\Gamma}$  contains one entry corresponding to every constraint in Eq. (4.27). Formally,

$$\mathbf{\Gamma} = \begin{bmatrix} \Gamma_{1,0} & \Gamma_{2,0} & \dots & \Gamma_{N,0} \\ \Gamma_{1,2} & \Gamma_{2,2} & \dots & \Gamma_{N,2} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{1,m-1} & \Gamma_{2,m-1} & \dots & \Gamma_{N,m-1} \end{bmatrix}. \quad (4.31)$$

It is important to note that **SOCP** problems can be handled in their native form, i.e., without having to transform the **Second-Order Cone (SOC)** constraints into quadratic ones, only by a handful of solvers, e.g., SCS, ECOS, and MOSEK. Many of the solvers require the **SOC** constraints to be transformed into quadratic ones, which, if applied to Eq. (4.27), renders it non-convex, unless an additional linear constraint is added, namely,  $\Gamma_{i,k} \geq 0 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f$ . By adding this additional constraint, many solvers, e.g., Gurobi and Knitro, recognize the transformed quadratic constraint as a second order cone, and treat it as such.

Tracking the behavior of the Second-Order Cone (SOC) constraint, Eq. (4.27), is quite

interesting since it behaves such that,

$$\|\bar{\mathbf{u}}_{i,k}^*\|_2 = \Gamma_{i,k}^* \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.32)$$

with  $(\cdot)^*$  being the optimal solution to the problem, due to the fact that Problem 4.4 is pushing  $\Gamma_{i,k}$  to be as low as possible (refer to the cost function of the problem), and since the lowest possible value for  $\Gamma_{i,k}$  is  $\|\bar{\mathbf{u}}_{i,k}\|_2$  according to Eq. (4.27).

Problem 4.4 can be taken a step further and be formulated as a **Linear Programming (LP)** problem, which makes it possible for the new formulation to be solved by the vast majority of the numerical optimization solvers. This transformation can be applied through implementing the same piece-wise linearization in equations (4.20) and (4.21), which approximates the **SOC** constraint (sphere) in Eq. (4.27) by multiple affine ones (polyhedron). Transforming Problem 4.4 into an **LP** one not only enables the possibility to use many solvers that cannot handle the **SOCP** form, but may also require less time to solve than that of the **SOCP** formulation, although not necessarily. A less solve time can be anticipated for the **LP** formulation due to the fact that **LP** is presumably the simplest form of a convex optimization problem, and also due to the existence of dedicated **LP** algorithms, e.g., primal and dual simplex, which have matured over the last few decades. However, transforming the **SOCP** formulation into an **LP** one, comes at the cost of adding new constraints to the problem.

The reformulation of the **SOCP** problem into an **LP** one can be written as follows:

**Problem 4.5** (LP formulation).

$$\min_{\mathbf{Y}, \bar{\mathbf{U}}, \mathbf{\Gamma}} \quad \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f} (\Delta t_k \Gamma_{i,k})$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.33)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.34)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_n, \quad (4.35)$$

$$\begin{bmatrix} 0 & \cos(\gamma_d) & \sin(\gamma_d) \end{bmatrix} \bar{\mathbf{u}}_{i,k} \leq \Gamma_{i,k} \cos(\gamma_{max}) \quad \forall d \in \mathcal{D}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.36)$$

$$\begin{bmatrix} \cos(\bar{\gamma}_d) & \sin(\bar{\gamma}_d) & 0 \\ \cos(\bar{\gamma}_d) & 0 & \sin(\bar{\gamma}_d) \end{bmatrix} \bar{\mathbf{u}}_{i,k} \leq \Gamma_{i,k} \cos(\bar{\gamma}_{max}) \quad \forall d \in \bar{\mathcal{D}}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.37)$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.38)$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA} \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 \quad \forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K}, \quad (4.39)$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA} \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \quad (4.40)$$

Similar to Problem 4.4, the optimal solution of Problem 4.5 lies at the boundaries of constraints (4.36) and (4.37). This implies that the second norm of the scaled input acceleration,  $\|\bar{\mathbf{u}}_{i,k}\|_2$ , lies on the surface of the polyhedron (depicted in Fig. 4.3 for  $\Gamma_{i,k} = a_c \bar{u}_{i,max}$ ) which relaxes the original feasibility sphere defined in Eq. (4.27).

It is to be emphasized that while the polyhedron that relaxes the acceleration norm constraint has projections on the T-N, the R-T, and the R-N planes that lie well inside the original feasibility sphere as seen in Fig. 4.3, some vertices of the polyhedron can be seen to slightly protrude from this sphere when visualized from different angles in the 3D space for the case of  $n_{dir} = 12$ , which can be seen clearly in Fig. 4.4. Although the relaxing polyhedron of Problem 4.5 can change in size depending on the value of  $\Gamma_{i,k}$ , and hence lie with its entirety within the original constraining sphere, the fact that it can extend beyond the limits of the sphere when  $\Gamma_{i,k} = a_c \bar{u}_{i,max}$  can be problematic. Interestingly, this almost never causes a problem for the QP formulation due to the fact that Problem 4.3 is very close to minimizing the maximum L2 norm of the input acceleration vectors. Moreover, the solution of Problem 4.3 almost never lies at the boundaries of the polyhedron, and even in the rare cases when it does, it typically does not reach the protruding regions.

As the solution of Problem 4.5 may include points where  $\Gamma_{i,k} = a_c \bar{u}_{i,max}$ , the event of  $\|\mathbf{u}_{i,k}\|_2$  getting slightly larger than  $u_{i,max}$  becomes a real possibility, especially since the

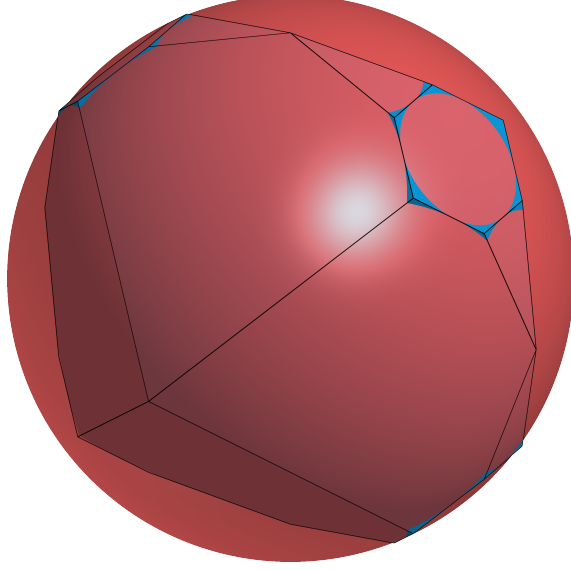


Figure 4.4: Control input feasibility regions of Problem 4.4 (red) and Problem 4.5 (blue)

solution has to lie on the boundaries of constraints (4.36) and (4.37) as established earlier. While the effect of this slight violation of the original constraints can be tolerated for short maneuvers, relative orbit correction maneuvers which require extended periods of time can deviate from their set points by tens or even hundreds of meters in the dimensional ROE space as will be discussed in Section 4.5. It is for this reason that constraints (4.36) and (4.37) are modified such that the approximating polyhedron is uniformly scaled down so that it can be contained entirely by the original sphere regardless of the value of  $\Gamma_{i,k}$ . The resulting problem after this scaling is written as,

**Problem 4.6** (LP formulation with scaled feasibility region).

$$\min_{\mathbf{Y}, \mathbf{U}, \mathbf{\Gamma}} \quad \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f} (\Delta t_k \Gamma_{i,k})$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.41)$$

$$\mathbf{y}_{i,k+1} = \mathbf{\Phi}_k \mathbf{y}_{i,k} + \mathbf{\Psi}_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{K}, \quad (4.42)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{K}_n, \quad (4.43)$$

$$\begin{bmatrix} 0 & \cos(\gamma_d) & \sin(\gamma_d) \end{bmatrix} \mathbf{c} \bar{\mathbf{u}}_{i,k} \leq \Gamma_{i,k} \cos(\gamma_{max}) \quad \forall d \in \mathcal{D}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.44)$$

$$\begin{bmatrix} \cos(\bar{\gamma}_d) & \sin(\bar{\gamma}_d) & 0 \\ \cos(\bar{\gamma}_d) & 0 & \sin(\bar{\gamma}_d) \end{bmatrix} \mathbf{c} \bar{\mathbf{u}}_{i,k} \leq \Gamma_{i,k} \cos(\bar{\gamma}_{max}) \quad \forall d \in \bar{\mathcal{D}}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.45)$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}_f, \quad (4.46)$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA} \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 \quad \forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K}, \quad (4.47)$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA} \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.48)$$

where  $c$  is a constant scaling factor that guarantees that the polyhedron of Fig. 4.4 is situated entirely within the sphere. The protrusion of the polyhedron beyond the constraining sphere is truly minimal, and could be quantified with the help of a 3D modeling software. In fact, the distance from the center of the sphere to the furthest vertex of the polyhedron is measured to be approximately 1.7% larger than the radius of the original sphere (for  $n_{\text{dir}} = 12$ ). Appendix C provides an overview of various methods for calculating the scaling factor corresponding to different values of  $n_{\text{dir}}$ , and presents the scaling factors obtained for each scenario. It has been demonstrated that  $n_{\text{dir}} = 12$  is one of the most effective choices, as it requires the smallest scaling factor among all the  $n_{\text{dir}}$  values that cover at least 95% of the circumcircle area in the T-N plane (refer to Appendix C). In the following discussions, the value of  $n_{\text{dir}}$  is set to 12 while the constant  $c$  is fixed to 1.017.

It is important to note that while mission operators usually favor closed-form solutions over numerical optimization-based ones due to the risk of running into an infeasible situation, infeasibility is never a problem in our context. A feasible solution can always be obtained when the maneuver is allowed enough time regardless of how hard the reconfiguration is, as long as the initial and final states are collision-free. Indeed, the maneuver duration is a user-input that can be planned and controlled by the ground operators.

### 4.3 Problem normalization

In most optimization problems, normalization of the cost function as well as the constraints improves numerical stability of the solver and increases its convergence velocity. This fact still remains true in the context of this thesis, even for the introduced guidance problems which have been formulated in such a way that the decision variables are within almost the same order of magnitude. Normalization helps avoid issues of numerical instability due to poorly-scaled matrices which might cause ill-conditioning. Problem formulations involving badly-scaled matrices often result in poor solver convergence or in high error in a solution due to arithmetic on floating-point numbers with a limited number of digits [91]. By normalizing the cost function and, especially, constraints, such discrepancies are reduced, making the formulation easier to work with.

The objective functions of the proposed problems, specifically Problems 4.2, 4.3, 4.4, and 4.6, are either linear or quadratic. Their constraints fall under three categories: quadratic, linear, and second-order cone constraints. This section provides a brief discussion on the normalization techniques adopted in this thesis, for both types of objective functions and the three categories of constraints. Indeed, more sophisticated scaling techniques can be used, however, the simplest normalization methods often prove to be the most effective [91].

#### 4.3.1 Quadratic objective functions

A quadratic cost function can be written in its general form as,

$$J_Q = \frac{1}{2} \mathbf{x}^\top \mathbf{Q}_o \mathbf{x} + \mathbf{f}_o^\top \mathbf{x}, \quad (4.49)$$

where  $\mathbf{x}$  is the vector that collates all the decision variables. This cost function is normalized by dividing both,  $\mathbf{Q}_o$  and  $\mathbf{f}_o$  by the Frobenius norm of  $\mathbf{Q}_o$ , which guarantees that all the values in  $\mathbf{Q}_o$  lie between  $-1$  and  $1$ .

### 4.3.2 Linear objective functions

A linear cost function can be written in its general form as,

$$J_L = \mathbf{f}_o^\top \mathbf{x}. \quad (4.50)$$

The linear cost function is normalized in this research by dividing  $\mathbf{f}_o$  by its  $L_\infty$  norm.

### 4.3.3 Quadratic constraints

A quadratic constraint can be written in its general form as,

$$\frac{1}{2} \mathbf{x}^\top \mathbf{Q}_c \mathbf{x} + \mathbf{f}_c^\top \mathbf{x} \leq b_u, \quad (4.51)$$

where the quadratic constraint is convex if and only if  $\mathbf{Q}_c$  is positive semi-definite, and where  $b_u$  is the upper bound of the constraint. Such constraint is normalized by dividing  $\mathbf{Q}_c$ ,  $\mathbf{f}_c$ , and  $b_u$  by the Frobenius norm of  $\mathbf{Q}_c$ .

### 4.3.4 Linear constraints

A collection of linear constraints can be written in the general form of,

$$\mathbf{b}_l \leq \mathbf{A}_c \mathbf{x} \leq \mathbf{b}_u, \quad (4.52)$$

where  $\mathbf{b}_l$  and  $\mathbf{b}_u$  are the vectors of lower and upper bounds, respectively. Equilibration is utilized to normalize these linear constraints, where each row of the matrix  $\mathbf{A}_c$  is divided by its  $L_\infty$  norm. The corresponding elements in the upper and lower bounds are divided by the same norm to keep the scaling non-effective in the constraints.

### 4.3.5 Second-order cone constraints

For a SOC constraint of the form,

$$\|\mathbf{A}_c \mathbf{x} - \mathbf{b}_c\|_2 \leq \mathbf{f}_c^\top \mathbf{x} - g_c, \quad (4.53)$$



the scaling is done by dividing  $\mathbf{A}_c$ ,  $\mathbf{b}_c$ ,  $\mathbf{f}_c$ , and  $g_c$  by the Frobenius norm of  $\mathbf{A}_c$ .

## 4.4 Parameters Sensitivity Analysis

Upon examining the proposed formulations of the guidance problems and their operational concepts in Fig. 4.1, it is evident that certain user-provided parameters may require tuning. These parameters include the time instances for thruster activation and deactivation, or alternatively, the time periods  $T_{f,l}$  and  $T_{n,l} \forall l \in \mathcal{L}$ , as well as the allocated maneuver time,  $t_f - t_0$ . The selection of the time periods  $T_{f,l}$  and  $T_{n,l}$  is typically constrained by mission requirements, such as avoiding thrust during eclipses, ground contact, or scientific experiments. In the absence of such constraints, it is natural to fix the time periods  $T_{f,l}$  and  $T_{n,l}$ , i.e.,  $T_{f,l} = T_f$ ,  $T_{n,l} = T_n \forall l \in \mathcal{L}$ . For this specific scenario, a sensitivity analysis was conducted to evaluate the feasibility of the optimization problem with varying values of  $T_f$  and  $T_n$ . In this analysis, Problems 4.2, 4.3, 4.4, and 4.6 were solved, excluding the collision avoidance constraints, for 100 randomly selected satellite numbers and initial conditions, i.e.,  $N$  and  $\mathbf{y}_{i,0}$ , across 70  $T_f$ - $T_n$  combinations drawn from  $T_n \in \{60, 90, \dots, 240\}$  s and  $T_f \in \{0.05, 0.1, 0.15, \dots, 0.5\}$  orbits. Furthermore, since the difference between the initial and final ROE vectors characterizes the maneuver, rather than the vector values themselves, only the initial values  $\mathbf{y}_{i,0}$  were chosen randomly, while the final values  $\bar{\mathbf{y}}_{i,f}$  were set to zero for the 7000 experiments. Specifically, the entries of the initial dimensional ROE vectors,  $\mathbf{y}_{i,0}$ , were randomly selected from the  $[-1 \ 1]$  km range, except for the initial relative mean argument of longitude,  $a_c \delta \lambda_{i,0}$ , which was randomly chosen from the  $[-100 \ 100]$  km range. The maneuver time was fixed at 20 orbits, and the number of deputies was randomly chosen from  $\{1, 2, \dots, 6\}$ . Additionally, the mean orbital elements vector of the chief at the beginning of the maneuver was fixed for all 7000 experiments to  $\boldsymbol{\alpha}_{c,0} = \begin{bmatrix} 6761.45 \text{ km} & 180 \text{ deg} & 1.5 \cdot 10^{-3} & 0 & 98 \text{ deg} & 0 \text{ deg} \end{bmatrix}^\top$  at  $t_0$ , while  $n_{\text{dir}}$  is fixed to 12. A histogram of the 100 randomly chosen number of deputies used in the 7000 guidance runs is presented in Fig. 4.5.

The output of the sensitivity analysis is the success and failure regions in the  $T_f$ - $T_n$

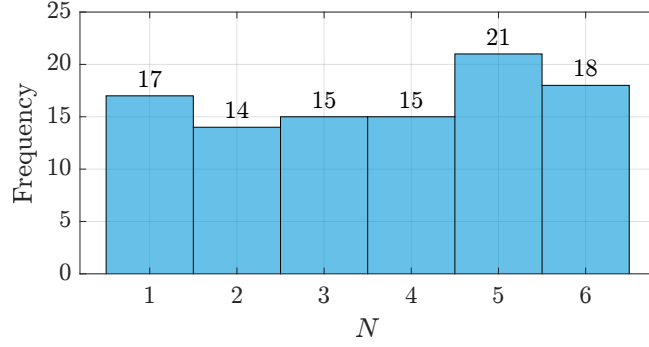


Figure 4.5: Formation sizes over the 100 randomly chosen formation scenarios

plane, where the success region is the one in which the optimizer succeeded to find a feasible solution for all the 100 random initial conditions and number of deputies. The success and failure regions of the aforementioned sensitivity study are depicted in Figures 4.6a, 4.7a, 4.8a, and 4.9a for the QCQP, the QP, the SOCP, and the LP formulations, respectively.

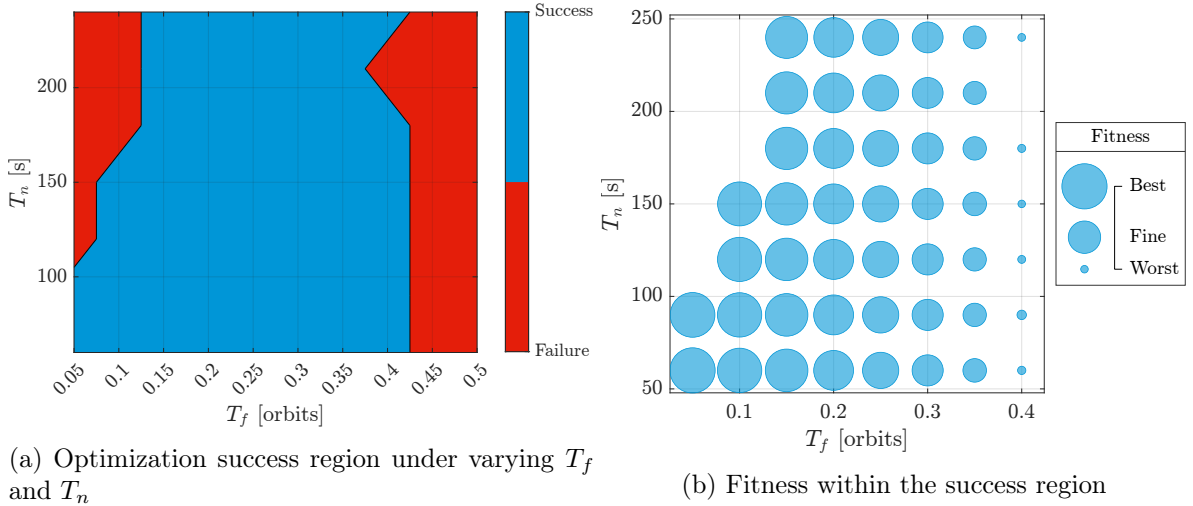


Figure 4.6: Sensitivity of the QCQP formulation (Problem 4.2) to  $T_f$  and  $T_n$  variations

Indeed, not all points within the success region are equivalent in terms of fuel efficiency, as variations in  $T_f$  and  $T_n$  alter the optimal Delta-V. To evaluate the effectiveness of each  $T_f$ - $T_n$  combination within the success region, a fitness function is introduced. This fitness function is defined as the reciprocal of the average optimal cost over the 100 initial conditions at each

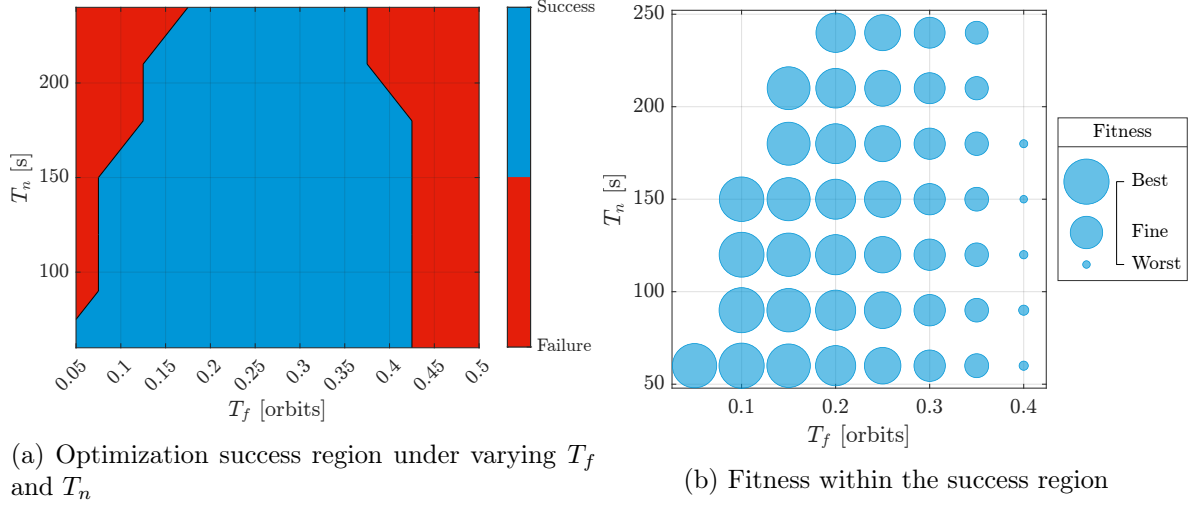


Figure 4.7: Sensitivity of the QP formulation (Problem 4.3) to  $T_f$  and  $T_n$  variations

$T_f$ - $T_n$  point. Formally, it is expressed as,

$$\text{Fitness} = N_{IC} \sum_{l=1}^{N_{IC}} \frac{1}{J_l^*}, \quad (4.54)$$

where  $N_{IC} = 100$  represents the number of initial conditions solved at each  $T_f$ - $T_n$  point, and  $J_l^*$  denotes the optimal cost of the problem (i.e., QCQP, QP, SOCP, or LP) for the  $l^{th}$  initial condition. The fitness is calculated using Eq. (4.54) at each point in the success region, with the results presented in Figures 4.6b, 4.7b, 4.8b, and 4.9b for the QCQP, QP, SOCP, and LP formulations, respectively.

It is unsurprising that the success region of the QCQP formulation (see Fig. 4.6a) mirrors that of the SOCP formulation (see Fig. 4.8a), as both problems explore the same control acceleration space. However, the success regions of the QP formulation (see Fig. 4.7a) and the LP formulation (see Fig. 4.9a) are smaller, as these formulations explore a more limited feasibility region of the control acceleration. Notably, all formulations exhibit the same sensitivity to changes in  $T_f$  (see Fig. 4.6b, Fig. 4.7b, Fig. 4.8b, and Fig. 4.9b).

The results depicted in Figures 4.6, 4.7, 4.8, and 4.9 align with intuition. Changes in

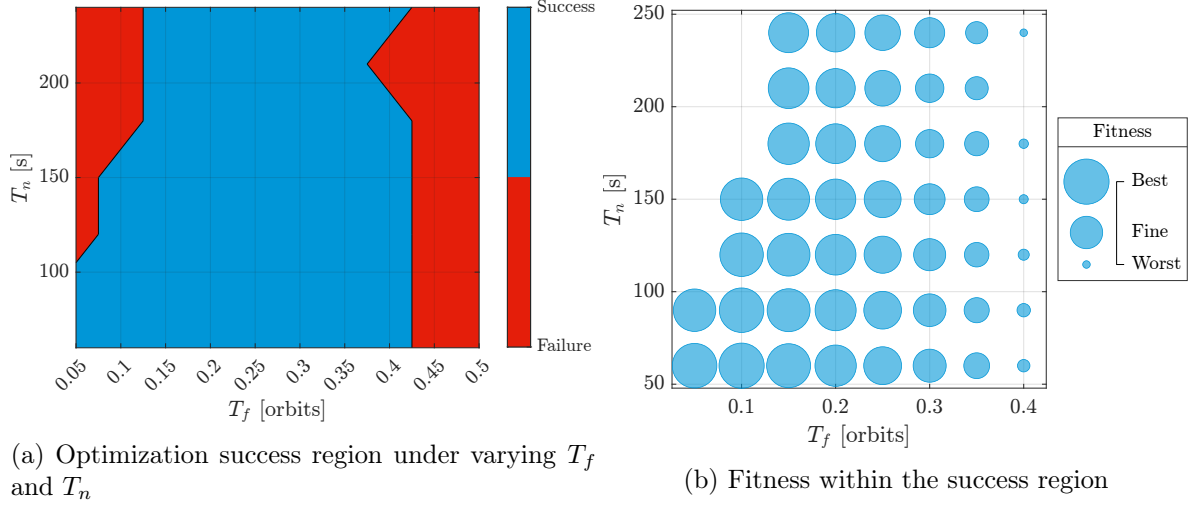


Figure 4.8: Sensitivity of the SOCP formulation (Problem 4.4) to  $T_f$  and  $T_n$  variations

$T_n$ , representing the duration of the coast arcs, minimally affect fitness for a fixed  $T_f$  value. However, if  $T_n$  becomes excessively large, the problem becomes infeasible. Conversely, a small  $T_f$  value allows for precise control over thrust timing, reflected in the optimal cost (higher fitness  $\equiv$  lower cost). A very large  $T_f$ , however, is undesirable from both fitness and feasibility perspectives. Although large  $T_f$  values reduce the number of variables and expedite problem-solving, they eventually lead to infeasibility, as the optimization problem formulations require constant thrust throughout the  $T_f$  period, which can be overly restrictive.

To choose a proper value for  $T_f$ , a compromise has to be made between the complexity of the problem from one side, and the fitness of the  $T_n - T_f$  from another. It is for this reason that the adopted value of  $T_f$  in many of the numerical experiments to follow is set to 0.2, which is a value that guarantees acceptable fitness, and at the same time, produces an acceptable number of decision variables. Conversely, choosing an adroit value for  $T_n$  is rather straightforward, since  $T_n$  is typically calculated analytically using the equality option in Eq. (4.1). For Triton-X [47], the maximum slew rate is 2deg/s, and hence  $T_n$  is calculated to be 100 s after choosing  $T_{\text{safety}} = 10$  s.

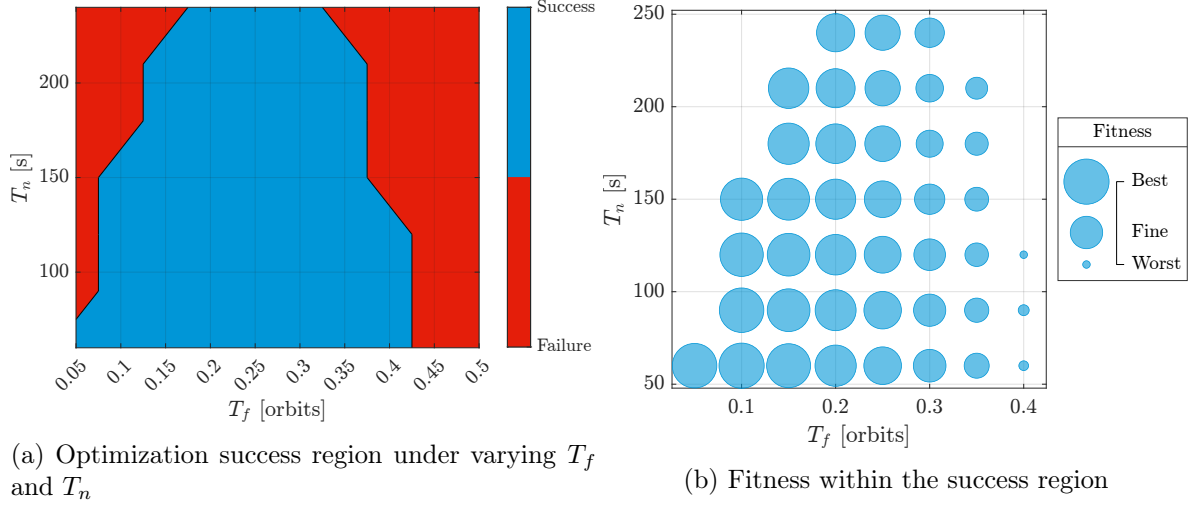


Figure 4.9: Sensitivity of the LP formulation (Problem 4.6) to  $T_f$  and  $T_n$  variations

## 4.5 Results and discussion

In this section, the proposed LP guidance scheme is validated through a case-study and the results of this case-study are analyzed. The LP formulation is the only scheme presented in this section as it offers more inherent details, such as the scaling factor, for discussion. Moreover, the ability of the proposed guidance strategies to adapt to mission requirements, including extended no-thrust-periods, is also validated. Furthermore, a comparison between the convex QCQP, the QP, SOCP, and the LP formulations is carried out, through a benchmark study of multiple solvers. The benchmark experiment allows for a deeper understanding of the limitations of the proposed formulations, and helps understanding which formulation is better suited for which reconfiguration scenarios.

### 4.5.1 Case study

To test the effectiveness of the guidance schemes, they were run over multiple simulation scenarios. One example of such scenarios is Reconfiguration 0 presented in Appendix D. Since the deputies in this reconfiguration scenario are assumed to be identical,  $u_{i,\max}$  are set to  $u_{\max}$  and  $\omega_{i,\max}$  are set to  $\omega_{\max}$  for all  $i \in \mathcal{I}$ . Furthermore, the simulation is set up such that there are no operational time constraints, and the durations of the coast arcs are thus set

to be all equal, i.e.,  $T_{n,l} = T_n \forall l \in \mathcal{L}$ , where  $T_n$  is calculated according to the equality option in Eq. (4.1) for  $\omega_{\max} = 2 \text{ deg/s}$  and  $T_{\text{safety}} = 10 \text{ s}$ . The forced motion periods are also fixed to a constant value, i.e.,  $T_{f,l} = T_f \forall l \in \mathcal{L}$ , which was tuned around the results of the sensitivity analysis presented in Section 4.4. A full list of the parameters used in the simulation is given in Table 4.1, which, if not tunable, correspond to that of Triton-X heavy [47].

Table 4.1: Parameters used in Problem 4.6 validation simulation

$t_f - t_0$	[orbits]	$T_f$	[orbits]	$T_n$	[s]	$u_{\max}$	$[\mu\text{m/s}^2]$	$R_{CA}$	[m]	$n_{\text{dir}}$	[-]	$c$	[-]
5		0.2		100		35		100		12		1.017	

Running the LP guidance scheme, Problem 4.6, over the described reconfiguration setting, profiles for the state vector, the control input vector, and the slack variables are obtained for each deputy at every step of the defined time vector. The trajectory followed by each of the deputies is depicted in Fig. 4.10, where the final relative orbit for all the deputies is seen to indeed resemble a Projected Circular Orbit (PCO) with a 300 m radius (refer to Appendix D). Note that Fig. 4.10 contains legends only for Sat. A to explain the line and marker shapes' convention in use. Legends for other satellites would have had the same shapes as those of Sat. A, yet with their respective colors.

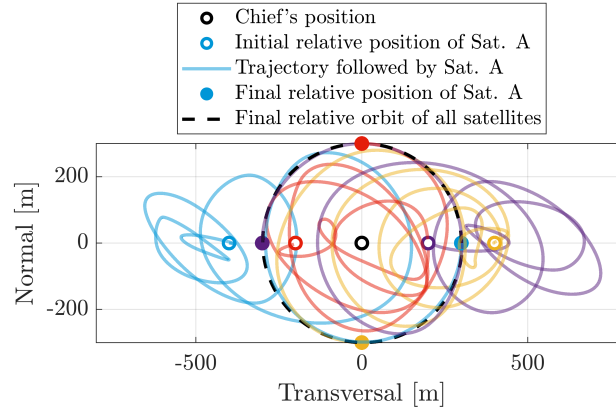


Figure 4.10: Trajectories followed by each of the deputies throughout the coplanar-to-PCO maneuver

Note that Fig. 4.10 shows the 2-dimensional view of the trajectory's projection onto the

T-N plane. In fact, Table D.0 suggests that the maneuver is a general one, which requires both, in-plane and out-of-plane corrections. This could be verified by looking either at the 3-dimensional visualization of the trajectories, or at the ROE profile of the maneuver. The ROE profile of the reconfiguration maneuver is depicted in Fig. 4.11, which not only shows corrections of the in-plane variables as well as the out-of-plane ones, but also demonstrates how each of the relative orbital elements of all the deputies matches its set-point at the final time of the maneuver.

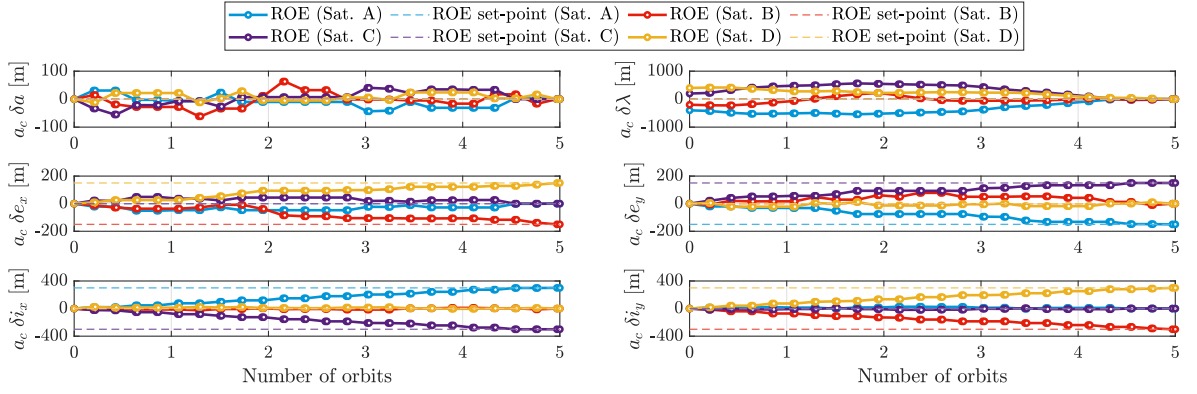


Figure 4.11: ROE profile over the coplanar-to-PCO maneuver

The optimality of the results were investigated through exploring the acceleration profiles of the maneuver. The radial, the transversal, and the normal components of the acceleration provided by each deputy are shown in Fig. 4.12.

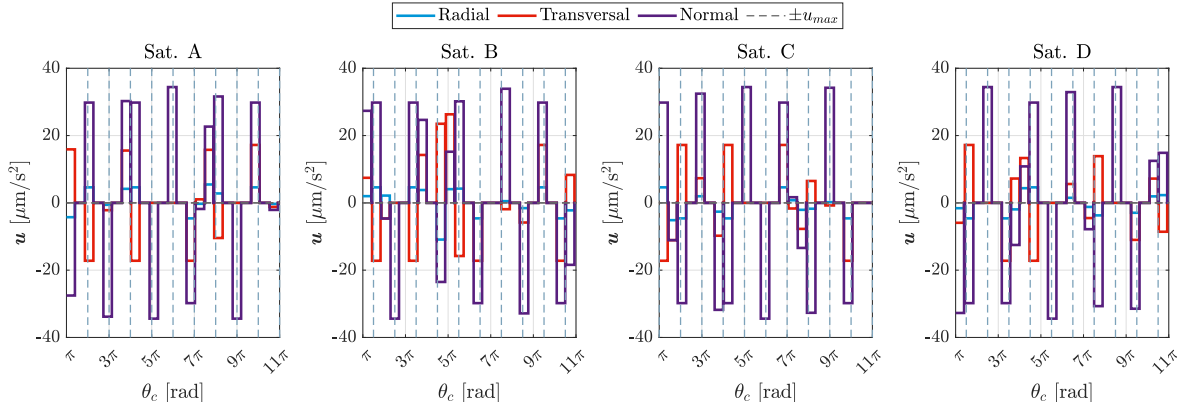


Figure 4.12: Control acceleration vector profile over the coplanar-to-PCO maneuver

One interesting aspect which can be seen in Fig. 4.12 is that the radial acceleration component is barely used even though it was not explicitly restricted in the formulation of Problem 4.6, simply because optimality dictates not exploiting it. Leveraging mostly the transversal and normal components does indeed coincide with our initial expectations, which formed a foundation for approximating the projections of the maximum acceleration constraint on the R-T and the R-N planes with only 4 affine constraints for each plane, unlike the 12 constraints which approximate the projection of the same constraint on the T-N plane (see Fig. 4.3). Moreover, the fact that the radial acceleration component is minimally utilized suggests that  $\delta\lambda$  corrections are not performed directly through thrust, but rather through varying the value of  $\delta a$  which, in-turn, changes  $\delta\lambda$  according to the natural dynamics (see Eq. (2.51)). The small corrections of the relative eccentricity vector are conceivably done using the transversal acceleration component since it is half as expensive as using the radial one from the Delta-V point of view [77]. Although the in-plane corrections are larger in magnitude, they are mostly done by exploiting the natural dynamics as discussed earlier. Thus, it comes as no surprise that the normal acceleration component, which is responsible only for out-of-plane corrections, is the most component that has been utilized. In Fig. 4.12, the dashed teal vertical lines represent the Delta-V-optimal locations to provide impulsive acceleration in the normal direction to achieve the required out-of-plane corrections [77]. That said, the obtained solution, once again, matches the expectations for an optimal control profile as the normal acceleration component is seen in Fig. 4.12 to act as a bang-bang controller around the optimal locations, while being zero away from them.

The difference between the two problems, 4.5 and 4.6, is only the scaling factor,  $c$ , which scales down the feasibility region of the maximum acceleration constraint. The effect of this scaling coefficient can be clearly seen by looking into the acceleration (norm) profiles throughout the maneuver, which are depicted in Figures 4.13a and 4.13b for Problems 4.5 and 4.6, respectively.

When  $c$  is set to unity, which is the case in Problem 4.5, the acceleration is prone to surpass the maximum level as seen in Fig. 4.13a. Conversely, in Fig. 4.13b, our claim that



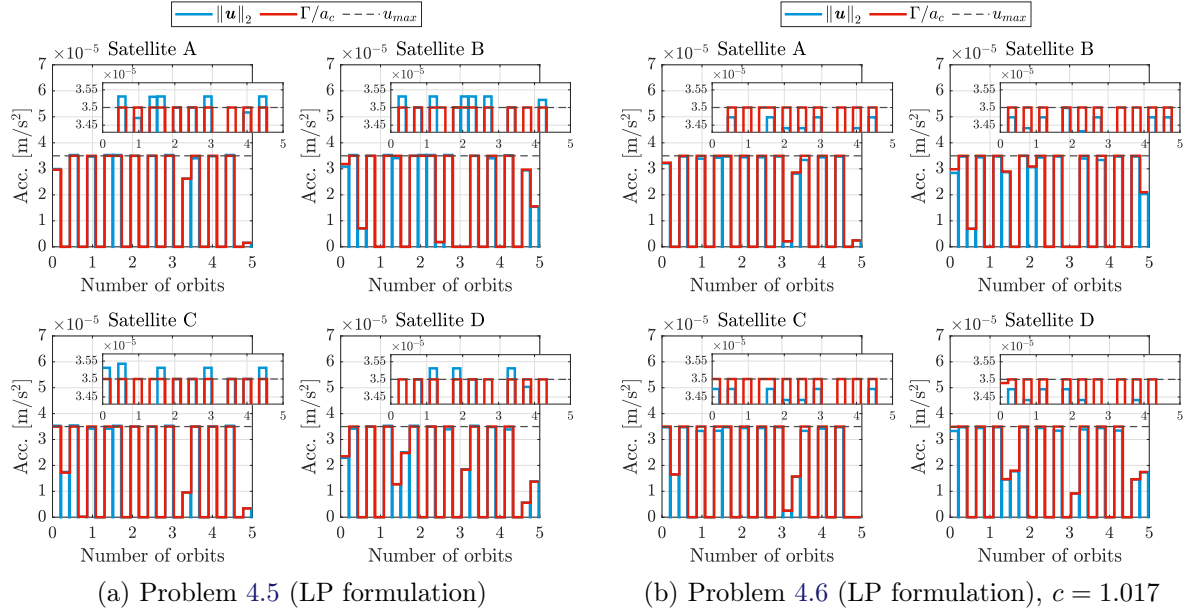


Figure 4.13: Control acceleration vector and slack variables over the coplanar-to-PCO maneuver

setting  $c$  to 1.017 (for  $n_{\text{dir}} = 12$ ) guarantees that the L2 norm of the acceleration never exceeds the maximum acceleration level is verified (see the blue line). It is important to note that in the former case, the acceleration will never surpass the maximum acceleration by more than 1.7% (for  $n_{\text{dir}} = 12$ ). Despite being minimal, this violation may lead to large errors over time in the ROE space which cannot be simply ignored when thrusters saturation is imposed, especially when the  $\delta a$  variations are used to drive the in-plane variables to their set points. Figures 4.13a and 4.13b also depict a scaled profile of the slack variable,  $\Gamma$ , throughout the reconfiguration maneuver (red lines), which verifies that the constraints in Eq. (4.46) never get violated. Note that  $\Gamma_{i,k}^* \neq a_c \|\mathbf{u}_{i,k}^*\|_2$  in the two LP formulations of the problem, which can be seen in Fig. 4.13, however,  $\Gamma_{i,k}^* = a_c \|\mathbf{u}_{i,k}^*\|_2$  in the SOCP formulation, as stated by Eq. (4.32). This can be verified by inspecting the profiles of the L2 norm of the control acceleration vector when Problem 4.4 is solved. These profiles are depicted for the case-study reconfiguration scenario in Fig. 4.14, which shows that, for all the forced-motion periods, the optimal (scaled)  $\Gamma$  is indeed equal to the acceleration level provided by each deputy.

A very important aspect of the modified guidance schemes, Problems 4.2 through 4.6, is

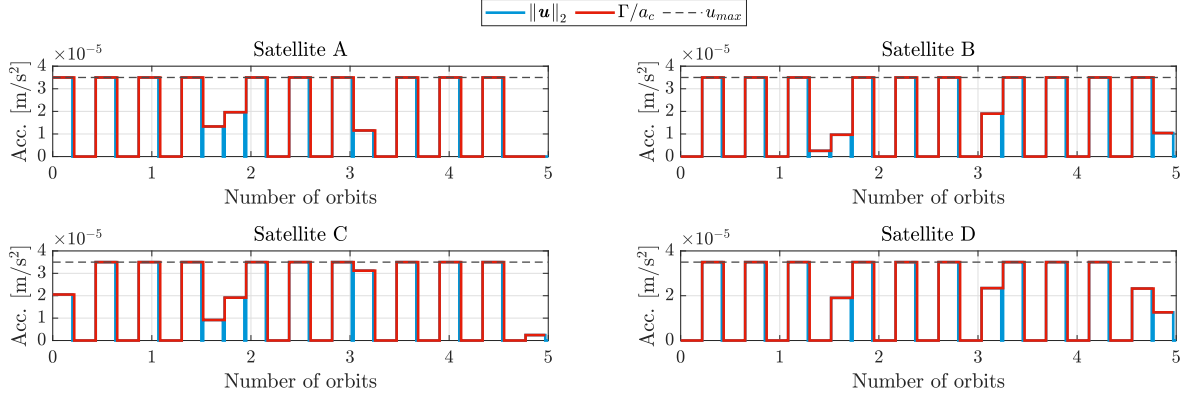


Figure 4.14: Control acceleration vector and slack variables over the coplanar-to-PCO maneuver, Problem 4.4 (SOCP formulation)

that they all rely on sequential convex programming, which may require the problem to be solved multiple times before an optimal solution is obtained. As illustrated by Fig. 4.2, the approach we adopted for solving Problem 4.2 through Problem 4.6 is to solve the problem first without the collision avoidance constraints to obtain estimates for  $\check{\mathbf{y}}_{i,k}$ . The problem is subsequently solved iteratively and the values of  $\check{\mathbf{y}}_{i,k}$  are updated at each iteration until any of the stopping criteria is met. The termination criteria were introduced in Section 4.2, but they are mentioned here once again to allow for a more elaborate discussion. The SCP termination criteria are,

- $\|\check{\mathbf{y}}_{i,k} - \mathbf{y}_{i,k}\|_2 \leq \epsilon$  at the current iteration, with  $\epsilon > 0$  being a user-defined threshold;
- The guidance profile of the current iteration is collision-free;
- The user-defined maximum number of iteration is reached, in which case, the solution trajectory is not guaranteed to be collision-free.

Indeed, adopting only the first criterion is guaranteed to result in an optimal guidance profile, if the problem is feasible to begin with, yet at the cost of computational time, since a large number of iterations might be required if  $\epsilon$  is chosen to be very small. Implementing the second criterion may lead to a sub-optimal solution, however, it considerably reduces the number of iterations required to solve the problem. The third criterion is nothing but a safeguard to ensure that the solver is not stuck in an infinite loop. In the context of our case-study,

two simulations were run (using the formulation of Problem 4.6) where the first adopted the two stopping criteria, the first and the last criteria, while the second adopted all of the three. The former required 7 iterations to solve the problem (using  $\epsilon = 1$  m), resulting in a solution which requires a total Delta-V of 1.8 m/s, while the latter needed only 1 iteration, providing a guidance profile which requires 1.82 m/s, which calls for only 1% increase in the required Delta-V. One thing which is worth noting is that, throughout our test simulations, adopting the second criterion almost always required a single iteration after the zeroth iteration in which the problem is solved without the collision avoidance constraints. This behavior is especially expected in cases where the solution of the zeroth iteration is almost collision-free.

The inter-satellite distances for our case-study are depicted in Fig. 4.15a for the zeroth SCP iteration and in Fig. 4.15b for the first SCP iteration, where the stars signify a location where collision avoidance is violated.

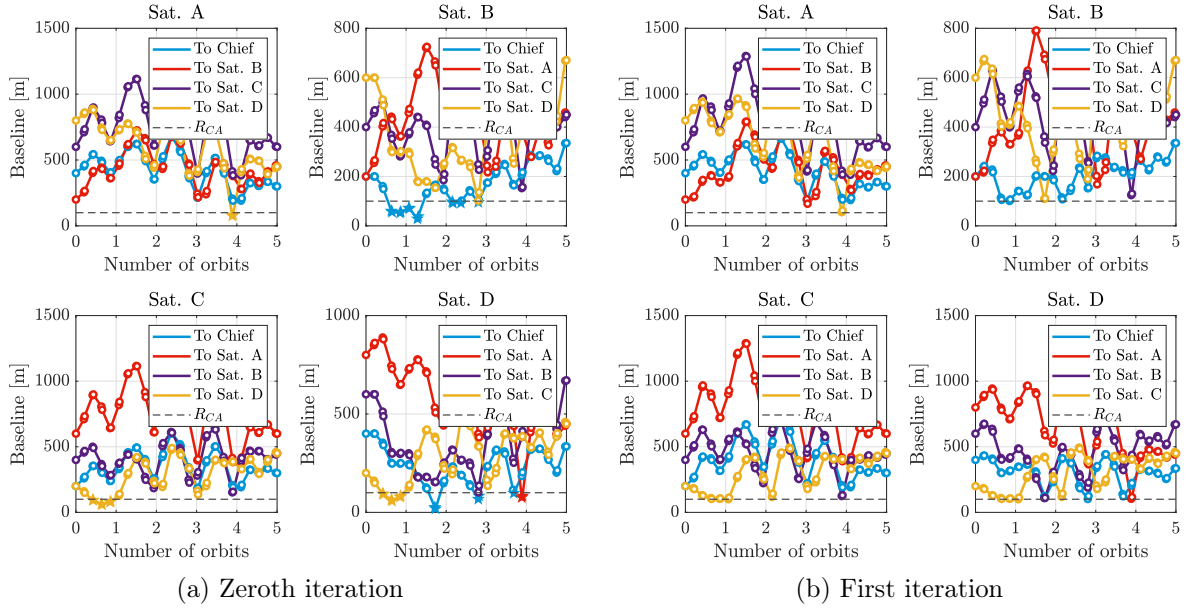


Figure 4.15: inter-satellite distances over the coplanar-to-PCO maneuver

### 4.5.2 Handling operational constraints

Satellite missions usually have constraints on the timing of the orbital maneuvers, which arise from not being able to perform any thruster firing, for example during ground contact, during eclipse periods, or during payload operations due to the need for a fully-operational [Attitude Determination and Control System \(ADCS\)](#) and/or because of the thruster vibrations which can compromise the pointing capabilities as well as the payload system. Unlike impulsive-thrust absolute/relative orbit correction maneuvers, low-thrust maneuvers may require too long to execute, days in some cases. It is for this reason that low-thrust guidance algorithms need to accommodate the no-thrust periods during the orbital maneuver itself. It is important to emphasize that the required thruster off-periods are generally known to the satellite/formation operator beforehand. One of the main contributions of this chapter is the ability of the proposed guidance schemes to adapt to different scenarios where the thruster is required to shut down for known extended periods without the need to change the structure of the problem. The duration of each coast arc,  $T_{n,l}$  (see Fig. 4.1), which is a user-input, is the only thing that needs to be adapted according to the operational constraints. To validate the ability of the trajectory optimization routines to accommodate long coast arcs, two different scenarios are introduced, where the reconfiguration in Table D.0 is requested in both scenarios. The maneuver duration as well as the initial and final [ROE](#) vectors are fixed for the two settings and only the no-thrust periods are varied. The simulation parameters for both cases are identical to those reported in Table 4.1 except for the maneuver time,  $t_f - t_0$ , which is set to 10 orbits. This extension of the maneuver period is motivated by the fact that, given the large durations of the coast arcs, the problem would become infeasible if the maneuver was required to be performed within 5 orbital revolutions. The thruster off-periods are chosen for each of the two scenarios and are reported in Table 4.2.

Table 4.2: Adopted no-thrust periods for two simulation scenarios

No-thrust periods [orbits]	
Scenario 1	$\{0.5-1, 1.5-2, \dots, 9.5-10\}$
Scenario 2	$\{0.25-1, 2.5-3, 4-4.75, 5-6, 8.5-9.5\}$

Namely, the first scenario corresponds to eclipse periods during which the electric thruster is turned off in order not to drain the batteries, where the satellite is assumed to experience eclipse during 50% of each orbit. Indeed, this is an overly conservative assumption for a satellite in LEO, yet the guidance problem is still solvable as will be shown. The second scenario considers arbitrarily chosen off-periods to emphasize that the no-thrust intervals need not to be chosen with a regular pattern.

The user-defined thruster no-thrust intervals for the two defined scenarios can be seen graphically in Fig. 4.16 where the expected behavior of the thruster is presented.

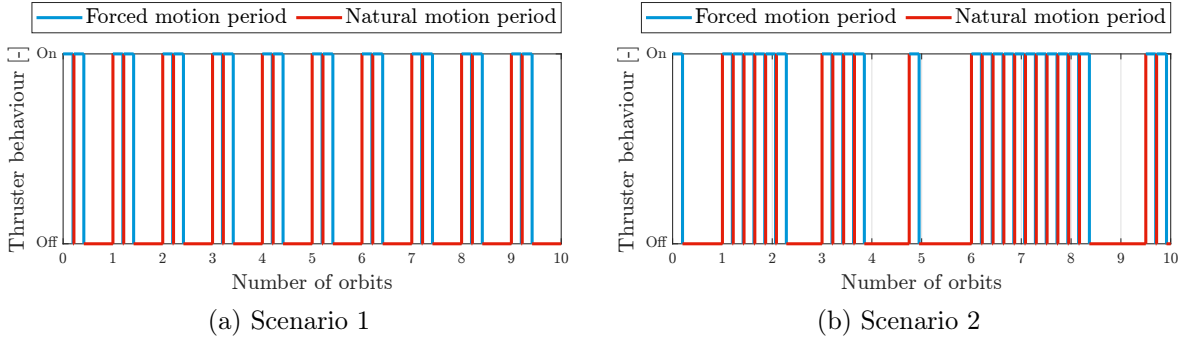


Figure 4.16: Expected behavior of the thruster in the presence of the thruster off-periods in Table 4.2

The dimensional ROE profile, predicted by the guidance scheme in Problem 4.6, is depicted in Fig. 4.17 and Fig. 4.18 for the first and the second scenarios, respectively, where the shaded areas represent the thruster off-periods. It is obvious that the  $\delta\lambda$  signal is conceivably evolving, in both cases, even when no thrust is provided, since it can be manipulated not only directly through input thrust, but also indirectly through the non-zero value of  $\delta a$ . Notably, the total Delta-V is different for each of the scenarios, as it counts to 2.193 m/s in the first scenario and to 1.774 m/s in the second.

The acceleration level, calculated by the guidance algorithm in Problem 4.6, is depicted in Fig. 4.19 and Fig. 4.20 for the first and the second scenarios, respectively, where it is clear that no acceleration is being provided to any deputy during the no-thrust periods.

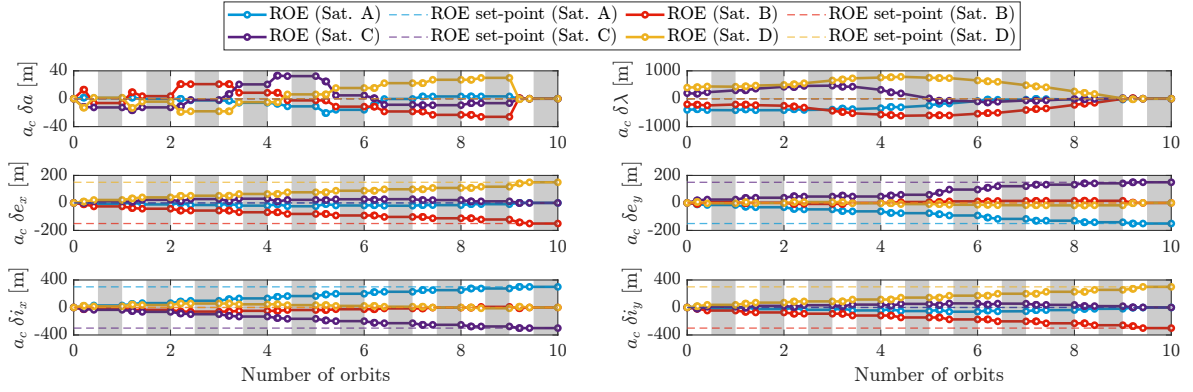


Figure 4.17: Guidance dimensional ROE profile under operational constraints - Scenario 1

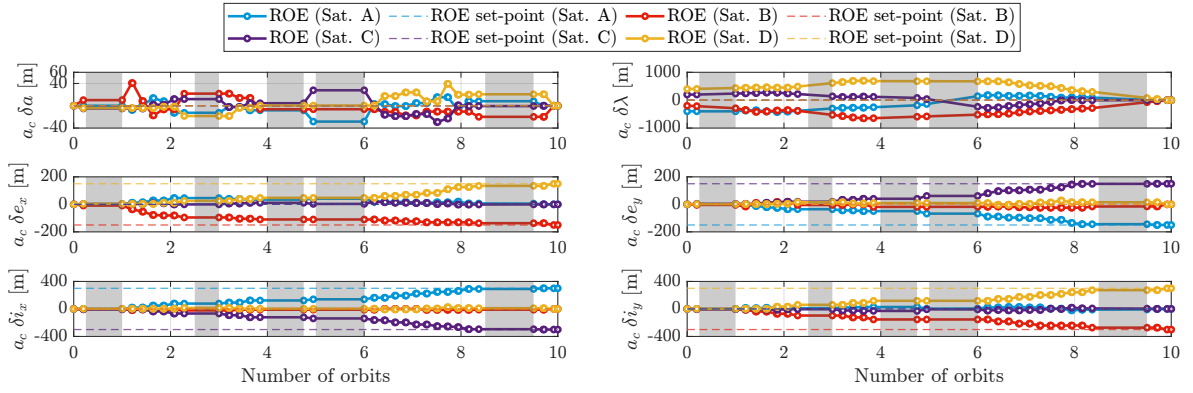


Figure 4.18: Guidance dimensional ROE profile under operational constraints - Scenario 2

### 4.5.3 Solvers benchmark

The total Delta-V cost of the SOCP formulation, Problem 4.4, is expected to be less than what results from solving the convex QCQP formulation, Problem 4.2, as well as that of QP formulation, Problem 4.3. For this obvious reason, the QCQP scheme had to be modified into the SOCP formulation. Approximating the SOCP problem by an LP formulation, Problem 4.6, could be justified by many motives, among which is the fact that this relaxation puts the problem in the simplest form of convex programming, which is the easiest to implement. Moreover, the LP formulation is expected to require less solve time, especially for lower dimensional reconfiguration scenarios, i.e., those scenarios that involve a low number of sampling instances, and consequently involve a low number of constraints. The LP formulation is not expected to be very fast for higher dimensional reconfiguration scenarios, since one SOCP

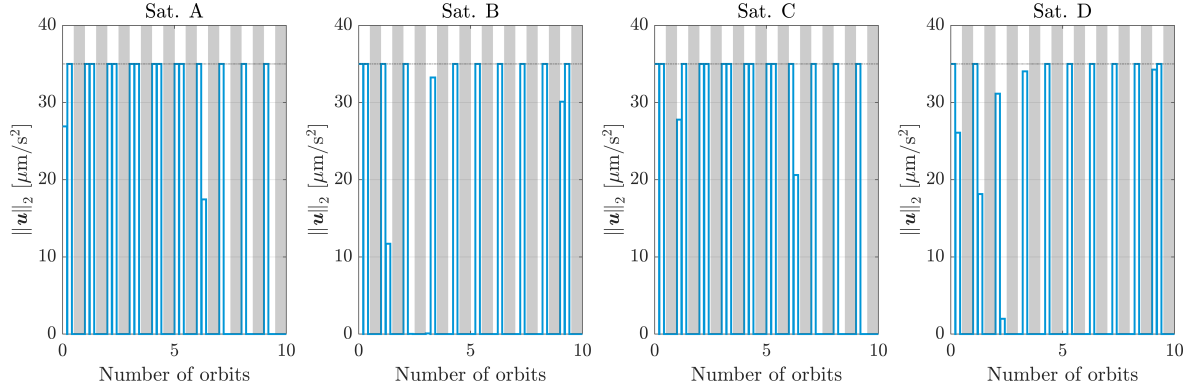


Figure 4.19: Guidance acceleration profile under operational constraints - Scenario 1

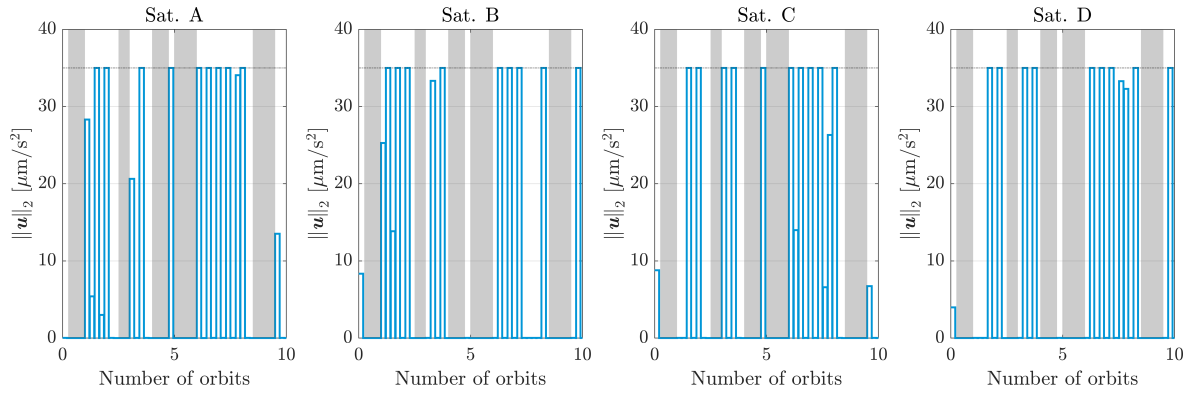


Figure 4.20: Guidance acceleration profile under operational constraints - Scenario 2

constraint is approximated by 20 linear constraints; 12 in the T-N plane, 4 in the R-N plane, and 4 in the T-R plane. One other advantage of transforming an [SOCP](#) problem into an approximated linear program is to use dedicated solvers for linear programming that could not be possibly used in an [SOCP](#) context. This opens the door to many open-source non-commercial solvers that can handle only linear constraints such as GLPK, HiGHS, OSQP, and many others. Furthermore, some solvers which can handle [SOCP](#) problems still use dedicated algorithms for linear programs that are not suitable for solving [SOCP](#) problems. Examples of such solvers include CPLEX, Gurobi, and Xpress, which usually favor simplex methods for linear programs, while using interior-point algorithms for [QCQP](#), [QP](#), and [SOCP](#) problems.

To get a clearer insight into the key differences between Problems [4.2](#), [4.3](#), [4.4](#), and [4.6](#),

four different reconfiguration scenarios were identified (Reconfigurations 1-4 in Appendix D), and were used for the purpose of benchmarking different solvers when run over each of these formulations. In the four scenarios, the deputies are assumed identical, and consequently,  $u_{i,max} = u_{max} \forall i \in \mathcal{I}$ . Moreover, the four reconfigurations assume no temporal constraints, and hence  $T_{n,l} = T_n \forall l \in \mathcal{L}$ , and  $T_{f,l} = T_f \forall l \in \mathcal{L}$ . The simulation parameters that are used in the benchmark experiment are set to their corresponding values from Table 4.1, and are shared in the four reconfigurations, except for the maneuver duration, which is defined separately for each reconfiguration scenario.

The four scenarios were carefully selected to reflect wide ranges of formation and problem sizes, i.e., inter-satellite distances, and number of variables and constraints, while being relevant from the applications point of view. The reader is referred to Appendix D for the full details of the reconfiguration scenarios. It is to be emphasized that the assumed configuration geometries, i.e., PCO, pendulum, cartwheel, and helix, are all of a great interest for remote sensing applications [3, 92], and the identified reconfigurations may resemble a formation geometry change within a multi-static Synthetic Aperture Radar (SAR) mission.

In the benchmark test, 15 of the most commonly used solvers for the types of problems in hand, according to the Mittelmann's benchmark<sup>1</sup> and according to the statistics of NEOS server<sup>2</sup>, were compared. An overview of the adopted 15 solvers is demonstrated in Table 4.3, where the problem types that could be handled by a solver are presented, together with information on whether the solver can be freely used in a commercial setting, or a software license needs to be purchased (on the date of writing this study, in June 2024). A more detailed discussion about these solvers is presented in Appendix E. Note that every QCQP or QP problem can be formulated as an SOCP one [90], nonetheless, the solvers that cannot handle QCQPs and QPs in their native form but can handle SOCP problems, e.g., SCS and

---

<sup>1</sup>This benchmark used to act a decision tree for optimization software. It can be accessed through: <https://plato.asu.edu/bench.html>, accessed on 01/06/2024

<sup>2</sup>The statistics of NEOS server can be accessed through: <https://neos-server.org/neos/report.html>, accessed on 01/06/2024



ECOS, are marked with an  $\times$  mark in the QCQP and in the QCQP columns. The opposite is not true, however. Solvers which do not accept SOC constraints in their native form but rather need a reformulation of these constraints into the quadratic form are still marked with a  $\checkmark$  mark in the SOCP column, since they still recognize the SOC nature of the reformulated constraints and treat them as such.

Table 4.3: Solvers overview

Solver	Free	QCQP	QP	SOCP	LP
GLPK [93]	$\checkmark$	$\times$	$\times$	$\times$	$\checkmark$
CLP [94]	$\checkmark$	$\times$	$\checkmark$	$\times$	$\checkmark$
OSQP [95]	$\checkmark$	$\times$	$\checkmark$	$\times$	$\checkmark$
OOQP [96]	$\checkmark$	$\times$	$\checkmark$	$\times$	$\checkmark$
SCS [97]	$\checkmark$	$\times$	$\times$	$\checkmark$	$\checkmark$
ECOS [98]	$\checkmark$	$\times$	$\times$	$\checkmark$	$\checkmark$
IPOPT [99]	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
SCIP [100]	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
MOSEK [101]	$\times^a$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
Gurobi [102]	$\times^a$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
CPLEX [103]	$\times^a$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
COPT [104]	$\times^a$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
Knitro [105]	$\times^b$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
Xpress [106]	$\times^c$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$
Matlab [107]	$\times$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$

<sup>a</sup> The solver offers a free academic license.

<sup>b</sup> The solver offers a free academic licenses for the professor and the students during the time of the course.

<sup>c</sup> The solver offers a community license which is limited to a maximum of 5000 variables and constraints.

Using their default parameters, each solver was run 10 times over each of the four re-configuration scenario (refer to Reconfigurations 1-4 in Appendix D) using the four problem formulations developed in this chapter, i.e., the QCQP formulation, Problem 4.2, the QP formulation, Problem 4.3, the SOCP formulation, Problem 4.4, and the LP formulation, Problem 4.6. The average time it took a solver to complete each of the 10 iterations was then recorded. The benchmark was conducted on a Windows-PC which comprises an Intel Core i9-10885H

Table 4.4: Benchmark results

	Reconfiguration 1				Reconfiguration 2				Reconfiguration 3				Reconfiguration 4			
	QCQP	QP	SOCQP	LP	QCQP	QP	SOCQP	LP	QCQP	QP	SOCQP	LP	QCQP	QP	SOCQP	LP
Variables	1608	1608	1696	1696	2412	2412	2544	2544	1608	1608	1696	1696	4356	4356	4596	4596
Constraints	1594	2210	1594	3266	2661	3585	2661	5169	1144	1760	1594	3266	4821	6501	4821	9381
$\Delta V$ [m/s]	1.11	1.11	0.96	0.98	3.03	3.03	2.66	2.76	1.73	1.73	1.68	1.73	4.65	4.67	3.99	4.12
SCP iter.	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
Solver	Solve time [s]															
GLPK	-	-	-	0.29	-	-	-	0.58	-	-	-	0.20	-	-	-	2.08
CLP	-	6.99	-	0.54	-	I.L. <sup>a</sup>	-	1.91	-	1.44	-	0.54	-	I.L. <sup>a</sup>	-	I.L. <sup>a</sup>
OSQP	-	0.02	-	0.29	-	0.02	-	0.35	-	0.00	-	0.27	-	0.19	-	0.43
OOQP	-	0.12	-	0.25	-	0.20	-	0.41	-	0.03	-	0.20	-	0.56	-	0.85
SCS	-	-	0.24	1.51	-	-	0.23	1.18	-	-	0.16	3.87	-	-	1.55	4.58
ECOS	-	-	0.04	0.08	-	-	0.08	0.14	-	-	0.05	0.06	-	-	0.15	0.24
IPOPT	1.65	0.55	20.25	1.18	4.04	0.75	T.L. <sup>b</sup>	1.76	0.58	0.21	10.12	0.92	T.L. <sup>b</sup>	1.52	T.L. <sup>b</sup>	3.24
SCIP	T.L. <sup>b</sup>	0.65	T.L. <sup>b</sup>	0.34	T.L. <sup>b</sup>	0.74	T.L. <sup>b</sup>	0.51	T.L. <sup>b</sup>	0.20	15.46	0.29	T.L. <sup>b</sup>	2.57	T.L. <sup>b</sup>	1.81
MOSEK	0.07	0.15	0.06	0.13	0.14	0.16	0.12	0.18	0.02	0.04	0.07	0.10	0.20	0.31	0.17	0.34
Gurobi	0.48	0.17	0.15	0.19	0.38	0.18	0.18	0.31	0.34	0.09	0.28	0.25	0.71	0.24	0.27	0.42
CPLEX	0.39	0.29	N.I. <sup>c</sup>	0.46	0.47	0.27	N.I. <sup>c</sup>	0.51	0.18	0.13	N.I. <sup>c</sup>	0.45	0.74	0.33	N.I. <sup>c</sup>	0.54
COPT	0.11	0.07	0.08	0.14	0.16	0.11	0.09	0.20	0.04	0.03	0.07	0.14	0.31	0.20	0.16	0.32
Knitro	0.06	0.07	0.21	0.12	0.09	0.11	0.32	0.19	0.02	0.02	0.19	0.10	0.17	0.31	0.49	0.41
Xpress	0.20	0.16	0.18	0.18	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>	0.08	0.08	0.18	0.22	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>
Matlab	2.22	0.21	0.23	0.14	6.38	0.35	0.55	0.54	0.79	0.05	0.23	0.19	13.75	0.96	1.98	1.25

<sup>a</sup> The solver reached the Iteration Limit (I.L.) before finding a feasible solution.

<sup>b</sup> The solver reached the Time Limit (T.L.) (20 seconds) before finding a feasible solution.

<sup>c</sup> The solver is Not Interfaced (N.I.) for the type of problem in question. Refer to Appendix E for the reason.

<sup>d</sup> The solver did not solve the problem because of License Limitations (L.L.). Note that the community license of Xpress is limited to a maximum of 5000 variables and constraints.

CPU with 16 cores and a clock speed of 2.4 GHz. The language that was used to formulate the problems is Matlab, and the problems were passed to each solver either through its native Matlab [Application Programming Interface \(API\)](#), e.g., OSQP and SCS, or through a third-party Matlab interface, e.g., GLPK and CLP. Some interesting details on the 15 adopted solvers and on how they were interfaced with Matlab are presented in [Appendix E](#). It is of utmost importance to declare that the purpose of this benchmark is not to compare the performance of the selected solvers. After all, tweaking the parameters of a solver may very probably result in a different performance. The purpose is rather to draw some recommendations as to which formulation is better used under which conditions, and also to showcase that using the different proposed formulations for the guidance problem has a major effect on both, the solve time of the problem, and the total Delta-V required for the maneuver. The results of the benchmark are presented in [Table 4.4](#). The table also conveys some of the problem properties which are solver-independent. Namely the following properties are reported; the number of decision variables, the number of constraints excluding decision variables' bounds, the total Delta-V, which is the summation of the Delta-V required by each deputy to complete the maneuver, and the number of required [SCP](#) iterations after the zeroth iterations in order to arrive to a collision-free reconfiguration. It is crucial to highlight that the Delta-V values presented in [Table 4.4](#) are drawn solely from the results of MOSEK, rather than representing the average Delta-V across all solvers. While it is generally expected that all solvers will yield nearly identical Delta-V costs for a given convex problem, variations can occur due to the numerical issues a solver may encounter when applied to specific problems. Furthermore, since not all solvers are capable of producing a solution for every reconfiguration scenario, reporting the Delta-V from a single solver ensures a fair comparison of the required Delta-V across the four reconfigurations. The log files for the benchmark experiment are available in [\[108\]](#).

[Table 4.4](#) presents, in a quantitative manner, how much of the total Delta-V can be saved simply by adopting the [SOCP](#) or the [LP](#) formulations. Conversely, adopting the [QP](#) formulation requires more Delta-V than that which is required by the [QCQP](#) problem. This is shown graphically in [Fig. 4.21](#), where the Delta-V savings (in percentage with respect to

the Delta-V required by the **QCQP** problem) are depicted for each formulation and for the four reconfiguration scenarios employed in the benchmark experiment.

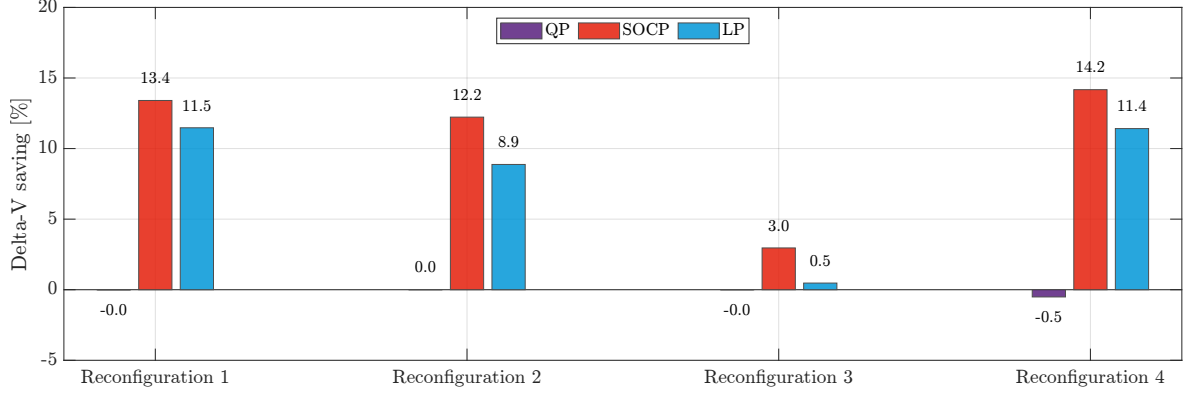


Figure 4.21: Delta-V saving by adopting a different formulation from the **QCQP**

It comes as no surprise that the **SOCP** formulation is requiring less total Delta-V than the **LP** formulation, since the **SOCP** problem is exploring a much larger action space than the **LP** is. The same can be said for the **QP** problem which explores a smaller action space than the **QCQP** (refer to Figures 4.3 and 4.4), while using the same cost function. It appears that the effect of using the cost function of the **LP** formulation on the total Delta-V is much larger than the effect of using a smaller feasibility region, which is why the total Delta-V of the **LP** formulation is smaller than the one required by the **QCQP** formulation in all of the four reconfiguration scenarios.

It is also clear from Table 4.4 how the number of variables is identical for the **SOCP** and the **LP** formulations, since they both consider all the entries of the  $\mathbf{\Gamma}$  matrix as decision variables. The same is true when comparing the number of variables of the **QP** problem with that of the **QCQP** one. Conversely, the number of constraints for the **QP** as well as the **LP** formulations are always much higher than the identical number of constraints of the **QCQP** and the **SOCP** formulations. The reason for this is that each instance of Eq. (4.13) (or Eq. (4.27)) is approximated by 20 linear constraints, represented by Equations (4.20) and (4.21) in the **QP** formulation (or Equations (4.44) and (4.45) in the **LP** formulation). The effect of adopting the various problem formulations on the number of required **SCP** iterations

(after the zeroth one) is unclear, mainly due to the fact that [SCP](#) is set to terminate once a collision-free reconfiguration is obtained, which is a sub-optimal approach that turns out to be much faster, as has been pointed out earlier. However, for Reconfiguration 3, the solution of the zeroth iteration of the [QCQP](#) and the [QP](#) problems is collision-free, while it required one more iteration from the [SOCP](#) and the [LP](#) formulations to ensure a collision-free maneuver.

To get a better idea on how fast the different solvers return the optimal solution to the problem in hand, the solve time data, reported in Table 4.4, were averaged across the four reconfiguration scenarios, and the mean is depicted for each solver in Fig. 4.22. For a fair comparison, any solver that failed to produce a solution for one or more reconfiguration scenarios, due to reasons such as time limits or license restrictions, was excluded from the summary for the specific formulation where it was unable to generate a feasible result.

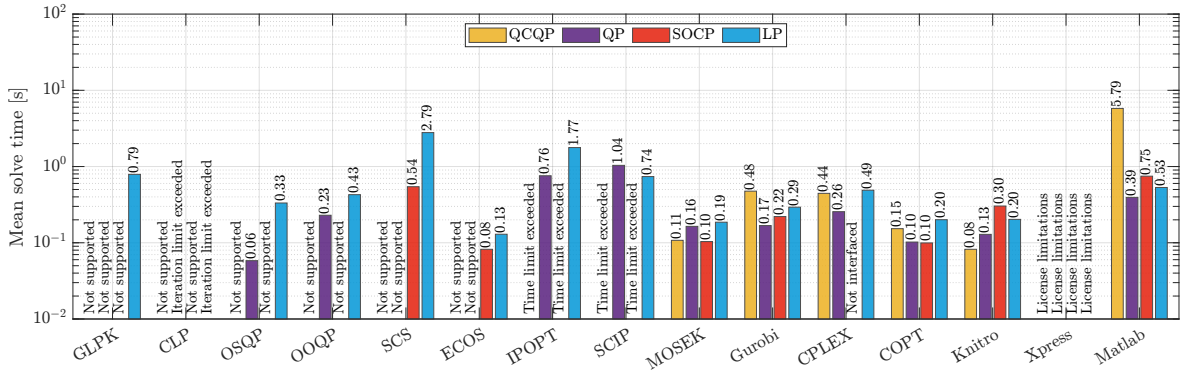


Figure 4.22: Benchmark summary

At first glance on Fig. 4.22, it becomes very clear that, among all the solvers, those which are not dedicated to convex optimization, i.e., IPOPT, SCIP, and Matlab in the [QCQP](#) case, are the slowest, although SCIP perform competitively in the [QP](#) and [LP](#) cases due to the simplicity of the problems. It is to be noted that many of the benchmarked solvers can still handle [Non-Linear Programming \(NLP\)](#) problems, e.g., CPLEX and Gurobi, however, they do recognize convex optimization problems, and hence, use dedicated algorithms that work best for the convex cases. The free solvers which are dedicated to the [QP](#) and the [LP](#) cases, i.e., GLPK, OSQP, and OOQP, present themselves as plausible candidates. Moreover, it is

hardly surprising that dedicated conic solvers, namely SCS and ECOS, perform generally faster for the conic case, while the dedicated QP solvers, i.e., OSQP and OOQP, perform better for the Quadratic Programming (QP) case. In fact, OSQP and ECOS appear to perform exceptionally well for the QP and the SOCP cases, respectively, even better than the commercial solvers for these two types of problems.

There is little to no conclusion that can be drawn based on the results of the commercial solvers. For instance, some solvers perform the best for the QCQP formulation, some are favored for the SOCP case, and some are faster when handling the QP problems. The main recommendation from Table 4.4 as well as Figures Fig. 4.21 and 4.22 is to avoid the QCQP formulation since it is Delta-V intensive, while not being the fastest to solve, despite involving the least number of variables and constraints. One other takeaway which can be observed by looking into the details of Table 4.4 is that the performance of the solvers over the LP problems is better when the number of constraints of the original SOCP problem is small, e.g., Reconfigurations 1 and 3. This claim is supported by the results of Knitro, and Gurobi. Conversely, for larger problem sizes, the SOCP problems are generally solved faster. It is for this reason that the LP formulation could only be recommended for small problem sizes, while the SOCP formulation is recommended for larger problems, although being harder to implement. In spite of the fact that the QP formulation is typically fast to solve by many of the solvers, this formulation is not recommended for a real mission scenario, since it is the most Delta-V intensive as outlined by Fig. 4.21.

To demonstrate the impact of the normalization described in Section 4.3 on both the numerical stability of the solvers and the time required to find a solution, the benchmark experiment was repeated without the normalization step. The solve time exploited by each solver for the four formulations together with some problem properties are presented in Table 4.5.

The average solve time of each solver across the four reconfiguration scenarios for the new benchmark experiment is shown graphically in Fig. 4.23.

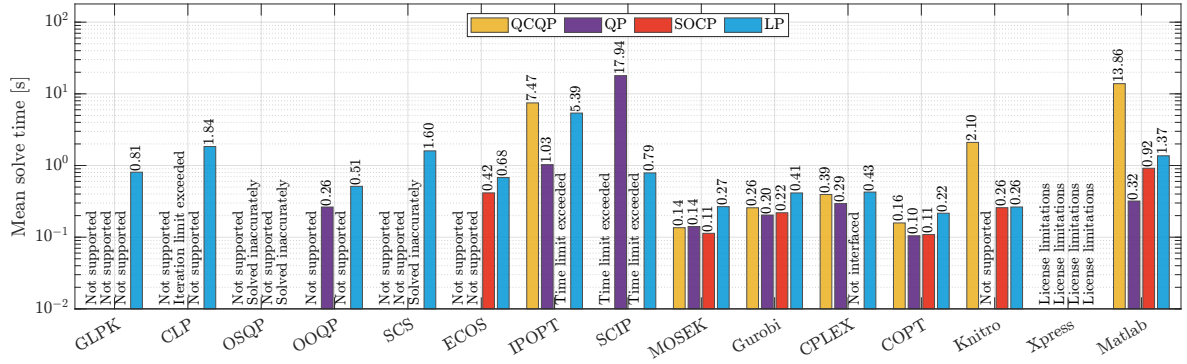


Figure 4.23: Summary of benchmark excluding the normalization step

Table 4.5: Benchmark results when the normalization step is excluded

	Reconfiguration 1				Reconfiguration 2				Reconfiguration 3				Reconfiguration 4			
	QCQP	QP	SOCP	LP	QCQP	QP	SOCP	LP	QCQP	QP	SOCP	LP	QCQP	QP	SOCP	LP
Variables	1608	1608	1696	1696	2412	2412	2544	2544	1608	1608	1696	1696	4356	4356	4596	4596
Constraints	1594	2210	1594	3266	2661	3585	2661	5169	1144	1760	1594	3266	4821	6501	4821	9381
$\Delta V$ [m/s]	1.11	1.47	0.96	0.98	3.03	3.29	2.66	2.75	1.73	1.80	1.68	1.73	4.65	5.20	3.99	4.10
SCP iter.	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
Solver	Solve time [s]															
GLPK	-	-	-	0.28	-	-	-	0.59	-	-	-	0.21	-	-	-	2.15
CLP	-	6.43	-	0.72	-	I.L. <sup>a</sup>	-	1.77	-	0.45	-	0.46	-	I.L. <sup>a</sup>	-	4.41
OSQP	-	I.S. <sup>e</sup>	-	0.17	-	I.S. <sup>e</sup>	-	-	-	I.S. <sup>e</sup>	-	I.S. <sup>e</sup>	-	I.L. <sup>a</sup>	-	0.51
OOQP	-	0.16	-	0.31	-	0.25	-	0.48	-	0.04	-	0.25	-	0.61	-	1.00
SCS	-	-	0.30	1.26	-	-	I.S. <sup>e</sup>	1.88	-	-	0.03	0.56	-	-	2.18	2.70
ECOS	-	-	0.30	0.34	-	-	0.37	0.71	-	-	0.09	0.41	-	-	0.90	1.26
IPOPT	3.77	0.72	20.25	2.04	5.84	1.10	20.27	5.61	0.85	0.26	10.12	2.68	19.43	2.04	T.L. <sup>b</sup>	11.24
SCIP	11.82	20.46	11.66	0.32	T.L. <sup>b</sup>	20.38	T.L. <sup>b</sup>	0.83	7.82	10.25	15.89	0.30	20.95	20.68	T.L. <sup>b</sup>	1.70
MOSEK	0.10	0.10	0.07	0.12	0.16	0.14	0.12	0.19	0.02	0.03	0.07	0.12	0.26	0.29	0.19	0.64
Gurobi	0.26	0.17	0.16	0.22	0.22	0.19	0.19	0.46	0.17	0.08	0.26	0.27	0.37	0.37	0.27	0.72
CPLEX	0.36	0.31	N.I. <sup>c</sup>	0.40	0.42	0.39	N.I. <sup>c</sup>	0.43	0.16	0.12	N.I. <sup>c</sup>	0.36	0.64	0.35	N.I. <sup>c</sup>	0.51
COPT	0.12	0.08	0.08	0.15	0.16	0.11	0.11	0.22	0.04	0.03	0.08	0.14	0.31	0.20	0.17	0.34
Knitro	4.40	0.20	0.13	0.16	1.60	0.75	0.21	0.17	0.02	0.02	0.23	0.11	2.39	I.S. <sup>e</sup>	0.46	0.62
Xpress	0.24	0.18	0.20	0.21	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>	0.09	0.08	0.19	0.21	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>	L.L. <sup>d</sup>
Matlab	10.28	0.18	0.27	0.18	10.92	0.28	0.74	1.24	1.45	0.04	0.29	0.28	32.80	0.78	2.36	3.77

<sup>a</sup> The solver reached the Iteration Limit (I.L.) before finding a feasible solution.

<sup>b</sup> The solver reached the Time Limit (T.L.) (20 seconds) before finding a feasible solution.

<sup>c</sup> The solver is Not Interfaced (N.I.) for the type of problem in question. Refer to Appendix E for the reason.

<sup>d</sup> The solver did not solve the problem because of License Limitations (L.L.). Note that the community license of Xpress is limited to a maximum of 5000 variables and constraints.

<sup>e</sup> The solver managed to find a solution for the problem, however, it was an Inaccurate Solution (I.S.).



By comparing the results in Fig. 4.23 with those in Fig. 4.22, it is evident that normalization enhances the solve times for almost all free solvers as well as some commercial ones. This improvement positions OSQP as the top performer for the QP case and ECOS as the best for the SOCP case. Without normalization, these solvers perform poorly. Specifically, OSQP and SCS fail to handle a subset of the reconfiguration scenarios (refer to Table 4.5). This is reflected in Fig. 4.23, where the OSQP solver is indicated to have inaccurately solved the QP and LP formulations, while the SCS solver is shown to have an inaccurate solution for the SOCP formulation. An inaccurate solution implies that the solver arrived at a solution with a different cost from the other solvers, and additionally, that solution was infeasible at the boundaries of the feasibility region.

The advantages of normalizing the objective and constraints matrices extend to enhancing the numerical stability of the solvers. This is most evident in the previously discussed inaccurate solutions of OSQP and SCS, and also in the required Delta-V for the QP formulation, where the non-normalized benchmark results show a much higher Delta-V compared to the normalized benchmark results (refer to Table 4.4 and Table 4.5). Conversely, the optimal Delta-V values for all other formulations remain nearly identical in both benchmark cases. Again, the reported Delta-V values are those required by MOSEK, which appear to exhibit some numerical issues when handling the QP formulation. While other solvers may exhibit greater numerical stability for the QP case, the primary objective here is not to compare the performance of each solver, but rather to demonstrate the effect of normalization on the numerical stability of the solvers, which has been clearly shown by the results in Table 4.4 and Table 4.5. As demonstrated for the normalized benchmark results, the effect of adopting the different problem formulations on the Delta-V is shown graphically in Fig. 4.24, where the Delta-V saving of each formulation with respect to the QCQP one is depicted.

The results in Fig. 4.24 are nearly identical to those in Fig. 4.21 for the SOCP and LP cases. However, the effect of normalization is most apparent in the QP case, as discussed previously.

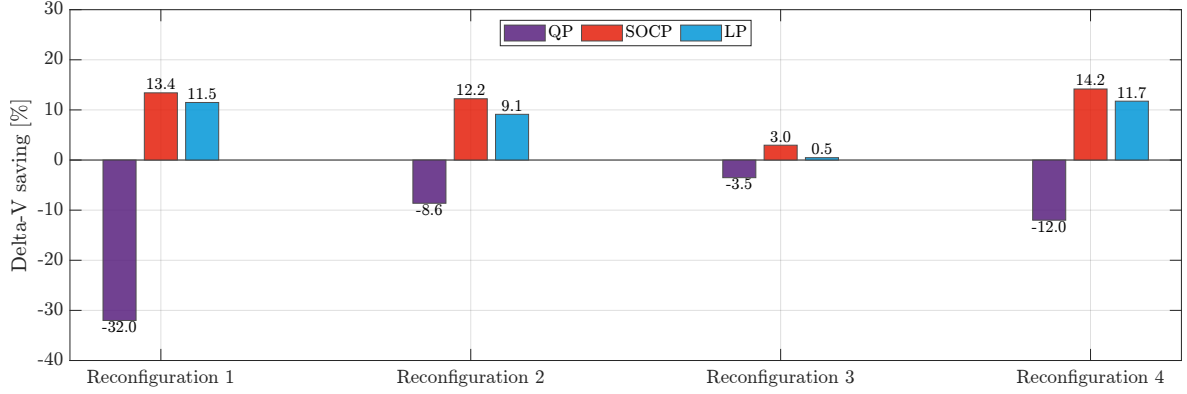


Figure 4.24: Delta-V saving by adopting a different formulation from the QCQP when normalization is not employed

## 4.6 Minimum thrust constraint

Many of the [Commercial-Off-The-Shelf \(COTS\)](#) electric thrusters, not to mention the one employed by Triton-X, are constrained not only by a maximum thrust bound, but also by a minimum one, beyond which the thruster is not throttleable. In the context of our low-thrust formation reconfiguration problem, where each deputy is equipped with a single thruster, this bound translates to a constraint on the  $L_2$  norm of the control thrust vector, or alternatively, on the  $L_2$  norm of the control acceleration vector. This can be formally written as,

$$\|\mathbf{u}_{i,k}\|_2 \geq u_{i,\min}, \quad (4.55)$$

where  $u_{i,\min}$  the minimum allowable acceleration level by the  $i^{\text{th}}$  deputy.

The constraint in Eq. (4.55) is a non-convex one, since it represents the outer region of a sphere with radius  $u_{i,\min}$ . If the minimum acceleration constraint is incorporated in any of the proposed convex guidance problems, it directly renders them non-convex. A lossless convexification technique was proposed [109, 110] for this non-convex constraint, yet the biggest drawback of this convexification proposal is that it relies on specific observability conditions, which are not met by the [ROE](#) dynamics. Namely, when the [ROE](#) dynamics are expressed in the standard state-space form as a [Linear Time-Varying \(LTV\)](#) system, the pair

comprising the system matrix and the input matrix must be fully observable, or equivalently, the system must be fully controllable. However, as demonstrated in [111], this is not true for the ROE dynamics. Here, the minimum acceleration constraint is convexified using a dynamics-independent affine relaxation, where the constraint in Eq. (4.55) is rewritten in an approximated convex form as follows:

$$\check{\mathbf{u}}_{i,k}^T \mathbf{u}_{i,k} \geq \|\check{\mathbf{u}}_{i,k}\|_2 u_{i,\min}, \quad (4.56)$$

where  $\check{\mathbf{u}}_{i,k}$  is assumed to be a known vector. A visual representation of this relaxation, using the projections of  $\check{\mathbf{u}}_{i,k}$  and  $\mathbf{u}_{i,k}$  into the T-N plane without loss of generality, is depicted in Fig. 4.25 together with the original non-convex constraint. The maximum acceleration constraint is also included in the figure for completeness.

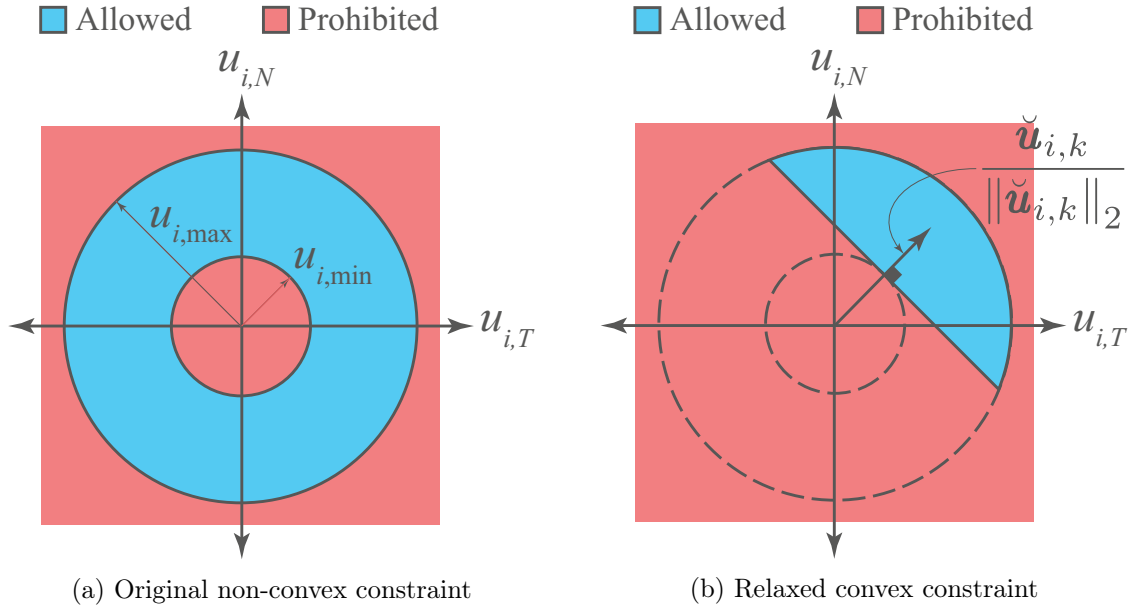


Figure 4.25: Feasibility region of the acceleration vector

The relaxation in Eq. (4.56) can be rewritten in terms of  $\bar{\mathbf{u}}_{i,k} = a_c \mathbf{u}_{i,k}$ , so that it could be added to the list of constraints of the guidance problem, as,

$$\check{\mathbf{u}}_{i,k}^T \bar{\mathbf{u}}_{i,k} \geq \|\check{\mathbf{u}}_{i,k}\|_2 a_c u_{i,\min}, \quad (4.57)$$

where, again,  $\check{\check{\mathbf{u}}}_{i,k}$  is assumed to be a known vector.

One of the most significant challenges in implementing the proposed relaxation in (4.57) lies in determining the linearization direction, denoted as  $\check{\check{\mathbf{u}}}_{i,k}/\|\check{\check{\mathbf{u}}}_{i,k}\|_2$ . This task is particularly complex because the linearization direction must be computed individually for each satellite and at every discrete time step, effectively acting as an initial guess for the affine relaxation process. The selection of this direction is critical, as it directly influences the feasibility and efficiency of the convexified optimization problem. To generate this initial guess, a two-step approach is proposed. The first step involves solving the problem without enforcing the minimum acceleration constraint given in Eq. (4.57). By doing so, an unconstrained solution for the acceleration profile,  $\mathbf{u}_{i,k}$ , is obtained for each satellite. This step namely incorporates the SCP scheme in Fig. 4.2. In the following step, an *acceleration pruning* process is performed. This procedure starts by evaluating the acceleration profile of each satellite. Specifically, if the mean acceleration across the entire profile is less than a predetermined threshold, the instances with the lowest acceleration values are set to zero. The rationale for this step lies in the fact that the applied acceleration can range from the minimum to maximum allowable levels, and can also be zero. Solving the problem while accounting for zero acceleration typically necessitates mixed-integer optimization techniques. By selectively pruning the smallest accelerations and re-solving the problem, a sub-optimal solution can be achieved without resorting to these computationally expensive methods. Additionally, the unpruned time instances are allowed to utilize higher acceleration levels, thereby redistributing the available capacity. This redistribution generally better satisfies the minimum acceleration constraint and increases the likelihood of obtaining a feasible solution when Eq. (4.57) is imposed in a later stage. Once the pruning process is complete, the pruned acceleration profile serves as the initial guess for the linearization direction in the affine relaxation. This refined guess is then employed in subsequent iterations of the optimization problem, which now includes the convexified version of the minimum acceleration constraint, Eq. (4.57). By incorporating these steps into the SCP strategy, Fig. 4.2 is modified, and the new version is depicted in Fig. 4.26.

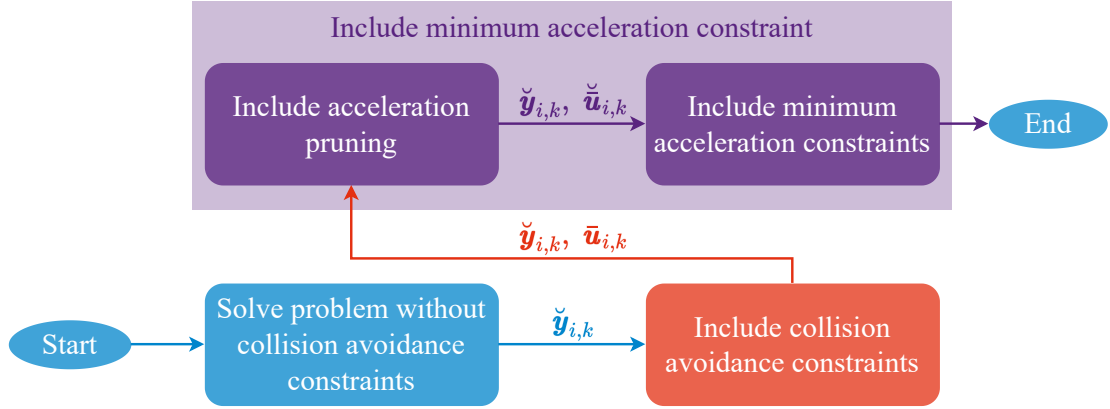


Figure 4.26: Graphical representation of the SCP scheme with minimum acceleration constraint

It is important to note that, when the minimum acceleration constraints are included, the collision avoidance constraints are also taken into account. The results from the previous iteration are used as an initial guess for the affine relaxation of the collision avoidance constraints.

The pruning process is formally introduced in Algorithm 1, where the sets of indices  $\mathcal{K}_i$  are identified for each deputy. These indices correspond to the time steps with the lowest acceleration values. The cardinality of each  $\mathcal{K}_i$  is determined based on the user-defined pruning factors,  $p_i$ , as well as the mean value of the acceleration profile for the deputy satellite in question, as illustrated by Algorithm 1.

---

**Algorithm 1:** Acceleration pruning

---

```
1 Solve the problem without the minimum thrust constraint; Obtain the solution  $\mathbf{u}_{i,k}$ 
2 foreach  $i \in \mathcal{I}$  do
3    $\mathcal{U}_i \leftarrow \{\|\mathbf{u}_{i,k}\|_2 : k \in \mathcal{K}_f\}$ 
4    $u_{i,\text{mean}} \leftarrow \text{mean}(\mathcal{U}_i)$ 
5   if  $(u_{i,\text{mean}} \leq 1.5 u_{i,\text{min}})$  then
6      $n_i \leftarrow \lfloor p_i \cdot \left(1 - \frac{u_{i,\text{mean}}}{u_{i,\text{min}}} \cdot |\mathcal{K}_f|\right) \rfloor$ 
7     if  $(n_i > |\mathcal{K}_f| - 2)$  then
8        $n_i \leftarrow |\mathcal{K}_f| - 2$ 
9      $\mathcal{K}_i \leftarrow \text{Set of } (k) \text{ indices corresponding to the lowest } n_i \text{ values in } \mathcal{U}_i$ 
10 end
```

---

Note that in Algorithm 1,  $|\mathcal{K}_f|$  represents the cardinality of the set  $\mathcal{K}_f$ , while  $p_i$  is the pruning factor of the  $i^{\text{th}}$  deputy. Moreover, the condition in line 7 is set to guarantee that at least two acceleration instances are not pruned. This is important because maintaining at least two control instances can be necessary since the action taken by the first acceleration instance, e.g., adjusting the value of  $\delta a$  in order to achieve a desired  $\delta \lambda$  behavior, might need to be corrected at a later point in the profile. Considering the logic described in Fig. 4.26, it is important to clarify that the identification of pruned instances, as detailed in Algorithm 1, is performed twice during the process. Initially, it is conceivably executed before the "Include acceleration pruning" block to identify the indices  $\mathcal{K}_i$  where the acceleration is constrained to zero. Subsequently, it is executed again prior to the "Include minimum acceleration constraints" block to ensure that the appropriate instances are pruned at this stage. It should be noted that the pruned instances in these two blocks are not necessarily the same.

From this point onward, all modifications to the guidance problem will focus exclusively on the SOCP formulation. This decision is justified by its proven fuel efficiency and computational effectiveness, as outlined in Section 4.5. Accordingly, the SOCP formulation of the guidance problem, incorporating the acceleration pruning constraints, is presented with the modifications to Problem 4.4 marked in red as follows:

**Problem 4.7** (SOCP formulation with acceleration pruning).

$$\min_{\mathbf{Y}, \bar{\mathbf{U}}, \Gamma} \quad \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} (\Delta t_k \Gamma_{i,k})$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.58)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{K}, \quad (4.59)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \quad \forall (i, k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \quad (4.60)$$

$$\|\bar{\mathbf{u}}_{i,k}\|_2 \leq \Gamma_{i,k} \quad \forall i \in \mathcal{I}, \quad \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.61)$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \quad \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.62)$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA} \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 \quad \forall i, j \in \mathcal{I}, \quad i \neq j, \quad \forall k \in \mathcal{K}, \quad (4.63)$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA} \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{K}. \quad (4.64)$$

As indicated by Fig. 4.26, the solution of Problem 4.7 is used as an initial guess for the minimum thrust constraint in Eq. (4.57). The full guidance problem, including the minimum thrust constraint, is written in the SOCP form with the modifications to Problem 4.7 marked in red as follows:

**Problem 4.8** (SOCP formulation with minimum acceleration constraints).

$$\min_{\mathbf{Y}, \bar{\mathbf{U}}, \Gamma} \quad \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} (\Delta t_k \Gamma_{i,k})$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad \mathbf{y}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \quad \forall i \in \mathcal{I}, \quad (4.65)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{K}, \quad (4.66)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \quad \forall (i, k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \quad (4.67)$$

$$\check{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \geq \|\check{\mathbf{u}}_{i,k}\|_2 a_c u_{i,min} \quad \forall i \in \mathcal{I}, \quad \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.68)$$

$$\|\bar{\mathbf{u}}_{i,k}\|_2 \leq \Gamma_{i,k} \quad \forall i \in \mathcal{I}, \quad \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.69)$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.70)$$

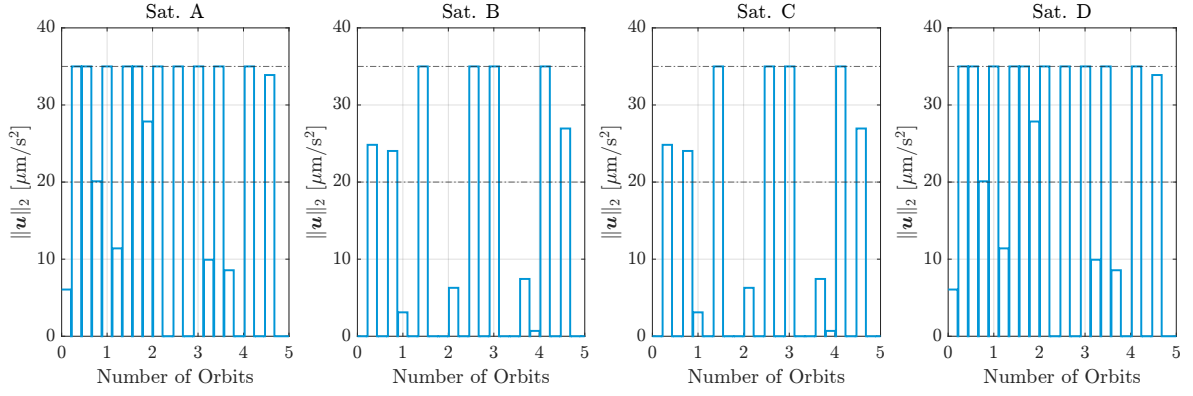
$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq R_{CA} \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 \quad \forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K}, \quad (4.71)$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq R_{CA} \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \quad (4.72)$$

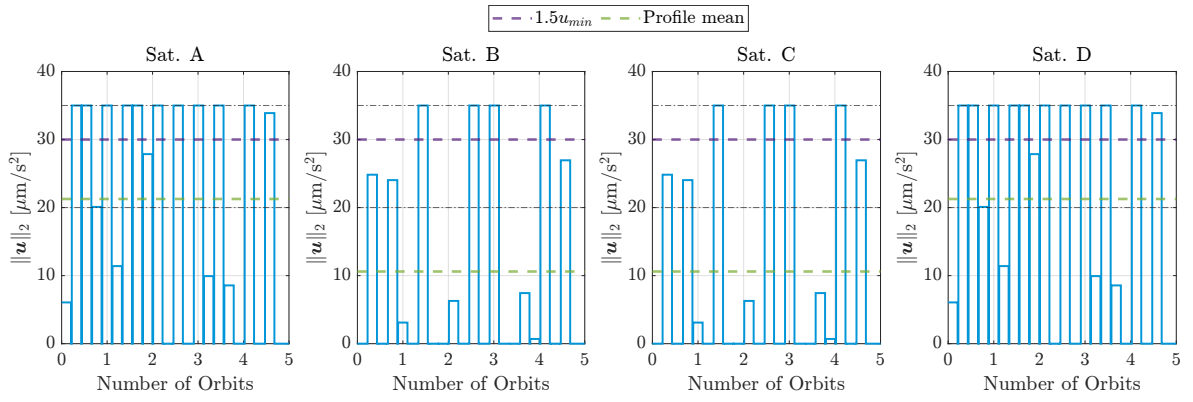
Although only the **SOCP** formulation is presented here, the relaxed minimum acceleration constraint can be applied to all other formulations, namely **QCQP**, **QP**, and **LP**, exactly as written in Eq. (4.68). Since this constraint is linear, adding it to any of the proposed formulations does not alter their structure; for instance, a **QCQP** problem will remain a **QCQP**.

It is important to emphasize that Problem 4.8 is solved sequentially, following the process depicted in Fig. 4.26. By applying this sequence to Reconfiguration 3 from Appendix D with  $u_{i,\min} = u_{\min} = 20 \mu\text{m/s}^2$  and  $p_i = p = 1 \forall i \in \mathcal{I}$ , a solution which satisfies all the constraints is obtained. The results of this simulation is used to further illustrate the mechanism according to which the pruning process is applied. The evolution of the acceleration profile across the **SCP** iterations is demonstrated in Fig. 4.27.

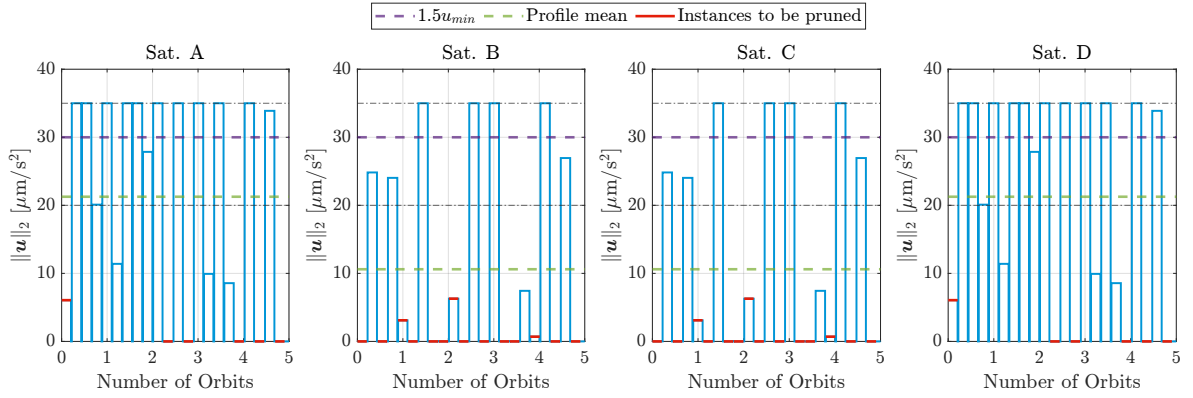




(a) End of inclusion of the collision avoidance constraints - End of SCP iteration 1



(b) Definition of the mean and the threshold for each profile - End of SCP iteration 1



(c) Determination of the instances to be pruned - End of SCP iteration 1

Figure 4.27: Evolution of the acceleration profiles for all deputies across Reconfiguration 3, illustrating the pruning process (Page 1)

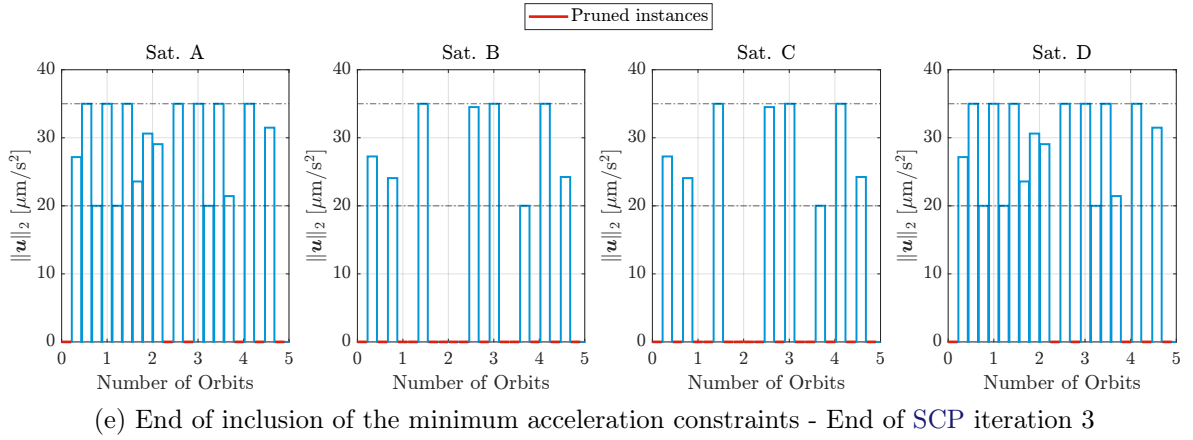
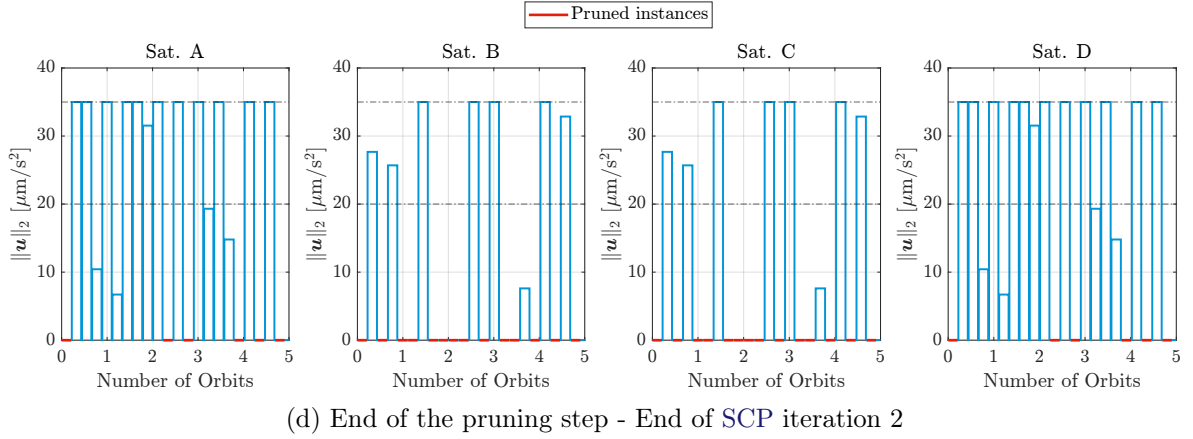


Figure 4.27: Evolution of the acceleration profiles for all deputies across Reconfiguration 3, illustrating the pruning process (Page 2)

In Fig. 4.27a, the control profiles after incorporating the collision avoidance constraints (at the end of the first SCP iteration, following the zeroth iteration) are examined to determine whether pruning should be applied to a particular profile. The investigation process begins by calculating the mean of each profile ( $\text{mean}(\mathcal{U}_i)$ ) and defining a threshold (set to  $1.5 u_{i,\min}$  in Algorithm 1). If the mean of the profile falls below this threshold, the pruning process is initiated for that profile. The computed mean values and thresholds are depicted in Fig. 4.27b. When pruning is triggered for a specific satellite control profile, the acceleration instances to be pruned are identified according to Algorithm 1. These instances are highlighted in red in Fig. 4.27c. Up to this stage, the problem incorporating the acceleration

pruning constraints has not yet been solved. After solving it, the resulting control profiles are shown in Fig. 4.27d, where the pruned instances are observed to have accelerations equal to zero. Finally, the problem that includes both pruning and minimum acceleration constraints is solved, and the resulting profiles are illustrated in Fig. 4.27. It is evident that the  $L_2$ -norms of the control acceleration either lie within the range  $[u_{\min}, u_{\max}]$  or are set to zero.

In this particular reconfiguration, three SCP iterations were required after the zeroth iteration. These include one iteration for collision avoidance, one for acceleration pruning, and a final iteration for incorporating the minimum acceleration constraints. It is worth noting that this four-iteration process is typical for the guidance scheme shown in Fig. 4.26. Generally, the step involving collision avoidance constraints is the only one that may require multiple SCP iterations, while the inclusion of minimum acceleration constraints strictly requires two iterations; one for including the pruning and another for incorporating the relaxed minimum acceleration constraints. Notably, the required Delta-V for solving Problem 4.8 for Reconfiguration 3 is 1.6883 m/s, which is only 0.3% larger than the Delta-V required for solving the guidance problem without the minimum acceleration constraints, which requires 1.6827 m/s.

While solving Problem 4.8 yields a feasible solution for Reconfiguration 3, there is no guarantee that it will always be feasible for other reconfiguration tasks. Unlike Problems 4.2 through 4.6, which are guaranteed to provide a feasible solution given a sufficiently large maneuver time, Problem 4.8 is heavily dependent on the choice of linearization directions  $\check{\mathbf{u}}_{i,k}$ . If these directions are poorly selected, the problem is prone to infeasibility, even when the allowed maneuver time is extended. The following section discusses techniques to mitigate the risk of infeasibility.

## 4.7 Infeasibility handling

To support autonomy of the formation control tasks, the proposed guidance strategy is meant to be solved recurrently within an MPC scheme, which makes use of the available sensor

measurements to perform the control task in an efficient manner while accounting for the disturbances and model uncertainties. It does so through optimizing the trajectory as well as the control profile over a period of time (horizon), then applies the optimized control profile up to a certain point (usually only the first control step is utilized). The details of two distinct MPC strategies are discussed later in Chapter 5.

One major risk for an MPC is arriving to a point where the guidance optimization problem, on which the MPC heavily depends, is infeasible. The mitigation of this risk is handled in our context inside the guidance layer, where all the constraints that may lead to infeasibility are softened in order to ensure that a feasible solution is always available for the MPC. This comes understandably at the cost of slight violations of the constraints if the problem is not initially feasible.

Before introducing methods to soften the constraints, those which are expected to lead to infeasibility must be first identified. As discussed earlier, the relaxed minimum thrust constraint, in Eq. (4.68), is responsible for potential infeasibility because of its dependence on the initial guess,  $\check{\mathbf{u}}_{i,k}$ . Indeed, a good guess for  $\check{\mathbf{u}}_{i,k}$  is obtained through the acceleration pruning process, nonetheless, infeasibility could still be expected mainly because the optimizer is not able to find a solution that satisfies the final state constraint, in Eq. (4.65), while respecting the dynamics constraints in Eq. (4.66) and at the same time being constrained by the thrust directions imposed by the constraints in Eq. (4.68). It is for this reason that both, the final state constraints (Eq. (4.65)), and the relaxed minimum acceleration constraints (Eq. (4.68)), are softened to reduce the risk of infeasibility. The process of softening the final state constraints involves broadening the original objective of these constraints. Initially, the goal was to align the final state of the optimized profile with the desired one. The revised objective, however, is for the state profile to approximate a user-defined profile at specific user-defined time instances. This approach is supported by the fact that many MPC schemes, not to mention the Fixed-Horizon MPC (FHMPC) that will be introduced in Chapter 5, rely on tracking a user-defined state profile which typically results from solving the trajectory optimization

problem for the whole maneuver. In this setting, the final state constraints in Eq. (4.65) should be anyway softened to address this tracking objective. Before the softening is formally introduced, other constraints are identified to be potential sources of infeasibility. Namely, the collision avoidance constraints. These constraints could induce infeasibility in a variety of ways, one of which is when a satellite is initially situated inside the **KOZ** of another satellite, which may happen during operation of the closed control loop due to a variety of reasons including unmodeled nonlinearities, navigation inaccuracies, or actuation errors. Indeed, the initial state is not controlled, and if the initial positions of the satellites are not collision-free, the whole guidance problem will be doomed infeasible. It is for this reason that the guaranteed feasible trajectory optimization problem will not constrain the first step to be collision-free. Another way the collision avoidance constraints could induce infeasibility is, again, when the initial states are not collision-free, and the maximum control acceleration is not enough to get the satellite that violates the collision avoidance criteria out of the **KOZ** within the given time period before the following one or more time steps. To address these concerns, the collision avoidance constraint is also softened for all the time steps to ensure feasibility.

The **SOCP** formulation, incorporating the minimum acceleration constraints, is softened and rewritten in Problem 4.9 with the modifications to Problem 4.8 marked in red. The methodology enabling this softening, along with approaches to soften the three other guidance reformulations, is described in Appendix F.

**Problem 4.9** (Softened **SOCP** formulation).

$$\begin{aligned}
& \min_{\mathbf{Y}, \mathbf{U}, \mathbf{\Gamma}, w, \mathcal{B}, \Upsilon} \quad \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} (\Delta t_k \Gamma_{i,k}) + w + \\
& \quad q_{umin} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} v_{i,k} + q_{ca} \sum_{i \in \mathcal{I}} \sum_{\substack{j \in \mathcal{I} \setminus \{i\} \\ \cup \{0\}}} \sum_{k \in \mathcal{K} \setminus \{0\}} \beta_{ij,k} \\
& \text{subject to,} \\
& \mathbf{y}_{i,0} = \mathbf{y}_{i,0} \quad \forall i \in \mathcal{I},
\end{aligned} \tag{4.73}$$

$$\left\| \left[ \left[ \sqrt{\mathbf{Q}} (\bar{\mathbf{y}}_{i,k} - \mathbf{y}_{i,k}) \right]_{i \in \mathcal{I}}^h \right]_{k \in \bar{\mathcal{K}}}^v \right\|_F \leq w, \quad (4.74)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (4.75)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \quad (4.76)$$

$$\check{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \geq \left\| \check{\mathbf{u}}_{i,k} \right\|_2 (a_c u_{i,min} - v_{i,k}) \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i) \quad (4.77)$$

$$0 \leq v_{i,k} \leq v_{max} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.78)$$

$$\|\bar{\mathbf{u}}_{i,k}\|_{\mathbf{R}} \leq \Gamma_{i,k} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.79)$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.80)$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq \left\| \mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k}) \right\|_2 (R_{CA} - \beta_{ij,k}) \quad (4.81)$$

$$\forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\},$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq \left\| \mathbf{T}_k \check{\mathbf{y}}_{i,k} \right\|_2 (R_{CA} - \beta_{i0,k}) \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \setminus \{0\}, \quad (4.82)$$

$$0 \leq \beta_{ij,k} \leq \beta_{max} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{I} \cup \{0\}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\}, \quad (4.83)$$

where  $w$  is a slack variable,  $q_{\min} \in \mathbb{R}^+$  and  $q_{ca} \in \mathbb{R}^+$  are positive weighting factors,  $v_{i,k}$  are slack variables that represent violations to the relaxed minimum acceleration constraints, and  $\beta_{ij,k}$  and  $\beta_{i0,k}$  are slack variables that represent violations to the affine collision avoidance constraints. These slack variables are collated in the sets  $\Upsilon$  and  $\mathcal{B}$ , respectively, which are defined as,

$$\Upsilon = \{v_{i,k} : i \in \mathcal{I}, (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i)\}, \quad (4.84)$$

$$\mathcal{B} = \{\beta_{ij,k} : i \in \mathcal{I}, j \in \mathcal{I} \cup \{0\}, i \neq j, k \in \mathcal{K} \setminus \{0\}\}.$$

Moreover,  $\bar{\mathcal{K}}$  is the set of indices for the time instances on which the user requires the state of the deputies to track certain reference states,  $\bar{\mathbf{y}}_{i,k} \in \mathbb{R}^6$  is the user-defined reference state required to be tracked by the  $i^{\text{th}}$  deputy at the time step  $t_k$ ,  $\|\cdot\|_F$  is the Frobenius norm operator,  $\mathbf{Q} \in \mathbb{R}^{6 \times 6}$  and  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  are diagonal positive semi-definite weighting matrices relating respectively to the minimization of the state deviation from the reference one and to the minimization of the total control effort (Delta-V). Note that  $\|\bar{\mathbf{u}}_{i,k}\|_{\mathbf{R}} = \left\| \sqrt{\mathbf{R}} \bar{\mathbf{u}}_{i,k} \right\|_2$ , where the diagonal entries of  $\mathbf{R}$  have to be strictly greater than or equal to unity, so that

the acceleration could never exceed the maximum allowable one. Furthermore, since both  $\mathbf{Q}$  and  $\mathbf{R}$  are diagonal matrices, their matrix square roots are equal to their element-wise square roots. Furthermore,  $[(\cdot)]_{i_{\text{indices}}}^h$  and  $[(\cdot)]_{i_{\text{indices}}}^v$  are defined as the horizontal and the vertical concatenation operators, respectively, such that  $[a_i]_{i \in \{1,2,\dots,n\}}^h = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}$  and  $[a_i]_{i \in \{1,2,\dots,n\}}^v = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}^\top$ .

Although the set  $\mathcal{B}$  collates the values corresponding to violations in the affine relaxation of the collision avoidance constraints, non-zero values of the elements of  $\mathcal{B}$  do not necessarily indicate violations to the original collision avoidance constraints, i.e., keeping one satellite outside the  $\text{KOZ}$  of another. This concept is illustrated visually in Fig. 4.28 (assuming the motion is taking place in 2D plane, without loss of generality) where it can be seen that while the affine relaxation of the collision avoidance constraint is allowed to be violated, the original constraint is still respected. The same can be said about violations to the relaxed minimum acceleration constraints. A non-zero  $v_{i,k}$  does not necessarily imply that the control acceleration required by the  $i^{\text{th}}$  deputy at time  $t_k$  is below the minimum acceleration threshold. It means, however, that one of the relaxed minimum acceleration constraint is violated, which might or might not lead to a violation in the original constraint, since the relaxation is overly restrictive.

Problem 4.9 is written in the epigraph form, which behaves such that,

$$\begin{aligned} w &= \left\| \left[ \left[ \sqrt{\mathbf{Q}} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \right]_{i \in \mathcal{I}}^h \right]_{k \in \bar{\mathcal{K}}}^v \right\|_F, \\ &= \sqrt{\sum_{i \in \mathcal{I}} \sum_{k \in \bar{\mathcal{K}}} ((\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k})^\top \mathbf{Q} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}))}. \end{aligned} \quad (4.85)$$

Note that Problem 4.9 reduces to a softened version of Problem 4.8 if and only if  $\bar{\mathcal{K}} = \{m+1\}$  and  $\bar{\mathbf{y}}_{i,m+1} = \bar{\mathbf{y}}_{i,f}$ . Otherwise, the two problems become unrelated. It is worth noting that the matrices  $\mathbf{R}$  and  $\mathbf{Q}$  can be adjusted to obtain specific desired solutions without changing the structure of the problem. These desired solutions may include ones that avoid using the radial acceleration component or those that track a given  $\text{ROE}$  profile, excluding,

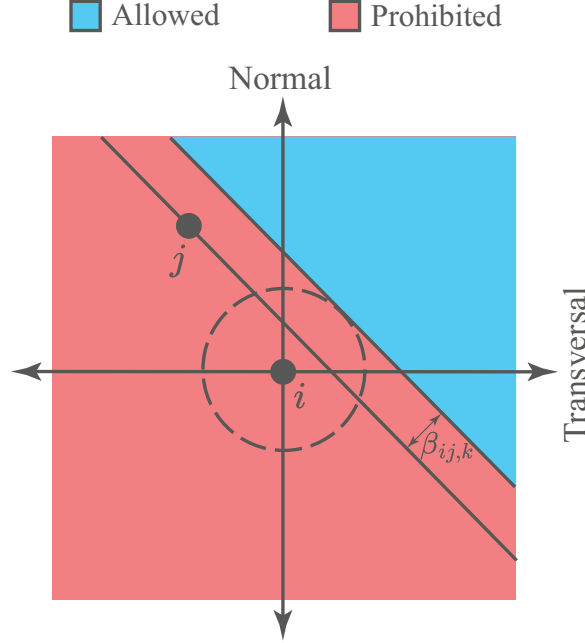


Figure 4.28: Violation of the affine relaxation of the collision avoidance constraint

for example, the relative mean argument of latitude. A third example is when the operator requires more emphasis on minimizing the errors of specific ROE elements, such as the relative semi-major axis, for being more critical than others.

In order to showcase the importance of softening the hard constraints, the SOCP guidance problem in their two settings, i.e., the hard- and the soft-constrained ones, in Problems 4.8 and 4.9 respectively, are run over two of the reconfiguration scenarios for comparison purposes. The initial and final states for the two reconfiguration scenarios are provided in Appendix D, with only the Reconfigurations 2 and 3 being used in this comparison. Once again, the deputy spacecraft are assumed identical in both scenarios, and hence  $u_{i,\max} = u_{\max}$ ,  $u_{i,\min} = u_{\min}$ ,  $p_i = p \ \forall i \in \mathcal{I}$ . Furthermore, since the maneuver duration,  $t_f - t_0$ , is defined for the two scenarios, the user-defined time vector,  $\mathbf{t}$ , is provided for both maneuvers such that the duration of the thrust arcs are identical throughout the maneuver. This also applies to the durations of the coast arcs. Formally,  $T_{f,l} = T_f$ ,  $T_{n,l} = T_n$ ,  $\forall l \in \mathcal{L}$ , where  $\mathcal{L} = \{1, 2, \dots, (m+1)/2\}$  is the list of indices of the different control cycles (refer to Fig. 4.1).



It is important to emphasize that the soft-constrained problem is solved with  $\bar{\mathcal{K}} = \{m + 1\}$  and with  $\bar{\mathbf{y}}_{i,m+1} = \bar{\mathbf{y}}_{i,f} \forall i \in \mathcal{I}$ . A full list of the parameters used in solving the guidance problems for both scenarios are reported in Table 4.6.

Table 4.6: Parameters used in the comparison experiment between the soft- and hard-constrained guidance problems

Parameter	Value	Parameter	Value
$T_f$ [orbits]	0.2	$T_n$ [s]	100
$u_{\max}$ [ $\mu\text{m}/\text{s}^2$ ]	35	$u_{\min}$ [ $\mu\text{m}/\text{s}^2$ ]	20
$R_{CA}$ [m]	100	$p$ [%]	100
$\mathbf{Q}$ [-]	$\mathbf{I}_3$	$\mathbf{R}$ [-]	$\mathbf{I}_6$
$q_{\text{umin}}$ [-]	$10^{-2}$	$v_{\max}$ [ $\text{m}^2/\text{s}^2$ ]	$\infty$
$q_{ca}$ [-]	1	$\beta_{\max}$ [m]	10

Reconfiguration 2 was chosen to showcase the significance of transforming Problem 4.8 into Problem 4.9. Specifically, when Problem 4.8 is solved according to the logic of Fig. 4.26 over Reconfiguration 2, it fails to produce a feasible solution, while it can be solved with a tolerable level of constraint violations when the formulation of Problem 4.9 is used. This makes Reconfiguration 2 an excellent example of how the softened formulation is beneficial. Reconfiguration 3, on the other hand, can be solved by both, the hard-constrained formulation as well as the soft-constrained one, and hence forms a basis for comparing the two methodologies. The results of the comparison experiment are summarized in Table 4.7 where the number of variables, the number of constraints, the total required Delta-V for the maneuver, the maximum final state error, and the maximum violation of the relaxed minimum acceleration constraints as well as that of the relaxed collision avoidance constraints across all time steps and for all satellites are reported. In the table, a column of only dashes (i.e., -) indicates that no solution could be found for the adopted formulation over a certain reconfiguration scenario.

The results in Table 4.7 indicate that the hard-constrained problem formulation did indeed fail to produce a feasible solution for Reconfiguration 2. The effectiveness of introducing slack variables becomes particularly evident in Reconfiguration 2, as the softened problem suc-

Table 4.7: Comparison of the soft- and the hard-constrained problems

	Reconfiguration 2		Reconfiguration 3	
	Hard	Soft	Hard	Soft
Variables	-	3547	1696	2185
Constraints	-	2719	1632	1633
$\Delta V$ [m/s]	-	2.77	1.69	1.69
$\max \left( \ \mathbf{y}_{i,f} - \bar{\mathbf{y}}_{i,f}\ _2 \right)$ [m]	-	0.00	0.00	0.00
$\max(\Upsilon)$ [m <sup>2</sup> /s <sup>2</sup> ]	-	0.0	-	0.0
$\max(\mathcal{B})$ [m]	-	0.0	-	0.0

cessfully identified a solution that satisfied all constraints, including the final state, collision avoidance, and minimum acceleration requirements. This suggests that the hard-constrained problem for Reconfiguration 2 was "almost" feasible, with only a minimal constraint violation needed to yield a valid solution. While it might be possible for the hard-constrained formulation to find a feasible solution for Reconfiguration 2 by adjusting parameters such as the pruning factor, such an approach is unsuitable for autonomous implementation due to its reliance on manual tuning.

The results of Reconfiguration 3 are interesting since the two variants of the problems, i.e., soft- and hard-constrained, were able to provide feasible solutions for it. Generally, the results from a soft-constrained problem are close to those of the hard-constrained one, but they are not necessarily identical. This is because, as shown in Fig. 4.26, a problem incorporating both collision avoidance and minimum acceleration constraints is typically solved in four iterations. The solution of each iteration, except the zeroth, depends on the solution from the previous iteration. Consequently, the "final result" of interest is the outcome of the last iteration, which relies on the preceding solutions. Given the complexity of the guidance problem, it is not guaranteed that the solutions from each iteration in the soft- and hard-constrained settings will match. Nevertheless, it can be observed that the two variants produce identical solutions for Reconfiguration 3, although this may not be the case for a general reconfiguration scenario.

The comparison between the soft- and hard-constrained problems, as presented in Table 4.7, was designed to showcase the softened guidance scheme’s capability to handle infeasibility. This issue is particularly relevant in closed-loop operations where guidance strategies are repeatedly employed, as will be discussed in Chapter 5. One of the key challenges in this context is ensuring collision-free trajectories throughout the maneuver. The guidance strategies developed so far guarantee collision avoidance only at discrete sampling times,  $t_k \forall k \in \mathcal{K}$ , leaving the safety of the relative trajectories between these sampling times uncertain. Using a control cycle duration of  $T_f = 0.2$  orbits poses a potential risk for closed-loop implementation. As the interval between successive sampling times increases, the probability of a satellite entering the KOZ of another satellite becomes significantly higher. To address this issue, the thrust arc length within each control cycle is reduced here, and in the discussions to follow, to  $T_f = 0.05$  orbits, roughly equivalent to 300 seconds for LEO satellites. This adjustment ensures finer control authority and minimizes the risk of collisions during the unmonitored intervals between sampling times. To evaluate the impact of this refined setting on the guidance layer, the comparison experiment between the hard- and soft-constrained guidance problems was repeated. The results, summarized in Table 4.8, reveal a notable increase in the number of variables and constraints, thereby significantly raising the computational complexity. Despite this added complexity, the shorter thrust arcs allowed the hard-constrained problem to achieve a feasible solution for Reconfiguration 2, demonstrating the benefits of enhanced control precision in this configuration.

## 4.8 Distributed guidance

Up to this point, all the presented guidance schemes have relied on a centralized processing unit, namely the chief spacecraft, to compute the guidance profiles. However, this centralized approach is not scalable, as the computational time required to solve the problem increases exponentially with the number of deputies, as demonstrated in [14]. To address this limitation, the guidance problem must be reformulated to allow parallel computation by each

Table 4.8: Comparison of the soft- and the hard-constrained problems

	Reconfiguration 2		Reconfiguration 3	
	Hard	Soft	Hard	Soft
Variables	8244	11554	5496	7115
Constraints	8925	8926	5362	5363
$\Delta V$ [m/s]	2.58	2.68	1.58	1.58
$\max \left( \ \mathbf{y}_{i,f} - \bar{\mathbf{y}}_{i,f}\ _2 \right)$ [m]	0.00	0.00	0.00	0.00
$\max(\Upsilon)$ [m <sup>2</sup> /s <sup>2</sup> ]	-	0.0	-	0.0
$\max(\mathcal{B})$ [m]	-	0.0	-	0.0

deputy spacecraft.

A closer inspection of Problem 4.9 reveals that the primary obstacle preventing a distributed solution lies in the constraints in Eq. (4.81). Specifically, this issue arises because the state profile of the  $j^{\text{th}}$  deputy at the current SCP iteration,  $\mathbf{y}_{j,k}$ , is not directly accessible onboard the  $i^{\text{th}}$  satellite. To resolve this, the state profile of the  $j^{\text{th}}$  deputy from the previous SCP iteration,  $\check{\mathbf{y}}_{j,k}$ , is used instead. With this modification, Problem 4.9 can be reformulated into a distributed framework and can be rewritten for the  $i^{\text{th}}$  deputy as follows:

**Problem 4.10** (Softened distributed SOCP formulation).

$$\min_{\mathbf{Y}_i, \bar{\mathbf{U}}_i, \Gamma_i, w, \mathcal{B}_i, \Upsilon_i} \frac{1}{a_c} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} (\Delta t_k \Gamma_{i,k}) + w + q_{umin} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} v_{i,k} + q_{ca} \sum_{j \in \mathcal{I} \setminus \{i\}} \sum_{k \in \mathcal{K} \setminus \{0\}} \beta_{ij,k}$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0}, \quad (4.86)$$

$$\left\| \left[ \sqrt{\mathbf{Q}} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \right]_{k \in \bar{\mathcal{K}}}^v \right\|_2 \leq w, \quad (4.87)$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall k \in \mathcal{K}, \quad (4.88)$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall (i, k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \quad (4.89)$$

$$\frac{\check{\bar{\mathbf{u}}}_{i,k}^\top}{\|\check{\bar{\mathbf{u}}}_{i,k}\|_2} \bar{\mathbf{u}}_{i,k} \geq a_c u_{i,min} - v_{i,k} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.90)$$

$$0 \leq v_{i,k} \leq v_{max} \quad \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.91)$$

$$\|\bar{\mathbf{u}}_{i,k}\|_{\mathbf{R}} \leq \Gamma_{i,k} \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.92)$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (4.93)$$

$$\frac{(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top}{\|\mathbf{T}_k(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2} \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \check{\mathbf{y}}_{j,k}) \geq R_{CA} - \beta_{ij,k} \quad \forall j \in \mathcal{I} \setminus \{i\}, \forall k \in \mathcal{K} \setminus \{0\}, \quad (4.94)$$

$$0 \leq \beta_{ij,k} \leq \beta_{max} \quad \forall j \in \mathcal{I} \setminus \{i\}, \forall k \in \mathcal{K} \setminus \{0\}, \quad (4.95)$$

where,

$$\begin{aligned} \mathbf{Y}_i &= [\mathbf{y}_{i,k}]_{k \in \mathcal{K} \cup \{m+1\}}^h, \\ \bar{\mathbf{U}} &= [\bar{\mathbf{u}}_{i,k}]_{k \in \mathcal{K}}^h, \\ \mathbf{\Gamma}_i &= [\Gamma_{i,k}]_{k \in \mathcal{K}_f}^h, \\ \Upsilon_i &= \{v_{i,k} : (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i)\}, \\ \mathcal{B}_i &= \{\beta_{ij,k} : j \in \mathcal{I} \setminus \{i\}, k \in \mathcal{K} \setminus \{0\}\}. \end{aligned} \quad (4.96)$$

It is important to highlight that the chief-deputy collision avoidance constraints were excluded from the distributed guidance strategy. This is because the distributed approach treats the chief spacecraft as a virtual point, eliminating the need for a central processing unit since guidance calculations are performed onboard each deputy. Despite being considered a virtual point, the chief is assumed to orbit the Earth considering the effect of the second zonal harmonic ( $J_2$ ) since the dynamics in Eq. (2.50) are developed with the second zonal harmonic effect taken into account.

Similar to Problem 4.9, Problem 4.10 is solved using a SCP scheme, where most of the convex sub-problems are solved simultaneously for all deputies. However, not all sub-problems can be solved in parallel due to the collision avoidance constraints. Each deputy is only aware of the optimized trajectories of the other deputies from the previous SCP step and not the current one. As a result, the computed trajectories are not guaranteed to be collision-free, even after multiple iterations. If all calculations are performed in parallel, a *ping-pong* situa-

tion may arise. To explain, consider Satellites A and B solving their trajectory optimization problems individually and in parallel. The resulting trajectories may be unsafe. If they re-solve their problems in parallel, Satellite A computes a new trajectory that is safe relative to the old trajectory of Satellite B, but not the new trajectory. Similarly, Satellite B computes a trajectory that is safe relative to the old trajectory of Satellite A. This process can repeat indefinitely, resulting in oscillations between unsafe trajectories. To mitigate this issue, some serial computations are introduced. In this approach, all satellites temporarily fix their trajectories while one satellite optimizes its path, considering the fixed trajectories of all others. This sequential process ensures that the updated trajectory is safe relative to the current trajectories of all other satellites. If the overall configuration remains unsafe, the process is repeated iteratively. The execution logic for this distributed SCP scheme is illustrated in Fig. 4.29, where the mechanism ensuring collision-free final guidance profiles is also detailed.

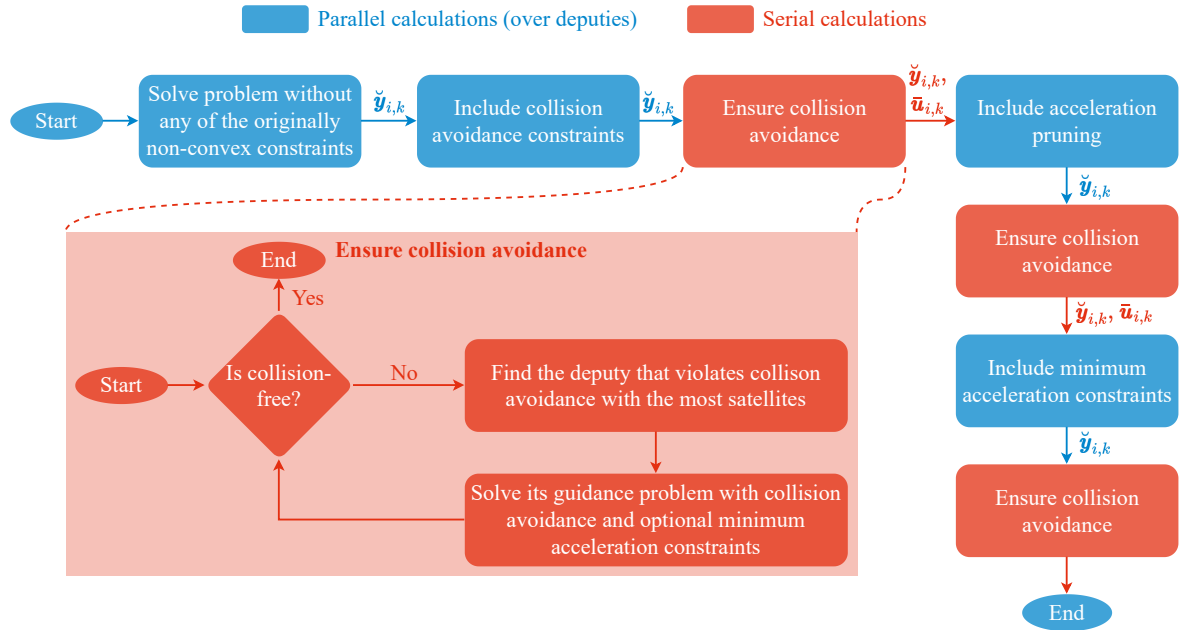


Figure 4.29: Graphical representation of the SCP scheme in the distributed setting

Clearly, the "Ensure collision avoidance cons" block in Fig. 4.29 addresses the challenge of avoiding the ping-pong effect by properly scheduling the calculations for each satellite to guarantee safe relative trajectories. Placing this block as the final step in the execution logic

is natural, as the final trajectories must be collision-free. However, this block is positioned after the collision avoidance and pruning blocks to improve the overall problem’s feasibility.

It is worth mentioning that while Problem 4.10 incorporates the collision avoidance, minimum acceleration, and final state constraints in their softened forms, there is a corresponding formulation where these constraints are strictly hard. However, this hard-constrained version is not explicitly discussed here for the sake of brevity and is left as an exercise for the reader. To ensure consistency with the centralized guidance schemes, the experiment comparing the hard- and soft-constrained formulations, with results presented in Table 4.8, is conducted for the distributed case as well. The outcomes of this distributed comparison are summarized in Table 4.9.

Table 4.9: Comparison of the soft- and the hard-constrained problems in the distributed setting

	Reconfiguration 2		Reconfiguration 3	
	Hard	Soft	Hard	Soft
Variables	1374	2145.67	1374	1851.5
Constraints	1703.5	1707.67	1412.5	1413.5
$\Delta V$ [m/s]	2.59	2.76	1.58	1.58
$\max \left( \ \mathbf{y}_{i,f} - \bar{\mathbf{y}}_{i,f}\ _2 \right)$ [m]	0.00	0.00	0.00	0.00
$\max (\Upsilon)$ [m <sup>2</sup> /s <sup>2</sup> ]	-	0.0	-	0.0
$\max (\mathcal{B})$ [m]	-	0.0	-	0.0

The reported numbers of variables and constraints in Table 4.9 represent the averages across all deputies for the final SCP iteration. Notably, the average values for the two reconfigurations are generally not whole numbers, indicating that the deputies did not have identical numbers of variables and constraints at the last SCP iteration, despite solving the same guidance problem. This discrepancy arises because the cardinality of  $\mathcal{K}_i$  varies among the deputies. In other words, differences in the number of pruned time instances for each satellite result in varying numbers of constraints. Consequently, the number of variables differs due to variations in the number of slack variables used to soften the minimum acceleration

constraints,  $|\Upsilon_i|$ , which might be unique to each satellite.

The results presented in Table 4.9 highlight the significant reduction in the number of variables and constraints when compared to the centralized case (see Table 4.8). Indeed, the number of variables and constraints increases linearly with the number of deputies, driven solely by the inclusion of collision avoidance constraints. Another notable observation, evident from the comparison of results in Tables 4.8 and 4.9, is that the distributed guidance strategy generally necessitates a higher Delta-V compared to the centralized strategy. Furthermore, as claimed earlier, the softened problem generally exhibits a distinct guidance profile compared to its hard-constrained counterpart, although the Delta-V requirements for the two trajectories do not differ substantially. The state profiles in the ROE space, corresponding to the solutions of both the hard- and soft-constrained distributed problems for Reconfiguration 3, are depicted in Fig. 4.30, illustrating the close resemblance between the trajectories followed under the two settings, but also showing that the profiles of the hard- and soft-constrained problems are not exactly the same.

To highlight the capabilities of the distributed approaches, an experiment was conducted where the number of deputies was varied from 1 to 20, while the formation was tasked to perform a Coplanar-to-PCO maneuver. In the initial configuration, the distance between every two consecutive deputies was set to 200 m, and the radius of the final PCO was set to 500 m. For a fair comparison, each of the 20 reconfigurations was allowed 10 orbits to complete.

The MOSEK solver was executed 10 times on Problems 4.9 and 4.10 for each of the 20 reconfiguration scenarios. The average solve time over the 10 runs is plotted against the number of deputies in Fig. 4.31. For the distributed case, the reported solve times correspond to the maximum solve time across all satellites, as calculations are performed in parallel.

It is evident, and unsurprising, that the solve time grows exponentially with the number of deputies in the centralized setting. Conversely, in the distributed setting, the solve time increases linearly. The observed increase in solve time for the distributed approach can be attributed to the inclusion of collision avoidance constraints, which requires additional time



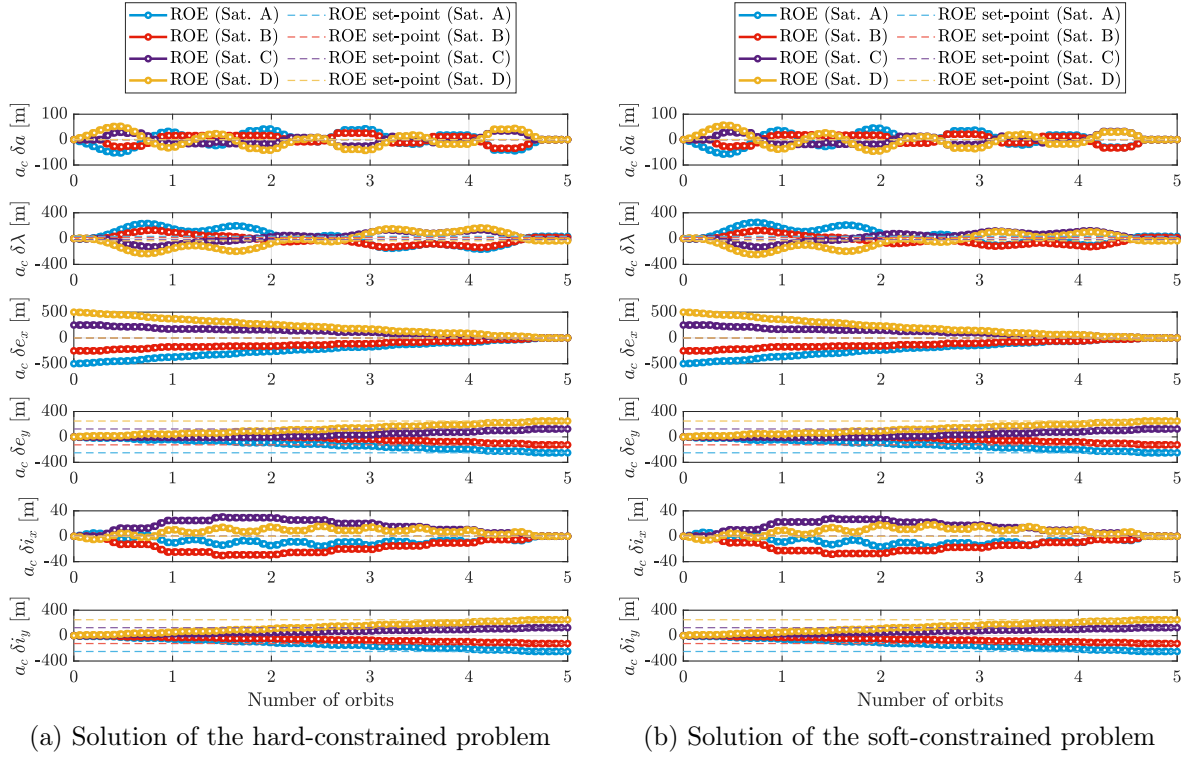


Figure 4.30: ROE profiles resulting from solving the distributed guidance problems over Reconfiguration 3

for the serial calculations in the "Ensure collision avoidance" blocks from Fig. 4.29 as the number of deputies increases. Without the collision avoidance constraints, the solve time for the distributed guidance scheme would remain nearly constant, irrespective of the number of deputies.

In line with this observation, Fig. 4.32 shows the number of SCP iterations required by both the centralized and distributed settings as the number of deputies increases. For the centralized setting, the number of iterations remains constant. As previously discussed, the centralized setting typically requires three iterations after the zeroth iteration: one to include the collision avoidance constraints, another to incorporate the acceleration pruning constraints, and a final iteration to enforce the minimum acceleration constraints, as depicted in Fig. 4.26. A reconfiguration requiring only two SCP iterations indicates that the step in

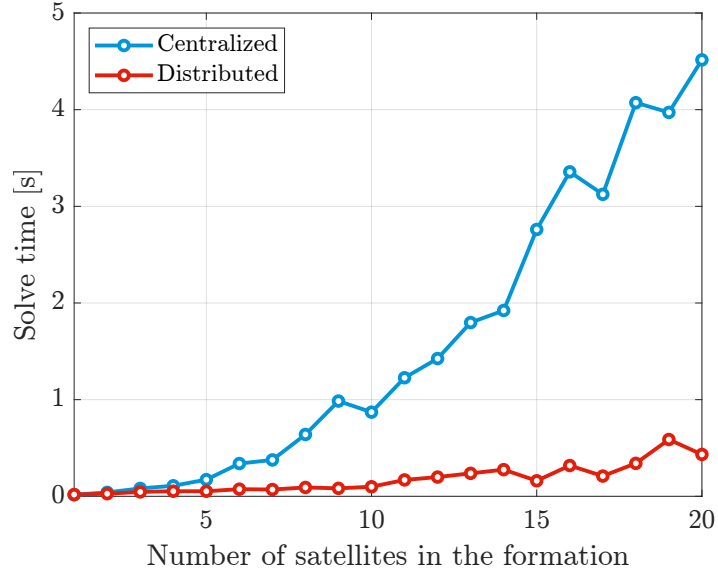


Figure 4.31: Solve time as the number of deputies increases

which collision avoidance constraints are included was not exploited because the trajectories from the zeroth iteration were already collision-free.

In contrast, the distributed case generally requires significantly more SCP iterations than the centralized case. This is due to the "Ensure collision avoidance" blocks, which demand additional iterations for a subset of the deputies. It is important to note that the reported number of SCP iterations for the distributed setting corresponds to the maximum across all deputies for a given reconfiguration scenario.

## 4.9 Conclusion

This chapter presented centralized and distributed guidance schemes designed for the efficient reconfiguration of multiple deputies' relative orbits around a chief satellite. A key feature of the proposed strategy is the reliance on single electric thrusters per deputy, with the chief satellite remaining uncontrolled. In the centralized approach, the chief functions as a physical satellite and serves as the central processing unit for optimizing the formation's trajectory. Conversely, in the distributed approach, the chief is treated as a virtual point orbiting the

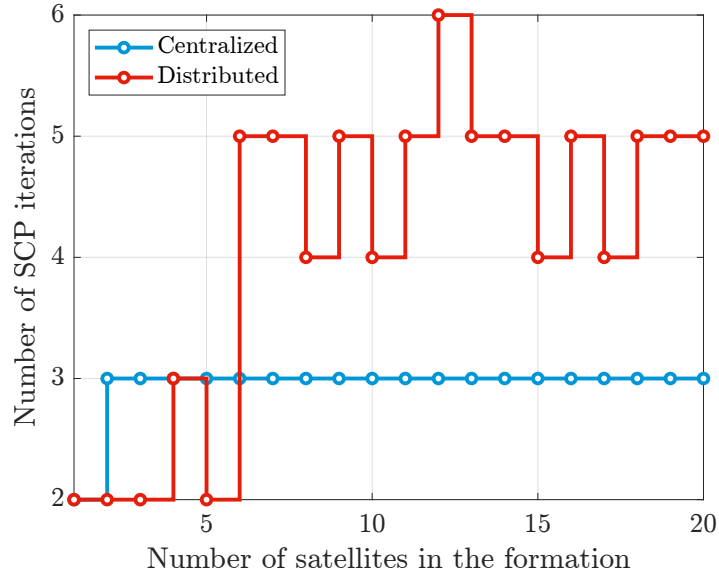


Figure 4.32: Number of required SCP iterations as the number of deputies increases

Earth, with each deputy independently calculating its own guidance profile using onboard computational resources.

The development of these guidance strategies incorporated critical considerations such as inter-deputy and deputy-chief collision avoidance (in the centralized case), as well as the inherent under-actuation of each deputy. These factors ensured safe and feasible execution of reconfiguration maneuvers. The chapter introduced a series of progressively refined numerical optimization formulations, each building on and relaxing constraints from the previous one. Four key formulations emerged based on their balance of computational efficiency and performance: [Quadratically Constrained Quadratic Programming \(QCQP\)](#), [Quadratic Programming \(QP\)](#), [Second-Order Cone Programming \(SOCP\)](#), and [Linear Programming \(LP\)](#).

To evaluate the effectiveness of these formulations, a comprehensive benchmarking experiment was conducted across four distinct reconfiguration scenarios using fifteen solvers. The results revealed that while the [QCQP](#) and [QP](#) formulations offered fast solve times, their high Delta-V requirements rendered them unsuitable for practical use. In contrast, the [SOCP](#)

formulation proved to be the most favorable, achieving minimal Delta-V while maintaining computational efficiency, making it ideal for complex, fuel-sensitive maneuvers. The LP formulation demonstrated exceptional efficiency and compatibility with various solvers, but its application was most appropriate for small-scale problems.

To address the physical limitations of the employed electric thrusters, the guidance formulations were extended to include a minimum thrust/acceleration constraint, below which the thruster becomes non-throttleable. As this constraint is inherently non-convex, an affine approximation was introduced to convexify it. Additionally, an acceleration pruning algorithm was developed to improve feasibility. However, the inclusion of the minimum acceleration constraint increased the likelihood of infeasibility for certain reconfiguration scenarios, particularly under strict hard constraints.

To mitigate these challenges, the chapter explored techniques for softening critical constraints, particularly for use in MPC-based schemes where guidance methods are applied iteratively in a closed-loop control framework. Softened formulations were shown to reliably produce feasible solutions for complex reconfiguration scenarios where hard-constrained approaches failed. Even when hard constraints were feasible, softened formulations delivered comparable results. This ability to generate approximate solutions when hard-constrained problems became infeasible underscores the robustness of softened formulations for real-time MPC applications.

Finally, methods for distributing the computational effort of solving the guidance problem were introduced, enabling each deputy to calculate its own guidance profile. These distributed approaches achieved performance comparable to centralized methods from the fuel efficiency point of view while offering scalability to larger satellite formations, making them suitable for missions with a large number of satellites.

# Control approaches for multi-satellite formation reconfiguration

## 5.1 Introduction

The guidance schemes discussed in Chapter 4 represent open-loop control systems. If applied directly to the formation reconfiguration system, these schemes may fail to achieve the desired control objectives due to unmodeled disturbances, linearization errors, and other factors. To mitigate these issues, it is essential to close the control loop by incorporating sensor measurements and navigation filters. A crucial element of this closed-loop system is the stepping control function, which leverages the guidance schemes to generate the appropriate control inputs based on the current state of the formation. This chapter explores different implementations for this stepping control function.

### 5.1.1 Problem Statement

The focus of this chapter is to introduce [Model-Predictive Control \(MPC\)](#) strategies that leverage the guidance strategies developed in Chapter 4 to close the control loop. The control system presented here represents a closed-loop extension of the problem described in Chapter 4. The following considerations are incorporated in the development and testing of the

closed-loop control system introduced in this chapter:

- The problem settings are consistent with those defined in Chapter 4. Specifically, a multi-satellite formation is tasked with reconfiguring from an initial configuration to a final configuration within a user-defined time frame, with the additional complexity that the satellites involved are underactuated.
- The control module takes the current state of the formation and outputs the optimal acceleration vector to be applied to each deputy from the current time up to the next sampling time;
- The control module is executed onboard the chief satellite in the centralized setting and onboard each deputy separately in the distributed case, thereby supporting the guidance strategies in both centralized and distributed settings;
- During the MPC operation, the control horizon is set equal to the prediction horizon. This approach is motivated by the absence of time-optimality requirements in the formation reconfiguration maneuver, and the well-known observation that the location of thrust arcs significantly impacts fuel efficiency [30];
- The Attitude Determination and Control System (ADCS) and the navigation system are assumed to be reliable, and hence their specific implementations are left out of the scope of this chapter. However, surrogate models for navigation and pointing errors are incorporated in the validation experiments to enhance the fidelity of the simulations.

### 5.1.2 State of the art

The literature on formation reconfiguration control mirrors that of guidance, presented at the beginning of Chapter 4, as most guidance strategies from the literature also include or rely on a specific control mechanism, often implemented through MPC. For instance, [81, 85, 112] highlight the advantages of MPC in realizing optimal trajectories for satellite formations while incorporating constraints such as collision avoidance and thrust limitations. MPC is particularly effective for continuous low-thrust applications due to its predictive capabilities

and ability to handle multiple constraints simultaneously. In [112], the efficiency of MPC was demonstrated in controlling 3D force-free formations, employing convex optimization to achieve real-time feasibility. Similarly, [85] extended the application of MPC by integrating both relative and absolute control for high-drag environments.

In addition to MPC, other control methods have been investigated. Nonlinear approaches, such as State-Dependent Riccati Equation (SDRE) controllers, address specific nonlinearities in system dynamics but often demand substantial computational resources, making them less suitable for onboard use [113]. Artificial potential fields [114, 115] provide intuitive solutions for collision avoidance, yet they provide a solutions which are neither time- or fuel-optimal. Despite these alternatives, MPC-based approaches remain the most practical and versatile for formation control, particularly when paired with convex optimization techniques. For instance, [81] demonstrated the effectiveness of fuel-optimal MPC in managing close-formation reconfigurations, ensuring collision-free trajectories while respecting thrust constraints.

However, challenges persist in adapting these approaches for formations of low-thrust underactuated satellites. This chapter addresses these challenges by introducing tailored MPC strategies that build on the guidance methods developed in Chapter 4, focusing on adaptability to underactuated satellite formations and practical operational constraints.

### 5.1.3 Methodology and contribution

In this chapter, two distinct stepping strategies (control modules) are utilized: a Shrinking-Horizon MPC (SHMPC) and a Fixed-Horizon MPC (FHMPC). The SHMPC employs a shrinking horizon, solving the guidance problem from the current time to the user-defined end of the maneuver. Conversely, the FHMPC relies on a pre-optimized trajectory, which serves as a reference for the MPC to track. It operates with a fixed-size receding horizon, solving the guidance problem to minimize fuel expenditure while maintaining close adherence to the reference trajectory. In both MPC settings, only the first control cycle from the control profile generated at each horizon is utilized.

While the SHMPC and FHMPC are not entirely novel strategies, the primary contribution of this chapter lies in the tailored implementation of these methods for the specific challenge of multi-satellite formation reconfiguration.

#### 5.1.4 Chapter outline

The next section introduces the MPC schemes adopted in this chapter, namely the SHMPC and the FHMPC. It also discusses the closed-loop operational settings of these two MPC strategies. To test the stability of the adopted control strategies, a Monte Carlo campaign is conducted, and its results are presented in Section 5.3. Additionally, the SHMPC and the FHMPC are benchmarked against a reference MPC from the literature, with the outcomes of this comparison also included in the same section. The key takeaways from the chapter are summarized in Section 5.4.

## 5.2 Model predictive control strategies

In this chapter, the Shrinking-Horizon MPC (SHMPC) and the Fixed-Horizon MPC (FHMPC) strategies are presented in detail. Additionally, the operational specifics of the Model-Predictive Control (MPC) within the closed-loop control framework are discussed, with a focus on their application to both centralized and distributed satellite formation reconfigurations.

### 5.2.1 Shrinking-horizon MPC

The shrinking-horizon MPC relies on optimizing the state and the control profiles over a time span that extends from the beginning of the current control cycle to the final time of the reconfiguration. In light of Fig. 4.1, the SHMPC starts by solving the guidance problem over the time span from  $t_0$  to  $t_f$ , while the second horizon extends from  $t_2$  to  $t_f$  and so on. The evolution of the shrinking horizons is illustrated graphically in Fig. 5.1, where the prediction horizon is shaded gray, starting at the current time  $t_k$ .

Although solving the guidance problem produces state and control profiles over the entire



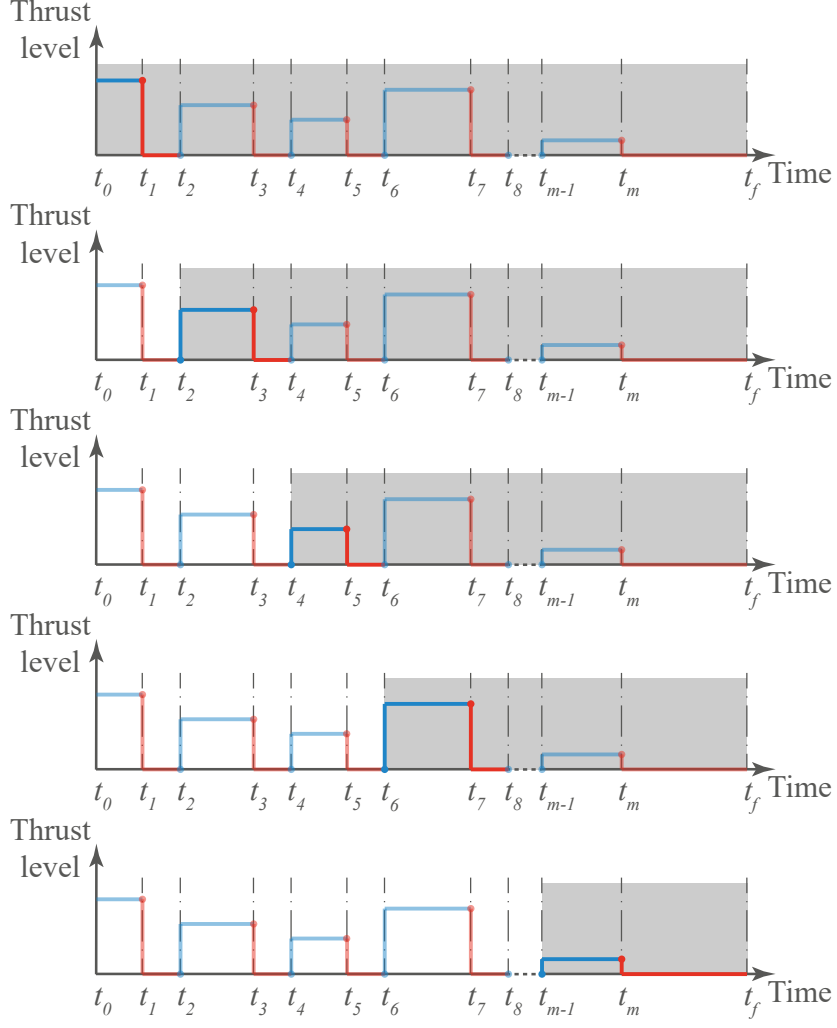


Figure 5.1: Evolution of the shrinking horizons

current horizon, only the first control cycle of each control profile is utilized and is provided as the output of the control stepping function.

### 5.2.2 Fixed-horizon MPC

In contrast to [SHMPC](#), [FHMPC](#) employs a fixed number of steps,  $N_h$  with  $N_h \ll m + 1$ , in each horizon over which [Problem 4.9](#) (or [Problem 4.10](#)) is solved. In order for the [FHMPC](#) to operate properly, a prior optimization of the trajectory of each deputy throughout the

formation reconfiguration maneuver is required, which is a one-time procedure that can be performed either online onboard the satellites or offline on the ground (see the yellow area in Fig. 5.2). The optimized trajectory is used as the reference trajectory that the guidance tries to track across all the horizons of the FHMPC. In other words,  $\bar{\mathcal{K}} = \{0, 1, 2, \dots, N_h - 1\}$  and the values of  $\bar{y}_{i,k} \forall k \in \bar{\mathcal{K}}$  are drawn from the optimized state profile throughout the maneuver. One unique requirement of the FHMPC is that the number of steps in each horizon needs to be an odd number, since a horizon contains an integer number of control cycles. The evolution of the horizons in the FHMPC scheme is shown in Fig. 5.2. In this figure, the yellow area represents the initial optimized trajectory, which is computed only once over the entire horizon, while the fixed receding horizons are displayed in gray. In the figure, it is assumed that each horizon consists of 5 steps, corresponding to 2 control cycles.

It is important to note that the size of each horizon is fixed in terms of the number of steps, and not in terms of the actual duration of a horizon, aligning each of them with the fixed steps introduced by the first whole optimization. To keep the size of each horizon fixed, specifically those which approach the end of the maneuver, some artificial steps must be introduced beyond the final time of the reconfiguration (refer to the last horizon in Fig. 5.2). The reference states for these artificial steps are drawn from propagating the linear dynamics in Eq. (2.50) with zero control input, and starting from  $t_f$ . The initial states for the propagation are set to  $\mathbf{y}_{i,m+1}$  for each satellite, which is the last step from the pre-optimized trajectory that was run over the whole maneuver.

### 5.2.3 MPC operation

The output of the control stepping function, whether the SHMPC or the FHMPC, consists of a set of acceleration vectors required to be executed by each deputy satellite's onboard propulsion system. Notably, these acceleration vectors are provided by either MPC scheme in the RTN frame, necessitating an additional layer for each deputy. This layer translates the acceleration vector into a reference attitude for the attitude control system to track, as well as determines the required thrust level. Moreover, since the  $L_2$  norm of the acceleration vector is allowed to be less than the minimum acceleration in the guidance layer as a result of

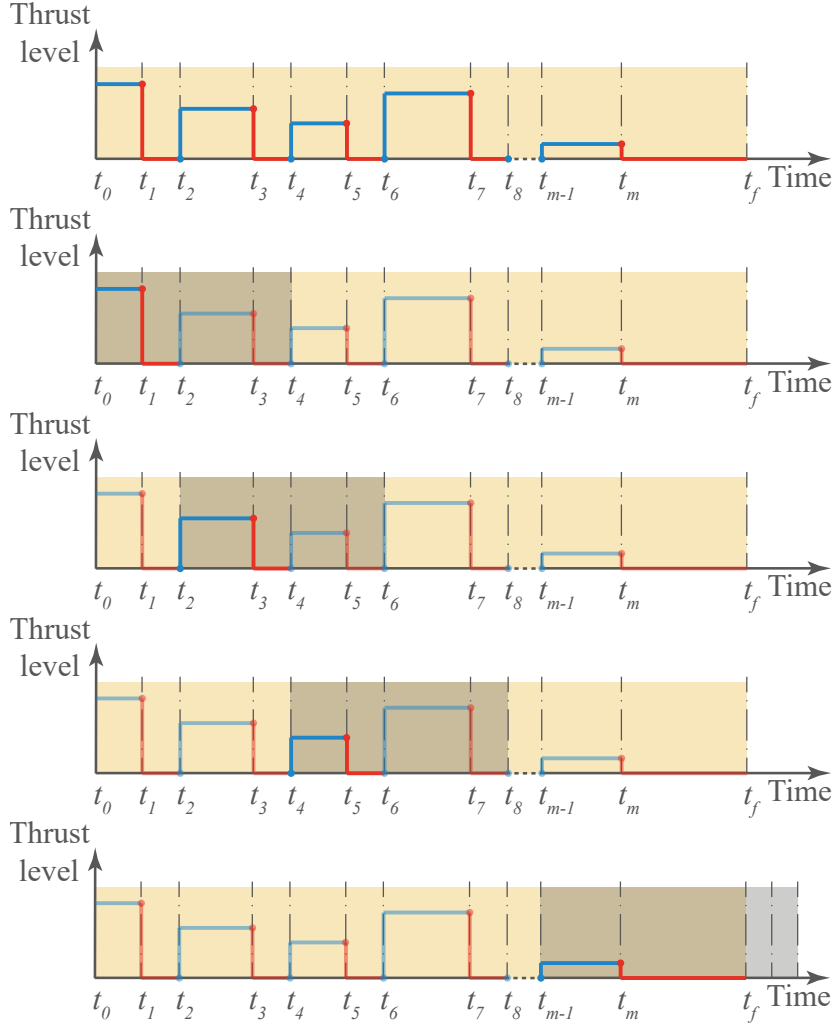


Figure 5.2: Evolution of the fixed horizons

the softening procedures, a saturation function is necessary to guarantee that the acceleration vector demanded by the control function is either bounded by the minimum and the maximum levels, or strictly zero. The adopted saturation scheme in this research is written as follows:

$$\text{Sat}(\mathbf{u}, u_{\min}, u_{\max}, \alpha) = \begin{cases} 0, & \|\mathbf{u}\|_2 \leq \alpha u_{\min} \\ \frac{\mathbf{u}}{\|\mathbf{u}\|_2} u_{\min}, & \alpha u_{\min} < \|\mathbf{u}\|_2 \leq u_{\min} \\ \mathbf{u}, & u_{\min} < \|\mathbf{u}\|_2 \leq u_{\max} \\ \frac{\mathbf{u}}{\|\mathbf{u}\|_2} u_{\max}, & u_{\max} < \|\mathbf{u}\|_2 \end{cases} \quad (5.1)$$

where  $0 \leq \alpha < 1$  is a user-defined parameter. In the context of the proposed guidance and control schemes, the last option of the saturation function (the fourth line) is never exploited since the guidance layer strictly constrains the  $L_2$  norm of the acceleration vector to be less than or equal to the maximum allowable acceleration (see Eqs. (4.79) and (4.92)), unlike the soft constraints on the minimum allowable acceleration (see Eqs. (4.77) and (4.90)).

For the numerical simulations, the absolute position and velocity are propagated individually for each deputy, and the relative states are later extracted from the absolute ones. Moreover, surrogate models are used to account for estimation and pointing errors that affect the formation control system, as the development of these systems is beyond the scope of this thesis. The adopted surrogate models are discussed in details in Appendix G.

### 5.3 Results and discussion

In order to validate the proposed MPC schemes, they were run over Reconfigurations 2 and 3 from Appendix D. These two scenarios are specifically interesting because they form a basis to compare the performance of the closed-loop system with that of the open-loop system, presented in Tables 4.8 and 4.9. The parameters used for the guidance scheme are consistent with those listed in Table 4.6, with the exception of the thrust arc duration, which is set to  $T_f = 0.05$  orbits. Parameters specific to the SHMPC and FHMPC schemes include a sampling time of 50 seconds for both MPC strategies, a fixed horizon size of  $N_h = 21$  steps for the FHMPC, and a parameter  $\alpha$  set to 40% for both schemes. Additionally, the parameters related to the surrogate models for navigation and pointing errors are set to identical values across all deputy satellites and the chief. Specifically, these parameters are:

- $\sigma_{\mathbf{r}_c} = \sigma_{\mathbf{r}_i} = \sigma_{\mathbf{r}} = \frac{2.5}{\sqrt{3}} \text{ m}, \forall i \in \mathcal{I},$
- $\sigma_{\mathbf{v}_c} = \sigma_{\mathbf{v}_i} = \sigma_{\mathbf{v}} = \frac{4.7}{\sqrt{3}} \text{ mm/s}, \forall i \in \mathcal{I},$
- $\sigma_{\Delta \mathbf{r}_i} = \sigma_{\Delta \mathbf{r}} = \frac{32}{\sqrt{6}} \text{ mm}, \forall i \in \mathcal{I},$
- $\sigma_{\Delta \mathbf{v}_i} = \sigma_{\Delta \mathbf{v}} = \frac{95}{\sqrt{6}} \mu\text{m/s}, \forall i \in \mathcal{I}.$

These values ensure that the surrogate absolute and relative navigation models (see Appendix G) align with the performance of the absolute and relative navigation systems developed for the PRISMA technology demonstration mission as presented in Table 3.12 of [6]. Furthermore, the pointing error  $\zeta_{pe}$  is set to 25 arc-seconds for all deputy satellites, which is consistent with the publicly available specifications of Triton-X [47].

Due to the inherent randomness in navigation and pointing errors, the two MPC strategies were evaluated over the two reconfiguration scenarios with 100 repeated runs for both the centralized and the distributed settings. For each run, the total required Delta-V, the final state errors, and the maximum intrusion of one satellite to the KOZ of another, were recorded and then averaged across the 100 trials. The results, presented in Table 5.1, summarize the total required Delta-V, the maximum and mean  $L_2$  norm of the state error at the final time of the maneuver, and the maximum collision avoidance violation, for each MPC strategy.

Table 5.1: Results of the proposed control strategies over the two case studies

	Centralized				Distributed			
	Reconfig. 2		Reconfig. 3		Reconfig. 2		Reconfig. 3	
	SH	FH	SH	FH	SH	FH	SH	FH
$\Delta V$ [m/s]	2.78	2.69	1.67	1.58	3.02	2.77	1.65	1.58
$\max \left( \ \mathbf{y}_{i,f} - \bar{\mathbf{y}}_{i,f}\ _2 \right)$ [m]	2.03	1.57	2.40	1.37	7.21	1.49	2.19	0.93
$\text{mean} \left( \ \mathbf{y}_{i,f} - \bar{\mathbf{y}}_{i,f}\ _2 \right)$ [m]	0.82	1.20	1.68	1.05	1.90	1.06	1.42	0.90
$\max (R_{CA} - \ \mathbf{y}_{i,k} - \mathbf{y}_{j,k}\ _2)$ [m]	0.13	6.90	0.17	5.29	12.00	1.02	0.04	0.00

The results in Table 5.1 are best understood in comparison to the open-loop results of the softened problem, presented in Tables 4.8 and 4.9. From a Delta-V perspective, both MPC schemes, SHMPC and FHMPC, demonstrate similar or slightly higher Delta-V requirements compared to the open-loop performance in both centralized and distributed settings. Notably, for these two specific reconfigurations, the SHMPC tends to demand slightly more Delta-V than the FHMPC, which exhibits strong tracking capabilities of the reference open-loop trajectory. However, as maneuver durations increase, the FHMPC begins to deviate more significantly from the reference trajectory, a trend that will be further explored later.

These deviations can be attributed to longer maneuvers being more susceptible to discrepancies between the linear model used in the guidance strategy and the nonlinear model employed for state propagation in validation simulations. In particular, these deviations are evident in the convolution matrix presented in Eq. (2.52), where certain assumptions inherent to its derivation play a significant role [28].

While the control accuracy of the **SHMPC** appears lower than that of the **FHMPC** for the specific reconfigurations examined here, this is not a general trend, as will be illustrated in subsequent discussions. The primary strength of the **SHMPC** lies in its adaptability. For example, it can dynamically improve performance when relative navigation accuracy is enhanced, such as during a rendezvous task when a vision-based navigation system is activated. Additionally, the **SHMPC** demonstrates superior resilience to elevated disturbance levels, including those arising from nonlinear dynamics not captured by the linear model. This adaptability is due to the **SHMPC**'s approach of optimizing the entire trajectory from the current epoch to the end of the maneuver, making it particularly suitable for longer or disturbance-prone maneuvers. Conversely, the **FHMPC** optimizes over a shorter, fixed horizon and focuses on tracking a pre-optimized trajectory. This approach makes the **FHMPC** more vulnerable to control inaccuracies or higher Delta-V demands in scenarios involving prolonged maneuvers, increased navigation errors, or disturbance-heavy conditions.

Regarding collision risk, the **FHMPC** shows strong tracking performance in the distributed setting, leading to safe trajectories throughout the Monte Carlo simulations. However, in the centralized setting, the **SHMPC** clearly outperforms the **FHMPC** by producing collision-free relative trajectories, even in the presence of navigation and pointing errors. Violations of collision avoidance criteria can be traced to two main factors: the softening of collision avoidance constraints and the piecewise safety guaranteed by the guidance layer, which only ensures collision-free trajectories at specific sampling times. As indicated in Table 5.1, although the safety violations are generally within acceptable levels—given that  $\beta_{max}$  was set to 10 m in Table 4.6—there may still be a desire to eliminate these violations entirely. This can be

achieved either by adopting a smaller sampling time in the guidance layer, which would increase computational demand, or by introducing an artificial margin to the [KOZ](#) radius for each satellite to further enhance safety.

The state error profiles for the six satellites involved in Reconfiguration 2 are presented in [Fig. 5.3](#), comparing the [SHMPC](#) and [FHMPC](#) approaches for a single Monte Carlo run out of the 100 conducted. For clarity and conciseness, only the results of the centralized approach are shown. It can be seen that each element of the dimensional [ROE](#) error vector gradually converges to zero by the end of the maneuver, achieving the primary control objective.

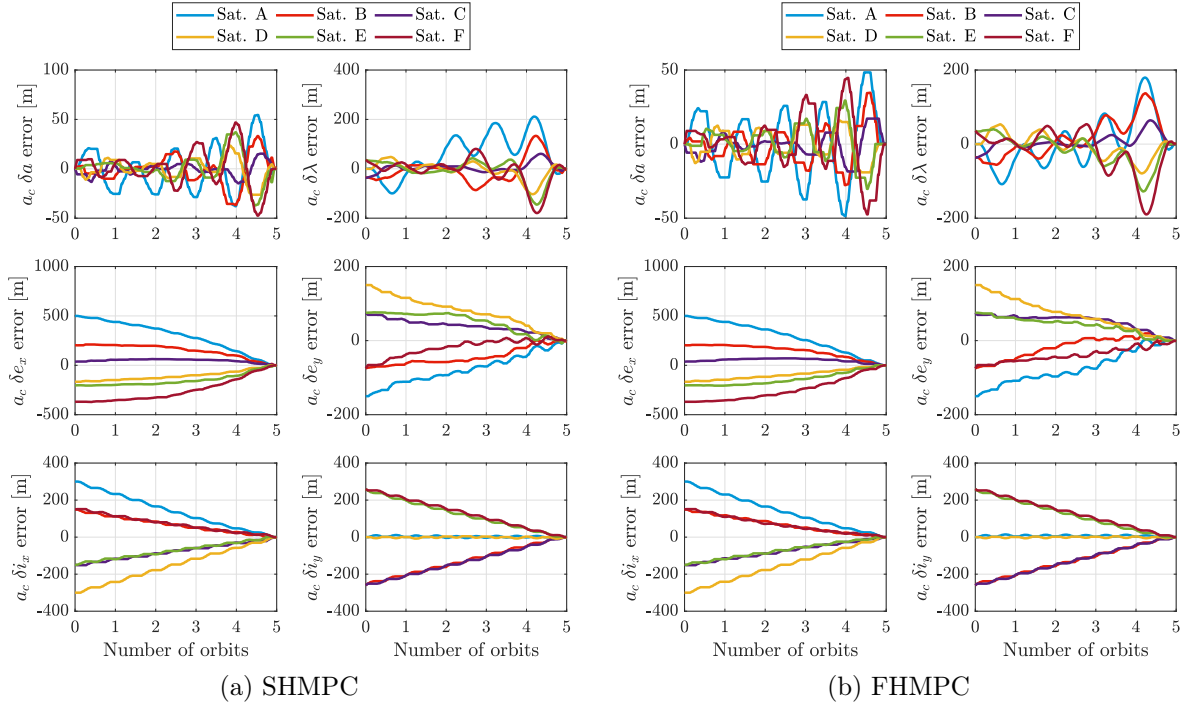


Figure 5.3: State error profiles for Reconfiguration 2 in the centralized setting.

Beyond verifying the achievement of the primary control objective, it is also necessary to assess the adherence to specific constraints, such as the minimum acceleration threshold and collision avoidance. These aspects are investigated here. The control acceleration profiles corresponding to the same Monte Carlo run as in [Fig. 5.3](#) are shown in [Fig. 5.4](#). The effect of the saturation function defined in [Eq. \(5.1\)](#) is evident, as the controller ensures that the

satellite never exceeds feasible acceleration limits, even though the guidance layer allows slight violations of the minimum acceleration constraint.

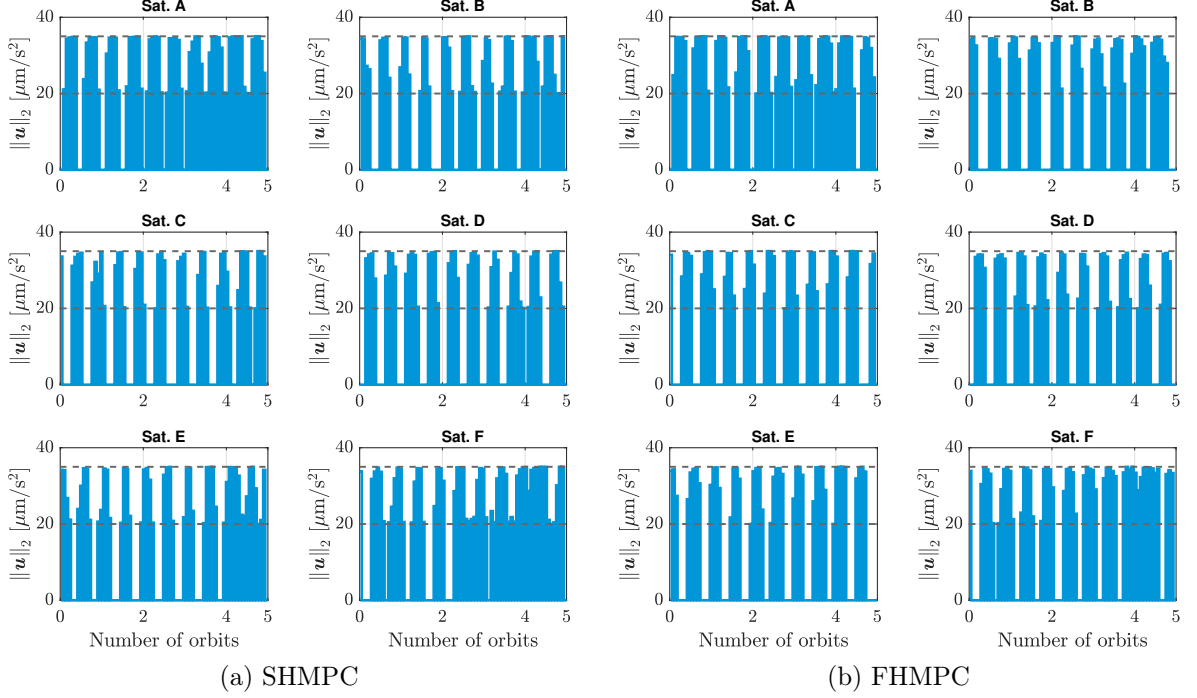


Figure 5.4: Control acceleration profiles for Reconfiguration 2 in the centralized setting.

Finally, the adherence to collision avoidance constraints is evaluated by examining the baselines between any two satellites throughout the maneuver. These baseline distances, corresponding to the same Monte Carlo run shown in Fig. 5.3 and Fig. 5.4, are depicted in Fig. 5.5. A star symbol highlights instances of collision avoidance violations. For this particular Monte Carlo run, the **FHMPC** exhibits a 5 m intrusion of satellite A into the **KOZ** of satellite B. This aligns with the results in Table 5.1, which indicate that, in the centralized setting, the **FHMPC** experiences larger violations of the collision avoidance threshold compared to the **SHMPC**. It is also worth noting that, as the results in Fig. 5.5 correspond to the centralized approach, the baselines of each satellite relative to the chief spacecraft are included. This is because, in the centralized setting, the chief spacecraft is treated as a physical satellite.

The **SHMPC** and **FHMPC** schemes, in both their centralized and distributed configurations, were benchmarked against other **MPC** approaches from the literature. Specifically, the



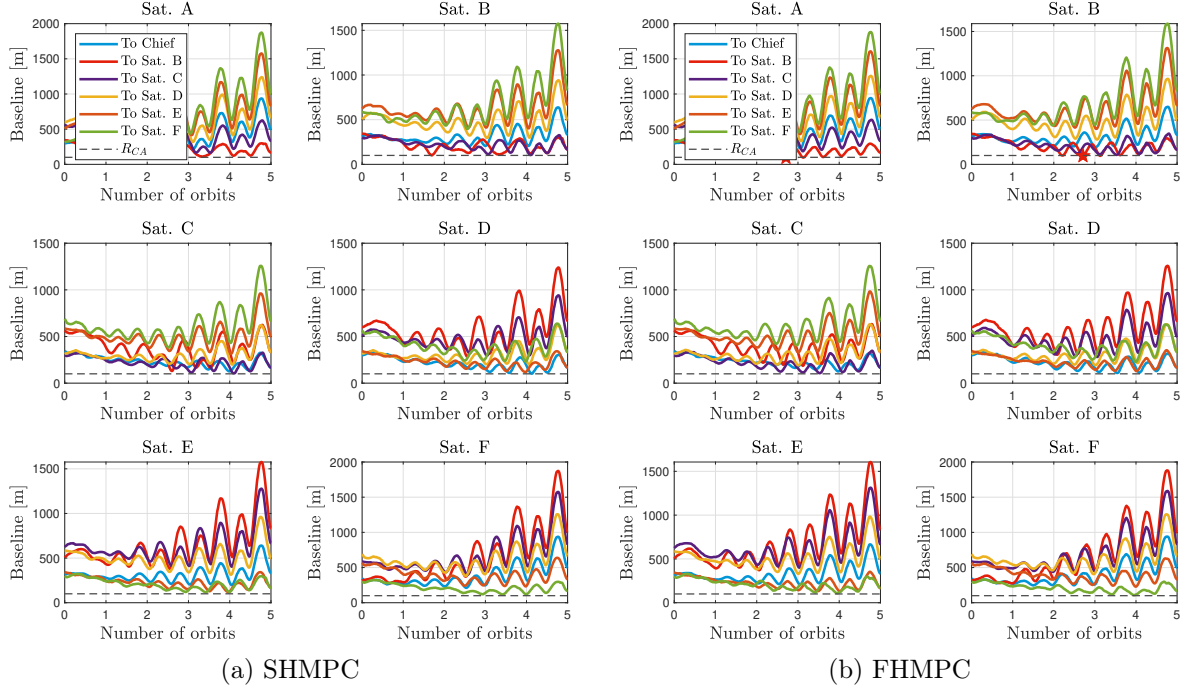


Figure 5.5: Baseline distances between satellites for Reconfiguration 2 in the centralized setting.

MPC algorithm presented in [85] was selected as a reference for comparison. This benchmark experiment was conducted using Reconfiguration 5, detailed in Appendix D, which was initially introduced in [85]. To ensure a fair comparison, both navigation and pointing errors were set to zero, consistent with the conditions in [85].

A summary of the results for the three MPC schemes is provided in Table 5.2. The table includes the final state errors for each satellite, the mean final state error across all satellites ( $\text{mean}(\|\mathbf{y}_{i,f} - \bar{\mathbf{y}}_{i,f}\|_2)$ ), the Delta-V requirements for each satellite, and the total Delta-V for the maneuver. Additionally, the table highlights the improvements achieved by the SHMPC and FHMPC schemes relative to the reference MPC, expressed as a percentage reduction in the mean final state error and the total Delta-V. The benchmark experiment was conducted using the same parameters listed in Table 4.6, except for the thrust arc duration, which was set to  $T_f = 0.1$  orbits and the weighting matrices for the guidance problem which were adjusted

to  $\mathbf{Q} = 100 \cdot \mathbf{I}_6$  and  $\mathbf{R} = 1.1 \cdot \mathbf{I}_3$ . The MPC parameters used are the same as the ones in the earlier comparison experiments. It is noteworthy that all generated trajectories during this benchmark experiment were collision-free.

Table 5.2: Benchmark summary

	$L_2$ Norm of the final dimensional ROE error [m]						Required Delta-V [m/s]					
	Sat. A	Sat. B	Sat. C	Sat. D	Mean	Improved	Sat. A	Sat. B	Sat. C	Sat. D	Total	Saved
Ref. MPC	2.96	4.46	7.42	3.39	4.56	0%	0.09	0.97	0.91	0.12	2.09	0%
Cent. SHMPC	1.08	0.13	0.11	1.08	0.60	<b>86.81%</b>	0.08	0.77	0.82	0.08	1.76	<b>15.82%</b>
Cent. FHMPC	5.76	0.90	0.34	7.32	3.58	<b>21.35%</b>	0.27	0.74	0.77	0.24	2.02	<b>3.30%</b>
Dist. SHMPC	1.07	0.12	0.10	1.08	0.59	<b>86.98%</b>	0.08	0.86	0.89	0.08	1.91	<b>8.41%</b>
Dist. FHMPC	5.74	1.13	0.94	7.48	3.82	<b>16.09%</b>	0.27	0.79	0.82	0.24	2.12	<b>-1.26%</b>

The results in Table 5.2 demonstrate that both SHMPC and FHMPC offer significant improvements over the reference MPC in terms of Delta-V efficiency and final state accuracy, particularly in the centralized setting. Despite the reference MPC using a much smaller sampling time for its guidance layer (50 seconds, compared to 0.1orbits  $\approx$  600 seconds for SHMPC and FHMPC), which permits the reference MPC a much finer control authority, the SHMPC outperformed it significantly in terms of both control accuracy and Delta-V demand. These findings validate the suitability of SHMPC for longer maneuvers, as its optimization approach dynamically adapts to discrepancies in the system states. The FHMPC also demonstrated measurable benefits over the reference MPC, particularly in the centralized configuration, where it achieved better final state accuracy and required less Delta-V. However, its performance ranked second to that of the SHMPC, which remains the most efficient strategy for the reconfiguration scenario considered in this study. Overall, the results confirm that adopting either SHMPC or FHMPC can significantly enhance satellite formation control performance compared to the reference MPC. Notably, SHMPC stands out as particularly advantageous for longer maneuvers, offering more precise control while demanding less fuel. Moreover, a comparison between the results of the centralized and distributed approaches reveals that the centralized setting consistently yields more fuel-efficient Delta-V trajectories, as is generally anticipated.

The state error profiles for Sat. A, across all three MPC schemes benchmarked in both the centralized and distributed settings, are depicted in Fig. 5.6. The figure highlights the distinct differences between the trajectories resulting from the centralized and distributed approaches.

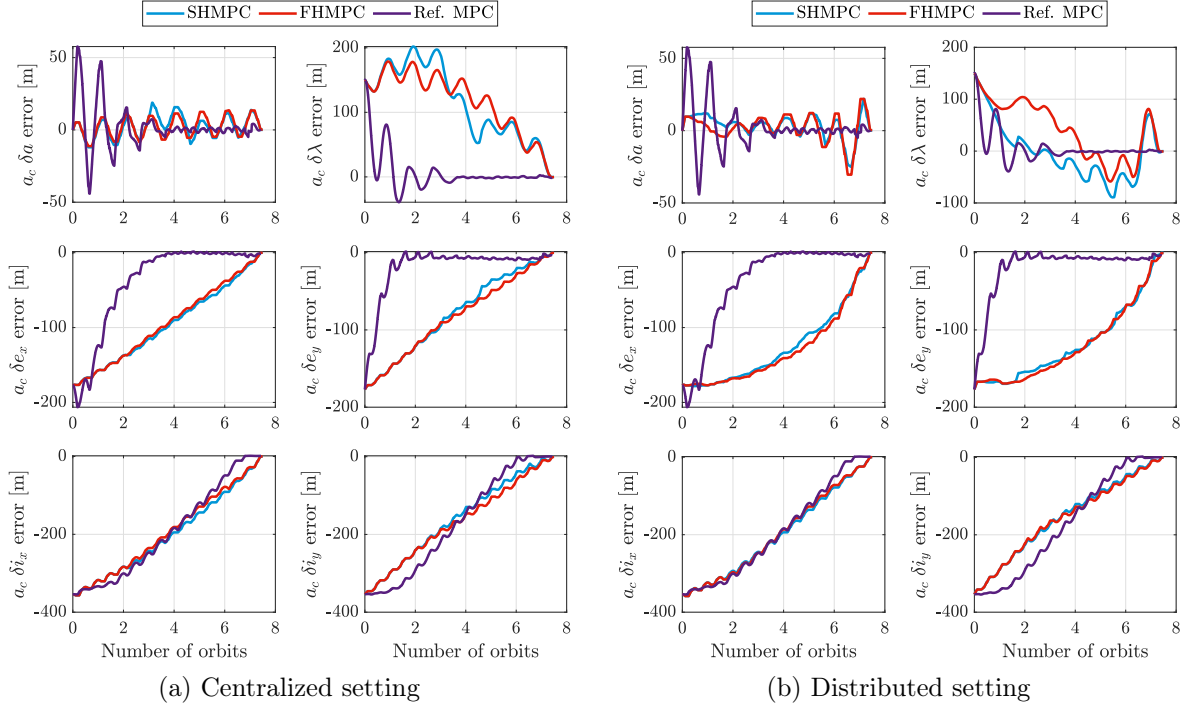


Figure 5.6: State error profiles of Sat. A in Reconfiguration 3

The fact that the reference MPC does not rely on a pre-optimized trajectory enhances its resilience to unexpected external disturbances, similar to the SHMPC. However, the absence of a constraint on the maneuver's final time, combined with its objective of tracking the required final state at each MPC horizon makes it naturally inclined toward quicker maneuvers rather than fuel-efficient, longer-duration ones, as depicted in Fig. 5.6. This tendency to prioritize rapid convergence to the final state from the start of the maneuver makes the reference MPC more suitable for formation keeping rather than formation reconfiguration, as discussed in [12]. While the FHMPC closely resembles the reference MPC, its adherence to an optimized reference trajectory makes it more fuel-efficient under standard conditions. However, this tracking approach also makes the FHMPC more sensitive to external disturbances than

the SHMPC or the reference MPC.

The benchmark experiment clearly demonstrates that, typically, the SHMPC not only is able to achieve more precise final states than the FHMPC, but also requires less Delta-V to complete the maneuver. However, this comes at the expense of increased computational complexity. Unlike the FHMPC, which maintains a fixed optimization horizon, the SHMPC utilizes a variable horizon that starts at a maximum length at the beginning of the maneuver and gradually decreases toward the end. A larger horizon length generally requires more time to solve its associated guidance problem. This is illustrated in Fig. 5.7, where the computation time for solving the guidance problem at each optimization horizon is shown for both the SHMPC and FHMPC in the centralized and the distributed settings. The absolute values presented in Fig. 5.7 are not crucial for our analysis, as they can vary depending on the onboard processor, the solver used, and other implementation factors such as the programming language. Instead, the focus is on the trend of each line. The solve time for each FHMPC horizon remains nearly constant, reflecting its fixed horizon length, whereas the solve time for each SHMPC horizon decreases as the horizon length reduces. For longer maneuvers, the SHMPC may become considerably more computationally demanding.

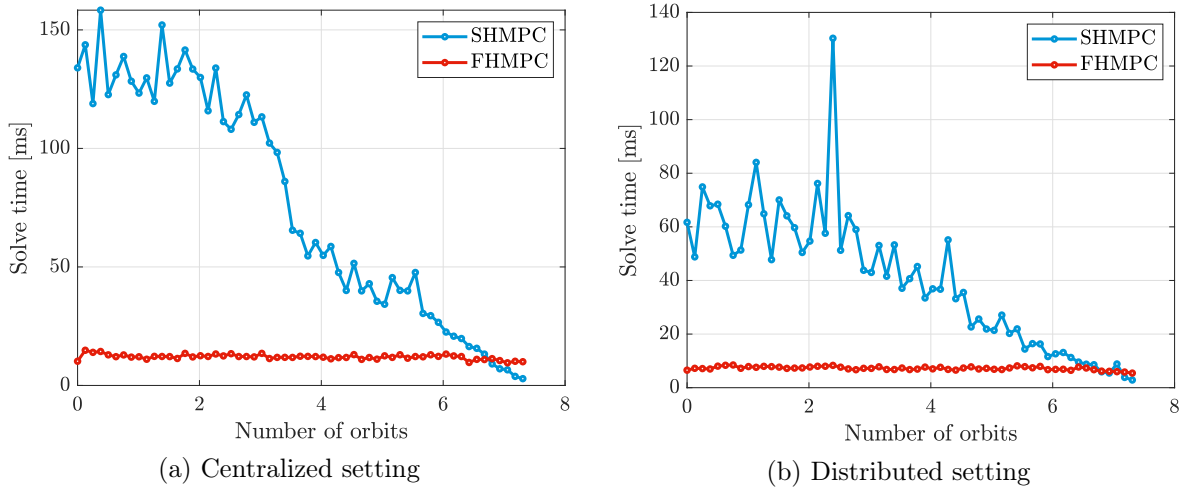


Figure 5.7: Solve times of the guidance problem over Reconfiguration 3

The advantages of the FHMPC extend beyond its stable and predictable computational

complexity as it is also well-suited for onboard implementation. By keeping the horizon length fixed, the size of the recurrent guidance problem is also fixed, which simplifies the writing/generation of embedded software without the need for dynamic memory allocation. This characteristic is especially beneficial in developing safety-critical software, such as that used onboard spacecraft, where dynamic memory allocation is often avoided to enhance reliability and safety [116]. Indeed, SHMPC can also be implemented using static memory allocation, however, this process is significantly more challenging and requires certain programmatic techniques to manage the variable horizon effectively.

## 5.4 Conclusion

In this chapter, two Model-Predictive Control (MPC) schemes were presented, building upon the guidance strategies introduced in Chapter 4. Specifically, the Shrinking-Horizon MPC (SHMPC) and Fixed-Horizon MPC (FHMPC) were described in detail.

The schemes were validated through a Monte Carlo campaign that focused on two representative reconfiguration scenarios. The simulations accounted for navigation inaccuracies and pointing errors to evaluate the robustness and reliability of the proposed methods. The SHMPC and FHMPC schemes were also benchmarked against a reference MPC strategy from the literature to provide a comparative analysis.

The results showed that the FHMPC performed well for short-duration reconfiguration maneuvers, excelling in tracking its pre-optimized reference trajectory. This strong tracking capability led to high control accuracy and reduced fuel consumption, making it suitable for short tasks. However, its performance declined in longer maneuvers due to increased sensitivity to modeling errors and disturbances.

In contrast, the SHMPC demonstrated superior adaptability and robustness in longer and disturbance-rich scenarios. By continuously re-optimizing the trajectory over a shrinking

horizon, the [SHMPC](#) managed discrepancies between the model and the actual system states effectively.

Both schemes outperformed the reference [MPC](#) in terms of control accuracy and fuel efficiency in the evaluated scenario.

The [FHMPC](#) was shown to be advantageous for onboard implementation as it can be implemented with static memory allocation, whereas the [SHMPC](#) requires additional techniques to avoid dynamic memory allocation during onboard use.

# Chapter 6

## Guidance and control strategies for absolute orbit keeping

### 6.1 Introduction

Maintaining a satellite in its desired orbit after achieving the targeted trajectory is a challenging task. Orbital perturbations, caused by natural dynamics, lead to both secular and non-secular drifts in certain orbital elements, as partly described in Eq. (2.19). To counter these effects, small corrective maneuvers are necessary to preserve the orbit over time. This chapter expands the scope of the thesis to include absolute orbit-keeping maneuvers, leveraging [Formation Flying \(FF\)](#) techniques to address this challenge.

#### 6.1.1 Problem statement

This chapter focuses on developing optimal guidance and control strategies for absolute orbit keeping of LEO satellites equipped with a single electric thruster. The objective is to align the satellite's orbital elements with a reference set of orbital elements. A key requirement of the absolute orbit-keeping control system is balancing maneuver speed and fuel efficiency. Unlike orbit transfer maneuvers, which are similar to formation reconfiguration maneuvers in multi-satellite systems and which can span long durations, orbit-keeping maneuvers must be

executed quickly while still considering fuel efficiency.

To simplify the design of the guidance and control methods, the minimum thrust constraint is not considered, and the satellite is hence assumed to have a fully throttleable electric thruster. The formal requirements for absolute orbit keeping are as follows:

- The system must autonomously guide the satellite from an initial orbit, defined by a set of initial orbital elements, to a nearby final orbit, defined by a set of target orbital elements;
- The orbit correction maneuver must achieve a trade-off between time-optimality and fuel efficiency, with a higher priority placed on time efficiency;
- The satellite is equipped with a single electric thruster that is fully throttleable from zero to its maximum thrust level.

### 6.1.2 State of the art

Many contributions on autonomous absolute orbit keeping for LEO satellites have been introduced over the last few decades [117, 118, 119, 120, 121]. Among these works, formation flying techniques were utilized for absolute orbit control in [121], where the satellite maintained its ground track close to a reference trajectory defined as the orbit of a virtual chief satellite. A specific set of variables parameterized the relative motion, and linear and quadratic optimal regulators were employed. All these studies assumed satellites with 3D thrusting capabilities and primarily adopted impulsive thrust models, rendering them unsuitable for satellites equipped with unidirectional electric thrusters.

A notable exception is the Autonomous Vision Approach Navigation and Target Identification (AVANTI) experiment [83], where a single impulsive thruster was utilized. In this mission, a specific guidance scheme incorporated no-control windows to enable nozzle re-orientation [84, 77]. Similarly, in [122], formation flying techniques were applied to develop a control scheme for orbit drift correction using a single impulsive thruster. The satellite



employed a passive magnetic attitude control system to align its longitudinal axis, where the thruster was mounted, with the local geomagnetic induction vector.

To the author’s knowledge, the only research addressing absolute orbit control for unidirectional low-thrust satellites is presented in [85]. However, this approach does not account for the coupled attitude evolution, which limits its suitability for rapid orbit corrections. Additionally, [85] does not consider navigation or actuator errors, further restricting its applicability for realistic mission scenarios.

### 6.1.3 Methodology and contribution

The absolute orbit keeping problem in this chapter is addressed using formation flying techniques, where the Triton-X satellite is assumed to be flying in formation with a virtual chief. Specifically, when a small orbit transfer maneuver is defined, the chief’s orbit serves as the target orbit, and the satellite must ultimately rendezvous with the virtual chief. Unlike the guidance methods presented in Chapter 4, the guidance problem here is formulated as a **Non-Linear Programming (NLP)** problem, which explicitly accounts for the coupling between the attitude and orbit control systems. This ensures that the thruster is consistently aligned with the desired firing direction. Such an approach is particularly relevant for orbit corrections that must be performed within a limited time frame, especially when operating at very low thrust levels.

For the closed-loop control layer, a **Model-Predictive Control (MPC)** scheme is proposed, which builds on the **NLP**-based guidance strategies and runs them recurrently to predict the satellite’s roto-translational state at each horizon. While the proposed **MPC** in this chapter is a **Fixed-Horizon MPC (FHMPC)**, it differs from the one discussed in Chapter 5. Instead of tracking a pre-optimized trajectory, it tracks the reference orbital elements at each time step. This design aligns with the requirement to perform orbit correction maneuvers that balance speed and fuel efficiency, with a preference for quicker maneuvers. Another distinction lies in the prediction and control horizons: the control horizon is set to be significantly

smaller than the prediction horizon to simplify the computations of the optimization problem.

The output of the [MPC](#) is a tuple comprising the optimal level of acceleration to be delivered by the thruster and the optimal desired attitude to be achieved by the attitude control system. The overall cost function at each receding horizon balances Delta-V optimality, time optimality, and attitude effort optimality, making it specifically tailored for small orbit changes (less than 1 km) as typically encountered in absolute orbit control problems around a reference orbit. To simplify calculations within the [NLP](#) formulation, a surrogate model of the attitude control system is introduced. This surrogate model not only reduces computational complexity but also generalizes the proposed methods for applicability to a wide range of satellites by tuning it to match the behavior of various attitude control systems.

The original contributions of the proposed approach are as follows:

- Accounting for the coupled attitude and relative orbit control problem within the control loop;
- Introducing a surrogate model for the attitude control system, which can be tuned to match the behavior of diverse attitude control systems;
- Proposing an [MPC](#) scheme for satellites with unidirectional propulsion systems, where thrust can also be applied during slew maneuvers.

#### 6.1.4 Chapter outline

The following section of this chapter introduces the formulation of the adopted surrogate model for the attitude control system. In Section 6.3, the [MPC](#) problem is formally established in the sense that the employed cost function is introduced and the inherent constraints of the problem in hand are presented. The problem of [MPC](#) parameter tuning as well as the module execution logic of the closed control loop are discussed in the same section. The stability of the proposed [MPC](#) scheme is then tested by means of a Monte Carlo simulation in Section 6.4, and the proposed [MPC](#) is benchmarked against that introduced in [85].

## 6.2 Surrogate model of the attitude control system

Given that the adopted satellite is equipped with only one thruster, continuous attitude maneuvers are necessary in order to direct the thruster to the desired direction during time-extended orbit maneuvers, which are likely to occur with the available low-thrust level. In this chapter, it is assumed that the attitude control system of Triton-X is capable of tracking a desired attitude profile. The specifications of the attitude control system are obtained from the publicly available Triton-X specifications [47]. A surrogate model for the [Attitude Determination and Control System \(ADCS\)](#) is presented here, but first the attitude kinematics are introduced.

Let  $\mathbf{q}^{xy} \in \mathbb{Q}$ , with  $\mathbb{Q} = \{\mathbf{q} \in \mathbb{R}^4: \|\mathbf{q}\|_2 = 1\}$  as the unit quaternion that describes the attitude of a general frame  $y$  with respect to another frame  $x$ . The vector rotation between these two frames can be performed according to,

$$\mathbf{v}^x = \mathbf{q}^{xy} \circ \mathbf{v}^y \circ \tilde{\mathbf{q}}^{xy}, \quad (6.1)$$

where  $\mathbf{v}^x \in \mathbb{F}^x$ ,  $\mathbf{v}^y \in \mathbb{F}^y$ ,  $\circ$  is the quaternion multiplication operator, and  $\tilde{\mathbf{q}}^{xy}$  is the quaternion conjugate of  $\mathbf{q}^{xy}$ . Equivalently, the attitude rotation can be performed using the [Direction Cosine Matrix \(DCM\)](#)  $\mathbf{A}(\mathbf{q}^{xy}) \in \mathbb{A}$ , with  $\mathbb{A} = \{\mathbf{A} \in \mathbb{R}^{3 \times 3}: \mathbf{A}^\top \mathbf{A} = \mathbf{I}_3\}$  (see Appendix A in [123]) as follows:

$$\mathbf{v}^x = \mathbf{A}(\mathbf{q}^{xy}) \mathbf{v}^y. \quad (6.2)$$

The attitude kinematics of a rigid body can be described as follows [124]:

$$\dot{\mathbf{q}}^{xy} = \frac{1}{2} \mathbf{q}^{xy} \circ \boldsymbol{\omega}^y, \quad (6.3)$$

where  $\boldsymbol{\omega}^y \in \mathbb{R}^3$  is the angular velocity of frame  $y$  with respect to frame  $x$ , expressed in  $\mathbb{F}^y$ .

Letting  $\mathbf{q} \equiv \mathbf{q}(t) = \mathbf{q}^{rb}(t)$  and  $\boldsymbol{\omega} \equiv \boldsymbol{\omega}(t) = \boldsymbol{\omega}^b(t)$ , and letting  $\mathbf{q}_r \equiv \mathbf{q}_r(t)$  be the de-

sired/reference quaternion at time  $t$ , Eq. (6.3) can be written as

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \circ \boldsymbol{\omega}, \quad (6.4)$$

and the error quaternion,  $\mathbf{q}_e$ , is defined, in terms of the current attitude and the reference one, as

$$\mathbf{q}_e = \begin{bmatrix} q_{e,0} \\ \mathbf{q}_e \end{bmatrix} = \tilde{\mathbf{q}}_r \circ \mathbf{q}. \quad (6.5)$$

The ultimate behavior of a stable ADCS would be to force the error quaternion to asymptotically approach either  $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top$ , which could be achieved through forcing the angular velocity,  $\boldsymbol{\omega}$ , profile to follow:

$$\boldsymbol{\omega} = -K q_{e,0} \mathbf{q}_e, \quad (6.6)$$

where  $K$  is a positive control gain. The rationale behind why the  $\boldsymbol{\omega}$  profile is plausible is given in Appendix H.

An additional operational constraint is imposed on the ADCS that dictates that the angular rate around any axis does not exceed the maximum allowable angular rate, i.e.,  $\omega = \|\boldsymbol{\omega}\|_2 \leq \omega_{max}$ . To be compliant with the attitude control specifications of Triton-X [47], the value of  $K$  is set to 0.2, while the value of  $\omega_{max}$  is set to 2 deg/s. Indeed,  $\omega_{max}$  could be found directly in the data sheets. The numerical value of  $K$ , instead, has been assessed through numerical simulations in order to properly emulate the behavior of Triton-X ADCS, based on the time duration to perform a 60-degree slew.

Substituting from Eq. (6.6) into Eq. (6.4), the adopted surrogate model for the ADCS is obtained:

$$\dot{\mathbf{q}} = -\frac{1}{2} K q_{e,0} \mathbf{q} \circ \mathbf{q}_e. \quad (6.7)$$

One thing that has to be acknowledged is that the attitude and angular velocity profiles of the actual satellite could differ slightly from the profiles generated by the surrogate model

assumed here, depending on the exact quaternion feedback regulator used by the ADCS as well as the exact inertia tensor of the satellite. Nonetheless, changing the value of  $K$  could bring the attitude and angular velocity profiles of the surrogate model very close to those of the actual ADCS. This claim could be verified through benchmarking the surrogate model against a full attitude control system, which uses one of the commonly used feedback regulator proposed in [125], and the results are depicted in Fig. 6.1. It is to be noted that the initial and reference quaternions are randomly chosen for the simulation in Fig. 6.1. Moreover, the subscript  $(\cdot)_{\text{full}}$  refers to a quantity relating to the full attitude control model, while the subscript  $(\cdot)_{\text{surr}}$  refers to surrogate model quantities.

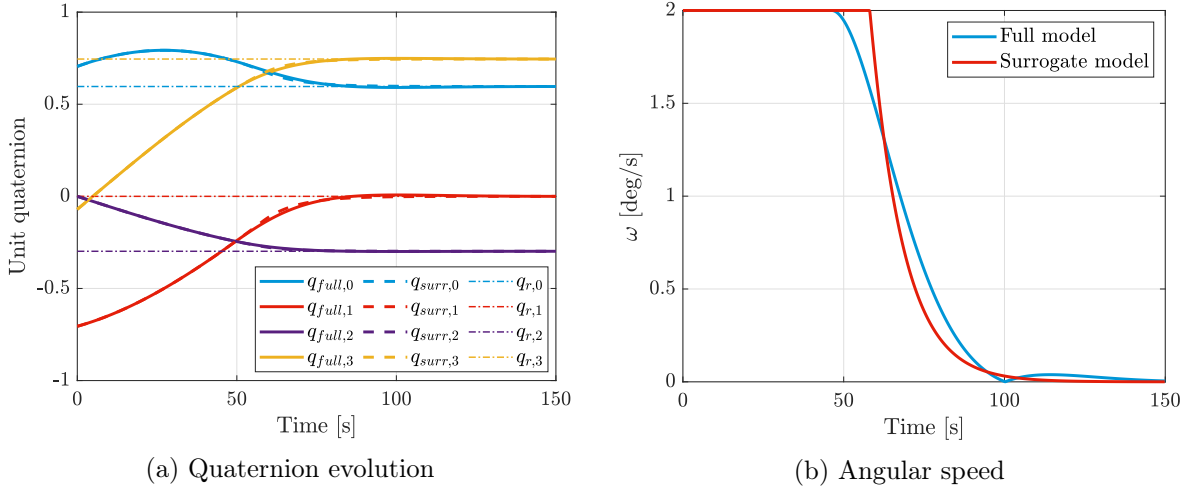


Figure 6.1: Validation of the surrogate ADCS model through random initial and desired attitudes

## 6.3 Model Predictive Control

### 6.3.1 Guidance layer

In the context of our problem, attitude dynamics are coupled with the ROE-based relative translational motion, and the state vector is set to

$$\mathbf{s} = \begin{bmatrix} \mathbf{y} \\ \mathbf{q} \end{bmatrix}, \quad (6.8)$$

where the dynamics of the state vector are described by Eq. (2.50) and Eq. (6.7). The control input, on the other hand, is set to

$$\mathbf{u} = \begin{bmatrix} u \\ \mathbf{q}_r \end{bmatrix}, \quad (6.9)$$

where  $u$  is the magnitude of the piece-wise constant input acceleration provided by the single thruster, and  $\mathbf{q}_r$  is the desired quaternion. The state vector as well as the control vector are collated over the prediction and the control horizons in the two matrices  $\mathbf{S}$  and  $\mathbf{U}$  respectively, such that

$$\begin{aligned} \mathbf{S} &= \begin{bmatrix} \mathbf{s}_0 & \mathbf{s}_1 & \dots & \mathbf{s}_{N_p} \end{bmatrix}, \\ \mathbf{U} &= \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \dots & \mathbf{u}_{N_u-1} \end{bmatrix}, \end{aligned} \quad (6.10)$$

with  $N_p$  being the length of the prediction horizon, and  $N_u$  being the length of the control horizon, where  $N_p \geq N_u$ . It is to be noted that the notations  $\mathbf{s}_k = \mathbf{s}(t_k)$ , where  $k \in \{0, 1, \dots, N_p\}$ , and  $\mathbf{u}_j = \mathbf{u}(t_j)$ , where  $j \in \{0, 1, \dots, N_u - 1\}$ , are used, and the time difference between any two consecutive instances is constant and is equal to the user-defined sampling time, i.e.,  $t_{k+1} - t_k = T_s$ . Moreover, after the control horizon is over, the input acceleration is set to zero (i.e.,  $u_k = 0 \ \forall k \in \{N_u, N_u + 1, \dots, N_p\}$ ) and the desired attitude is left to be decided by the mission-specific mode management logic of the ADCS. For the sake of simulating the attitude evolution, the reference attitude at an arbitrary time step beyond the control horizon is set to the quaternion from the previous step. Formally,  $\mathbf{q}_r(t_k) = \mathbf{q}_r(t_{k-1}) \ \forall k \in \{N_u, N_u + 1, \dots, N_p\}$ .

It is also important to note that the adopted satellite is assumed, without loss of generality of the approach, to have its thruster directed along the z-direction of its body frame (with the thrust vector passing by the center of mass of the satellite), and hence the input acceleration vectors in the body and in the RTN frames can be related as

$$\begin{aligned}\mathbf{u}^r &= \mathbf{A} \left( \mathbf{q}^{rb} \right) \mathbf{u}^b, \\ \mathbf{u}^b &= \begin{bmatrix} 0 & 0 & u \end{bmatrix}^\top.\end{aligned}\tag{6.11}$$

After having introduced the necessary notations, the guidance optimization problem can be formulated as follows:

**Problem 6.1.**

$$\min_{\mathbf{S}, \mathbf{U}} \quad J_{\mathbf{y}} + J_u + J_{\delta \mathbf{q}}$$

*Subject to*

$$\mathbf{s}_0 = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{q}_0 \end{bmatrix}, \tag{6.12}$$

$$\mathbf{y}_{k+1} = \begin{cases} \Phi_k \mathbf{y}_k + a_c \Psi_k \bar{\mathbf{u}}_k^r, & k \in \{0, 1, \dots, N_u - 1\} \\ \Phi_k \mathbf{y}_k, & k \in \{N_u, N_u + 1, \dots, N_p - 1\} \end{cases} \tag{6.13}$$

$$\mathbf{q}_{k+1} = RK_4(t_k, t_{k+1}, \mathbf{q}_k, \mathbf{q}_r(t_k)) \quad \forall k \in \{0, 1, \dots, N_p - 1\} \tag{6.14}$$

$$\|\mathbf{q}_k\|_2^2 = 1 \quad \forall k \in \{1, \dots, N_p - 1\}, \tag{6.15}$$

$$\|\boldsymbol{\omega}_k\|_2^2 \leq \omega_{max}^2 \quad \forall k \in \{1, \dots, N_p - 1\}, \tag{6.16}$$

$$0 \leq u_k \leq u_{max} \quad \forall k \in \{0, 1, \dots, N_u - 1\}, \tag{6.17}$$

$$\|\mathbf{q}_r(t_k)\|_2^2 = 1 \quad \forall k \in \{0, 1, \dots, N_u - 1\}, \quad (6.18)$$

where  $t_0$ ,  $\mathbf{y}_0$ , and  $\mathbf{q}_0$ , are the time, dimensional ROE vector, and quaternion vector, respectively, at the beginning of each horizon. Note that these are receding quantities that are specific to each prediction horizon. Accordingly, they coincide with the quantities at the beginning of the simulation only once, at the time of the first prediction. Moreover,  $u_{max}$  is the maximum applicable acceleration that can be provided by the thruster, and  $\omega_{max}$  is the maximum allowable angular rate around any axis.

According to the solution of ROE dynamics in Eq. (2.50), a constant value for the input acceleration vector,  $\mathbf{u}_k^r$ , has to be fed to the dynamics constraint in Eq. (6.13). However, the attitude dynamics are much faster than those of the ROE, and since the vector  $\mathbf{u}_k^r$  is attitude-dependent, its value can change significantly from  $t_k$  to  $t_{k+1}$ , even when  $\mathbf{u}_k^b$  is kept constant. Therefore, a prediction of  $\mathbf{u}_k^r$ ,  $\bar{\mathbf{u}}_k^r$  had to be fixed and fed to the dynamics constraint in Eq. (6.13). The value of  $\bar{\mathbf{u}}_k^r$  is set to that of  $\mathbf{u}_k^r$  at time  $(t_k + t_{k+1})/2$ , with its formal definition being as follows:

$$\begin{aligned} \bar{\mathbf{u}}_k^r &= \mathbf{A} \left( \mathbf{q} \left( \frac{t_k + t_{k+1}}{2} \right) \right) \mathbf{u}_k^b, \\ \mathbf{u}_k^b &= \begin{bmatrix} 0 & 0 & u_k \end{bmatrix}^\top. \end{aligned} \quad (6.19)$$

The attitude kinematics in Eq. (6.7) are discretized through a symbolic Runge-Kutta fourth-order (RK4) scheme and the discrete kinematics are imposed in Eq. (6.14). Moreover, the discrete angular velocity vector in Eq. (6.16),  $\boldsymbol{\omega}_k$ , could be obtained by applying the formula in Eq. (6.6) using the discrete unit quaternion signal  $\mathbf{q}_k$ . It is also worth noting that the squared norms ( $\|\mathbf{q}_k\|_2^2$ ,  $\|\boldsymbol{\omega}_k\|_2^2$ , and  $\|\mathbf{q}_r(t_k)\|_2^2$ ) are constrained instead of the norms themselves. This formulation is adopted to facilitate the job of the optimizer since it allows the Jacobian of the constraints to be defined at all values of  $\mathbf{S}$  and  $\mathbf{U}$ .



The components of the cost function of Problem 6.1,  $J_{\mathbf{y}}$ ,  $J_u$ , and  $J_{\delta \mathbf{q}}$ , are defined as

$$\begin{aligned}
J_{\mathbf{y}} &= (\mathbf{y}_{N_p} - \mathbf{y}_r)^\top \mathbf{P} (\mathbf{y}_{N_p} - \mathbf{y}_r) \\
&\quad + \sum_{k=0}^{N_p-1} (\mathbf{y}_k - \mathbf{y}_r)^\top \mathbf{Q} (\mathbf{y}_k - \mathbf{y}_r), \\
J_u &= R_u \sum_{k=0}^{N_u-1} u_k^2, \\
J_{\delta \mathbf{q}} &= R_{\delta \mathbf{q}} \sum_{k=0}^{N_u-1} \delta \mathbf{q}_k^\top \delta \mathbf{q}_k, \quad \delta \mathbf{q}_k = \mathbf{q}_r(t_k) - \mathbf{q}(t_k),
\end{aligned} \tag{6.20}$$

where  $\mathbf{y}_r$  is the reference dimensional ROE vector, while  $\mathbf{P} \in \mathbb{R}^{6 \times 6}$ ,  $\mathbf{Q} \in \mathbb{R}^{6 \times 6}$ ,  $R_u \in \mathbb{R}$ , and  $R_{\delta \mathbf{q}} \in \mathbb{R}$  are positive-definite MPC gains.

It is important to note that the ultimate goal of autonomous orbit keeping is to rendezvous with a virtual target then track its absolute orbit, i.e.,  $\mathbf{y}_r = \mathbf{0}$ . Moreover, the matrices  $\mathbf{P}$  and  $\mathbf{Q}$  are chosen to be diagonal matrices, which renders  $J_{\mathbf{y}}$  a summation over the squared weighted  $L_2$ -norms of the error vector. This ultimately means that  $J_{\mathbf{y}}$  is a measure of distance between  $\mathbf{y}_k$  and  $\mathbf{y}_r$  in a scaled-ROE space. Indeed, reasoning  $J_{\mathbf{y}}$  in terms of the magnitude of the error vector in a scaled-ROE space does not only allow to eventually drive  $\mathbf{y}$  to  $\mathbf{y}_r$ , and hence for the satellite to rendezvous and then track the orbit of the virtual target, but also allows to indirectly minimize the total Delta-V cost, since the distance in the ROE space, or in a scaled-ROE space for that matter, can be used as a measure for the total Delta-V cost [77, 84]. A closer look at  $J_{\mathbf{y}}$  reveals that while minimizing it indeed drives the error signal to zero, it is, in fact, an implicit trade-off between fuel and time optimality, depending on the ratio between the traces of  $\mathbf{P}$  and  $\mathbf{Q}$ , respectively. The greater the value of this ratio, the more inclined towards fuel optimality the cost function becomes, while the lower the ratio, the more the scheme leans towards time optimality.

Adding  $J_u$  to the compound cost function directly minimizes the total required thrust from the onboard single thruster. The adopted convex form of  $J_u$  is the fuel-optimal form of a

cost function, which also coincides with the Delta-V-optimal form for single-thruster satellites [9].

Finally, the purpose of adding the last component of the cost function,  $J_{\delta q}$ , is to softly constrain the desired attitude to be as close as possible to the actual attitude in order to avoid unnecessary large slews, which in turn minimizes the attitude control effort. This soft constraint allows for large maneuvers, depending on the choice of the MPC gains, only when it is meaningful from the point of view of the ROE error, i.e.,  $J_y$ , and fuel cost, i.e.,  $J_u$ .

### 6.3.2 Parameter Tuning

In the proposed formulation of the MPC guidance layer, there are many parameters and gains to be chosen, which can affect the overall performance and robustness of the MPC. In many industrial applications, such parameters are chosen through engineering experience. In this application, the initial guess for the prediction horizon has been formulated based on the literature. Since the Delta-V-optimal locations for ROE changes through impulsive Delta-V increments are known to occur, at most, every half-orbit [77], and since the input acceleration of the low-thrust propulsion system should be distributed around these Delta-V-optimal locations (if Delta-V optimality is sought), the prediction horizon,  $N_p T_s$ , is chosen to be at least half the orbital period. In this setting, the MPC is able to foresee all the Delta-V-optimal locations throughout a certain orbit, which hence leads to optimal allocation of the available thrust. Overly increasing the prediction horizon is expected to only overwhelm the onboard processor, while not having much effect on the optimality of the solution.

The choice of the control horizon,  $N_u T_s$ , on the other hand, is decided through trial and error, which is an iterative process as all the other parameters have to be fixed before the tuning process starts. Nonetheless, regarding the choice of the  $N_u$  value, it has to be a small positive integer less than or equal to  $N_p$ . Common guidelines for choosing the prediction and control horizons can be found in [126].

Tuning of the sampling time,  $T_s$ , has also been performed through trial and error. In the context of Problem 6.1, although the dynamics of the ROE are slow and the sampling time could be chosen as a large value in order to not solve the optimization problem so frequently, the following important consideration has to be taken into account to carry out the tuning process. Since the attitude dynamics are much faster than that of the ROE, one cannot ignore that a fixed value for the input acceleration vector in the RTN frame,  $\bar{\mathbf{u}}_k^r$ , is fed to the linearized ROE dynamics constraint in Eq. (6.13), which renders the model used in the MPC not accurate unless a small value is chosen for the sampling time. Indeed, including  $J_{\delta q}$  in the compound cost function helps in slowing down the attitude dynamics. Still, the need to choose a scant value for the sampling time is meaningful. A compromise has to be found between choosing a small  $T_s$  and having a more accurate model for the MPC on one hand, and choosing a large  $T_s$  that allows the optimization problem to be solved less frequently on the other.

The cost function gains (i.e.,  $\mathbf{P}$ ,  $\mathbf{Q}$ ,  $R_u$ , and  $R_{\delta q}$  in this chapter) are more challenging to tune, since the cost function of the MPC is problem-dependent and no general guidelines exist for its weighting. Thus, more focus is put on analyzing the effect of choosing different cost function weights and on actually selecting an optimal set of these weights.

In order to reduce the number of tunable parameters, the components of the cost function, i.e.,  $J_{\mathbf{y}}$ ,  $J_u$ , and  $J_{\delta q}$ , are related to the main weighting matrix,  $\mathbf{Q}$ , such that

$$\begin{aligned}\mathbf{P} &= f_{\mathbf{P}}\mathbf{Q}, \\ \bar{J}_u &= f_u\bar{J}_{\mathbf{y}}, \\ \bar{J}_{\delta q} &= f_{\delta q}\bar{J}_{\mathbf{y}},\end{aligned}\tag{6.21}$$

where

$$\begin{aligned}\bar{J}_{\mathbf{y}} &= (N_p + f_{\mathbf{P}}) \bar{\mathbf{y}}^\top \mathbf{Q} \bar{\mathbf{y}}, \\ \bar{J}_u &= N_u R_u \bar{u}^2, \\ \bar{J}_{\delta \mathbf{q}} &= N_N R_{\delta \mathbf{q}} \bar{\delta \mathbf{q}}^\top \bar{\delta \mathbf{q}},\end{aligned}\tag{6.22}$$

with  $\bar{\mathbf{y}}$  being the expected value of  $\mathbf{y}$  over the prediction horizon, which is approximated by the ROE vector at the beginning of each horizon, and  $\bar{u} \approx 0.5u_{\max}$  and  $\bar{\delta \mathbf{q}} \approx [0.1 \ 0.1 \ 0.1 \ 0.1]^\top$  being predictions for the values of  $u$  and  $\delta \mathbf{q}$  over the control horizon. The approximated expected values,  $\bar{\mathbf{y}}$ ,  $\bar{u}$ , and  $\bar{\delta \mathbf{q}}$ , are illustrated graphically over one prediction horizon in Fig. 6.2. For imaging purposes, each of them is taken to be one-dimensional. Indeed, the values assumed for  $\bar{u}$  and  $\bar{\delta \mathbf{q}}$  are merely the author's predictions for the expected values of these quantities over the whole simulation; however, choosing other values for these variables would not affect the final MPC gains, as will be elaborated later in the text.

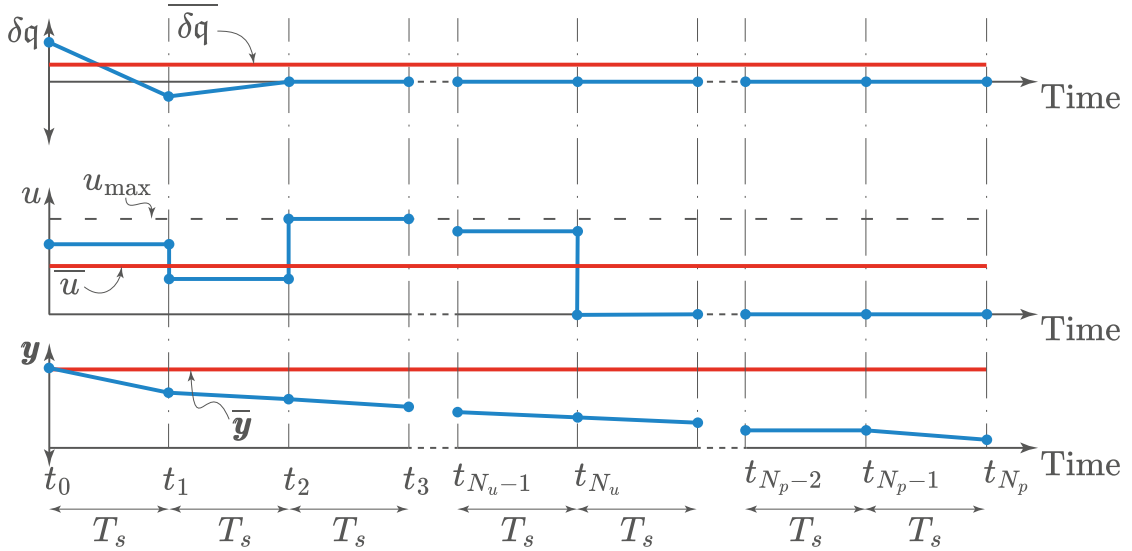


Figure 6.2: Assumed expected values  $\bar{\mathbf{y}}$ ,  $\bar{u}$ , and  $\bar{\delta \mathbf{q}}$  over one prediction horizon

Substituting from Eq. (6.22) into Eq. (6.21), the MPC gains can be related to  $\mathbf{Q}$ ,  $\bar{\mathbf{y}}$ ,  $\bar{u}$ , and  $\bar{\delta \mathbf{q}}$  as

$$\begin{aligned}R_u &= \frac{f_u (N_p + f_{\mathbf{P}})}{N_u \bar{u}^2} \bar{\mathbf{y}}^\top \mathbf{Q} \bar{\mathbf{y}}, \\ R_{\delta \mathbf{q}} &= \frac{f_{\delta \mathbf{q}} (N_p + f_{\mathbf{P}})}{N_u \bar{\delta \mathbf{q}}^\top \bar{\delta \mathbf{q}}} \bar{\mathbf{y}}^\top \mathbf{Q} \bar{\mathbf{y}},\end{aligned}\tag{6.23}$$

where  $R_u$  and  $R_{\delta q}$  change for every prediction horizon.

It is conceivable that when  $y = \|\mathbf{y}\|_2$  is very small (i.e., the rendezvous of the satellite with its virtual target has already taken place), the priority should be given to  $J_{\delta q}$  in order for the ADCS not to hyper-react to the small errors in  $\mathbf{y}$ . Indeed, the phenomenon of hyper-reaction of the ADCS when  $y$  becomes too small has been verified via numerical simulation for a system using the definition of  $R_{\delta q}$  in Eq. (6.23). It is for this reason that  $R_{\delta q}$  has to be redefined in order to steer the priority to  $J_{\delta q}$  when  $y$  is approaching zero, such that

$$R_{\delta q} = \text{Sat} \left( \frac{f_{\delta q} (N_p + f_{\mathbf{P}})}{N_u \delta \mathbf{q}^\top \delta \mathbf{q}} \bar{\mathbf{y}}^\top \mathbf{Q} \bar{\mathbf{y}}, R_{\delta q_{min}}, \infty, 0 \right), \quad (6.24)$$

where  $\text{Sat}(x, x_{min}, x_{max}, \alpha)$  represents the same saturation mechanism described in Eq. (5.1), with the distinction that it is applied here to a scalar value rather than a vector.

Looking at Eqs. (6.21)-(6.24), it is clear that in order to define all the MPC gains, one has to choose  $\mathbf{Q}$ ,  $f_{\mathbf{P}}$ ,  $f_u$ ,  $f_{\delta q}$ , and  $R_{\delta q_{min}}$ .

The choice of  $\mathbf{Q}$  is rather simple since all the ROEs can be weighted equally. However, in order to enhance the stability of the final relative orbit (i.e., to keep the obtained orbit for as long as possible without it being affected much by the perturbations), an emphasis is put on minimizing  $\delta a$  when the ROEs are close to the target ones, and  $\mathbf{Q}$  is defined as

$$\mathbf{Q} = \begin{cases} Q \mathbf{I}_6, & y \leq 1 \text{ m} \\ Q \begin{bmatrix} 100 & 0 \\ 0 & \mathbf{I}_5 \end{bmatrix}, & y > 1 \text{ m} \end{cases}, \quad (6.25)$$

where  $Q$  is a tuning parameter that controls the order of magnitude of the cost function.

Fixing the value of  $Q$  to  $10^5/a_c^2$ , of  $f_{\mathbf{P}}$  to 10, and of  $R_{\delta q_{min}}$  to  $10^{-5}$ , a brute force sen-

sitivity analysis over  $f_u$  and  $f_{\delta q}$  is carried out to choose their adroit values. Thanks to the [High-Performance Computer \(HPC\)](#) of the University of Luxembourg [127], 726 closed-loop MPC could be run simultaneously where, in addition to interchanging the values of  $f_u$  and  $f_{\delta q}$ , the seed of the [Random Number Generator \(RNG\)](#), which controls the initial state at the beginning of the simulation, could also be taken into account. Note that the details of the closed-loop system are yet to be presented in Section 6.3.3. In these simulations, and in the discussions to follow, warm-starting is employed at the beginning of each prediction horizon, meaning that the optimized state and control profiles from the previous prediction horizon are utilized to construct the initial guess for the current one.

Before starting the simulations, it was noted that the admissible values for  $f_u$  and  $f_{\delta q}$  belong to the period  $[0, 1]$ , since the most important component of the cost function is  $J_y$  and since the two other components are normalized with respect to  $J_y$  (see Eq. (6.21)). One simulation would have its [RNG](#) seed,  $f_u$ , and  $f_{\delta q}$  as one possible combination from the following sets:

$$\begin{aligned} \text{seed} &\in \{0, 1, 2, 3, 4, 5\}, \\ f_u, f_{\delta q} &\in \{0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.3, 0.6, 0.7, 0.9, 1\}, \end{aligned} \tag{6.26}$$

To this end, a fitness function is introduced in order to assess how good the output of the simulation is. Such fitness function in our context needed to address four criteria (performance metrics):

1. Driving the [ROE](#) vector to zero, which is the main goal of the [MPC](#) scheme to achieve orbit keeping. This is assessed through observing the [Root Mean Square \(RMS\)](#) of the dimensional [ROE](#) over the *finale* period of each simulation, denoted as  $y_{\text{fin}}$ .
2. Enhancing the stability of the relative orbit *finale* which, in other words, is minimizing the [RMS](#) of the last portion of the relative semi-major axis profile over time, denoted as  $\delta a_{\text{fin}}$ .
3. Minimizing the total Delta-V,  $\Delta V$  over the whole maneuver.

4. Reducing attitude effort, which is quantified by the mean angular rate of the satellite throughout the orbit correction maneuver,  $\omega_{\text{mean}}$ .

It is necessary to state that the term “finale” in this chapter refers to the last 10% of the simulation time span. Consequently, the formal definition of  $y_{\text{fin}}$  can be written as,

$$y_{\text{fin}} = \text{RMS}(\mathbf{y}(t_k)) \quad \forall t_k \in [t_0 + 0.9(t_f - t_0), t_f], \quad (6.27)$$

where  $t_0$  and  $t_f$  are the initial and final times of the closed-loop simulation, respectively. Similarly,  $\delta a_{\text{fin}}$  is formally defined as,

$$y_{\text{fin}} = \text{RMS}(\delta a(t_k)) \quad \forall t_k \in [t_0 + 0.9(t_f - t_0), t_f]. \quad (6.28)$$

Although this research is concerned with low-thrust propellers, the total Delta-V is considered instead of the total thrust in order to enable comparisons between the proposed scheme and those from the literature. The total Delta-V in our context can be calculated using the following formula:

$$\Delta V = \int_{t_0}^{t_f} u \cdot dt. \quad (6.29)$$

Given that the input thrust/acceleration from the single thruster is piece-wise-constant, the total Delta-V can be rewritten as follows:

$$\Delta V = \sum_{j=1}^{N_h} \left( \sum_{k=0}^{N_u-1} u_k T_s \right)_j, \quad (6.30)$$

with  $N_h$  being the number of receding horizons being processed within a simulation,  $j$  the horizon index, and  $T_s$  being the fixed sampling time.

Having introduced the fitness criteria, the adopted overall fitness function can then be written as,

$$\begin{aligned} \phi_{\text{tot}} = & K_y \cdot \underbrace{\text{fitness}(y_{\text{fin}})}_{\phi_y} + K_{\delta a} \cdot \underbrace{\text{fitness}(\delta a_{\text{fin}})}_{\phi_{\delta a}} \\ & + K_{\Delta V} \cdot \underbrace{\text{fitness}(\Delta V)}_{\phi_{\Delta V}} + K_y \cdot \underbrace{\text{fitness}(\omega_{\text{mean}})}_{\phi_{\omega}}, \end{aligned} \quad (6.31)$$

where  $K_y$ ,  $K_{\delta a}$ ,  $K_{\Delta V}$ , and  $K_y$  are weights that determine the importance of each of the four criterion, and the function  $\text{fitness}(\cdot)$  is defined as

$$\text{fitness}(x) = \frac{\frac{1}{x} - \min\left(\frac{1}{x}\right)}{\max\left(\frac{1}{x}\right) - \min\left(\frac{1}{x}\right)}, \quad (6.32)$$

with  $\min(\cdot)$  and  $\max(\cdot)$  being the minimum and maximum functions over all the 726 simulations. It is clear that this form of the fitness function only shoots out values between 0 and 1.

Running the simulations for the 726 combinations in Eq. (6.26) for a fixed initial orbit of the chief, which is parameterized by  $\tilde{\alpha}_{c,0} = \begin{bmatrix} 7121\text{km} & 0 \text{ deg} & 10^{-5} & 0 & 45 \text{ deg} & 0 \text{ deg} \end{bmatrix}^T$ , and applying the definition of the fitness function in Eq. (6.32) to all the 726 simulations and to the four performance metrics, and fixing the values of the weights to  $K_y = 5$ ,  $K_{\delta a} = 1$ ,  $K_{\Delta V} = 1$ , and  $K_y = 1$ , which reflect the high importance of minimizing  $y_{\text{fin}}$  in comparison to the other metrics, the heat maps that summarize all the simulations could be obtained and are presented in Fig. 6.3. The heat maps in Fig. 6.3 could only be obtained after averaging the fitness over each RNG seed in Eq. (6.26), and after adopting a scattered data interpolant in order to smooth the heat maps.

It is clear from the figures that the smaller  $f_u$  is, the better  $\phi_y$  and  $\phi_{\delta a}$  become, while this relation is conceivably inverted for  $\phi_{\Delta V}$ . Furthermore,  $\phi_{\Delta V}$  and  $\phi_{\omega}$  could be noticed to be positively correlated, and are both negatively correlated with  $\phi_y$  and  $\phi_{\delta a}$ . Further investigations on the simulations where  $\phi_{\Delta V}$  and  $\phi_{\omega}$  are both large revealed that minimal input acceleration is provided at an almost stagnant attitude of the satellite, while the effect of this acceleration on the orbit correction is, as well, minimal. Nonetheless, for these simulations, the ROE vector is approaching its set point, although very slowly.



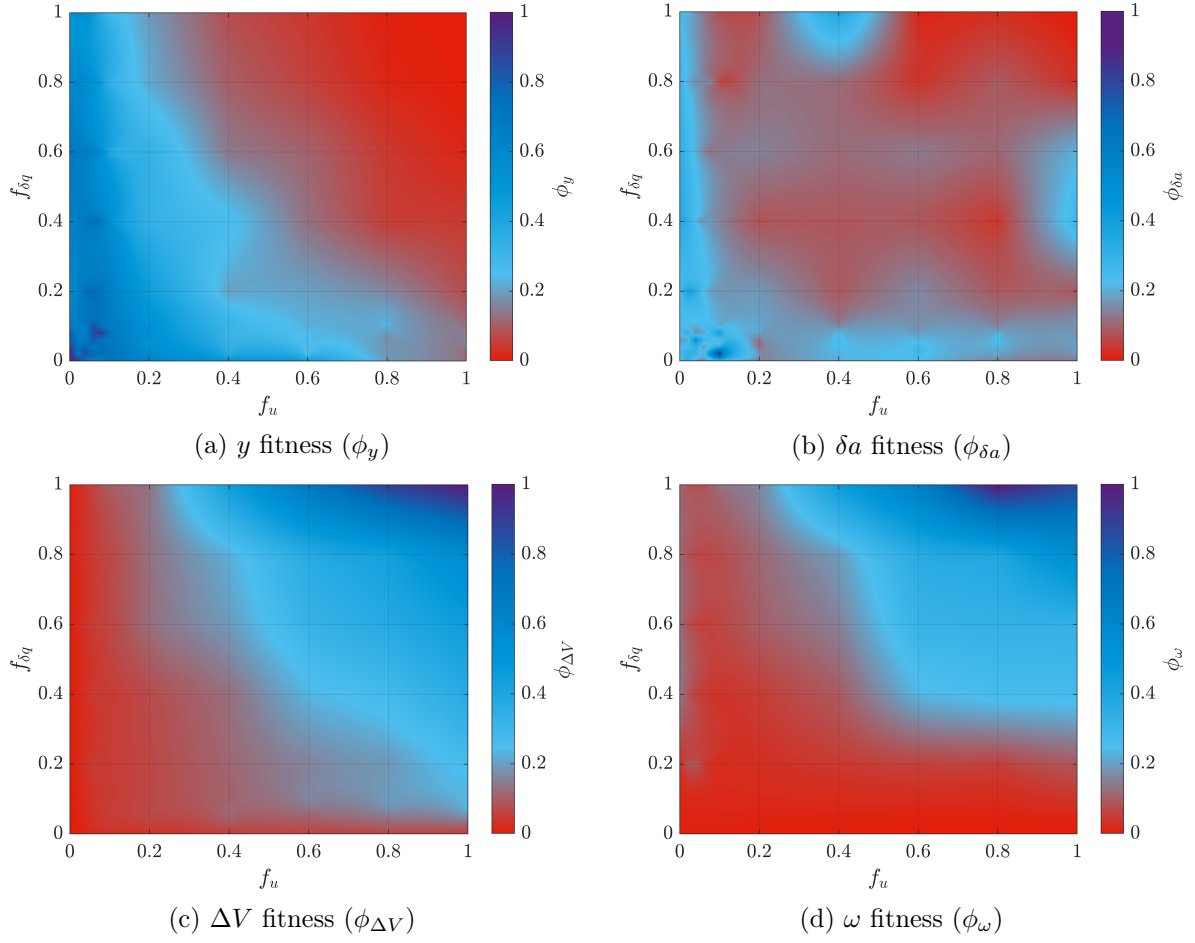


Figure 6.3: Fitness of the four performance metrics

The overall fitness function,  $\phi_{\text{tot}}$ , is depicted in Fig. 6.4 and the fittest point, i.e., the point with maximum overall fitness, ( $f_u = 0$  and  $f_{\delta q} = 0.02$ ) is marked with the gray circle on the heat map. It is clear from the fittest  $f_u - f_{\delta q}$  combination that, it was not necessary to include the direct Delta-V penalty,  $J_u$ , in the cost function, since  $J_y$  was already indirectly accounting for Delta-V cost as previously mentioned. One has to acknowledge that the fittest combination of  $f_u$  and  $f_{\delta q}$  in Fig. 6.4 is not necessarily universal for any arbitrary initial chief's orbit, and the selection of the optimal  $f_u - f_{\delta q}$  combination needs to be performed again for each initial orbit of the virtual chief. This does not represent a limitation of the approach, since the reference orbit of the chief is fixed once as soon as the satellite mission is designed.

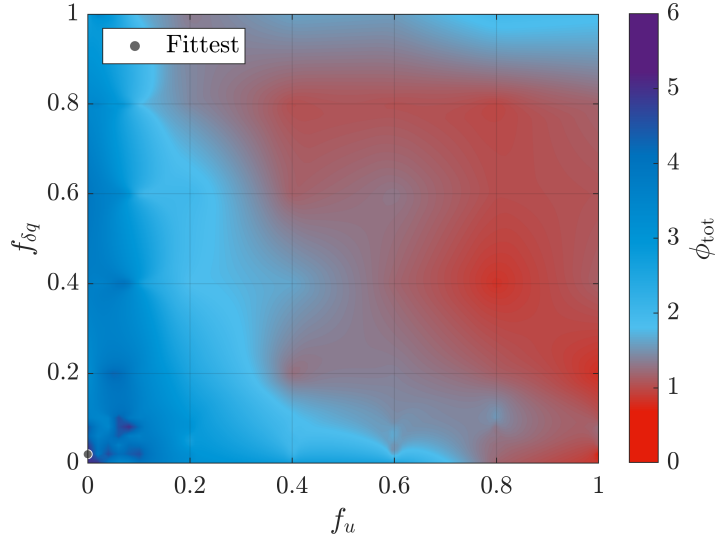


Figure 6.4: Overall fitness ( $\phi_{\text{tot}}$ )

The employed approach served the need to identify an adroit set of combinations of  $f_u$  and  $f_{\delta q}$ . Future investigations may focus on the setup of a heuristic approach, e.g., genetic algorithms, to search for the combination that globally maximizes the overall fitness function. In this context, note that the definition of the fitness functions and of the related parameters impact the global optimum.

Finally, the values of the optimized  $f_u$  and  $f_{\delta q}$  are heavily dependant on the authors' predictions of  $\bar{u}$  and  $\bar{\delta q}$ , and another choice of these quantities would have definitely led to different optimal  $f_u$  and  $f_{\delta q}$ ; however, the values of  $R_u$  and  $R_{\delta q}$  will not change since  $f_u$  is directly divided by  $\bar{u}^2$  to obtain  $R_u$  and the same dynamics apply for  $f_{\delta q}$  with the squared  $L_2$ -norm of  $\bar{\delta q}$  to obtain  $R_{\delta q}$  (see Eq. (6.23)).

### 6.3.3 Closing the Control Loop

The MPC described so far is one key module of the whole control loop, which, at practical implementation, requires feedback to operate properly. This feedback is provided by the navigation system, which, in turn, relies on sensors as well as a state estimation filter. The full closed loop, which illustrates the module execution logic onboard the deputy spacecraft,

is depicted in Fig. 6.5. There, Osc2Mean is the function that transforms osculating orbital elements to mean ones [17], Body2Inert is the method that rotates any vector from the body frame to the inertial frame, and OE2ROE is the function that transforms the absolute orbital elements of both the chief and the deputy to a ROE vector. The breve accent,  $(\breve{\cdot})$ , signifies a quantity which is affected by either one or a combination of the following sources of errors:

- Estimation errors (e.g.,  $\breve{\alpha}_d$ );
- Actuator errors (e.g.,  $\breve{q}$ );
- Physical constraints (e.g.,  $\breve{u}^b$ ).

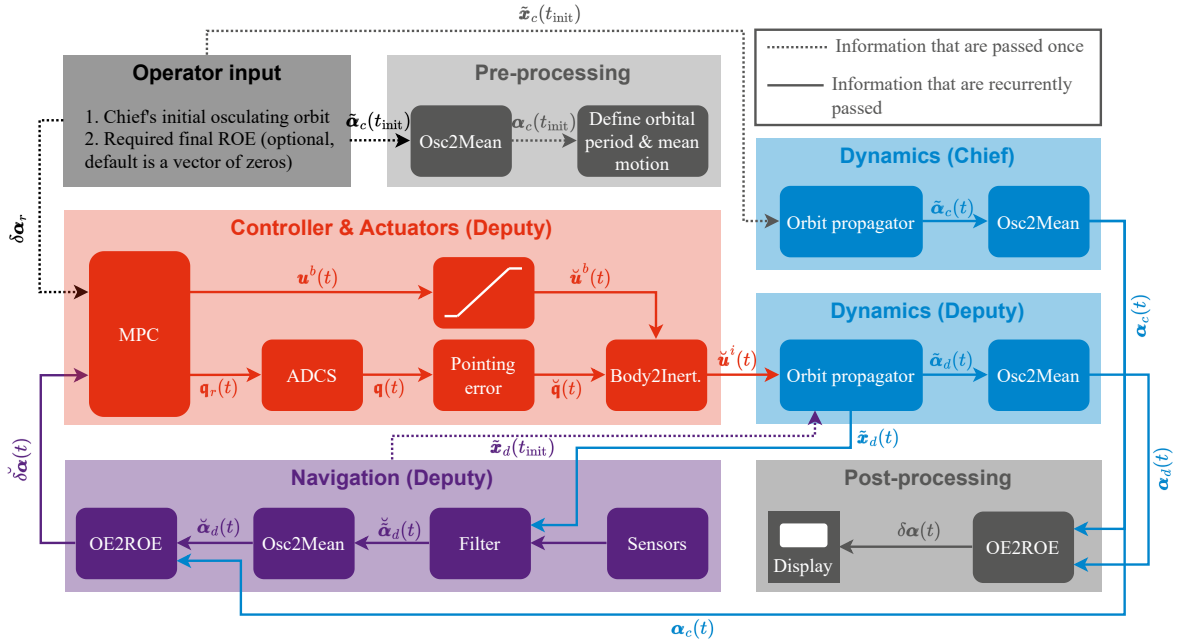


Figure 6.5: Closed loop of the deputy spacecraft

The proposed MPC is validated by means of numerical simulations, emulating the performance of the navigation module and the effects of the physical limitations of the adopted actuators. The physical limitations are only present in the saturation block (see Fig. 6.5). In fact, this saturation is taken into account in the guidance layer, but the saturation block after the MPC is implemented anyway as a safeguard in case the MPC computes an infeasible

solution. The "Navigation" set of blocks as well as the "Pointing error" block in Fig. 6.5 are modeled according to the surrogate models presented in Appendix G.

It is important to note that although the chief, in this context, is a virtual point, it is propagated using the perturbed dynamics for three main reasons:

- The dynamics in Eq. (2.48) are derived for a perturbed chief orbit;
- In many applications, the tracking of the reference orbit and the absolute control of the reference orbit are handled separately;
- The proposed scheme can be directly applied to the rendezvous with an actual spacecraft.

As a result, the autonomous orbit keeping compensates errors and perturbations acting on the relative dynamics.

## 6.4 MPC Validation

In order to test the stability of the proposed MPC scheme, and thanks to the parallelization capabilities of the HPC facilities of the University of Luxembourg, the algorithm was run over 500 different simulations, each with a different initial ROE vector. Specifically, the entries of the initial dimensional ROE vectors,  $\mathbf{y}_{i,0}$ , were randomly selected from the  $[-0.1 \ 0.1]$  km range, except for the initial relative mean argument of longitude,  $a_c \delta \lambda_{i,0}$ , which was randomly chosen from the  $[-1 \ 1]$  km range. Moreover, the initial orbit of the chief is fixed for all the 500 Monte Carlo runs, and is parameterized by  $\tilde{\boldsymbol{\alpha}}_{c,0} = [7121\text{km} \ 0 \text{ deg} \ 10^{-5} \ 0 \ 45 \text{ deg} \ 0 \text{ deg}]^T$ .

The simulation and MPC parameters employed in the 500-run Monte Carlo campaign are listed in Table 6.1. Notably, the parameters  $\sigma_{\mathbf{r}}$ ,  $\sigma_{\mathbf{v}}$ ,  $\sigma_{\mathbf{y}}$ , and  $\zeta_{\text{qpe}}$  are directly associated with the surrogate models for navigation and pointing errors, as detailed in Appendix G. For a detailed description of these parameters, the reader is encouraged to consult the aforementioned

appendix.

Table 6.1: Monte-Carlo simulation and MPC parameters

	$T_s$ [s]	$N_p$ [ $T_s$ ]	$N_u$ [ $T_s$ ]	$u_{max}$ [m/s <sup>2</sup> ]	$\omega_{max}$ [deg/s]
MPC parameters	50 s	60	15	$3.5 \times 10^{-5}$	2
	$Q$	$f_P$	$f_u$	$f_{\delta q}$	$R_{\delta q_{min}}$
	$10^5$	10	0.0	0.02	$10^{-5}$
Miscellaneous	$K$	$\sigma_r$ [m]	$\sigma_v$ [m/s]	$\sigma_y$ [m]	$\zeta_{q_{pe}}$ [arcs]
	0.2	10	0.5	1	25

It is to be noted that although the sampling time used in predicting the MPC states,  $T_s$ , is 50 s, the attitude dynamics are propagated each second in the simulations since the attitude changes much faster than the ROEs. Moreover, the value of the maximum allowable acceleration,  $u_{max}$ , was obtained from the information on Triton-X brochure [47] on both, the maximum thrust, which is set equal to 7 mN, and the mass of the satellite, which is assumed to be constant, 200 kg, throughout the maneuver.

The main performance metrics over the 500 Monte Carlo runs are depicted in Fig. 6.6, while the statistics of these metrics are presented in Table 6.2.

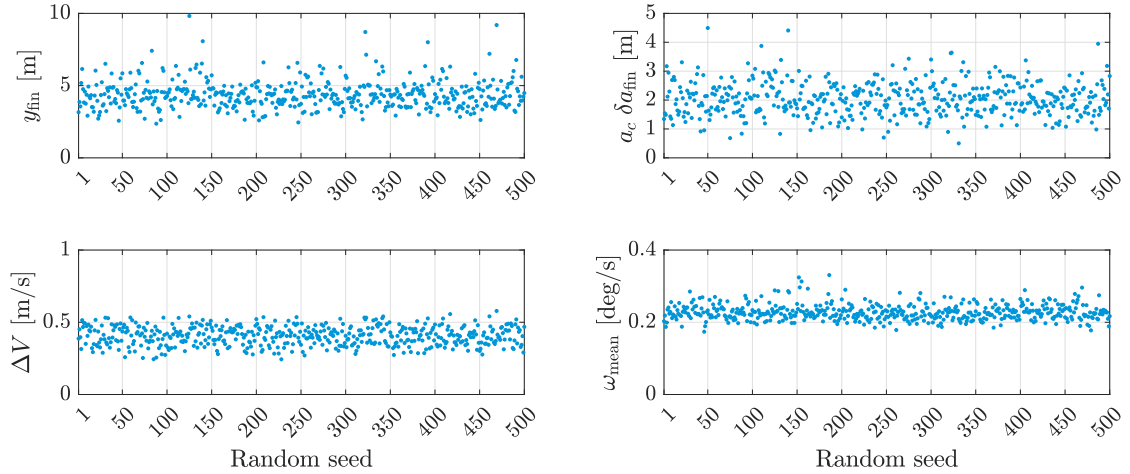


Figure 6.6: Performance metrics over the Monte Carlo simulation

Table 6.2: Performance metrics over 500 simulations

	$y_{\text{fin}}$ [m]	$a_c \delta a_{\text{fin}}$ [m]	$\Delta V$ [m/s]	$\omega_{\text{mean}}$ [deg/s]
Mean	4.38	2.05	0.406	0.228
Median	4.29	2.02	0.409	0.226
Max	9.82	4.49	0.58	0.331

In order to gain an insight into how the proposed scheme performs, the results of an arbitrary simulation out of the 500 simulations are presented. The randomly selected initial dimensional ROE vector is  $\mathbf{y}_0 = \begin{bmatrix} 64.62 & -947.77 & -57.84 & 23.68 & -80.35 & 24.03 \end{bmatrix}^\top$  m, while the the reference dimensional ROE vector is  $\mathbf{y}_r = \mathbf{0}$ .

The ROE states are seen to approach their set points (zero for rendezvous with the virtual target) in Fig. 6.7. It is of significance to mention that the chief's mean orbital period, which is the unit of time in Fig. 6.7, is 5974.46 s.

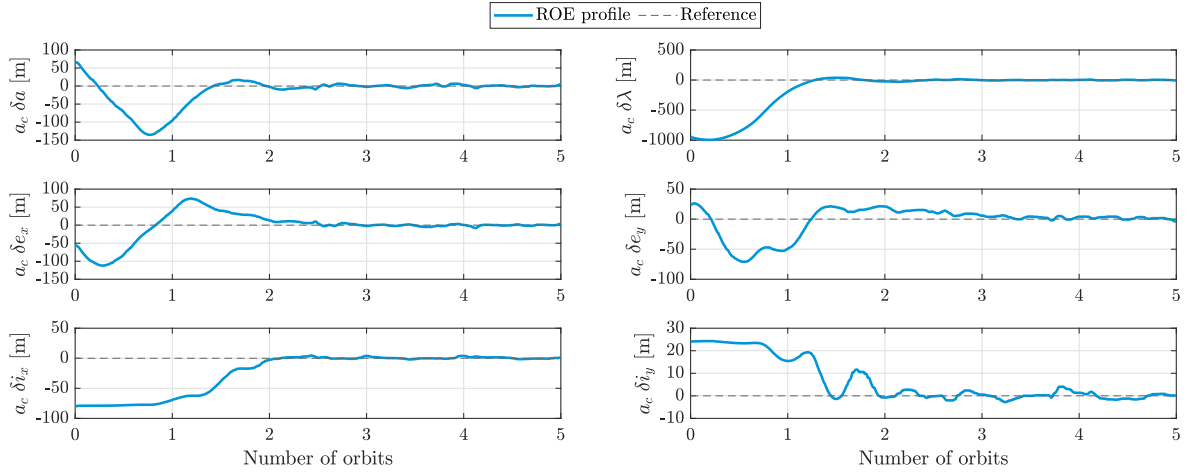


Figure 6.7: ROE profile of the selected Monte Carlo run

The trajectory followed by the satellite is depicted in Fig. 6.8 in both the Transversal-Radial and the Normal-Radial planes. This trajectory could be obtained by transforming the ROE to the relative Cartesian states in the RTN frame using the transformation in Eq. (2.56). It can be seen from Fig. 6.8a that the orbit maneuver is taking place gradually thanks to the

continuous firing of the thruster, in contrast to the impulsive-thrust maneuvers where the velocity changes at a handful of points on the orbit.

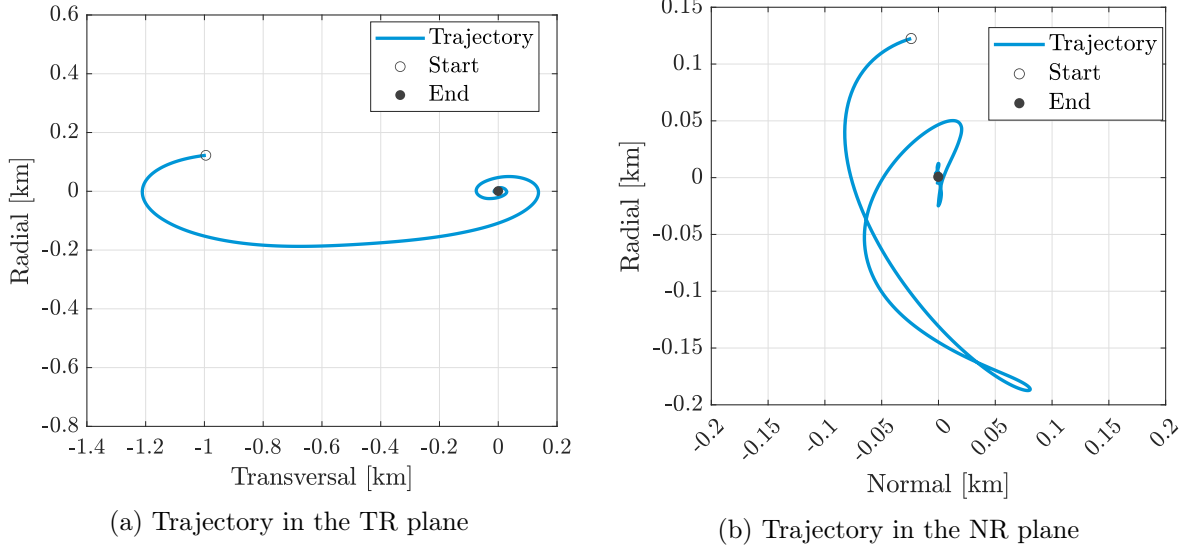


Figure 6.8: Trajectory followed by the satellite of the selected Monte Carlo run

The thrust exerted by the onboard thruster is depicted in Fig. 6.9. This thrust is also projected on the RTN frame and the projection is presented in Fig. 6.10. It is interesting to see that the MPC does not follow a trajectory that minimizes the radial burns, which has been a usual constraint imposed on the MPC in earlier studies [85]. Although firing in the radial direction to correct  $\delta\lambda$  errors is known to be more Delta-V-expensive than firing in the transversal direction to build up  $\delta a$  drift, which takes care of correcting the  $\delta\lambda$  error by exploiting the natural dynamics, this is not the case for close-proximity maneuvers where the initial  $\delta\lambda$  error is small. This can be verified by looking at the solution to the linearized forced Gauss variational equations [28].

In Fig. 6.11a, the error quaternion angle signal suggests that at each optimization step, the attitude starts from a value that does not coincide with the reference attitude. Nevertheless, by the end of each step, the attitude gradually converges towards the reference one, resulting in the error quaternion angle approaching a value close to zero. A very interesting feature of the proposed scheme is that the propulsion system is always turned on even when attitude redirection maneuvers are taking place. The optimizer calculates the reference attitude know-

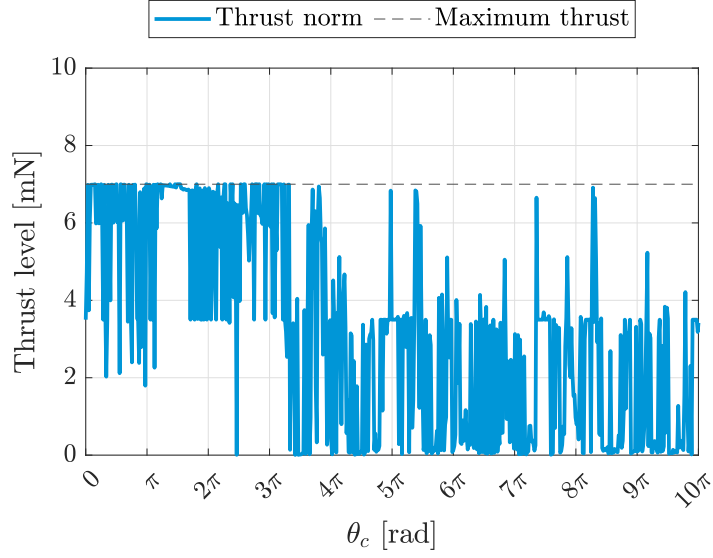


Figure 6.9: Thrust output of the single thruster

ing that the satellite will not necessarily be aligned with it for the whole upcoming control step.

It is worth noting that the angle of any quaternion (e.g., the error quaternion,  $\mathbf{q}_e$ , or the current quaternion,  $\mathbf{q}$ ) can be found as follows:

$$\zeta_{\mathbf{q}} = \arctan\left(\frac{\|\mathbf{q}\|_2}{q_0}\right), \quad (6.33)$$

where  $\mathbf{q}$  and  $q_0$  are the vector and the scalar parts of the quaternion  $\mathbf{q}$ , respectively. Lastly, the effect of adding the  $J_{\delta\mathbf{q}}$  term to the cost function of Problem 6.1 could be seen clearly in Fig. 6.11b, where the quaternion angle rate is depicted. The quaternion angle rate, which can be thought of as a signed version of  $\omega$ , is seen to be much lower than its maximum allowable values,  $2^\circ/\text{s}$ . Our preliminary simulations, which did not contain the term  $J_{\delta\mathbf{q}}$  on their cost function, always had the attitude rate saturating at its maximum value, which is a hyper-reaction that is not desired in a real mission.

In order to test the efficiency and the optimality of the proposed MPC scheme, it is benchmarked against the approach proposed in [85] for a similar problem, where an Out-Of-Plane (OOP) relative orbit correction maneuver had to be performed. The initial orbit of the chief



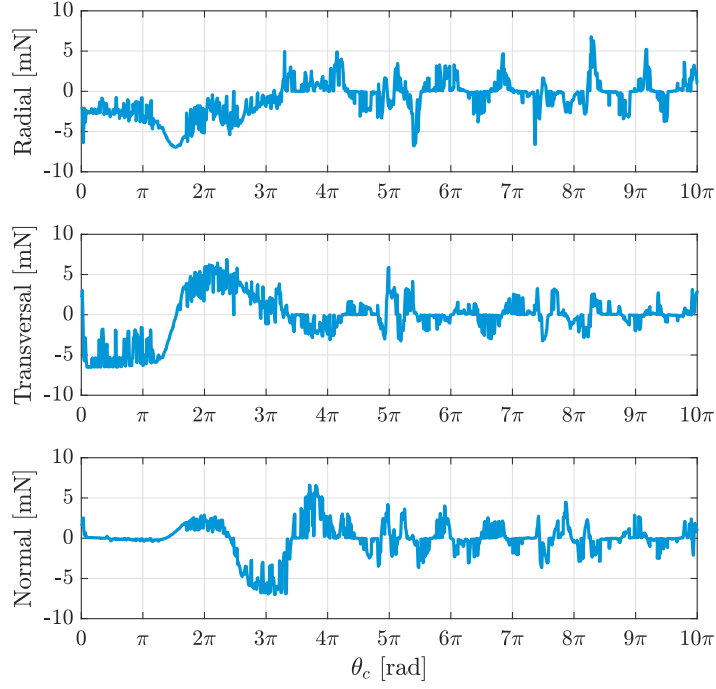


Figure 6.10: Thrust vector in the RTN frame

in this benchmark simulation is set to  $\tilde{\alpha}_{c,0} = \begin{bmatrix} 6828\text{km} & 0 \text{ deg} & 10^{-5} & 0 & 78 \text{ deg} & 0 \text{ deg} \end{bmatrix}^T$ , where the initial and the reference dimensional ROE are set to  $\mathbf{y}_0 = \begin{bmatrix} 0 & 0 & 273 & 0 & 10 & 70 \end{bmatrix}^T$  and  $\mathbf{y}_r = \begin{bmatrix} 0 & 0 & 273 & 0 & 400 & 120 \end{bmatrix}^T$ . The rest of the parameters are set to their values from Table 6.1, except for the maximum allowable acceleration, which is set to  $u_{max} = 32 \mu\text{m/s}^2$ . Applying this maximum acceleration to the satellite employed in [85], which has a mass of 20 kg, renders the maximum thrust as 0.64 mN.

The setting of this maneuver is interesting not only because it allows two different MPC schemes to be compared, but also because OOP maneuvers require only normal acceleration. In fact, the Delta-V-optimal locations for impulsive normal burns for such maneuvers could be analytically calculated [77]. For the case of electric propulsion, a controller is said to be Delta-V-optimal for out-of-plane maneuvers if the thrust in the normal direction is bang-bang-like and is distributed almost evenly around the Delta-V-optimal locations. The thrust profile of the proposed MPC is depicted in Fig. 6.12a, which reveals that the proposed MPC

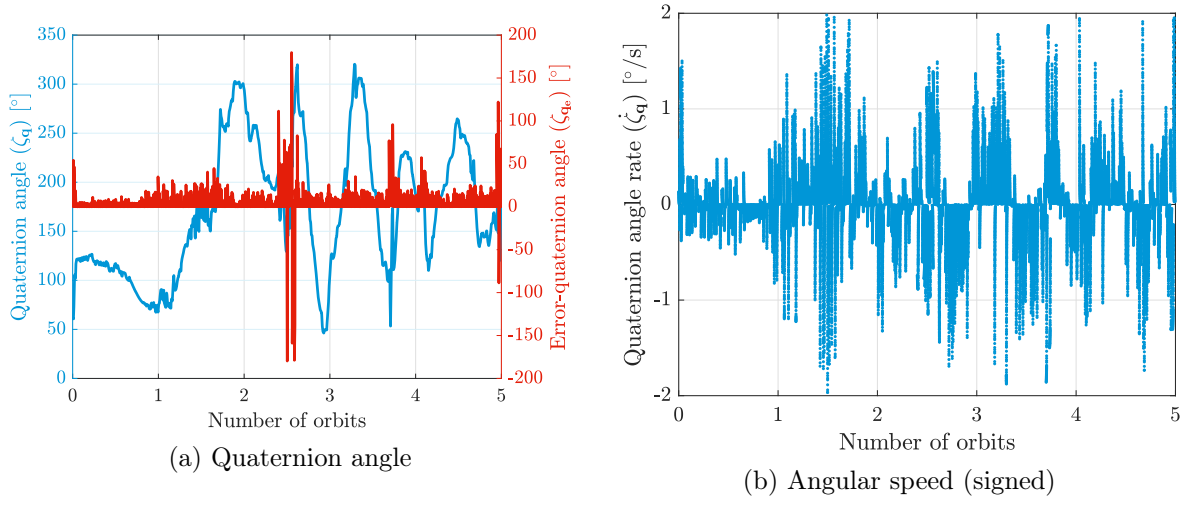


Figure 6.11: Attitude profile of the selected Monte Carlo run

behaves as a fuel-efficient scheme would be expected to behave.

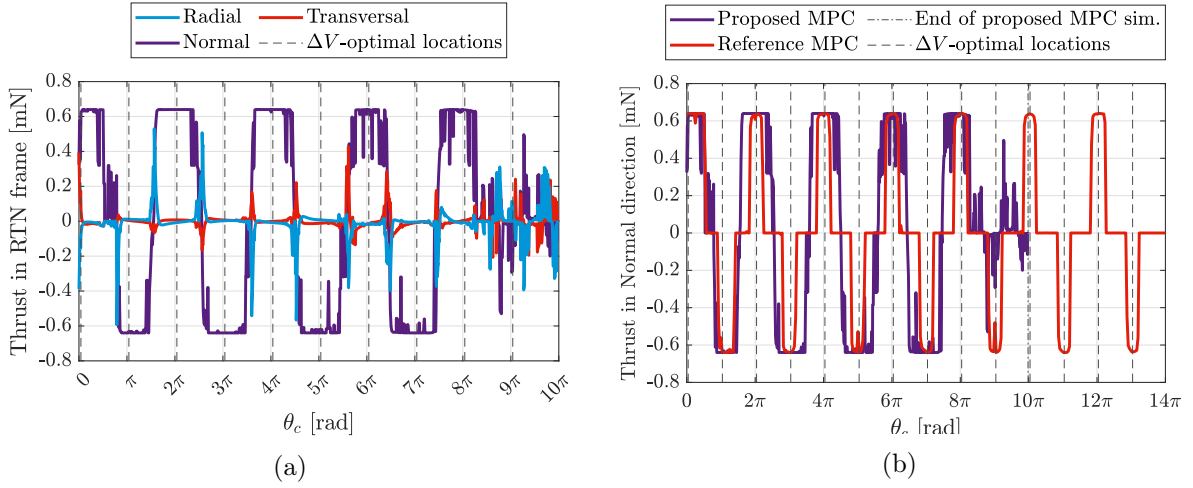


Figure 6.12: Thrust level for the benchmark simulation: (a) thrust vector in the RTN frame (b) comparison of the thrust in the normal direction

It can also be seen from Fig. 6.12a that the thrust components in the radial and the transversal directions are not exactly zero due to in-plane perturbation compensation and also owing to the fact that the thrust is provided during slew maneuvers. Therefore, the radial and transversal thrust components become more visible when the thrust direction is required to flip from the positive to the negative normal direction. In this situation, the

satellite has to pass by many transient attitudes for which the radial and the transversal components are not necessarily zero. Fig. 6.12b depicts a comparison between the normal component of the thrust profile of the proposed MPC scheme and the MPC scheme proposed by [85].

The dimensional ROE profiles of the two MPCs are also shown in Fig. 6.13, and a brief comparison between the performance of both schemes is presented in Table 6.3, where the convergence time is defined as the time it takes all elements in the error-dimensional ROE vector,  $\mathbf{y}_{\text{err}} = \mathbf{y} - \mathbf{y}_r$ , to be less than 5 [m].

Table 6.3: Comparison between the proposed and the reference MPCs

	Convergence time [orbits]	Terminal $\mathbf{y}_{\text{err}}$ [m]	$\Delta V$ [m/s]
Proposed MPC	4.1	$[0.9 \quad -0.4 \quad 1.9 \quad -0.33 \quad 0.0 \quad -0.5]^\top$	0.664
Reference MPC	6.6	$[-3.6 \quad 9.2 \quad -1.4 \quad 2.0 \quad -2.9 \quad 1.6]^\top$	0.501

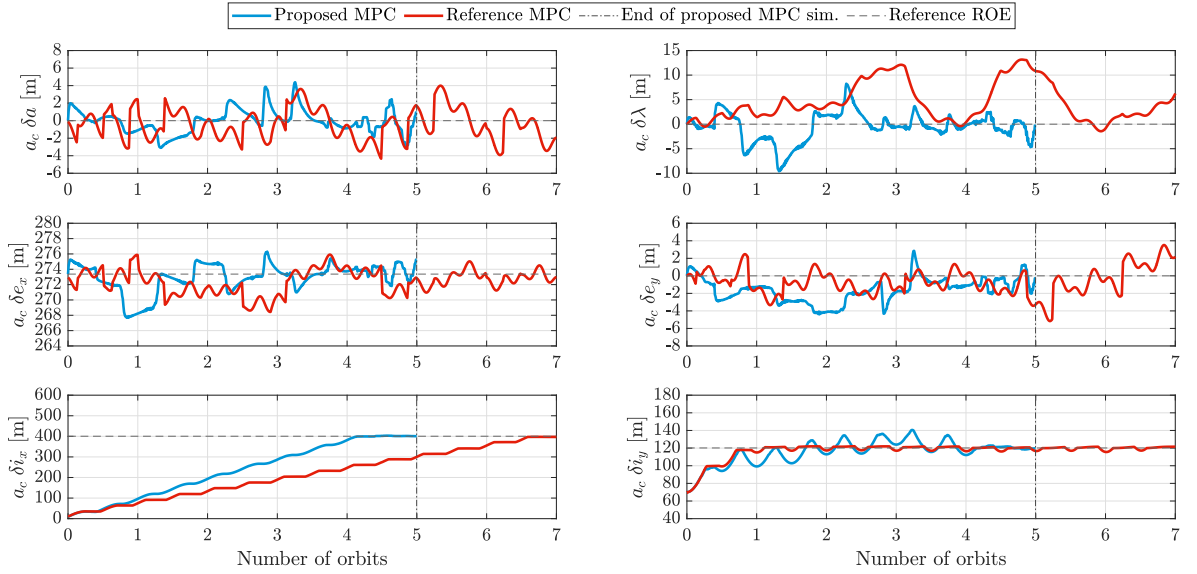


Figure 6.13: ROE profile of the benchmark maneuver

It is clear from Table 6.3 as well as Fig. 6.13 that, for the benchmark simulation, the proposed scheme is much faster, in fact 1.61 times as fast as that of [85]. Furthermore,

the proposed MPC appears to be much more precise than the reference one; however, it consumes 24.04% more Delta-V, as seen in Table 6.3. Indeed, consuming more Delta-V than the reference MPC is conceivable since the proposed MPC is not designed to be Delta-V-optimal in the first place. Again, the cost function of Problem 6.1 has two components (after eliminating  $J_u$  as discussed in Section 6.3.2): the first one,  $J_y$ , deals with gradually approaching the reference ROE vector, and implicitly implies a trade-off between fuel and time optimality (depending on  $f_P$ ), while the second component,  $J_{\delta q}$ , is added to minimize the attitude change throughout the maneuver. It is believed that a smaller total Delta-V could be achieved using a different set of MPC gains; however, it would be at the cost of more abrupt attitude changes and/or more maneuvering time. The ROE profile in Fig. 6.13 implies the evolution of the OOP variables,  $\delta i_x$  and  $\delta i_y$ , from their initial values to their set points, while the in-plane variables fluctuate slightly around their initial/reference values as a response to the relative orbital perturbations as well as to the unavoidable radial and transversal accelerations during each attitude flip from the positive to the negative normal direction and vice-versa.

Lastly, Fig. 6.14 relates to the attitude evolution of the system, where Fig. 6.14a shows how the attitude is changing smoothly to recurrently flip the thrust direction from/to the positive to/from the negative normal direction, while the error quaternion angle is starting from a nonzero value only when the thrust direction is required to be flipped, and is seen to always approach zero. Moreover, although the objective is clearly to flip the thrust direction, the error quaternion angle is never  $180^\circ$ , which could only mean that the optimizer chooses the reference quaternion to evolve smoothly until it reaches the flipped attitude so that a hyper-reaction of angular velocity can be avoided. The quaternion angle rate is seen in Fig. 6.14b to turn swiftly and almost reach its full potential, i.e.,  $\pm\omega_{max}$ , when the thrust direction is set to change from/to the positive to/from the negative normal direction.

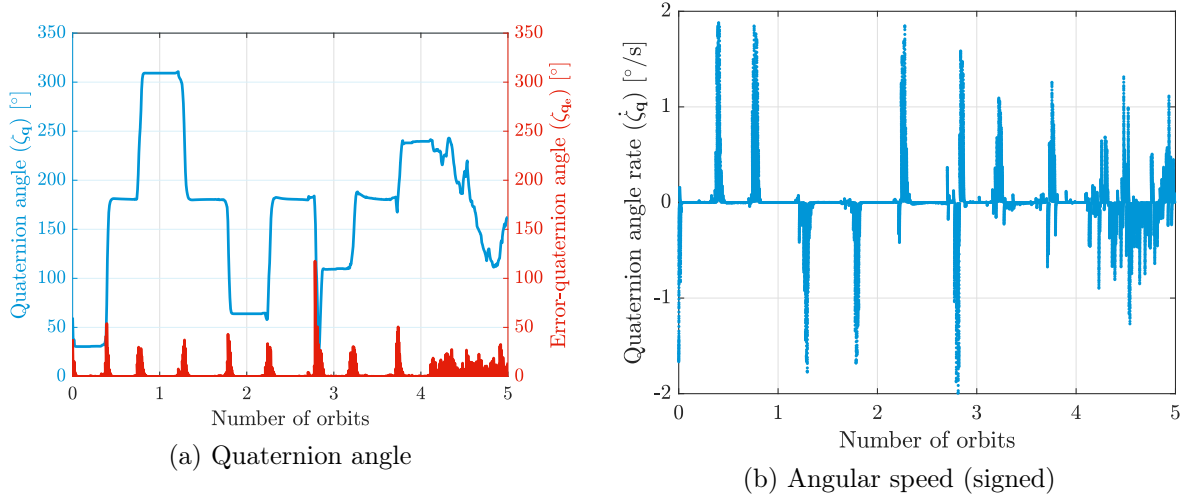


Figure 6.14: Attitude profile of the benchmark maneuver

## 6.5 Conclusions

This chapter presents an [MPC](#) scheme to solve the problem of optimal absolute orbit keeping for underactuated satellites equipped with electric propulsion systems. Many small-satellite platforms are currently equipped with unidirectional thrusters, which require continuous attitude slews during extended orbit maneuvers to redirect the thrust in the desired propulsion direction. Although the controller is designed for a single satellite, formation flying techniques are adopted by assuming the satellite flies in formation with a virtual spacecraft (following the reference trajectory). The satellite is tasked with maintaining a predefined relative orbit with respect to this virtual chief, typically achieving rendezvous. This approach enables the direct extension of the proposed algorithm to multi-satellite relative orbit-keeping scenarios.

The proposed [MPC](#) is designed to balance fuel and time optimality by tuning the controller gains. Importantly, it couples the attitude dynamics and relative orbital dynamics, ensuring that thrust is provided even during attitude redirection maneuvers. The control scheme's stability is validated through Monte Carlo numerical simulations that emulate navigation errors, hardware inaccuracies, and physical constraints.

To evaluate the controller's performance, the proposed MPC is compared to a reference MPC from the literature. The results demonstrate that the proposed scheme achieves more accurate orbit keeping in a shorter time frame. This performance improvement comes at a marginal increase in Delta-V, approximately 24% of the total maneuver cost. However, this increase is considered negligible at the mission level, given the significant gains in orbit-keeping accuracy and time efficiency.

## Conclusion

This thesis presents a comprehensive [Guidance, Navigation, and Control \(GNC\)](#) toolbox developed for formation flying missions of the Triton-X multi-mission satellite platform and similar satellites. The research addresses the key challenges imposed by the hardware specifications of Triton-X, which significantly influence the design of the [GNC](#) system. Specifically, the thesis develops novel relative navigation algorithms, formation reconfiguration guidance and control approaches, and absolute/relative orbit-keeping strategies.

The navigation system faces challenges due to Triton-X's reliance on a single-frequency [Global Navigation Satellite System \(GNSS\)](#) receiver. For close formations with intersatellite distances under 10 km, existing [GNSS](#)-based relative navigation algorithms are suitable, as this area has been extensively studied. However, for widely separated spacecraft with single-frequency receivers, ionospheric delay becomes a dominant bias. To address this, the thesis presents a precise bi-linear ionospheric delay model integrated into a novel [Extended Kalman Filter \(EKF\)](#). Using real sensor data from the [European Space Agency \(ESA\)](#) Swarm mission under the best and worst ionospheric conditions in the [24<sup>th</sup> Solar Cycle \(SC24\)](#), the filter achieved consistent performance. Even under the worst conditions, the filter achieved [Root Mean Square \(RMS\)](#) errors of 26 cm in relative distance and 4 cm/s in relative speed, demonstrating exceptional accuracy for many technology demonstration missions.

The guidance and control system faces the challenge of Triton-X’s single, ungimbaled electric thruster, which renders the orbit control system underactuated. This configuration requires attitude redirection maneuvers before each thrust activation. To address this, the maneuver span is split into multiple control cycles. Each control cycle consists of a thrust arc followed by a coast arc. During the latter, an attitude redirection maneuver is performed to align the thruster nozzle with the desired direction. The formation reconfiguration problem is divided into guidance and control subproblems, where the guidance problem is formulated as a convex trajectory optimization problem that incorporates operational constraints, including collision avoidance and thrust limits. Multiple formulations of the guidance problem are proposed, varying in computational complexity and fuel efficiency. Centralized and distributed frameworks are developed, and benchmarking across more than a dozen solvers reveals that the [Second-Order Cone Programming \(SOCP\)](#) formulation is the most fuel-efficient while maintaining computational efficiency. This formulation is adopted as the prediction function for two [Model-Predictive Control \(MPC\)](#) strategies: the [Shrinking-Horizon MPC \(SHMPC\)](#) and the [Fixed-Horizon MPC \(FHMPC\)](#).

The [SHMPC](#) and [FHMPC](#) approaches were evaluated in both centralized and distributed settings. The [FHMPC](#) excels in tracking pre-optimized reference trajectories for short maneuvers, achieving high accuracy and low Delta-V consumption. However, the [SHMPC](#) demonstrated greater flexibility, adapting to disturbances and achieving superior performance in long-duration or disturbance-rich scenarios. Consequently, [SHMPC](#) is recommended for long reconfiguration maneuvers or when external disturbances are significant. Both [MPC](#) strategies outperform a reference [MPC](#) from the literature in terms of final control accuracy and Delta-V efficiency.

The thesis also addresses absolute/relative orbit-keeping, prioritizing time efficiency over fuel-optimality. To this end, an [MPC](#) strategy is proposed that allows thrust application during attitude maneuvers. This strategy employs a [Non-Linear Programming \(NLP\)](#) problem



with a multi-objective cost function, balancing fuel efficiency, attitude control effort, time-optimality, and precision. Compared to a reference [MPC](#), the proposed strategy is shown to achieve more precise orbit-keeping in less time, albeit with a slight increase in Delta-V consumption.

The primary outcome of the research in this thesis is a Matlab toolbox that is readily available for use by LuxSpace. The developed codes can be automatically transformed into embedded software, making them suitable for integration as part of the complete Triton-X software framework. In fact, embedded C-code was successfully generated for one formulation of the guidance problem for multi-satellite formations, alongside the [FHMPC](#) approach for closing the control loop. This C-code was staged for implementation on the [Digital Mother Boards \(DMBs\)](#), which act as the onboard computers for multiple Triton-X satellites. However, the implementation could not be realized within the time frame of this doctoral study due to the unavailability of the [DMBs](#), as they were allocated for [Verification and Validation \(V&V\)](#) experiments related to another mission. The implemented Matlab toolbox allows for automatic code generation using *Matlab Coder*. The only remaining work involves modifying the toolbox interfaces to ensure compatibility with the complete Triton-X software architecture.

## Part III

# Appendices

## Extended Kalman Filter

The [Extended Kalman Filter \(EKF\)](#) is an extension of the well-known [Kalman Filter \(KF\)](#). Unlike the [KF](#), the [EKF](#) combines information from the process and the observation models which can be nonlinear models of the states. An [EKF](#) aims at giving an estimate of the state vector  $\mathbf{x}$  assuming the following nonlinear process and measurement models.

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \quad (\text{A.1})$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \quad (\text{A.2})$$

where  $\mathbf{f}$  and  $\mathbf{h}$  are the state transition and the measurement models, respectively, which are generally nonlinear. If  $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$  and  $\mathbf{h}(\mathbf{x}_k)$  were linear mappings, the [EKF](#) reduces to a [KF](#). The vector  $\mathbf{x}_k$  represents the state vector at time instant  $k$  which is to be estimated by the [EKF](#),  $\mathbf{u}_k$  is the known/estimated control vector, and  $\mathbf{w}_k$  and  $\mathbf{v}_k$  are the process and the measurement noises, respectively, which are assumed normally distributed multivariate random variables, i.e.,  $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$  and  $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ , with  $\mathcal{N}(\mu, \sigma^2)$  being the normal distribution with a mean  $\mu$  and a variance  $\sigma^2$ .

In the following discussion, the breve accent  $\breve{(\cdot)}$  signifies a quantity that is produced by the filter (i.e. an estimation), while the double subscript  $(\cdot)_{m|n}$  represents the estimation of

$(\cdot)$  at the time instant  $m$  given the measurements up to and including the time instant  $n$ , where  $n \leq m$ .

The [EKF](#) is typically divided into two successive phases, the prediction and the update. The prediction phase is model-dependant, and is depicted in the following set of equations.

$$\begin{aligned}\check{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\check{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \\ \check{\mathbf{P}}_{k|k-1} &= \mathbf{F}_k \check{\mathbf{P}}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k,\end{aligned}\tag{A.3}$$

where,

$$\mathbf{F}_k := \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\check{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k}.\tag{A.4}$$

The update phase can as well be summarised by the following equations.

$$\begin{aligned}\mathbf{K}_k &= \check{\mathbf{P}}_{k|k-1} \mathbf{H}_k^\top \left( \mathbf{H}_k \check{\mathbf{P}}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \right)^{-1} \\ \check{\mathbf{x}}_{k|k} &= \check{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\check{\mathbf{x}}_{k|k-1})) \\ \check{\mathbf{P}}_{k|k} &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_{k|k-1},\end{aligned}\tag{A.5}$$

where

$$\mathbf{H}_k := \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\check{\mathbf{x}}_{k|k-1}}.$$

# Appendix B

## Linearization of a general difference function

Consider the general vector valued function  $\mathbf{f}(\mathbf{x})$  and the following difference function,

$$\Delta \mathbf{f} = \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_0). \quad (\text{B.1})$$

The differential vector valued function  $\Delta \mathbf{f}$  can be approximated by Taylor expansion around  $\mathbf{x}_0$ . Taking into account that  $\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_0$ , and ignoring the higher order terms, the Taylor approximation of  $\Delta \mathbf{f}$  around  $\mathbf{x}_0$  can be written as,

$$\Delta \mathbf{f} \approx \left( \mathbf{f}(\mathbf{x}_0) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0} \Delta \mathbf{x} \right) - \mathbf{f}(\mathbf{x}_0) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0} \Delta \mathbf{x}, \quad (\text{B.2})$$

which, assuming  $\frac{\partial \Delta \mathbf{f}}{\partial \Delta \mathbf{x}} \approx \frac{\Delta \mathbf{f}}{\Delta \mathbf{x}}$ , suggests that,

$$\frac{\partial \Delta \mathbf{f}}{\partial \Delta \mathbf{x}} \approx \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_0}. \quad (\text{B.3})$$

This approximation is valid as long as  $\|\Delta \mathbf{x}\|$  is small in comparison to  $\|\mathbf{x}_0\|$ .

## Estimation of the scaling factor used in the affine relaxation of the maximum acceleration constraints

In order to put Problem 4.4 into the [Linear Programming \(LP\)](#) form, the [Second-Order Cone \(SOC\)](#) constraint in Eq. (4.27) had to be transformed into multiple piece-wise affine constraints. Specifically, Eq. (4.27) defines a constraining sphere with circular projections on the Transversal-Normal (T-N), Radial-Transversal (R-T), and Radial-Normal (R-N) planes. These circular projections were approximated by regular polygons, where the polygons in the R-T and R-N planes form rhombuses, and the number of sides for the polygon in the T-N plane,  $n_{\text{dir}}$ , is left to the user to choose. This transformation is graphically presented in Fig. 4.3. It is important to highlight that the union of all the affine constraints forms a constraining polyhedron in the 3D space, which is depicted in Fig. 4.4 for  $n_{\text{dir}} = 12$ . A notable aspect of this piece-wise affine relaxation is that, although the approximating polyhedron is designed to have projections on the three planes that remain within the original constraining sphere, the entire polyhedron is not necessarily contained within the sphere. Therefore, in Eq. (4.44) and Eq. (4.45), and in order to ensure that the constraining polyhedron lies with its entirety inside the original constraining sphere, the relaxed constraints had to be scaled

using a constant scaling factor,  $c$ , which represents the ratio of the distance from the center of the sphere to the furthest point on the polyhedron relative to the radius of the sphere. In this appendix, a method to estimate the scaling factor,  $c$ , is presented for different values of  $n_{\text{dir}}$ .

Given the value of  $n_{\text{dir}}$ , one approach to calculating the scaling factor involves using a 3D modeling package to model both the polyhedron and the sphere. By doing so, the distance from the center of the sphere to the furthest vertex of the polyhedron can be measured directly. However, this method is not scalable, as it requires constructing a 3D model for each specific value of  $n_{\text{dir}}$ . A more precise method takes advantage of the fact that the polyhedron is formed by the intersection of the  $n_{\text{dir}} + 8$  planes that define the piece-wise linear constraints. The scaling factor  $c$  can be calculated accurately by constructing a convex hull from the vertices created by the intersection of each pair of planes with one of the three principal planes (R-T, R-N, and T-N). While this method guarantees precision, it requires implementing a 3-plane intersection algorithm. An alternative method involves a Monte Carlo simulation, where a large number of random points is generated within the cube that circumscribes the constraining sphere. Each point is tested to determine whether it lies within the polyhedron. The points that are inside the polyhedron are kept, and their distances from the origin are calculated. The maximum distance found is divided by the radius of the sphere to calculate the scaling factor. This method offers a rather easy implementation, however, it does not guarantee the precision of the scaling factor. Therefore, in this study, the Monte Carlo method is adopted, with an additional safety margin added to the calculated scaling factor to account for any potential inaccuracies.

The Monte Carlo simulation was run for 9 different values of  $n_{\text{dir}}$ , namely  $\{4, 6, \dots, 20\}$ , and the results of these experiments are summarized in Table C.1. The table also includes the ratio of the area of relaxing polygon in the T-N plane relative to the area of the original constraining circle.

Table C.1 indicates that the rhombuses used to approximate the circles in the R-T and

Table C.1: Monte Carlo results over different values of  $n_{\text{dir}}$

$n_{\text{dir}}$	Area ratio [%]	$c$
4	63.7	0.999
6	82.7	1.008
8	90.0	1.041
10	93.5	1.018
12	95.5	1.016
14	96.7	1.027
16	97.4	1.040
18	98.0	1.032
20	98.4	1.031

R-N planes cover approximately 64% of the area of the corresponding constraining circles. In contrast, the smallest polygon that covers at least 95% of the area of the constraining circle in the T-N plane is the polygon with 12 sides. The choice of  $n_{\text{dir}} = 12$  has additional advantages beyond just covering 95% of the constraining area in the T-N plane; it also requires the least amount of scaling among all the cases that cover at least 95% of the circumcircle.

Fig. C.1 depicts the results of the Monte Carlo simulation over the case when  $n_{\text{dir}} = 12$ . The resulting scaling factor of this experiment is 1.016, which is very close to the one calculated by the 3D modeling technique in Chapter 4, which was calculated to be 1.0166, and was approximated to 1.017. Indeed, the Monte Carlo result is less precise.



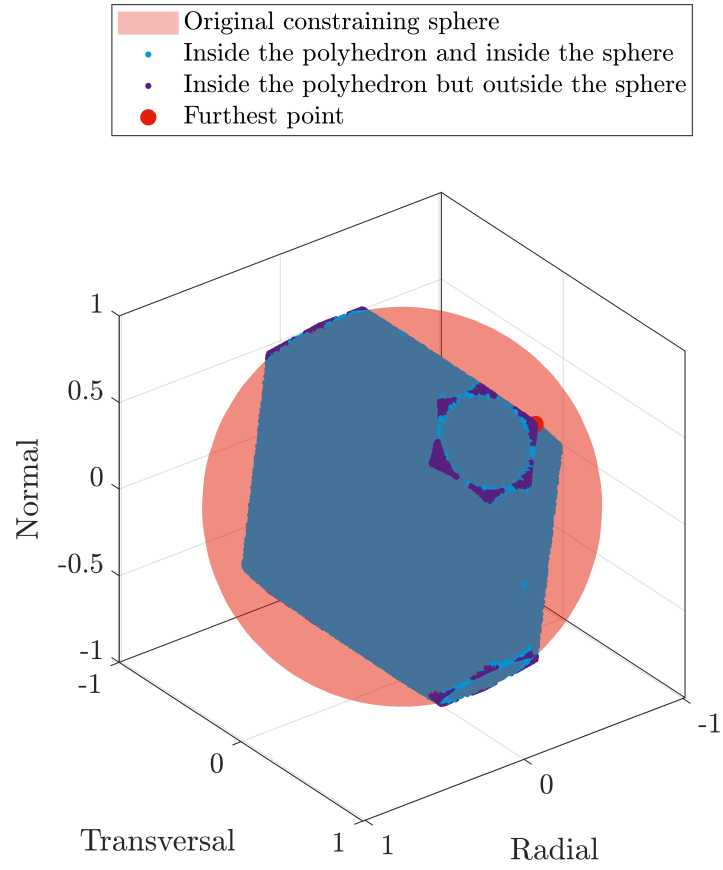


Figure C.1: Monte Carlo results for  $n_{\text{dir}} = 12$

## Reconfiguration scenarios used throughout the thesis

This appendix outlines the reconfiguration scenarios utilized throughout this thesis. Each scenario is assigned a unique identifier, referred to as Reconfiguration  $i$  or the  $i^{\text{th}}$  reconfiguration scenario, which is used consistently throughout the text.

### Reconfiguration 0 - Coplanar to PCO

This scenario involves a formation of four identical satellites flying in close proximity to a chief satellite. Initially, the deputies are distributed equidistantly along the transversal axis, forming a trailing or coplanar configuration, with the chief at the center of the formation. The goal is to reconfigure the formation so that the satellites are distributed along a [Projected Circular Orbit \(PCO\)](#) by the final time. The parameterization of the coplanar and PCO configurations using mean [ROE](#) is discussed in [\[79\]](#). The initial and final states for each satellite are summarized in [Table D.0](#).

In this case, the initial inter-satellite distance in the trailing formation is 200 m, while the radius of the final [PCO](#) is 300 m. The orbit of the chief is defined by  $\alpha_{c,0} = \begin{bmatrix} 6761.45 \text{ km} & 180^\circ & 1.5 \cdot 10^{-3} & 0 & 98^\circ & 0^\circ \end{bmatrix}^\top$  at  $t_0$ . The reconfiguration maneuver is allocated 5 orbital periods.

Table D.0: Initial and final (required) states for each deputy

Satellite	$\mathbf{y}_{i,0}$ [m]	$\mathbf{y}_{i,f}$ [m]
Sat. A	$\begin{bmatrix} 0 & -400 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 0 & -150 & 300 & 0 \end{bmatrix}^\top$
Sat. B	$\begin{bmatrix} 0 & -200 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & -150 & 0 & 0 & -300 \end{bmatrix}^\top$
Sat. C	$\begin{bmatrix} 0 & 200 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 0 & 150 & -300 & 0 \end{bmatrix}^\top$
Sat. D	$\begin{bmatrix} 0 & 400 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 150 & 0 & 0 & 300 \end{bmatrix}^\top$

The unforced relative orbits corresponding to the initial and final configurations are shown in Fig. D.0.

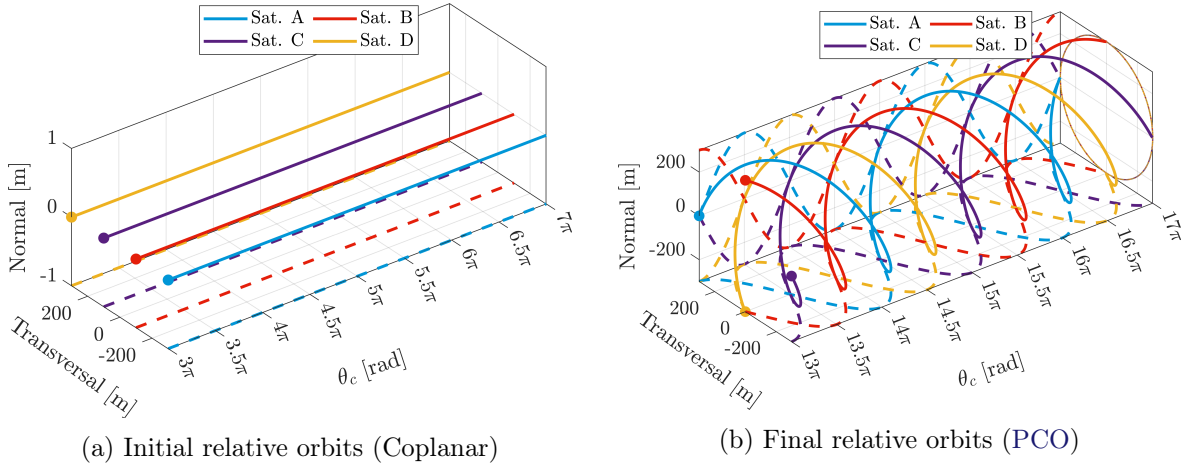


Figure D.0: Unforced relative orbits for Reconfiguration 0

## Reconfiguration 1 - Multi-pendulum to PCO

In this scenario, four identical deputy satellites initially form a multi-pendulum configuration. The objective is to rearrange the satellites into a PCO within  $t_f - t_0 = 4$  orbital periods. The term "multi-" indicates that the satellites are distributed across different pendulum relative orbits rather than sharing a single orbit. The chief's orbit is parameterized by  $\tilde{\alpha}_{c,0} = \begin{bmatrix} 6978 \text{ km} & 10^{-3} & 97.87^\circ & 0^\circ & 0^\circ & 90^\circ \end{bmatrix}^\top$  at  $t_0$ .

Table D.1: Initial and final (required) states for each deputy in Reconfiguration 1

Satellite	$\mathbf{y}_0$ [m]	$\mathbf{y}_f$ [m]
Sat. A	$\begin{bmatrix} 0 & -250 & 0 & 0 & 0 & -250 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 0 & -100 & 200 & 0 \end{bmatrix}^\top$
Sat. B	$\begin{bmatrix} 0 & -125 & 0 & 0 & 0 & -125 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & -100 & 0 & 0 & -200 \end{bmatrix}^\top$
Sat. C	$\begin{bmatrix} 0 & 125 & 0 & 0 & 0 & 125 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 0 & 100 & -200 & 0 \end{bmatrix}^\top$
Sat. D	$\begin{bmatrix} 0 & 250 & 0 & 0 & 0 & 250 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 100 & 0 & 0 & 200 \end{bmatrix}^\top$

The initial and final relative trajectories for this configuration are visualized in Fig. D.1.

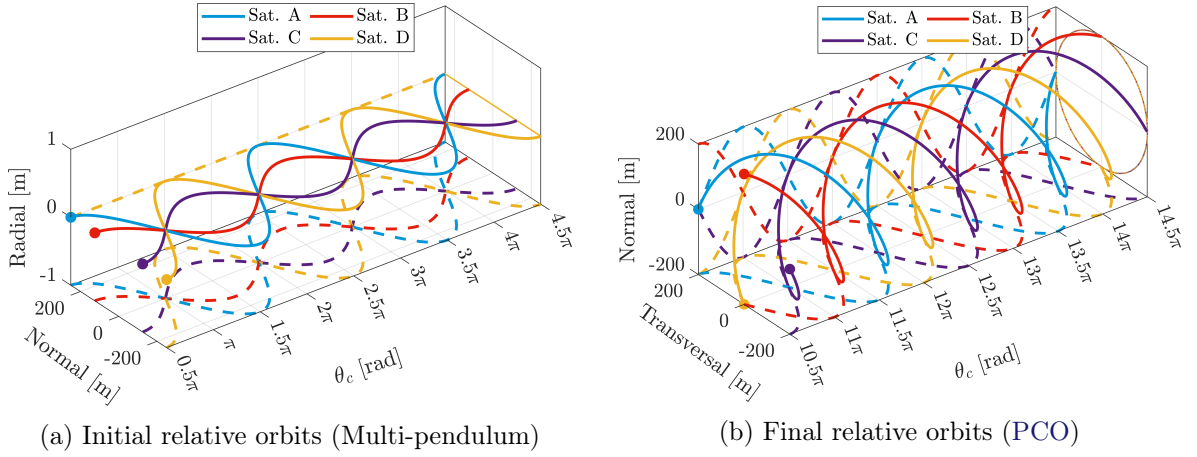


Figure D.1: Unforced relative orbits for Reconfiguration 1

## Reconfiguration 2 - Multi-PCO to Multi-cartwheel

In this reconfiguration, six identical satellites are initially organized in a multi-PCO formation. The goal is to transition these satellites into a multi-cartwheel configuration over  $t_f - t_0 = 5$  orbital periods. The term "multi-" reflects that the satellites occupy different PCO and cartwheel relative orbits. The chief's orbit is parameterized by  $\tilde{\mathbf{a}}_{c,0} = \begin{bmatrix} 6978 \text{ km} & 10^{-3} & 97.87^\circ & 0^\circ & 0^\circ & 90^\circ \end{bmatrix}^\top$  at  $t_0$ .

The unforced relative orbits corresponding to the initial and final configurations are depicted in Fig. D.2.

Table D.2: Initial and final (required) states for each deputy in Reconfiguration 2

Satellite	$\mathbf{y}_0$ [m]	$\mathbf{y}_f$ [m]
Sat. A	$\begin{bmatrix} 0 & 0 & 0 & -150 & 300 & 0 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & -500 & 0 & 0 & 0 \end{bmatrix}^\top$
Sat. B	$\begin{bmatrix} 0 & -35.91 & -129.90 & -75 & 150 & -259.81 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & -333.33 & 0 & 0 & 0 \end{bmatrix}^\top$
Sat. C	$\begin{bmatrix} 0 & -35.91 & -129.90 & 75 & -150 & -259.81 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & -166.67 & 0 & 0 & 0 \end{bmatrix}^\top$
Sat. D	$\begin{bmatrix} 0 & 0 & 0 & 150 & -300 & 0 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 166.67 & 0 & 0 & 0 \end{bmatrix}^\top$
Sat. E	$\begin{bmatrix} 0 & 35.91 & 129.90 & 75 & -150 & 259.81 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 333.33 & 0 & 0 & 0 \end{bmatrix}^\top$
Sat. F	$\begin{bmatrix} 0 & 35.91 & 129.90 & -75 & 150 & 259.81 \end{bmatrix}^\top$	$\begin{bmatrix} 0 & 0 & 500 & 0 & 0 & 0 \end{bmatrix}^\top$

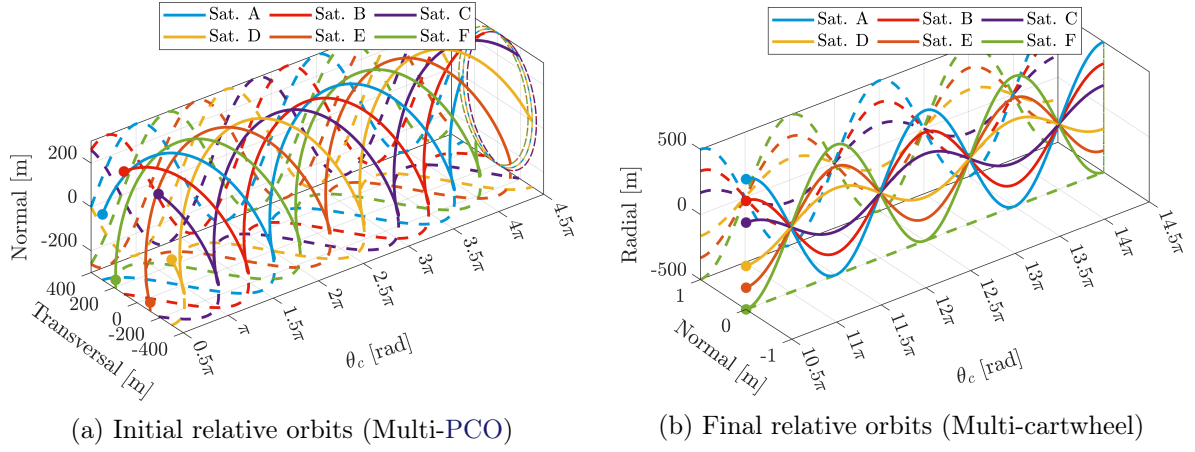


Figure D.2: Unforced relative orbits for Reconfiguration 2

### Reconfiguration 3 - Multi-cartwheel to Multi-helix

In this scenario, four identical satellites are initially in a multi-cartwheel configuration and are reconfigured into a multi-helix formation over  $t_f - t_0 = 4$  orbital periods. The chief's orbit is parameterized by  $\tilde{\alpha}_{c,0} = \begin{bmatrix} 6978 \text{ km} & 10^{-3} & 97.87^\circ & 0^\circ & 0^\circ & 90^\circ \end{bmatrix}^\top$  at  $t_0$ .

The unforced initial and final relative orbits are illustrated in Fig. D.3.

Table D.3: Initial and final (required) states for each deputy in Reconfiguration 3

Satellite	$\mathbf{y}_0$ [m]	$\mathbf{y}_f$ [m]
Sat. A	$\begin{bmatrix} 0 & 0 & -500 & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 & 34.56 & 0 & -250 & 0 & -250 \end{bmatrix}^T$
Sat. B	$\begin{bmatrix} 0 & 0 & -250 & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 & 17.28 & 0 & -125 & 0 & -125 \end{bmatrix}^T$
Sat. C	$\begin{bmatrix} 0 & 0 & 250 & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 & -17.28 & 0 & 125 & 0 & 125 \end{bmatrix}^T$
Sat. D	$\begin{bmatrix} 0 & 0 & 500 & 0 & 0 & 0 \end{bmatrix}^T$	$\begin{bmatrix} 0 & -34.56 & 0 & 250 & 0 & 250 \end{bmatrix}^T$

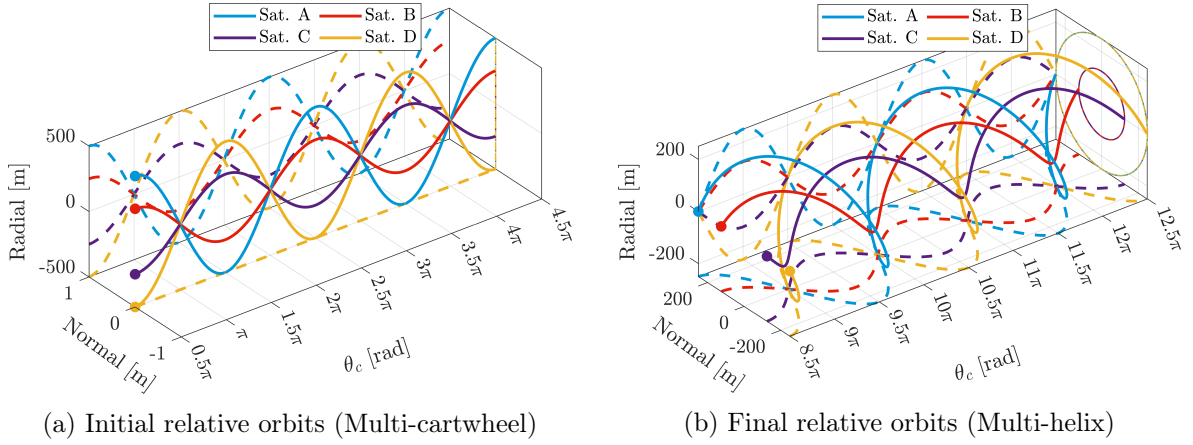


Figure D.3: Unforced relative orbits for Reconfiguration 3

## Reconfiguration 4 - Multi-helix to Multi-pendulum

In this scenario, six identical satellites are initially arranged in a multi-helix configuration. The reconfiguration goal is to transition the satellites into a multi-pendulum formation within  $t_f - t_0 = 9$  orbital periods. The chief's orbit is parameterized by  $\tilde{\mathbf{a}}_{c,0} = \begin{bmatrix} 6978 \text{ km} & 10^{-3} & 97.87^\circ & 0^\circ & 0^\circ & 90^\circ \end{bmatrix}^T$  at  $t_0$ .

The initial and final relative orbits are depicted in Fig. D.4. These represent the trajectories the spacecraft would follow in the absence of applied thrust or disturbances.

Table D.4: Initial and final (required) states for each deputy in Reconfiguration 4

Satellite	$\mathbf{y}_0$ [m]	$\mathbf{y}_f$ [m]
Sat. A	$\begin{bmatrix} 0 & 34.56 & 0 & -250 & 0 & -250 \end{bmatrix}^T$	$\begin{bmatrix} 0 & -1000 & 0 & 0 & 0 & -1000 \end{bmatrix}^T$
Sat. B	$\begin{bmatrix} 0 & 23.04 & 0 & -166.67 & 0 & -166.67 \end{bmatrix}^T$	$\begin{bmatrix} 0 & -666.67 & 0 & 0 & 0 & -666.67 \end{bmatrix}^T$
Sat. C	$\begin{bmatrix} 0 & 11.52 & 0 & -83.33 & 0 & -83.33 \end{bmatrix}^T$	$\begin{bmatrix} 0 & -333.33 & 0 & 0 & 0 & -333.33 \end{bmatrix}^T$
Sat. D	$\begin{bmatrix} 0 & -11.52 & 0 & 83.33 & 0 & 83.33 \end{bmatrix}^T$	$\begin{bmatrix} 0 & 333.33 & 0 & 0 & 0 & 333.33 \end{bmatrix}^T$
Sat. E	$\begin{bmatrix} 0 & -23.04 & 0 & 166.67 & 0 & 166.67 \end{bmatrix}^T$	$\begin{bmatrix} 0 & 666.67 & 0 & 0 & 0 & 666.67 \end{bmatrix}^T$
Sat. F	$\begin{bmatrix} 0 & -34.56 & 0 & 250.00 & 0 & 250.00 \end{bmatrix}^T$	$\begin{bmatrix} 0 & 1000.00 & 0 & 0 & 0 & 1000.00 \end{bmatrix}^T$

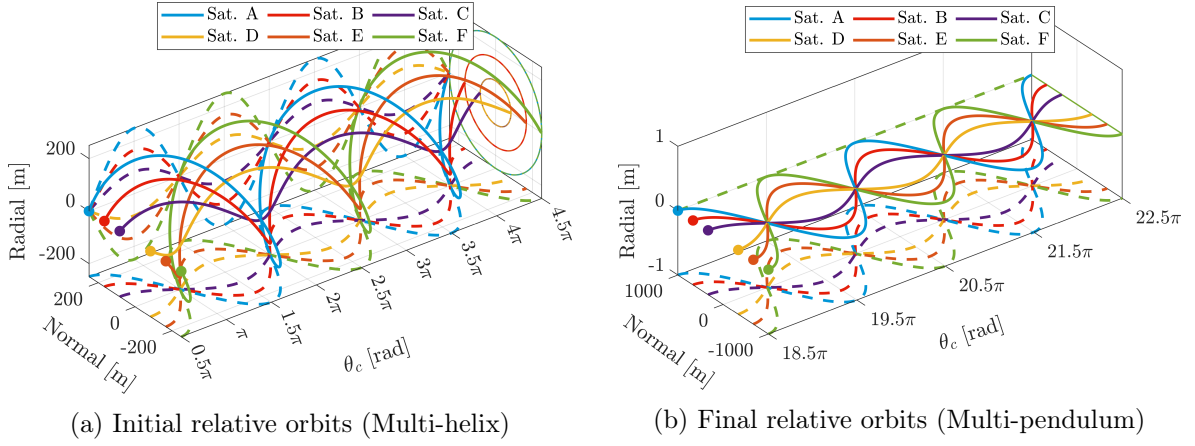


Figure D.4: Unforced relative orbits for Reconfiguration 4

## Reconfiguration 5 - Coplanar to Tetrahedron

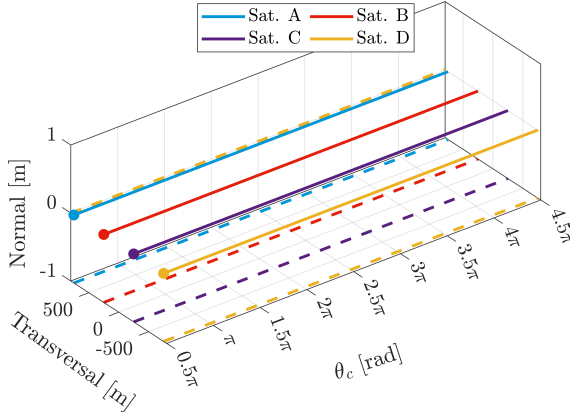
This scenario involves four identical satellites initially arranged in a trailing or coplanar configuration. The objective is to reconfigure the satellites into a tetrahedron formation over  $t_f - t_0 = 7.5$  orbital periods. The orbit of the chief is parameterized by the osculating orbital elements  $\tilde{\mathbf{a}}_c(t_0) = \begin{bmatrix} 6780.678 \text{ km} & 90^\circ & 0 & 029 & 97^\circ & 30^\circ \end{bmatrix}^T$

The initial and final relative trajectories are depicted in Fig. D.5. These illustrate the unforced motion of the satellites in the absence of thrust or external disturbances.

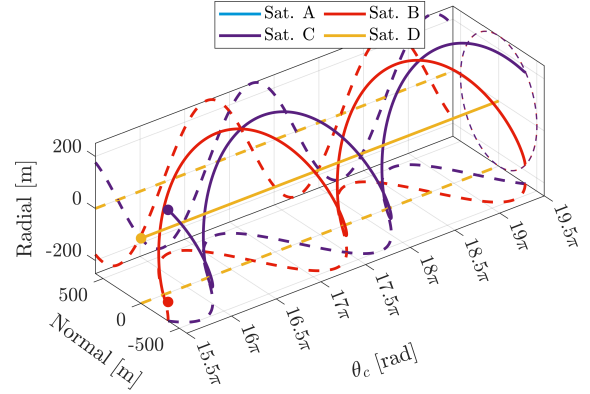
In the final tetrahedron configuration, Satellites A and D appear to follow identical relative

Table D.5: Initial and final (required) states for each deputy in Reconfiguration 5

Satellite	$\mathbf{y}_0^T$ [m]	$\mathbf{y}_f^T$ [m]
Sat. A	$\begin{bmatrix} 0 & 750 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 400 & 0 & 0 & 0 & 0 \end{bmatrix}$
Sat. B	$\begin{bmatrix} 0 & 250 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 100 & 177 & 177 & 354 & 354 \end{bmatrix}$
Sat. C	$\begin{bmatrix} 0 & -250 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -100 & -177 & 177 & -354 & 354 \end{bmatrix}$
Sat. D	$\begin{bmatrix} 0 & -750 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -400 & 0 & 0 & 0 & 0 \end{bmatrix}$



(a) Initial relative orbits (Coplanar)



(b) Final relative orbits (Tetrahedron)

Figure D.5: Unforced relative orbits for Reconfiguration 5

orbits when viewed from the Normal-Radial plane. However, they occupy distinct orbits due to their separation in the Transversal direction, as indicated by their different mean arguments of longitude in Table D.5. The tetrahedron configuration consists of two satellites in a helix formation (Satellites B and C) and two in a trailing configuration (Satellites A and D).



## A brief introduction to some relevant convex optimization solvers

Fifteen solvers were chosen for the benchmark using the four reconfiguration scenarios defined in [D](#). These solvers were chosen from the list of the most commonly used solvers on NEOS server in the category of LP and SOCP problems, and also from the solvers involved in the Mittelman's benchmark. Since NEOS server does not have a distinct category for QCQP, the solvers which are used for the SOCP problems are considered, since SOCP is the closest form of a convex programming formulation to the QCQP, and since any QCQP problem can be eventually formulated as an SOCP one. On top of the common solvers for each category, IPOPT was added for being one of the most commonly used free solvers for a variety of problem classes. Moreover, the solvers of the Matlab Optimization Toolbox were added to the list of solvers that are involved in our benchmark experiment under the name "Matlab". A brief overview of the 15 solvers is presented in the following bullet points:

- GLPK (GNU Linear Programming Kit): GLPK is intended for solving large-scale linear programming, mixed integer programming (MIP), and other related problems.
- CLP (COIN-OR Linear Program solver): CLP is part of the COIN-OR project and is a solver meant for linear programming problems.

- OSQP (Operator Splitting Quadratic Program solver): OSQP is a numerical optimization package for solving convex quadratic programs. Linear programs, as a subset of quadratic programming problems, can be handled by the solver.
- OOQP (Object-Oriented software for Quadratic Programs): OOQP is an object-oriented software package for solving convex quadratic programming problems.
- SCS (Splitting Conic Solver): SCS is primarily meant for large-scale convex quadratic cone problems.
- ECOS (Embedded Conic Solver): ECOS is a numerical software for solving convex second-order cone programs.
- IPOPT (Interior Point OPTimizer): IPOPT is vastly used for large scale nonlinear optimization of continuous systems.
- SCIP (Solving Constraint Integer Programs): SCIP is one of the fastest non-commercial solvers for mixed integer programming and mixed integer nonlinear programming (MINLP). SCIP uses the SoPlex solver internally to solve linear programming problems.
- MOSEK: MOSEK is a commercial solver with an emphasis on solving large-scale sparse problems. It supports a wide range of problem types, including linear, quadratic, and convex nonlinear programs (NLP).
- Gurobi: Gurobi is a commercial solver that performs very well in large-scale optimization settings. It can solve various types of optimization problems, including linear as well as nonlinear programming problems.
- IBM ILOG CPLEX Optimization Studio (CPLEX): CPLEX is high-performance optimization solver for linear, mixed-integer and quadratic programming. It was named after the simplex method implemented in the C programming language, although, at the moment, it supports other types of mathematical optimization and offers interfaces other than C.

- COPT (Cardinal optimizer): COPT is a mathematical optimization solver for large-scale problems. It supports many problem types including LP, SOCP, and Convex QCQP.
- Knitro: Knitro is a commercial software package for solving large scale nonlinear mathematical optimization problems. It supports a wide range of problem types, including linear and nonlinear programs, including non-convex NLPs.
- Xpress: Xpress is a mathematical optimization solver designed to solve linear programming, mixed integer programming, and other types of optimization problems.
- Matlab: Matlab is a high-level language and interactive environment that is used to perform computationally intensive tasks. In our context "Matlab" refers to the solvers offered by the Matlab Optimization Toolbox, which includes a dedicated linear programming solver, `linprog`, a dedicated conic programming function, `coneprog`, a general purpose NLP solver, `fmincon`, and many other solvers and capabilities.

Matlab has been used as a modelling language, and the problems were passed to each solver through an interface, either provided by the solver developer, by Matlab, or by a third party. Table E.1 contains the information about the version of each solver used in the benchmark, the interface that has been used between Matlab and the solver. The most interesting interfacing case is that of CPLEX, which is interfaced through the OPTI Matlab toolbox [128]. Although International Business Machines Corporation (IBM) used to provide a Matlab API for their CPLEX solver, this interface was discontinued, and the latest compatible Matlab version with that API was R2019b. Since the OPTI Matlab toolbox provides an interface that works flawlessly with newer Matlab versions, and since the authors

---

<sup>1</sup>The details of the interface as well as its source files are available on:<https://github.com/blegat/glpkmex>

<sup>2</sup>The details of the interface as well as its source files are available on:<https://github.com/jonathancurrie/OPTI>

<sup>3</sup>The details of the interface as well as its source files are available on:<https://github.com/bodono/scs-matlab>

<sup>4</sup>The details of the interface as well as its source files are available on:<https://github.com/ebertolazzi/mexIPOPT>

<sup>5</sup>The details of the interface as well as its source files are available on:<https://github.com/leavesgrp/COPT-MATLAB>

Table E.1: Benchmark solvers information

Solver	Version	Interface with Matlab
GLPK	v4.62	GLPKmex <sup>1</sup>
CLP	v1.16.11	OPTI <sup>2</sup>
OSQP	v0.6.2	Developer's <a href="#">API</a>
OOQP	v0.99.22	OPTI <sup>2</sup>
SCS	v3.2.3	SCS-Matlab <sup>3</sup>
ECOS	v2.0.10	Developer's <a href="#">API</a>
IPOPT	v3.14.4	mexIPOPT <sup>4</sup>
SCIP	v5.0.1	OPTI <sup>2</sup>
MOSEK	v10.1.15	Developer's <a href="#">API</a>
Gurobi	v11.0.2	Developer's <a href="#">API</a>
CPLEX	v12.10.0.0	OPTI <sup>2</sup>
COPT	v7.1.3	COPT-MATLAB <sup>5</sup>
Knitro	v14.0.0	Developer's <a href="#">API</a>
Xpress	v9.4.0	Developer's <a href="#">API</a>
Matlab	v9.2 (R2021b)	No interface needed

used a newer Matlab version to model the problems, with no guarantee that the same code can be used in Matlab2019b to produce the parameters of each problem, it was decided to interface CPLEX with Matlab through OPTI, instead of moving to an older version of Matlab. It is to be noted, however, that SOCP problems are not handled by OPTI. Another thing to note is that most solvers have multiple internal methods that the user can choose from according to the problem in hand. Many solvers offer the possibility to choose the most suitable algorithm for the problem in hand, according to the pre-solve results. In the benchmark experiment, the algorithm was set to the default, which in most cases means that the solve method is automatically chosen by the solver. The specific algorithms used in the benchmark experiment are reported, by most solvers, in the benchmark log files, which are available in [108].

## Softening of the different guidance formulations

This appendix presents the softened variants of all the hard-constrained guidance formulations introduced in Chapter 4. The category of each formulation is preserved during the softening process; for instance, the [QCQP](#) remains a [QCQP](#) and so on.

### F.1 Softening of the QCQP formulation

The [QCQP](#) formulation, including the minimum acceleration constraints, is softened and is rewritten as,

**Problem F.1** (Softened convex [QCQP](#) formulation).

$$\begin{aligned}
\min_{\mathbf{Y}, \bar{\mathbf{U}}, \mathbf{B}, \Upsilon} \quad & \frac{1}{a_c^2} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} \left( \Delta t_k^2 \bar{\mathbf{u}}_{i,k}^\top \mathbf{R} \bar{\mathbf{u}}_{i,k} \right) + \\
& \sum_{i \in \mathcal{I}} \sum_{k \in \bar{\mathcal{K}}} \left( (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k})^\top \mathbf{Q} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \right) + \\
& q_{umin} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} v_{i,k} + q_{ca} \sum_{i \in \mathcal{I}} \sum_{\substack{j \in \mathcal{I} \setminus \{i\} \\ \cup \{0\}}} \sum_{k \in \mathcal{K} \setminus \{0\}} \beta_{ij,k}
\end{aligned}$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0} \quad \forall i \in \mathcal{I}, \quad (\text{F.1})$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (\text{F.2})$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \quad (\text{F.3})$$

$$\bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \geq \left\| \bar{\mathbf{u}}_{i,k} \right\|_2 (a_c u_{i,min} - v_{i,k}) \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i) \quad (\text{F.4})$$

$$0 \leq v_{i,k} \leq v_{max} \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (\text{F.5})$$

$$\bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \leq a_c^2 u_{i,max}^2 \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (\text{F.6})$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq \left\| \mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k}) \right\|_2 (R_{CA} - \beta_{ij,k}) \quad (\text{F.7})$$

$$\forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\},$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq \left\| \mathbf{T}_k \check{\mathbf{y}}_{i,k} \right\|_2 (R_{CA} - \beta_{i0,k}) \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.8})$$

$$0 \leq \beta_{ij,k} \leq \beta_{max} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{I} \cup \{0\}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.9})$$

where  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  is a positive semi-definite weighting matrix associated with minimizing the control effort, specifically Delta-V.  $\bar{\mathcal{K}}$  denotes the set of indices corresponding to the time instances at which the user specifies that the deputies should track certain reference states. The term  $\bar{\mathbf{y}}_{i,k} \in \mathbb{R}^6$  represents the user-defined reference state that the  $i^{\text{th}}$  deputy is required to track at the time step  $t_k$ . The matrix  $\mathbf{Q} \in \mathbb{R}^{6 \times 6}$  is another positive semi-definite weighting matrix, this time associated with minimizing deviations of the deputies' states from their specified reference states. The constants  $q_{umin} \in \mathbb{R}^+$  and  $q_{ca} \in \mathbb{R}^+$  serve as positive weighting factors, while the variables  $v_{i,k}$  are slack variables representing the degree of violation allowed in the relaxed minimum acceleration constraints, and  $\beta_{ij,k}$  and  $\beta_{i0,k}$  are slack variables capturing the extent of violation permitted in the relaxed collision avoidance constraints. Finally,  $v_{max}$  and  $\beta_{max}$  represent the maximum allowable levels for the respective relaxed violations in minimum acceleration and collision avoidance constraints.

The aforementioned slack variables are collated in the sets  $\Upsilon$  and  $\mathcal{B}$ , respectively, which

are defined as,

$$\begin{aligned}\Upsilon &= \{v_{i,k} : i \in \mathcal{I}, (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i)\}, \\ \mathcal{B} &= \{\beta_{ij,k} : i \in \mathcal{I}, j \in \mathcal{I} \cup \{0\}, i \neq j, k \in \mathcal{K} \setminus \{0\}\}.\end{aligned}\tag{F.10}$$

## F.2 Softening of the QP formulation

The QP formulation is rewritten in Problem F.2 in a softened manner, similar to Problem F.1. The only difference between the softened QCQP formulation and the softened QP one is the transformation of the maximum acceleration constraint from the quadratic form in Eq. (F.6) into the piece-wise linear form in Eq. (F.16) and Eq. (F.17).

**Problem F.2** (Softened convex QP formulation).

$$\begin{aligned}\min_{\mathbf{Y}, \bar{\mathbf{U}}, \mathcal{B}, \Upsilon} \quad & \frac{1}{a_c^2} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} \left( \Delta t_k^2 \bar{\mathbf{u}}_{i,k}^\top \mathbf{R} \bar{\mathbf{u}}_{i,k} \right) + \\ & \sum_{i \in \mathcal{I}} \sum_{k \in \bar{\mathcal{K}}} \left( (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k})^\top \mathbf{Q} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \right) + \\ & q_{umin} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} v_{i,k} + q_{ca} \sum_{i \in \mathcal{I}} \sum_{\substack{j \in \mathcal{I} \setminus \{i\} \\ \cup \{0\}}} \sum_{k \in \mathcal{K} \setminus \{0\}} \beta_{ij,k}\end{aligned}$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0} \quad \forall i \in \mathcal{I}, \tag{F.11}$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \tag{F.12}$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \tag{F.13}$$

$$\bar{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \geq \left\| \check{\bar{\mathbf{u}}}_{i,k} \right\|_2 (a_c u_{i,min} - v_{i,k}) \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i) \tag{F.14}$$

$$0 \leq v_{i,k} \leq v_{max} \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \tag{F.15}$$

$$\begin{bmatrix} 0 & \cos(\gamma_d) & \sin(\gamma_d) \end{bmatrix} \bar{\mathbf{u}}_{i,k} \leq a_c u_{i,max} \cos(\gamma_{max}) \tag{F.16}$$

$$\forall d \in \mathcal{D}, \quad \forall i \in \mathcal{I}, \forall (i,k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i),$$

$$\begin{bmatrix} \cos(\bar{\gamma}_d) & \sin(\bar{\gamma}_d) & 0 \\ \cos(\bar{\gamma}_d) & 0 & \sin(\bar{\gamma}_d) \end{bmatrix} \bar{\mathbf{u}}_{i,k} \leq a_c u_{i,max} \cos(\bar{\gamma}_{max}) \quad (\text{F.17})$$

$$\forall d \in \bar{\mathcal{D}}, \quad \forall i \in \mathcal{I}, \quad \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i),$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 (R_{CA} - \beta_{ij,k}) \quad (\text{F.18})$$

$$\forall i, j \in \mathcal{I}, \quad i \neq j, \quad \forall k \in \mathcal{K} \setminus \{0\},$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 (R_{CA} - \beta_{i0,k}) \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.19})$$

$$0 \leq \beta_{ij,k} \leq \beta_{max} \quad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{I} \cup \{0\}, \quad i \neq j, \quad \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.20})$$

### F.3 Softening of the SOCP formulation

The softening of the **SOCP** formulation is more challenging compared to the **QCQP** and **QP** problems because the cost function must be strictly linear. In the cases of **QCQP** and **QP**, the summation related to minimizing the minimum acceleration violations as well as that which relates to minimizing the collision avoidance violations are already linear, and hence can be directly incorporated in the softened **SOCP** formulation. However, the quadratic terms in the cost function, which are responsible for minimizing the total Delta-V effort, as well as the deviation of the optimized states from the reference **ROE** profile, specifically,

$$\frac{1}{a_c^2} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} \left( \Delta t_k^2 \bar{\mathbf{u}}_{i,k}^\top \mathbf{R} \bar{\mathbf{u}}_{i,k} \right) + \sum_{i \in \mathcal{I}} \sum_{k \in \bar{\mathcal{K}}} \left( (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k})^\top \mathbf{Q} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \right),$$

are replaced by

$$\frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} \left( \Delta t_k \sqrt{\bar{\mathbf{u}}_{i,k}^\top \mathbf{R} \bar{\mathbf{u}}_{i,k}} \right) + \sqrt{\sum_{i \in \mathcal{I}} \sum_{k \in \bar{\mathcal{K}}} \left( (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k})^\top \mathbf{Q} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \right)}.$$

Although these new terms are still not linear functions of the decision variables, they allow the problem to be cast in a separable epigraph form [90] with additional **SOC** constraints as follows:



**Problem F.3** (Softened SOCP formulation).

$$\min_{\mathbf{Y}, \bar{\mathbf{U}}, \Gamma, w, \beta, \Upsilon} \quad \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} (\Delta t_k \Gamma_{i,k}) + w +$$

$$q_{umin} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} v_{i,k} + q_{ca} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I} \setminus \{i\} \cup \{0\}} \sum_{k \in \mathcal{K} \setminus \{0\}} \beta_{ij,k}$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0} \quad \forall i \in \mathcal{I}, \quad (\text{F.21})$$

$$\left\| \left[ \left[ \sqrt{\mathbf{Q}} (\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \right]_{i \in \mathcal{I}}^h \right]_{k \in \bar{\mathcal{K}}}^v \right\|_F \leq w, \quad (\text{F.22})$$

$$\mathbf{y}_{i,k+1} = \Phi_k \mathbf{y}_{i,k} + \Psi_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (\text{F.23})$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \quad (\text{F.24})$$

$$\check{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \geq \left\| \check{\mathbf{u}}_{i,k} \right\|_2 (a_c u_{i,min} - v_{i,k}) \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i) \quad (\text{F.25})$$

$$0 \leq v_{i,k} \leq v_{max} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (\text{F.26})$$

$$\|\bar{\mathbf{u}}_{i,k}\|_{\mathbf{R}} \leq \Gamma_{i,k} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (\text{F.27})$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (\text{F.28})$$

$$(\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) \geq \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 (R_{CA} - \beta_{ij,k})$$

$$\forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.29})$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 (R_{CA} - \beta_{i0,k}) \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.30})$$

$$0 \leq \beta_{ij,k} \leq \beta_{max} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{I} \cup \{0\}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.31})$$

where  $w$  is a slack variable,  $\|\cdot\|_F$  is the Frobenius norm operator, and  $[(\cdot)_i]_{\text{indices}}^h$  and  $[(\cdot)_i]_{\text{indices}}^v$  are defined as the horizontal and the vertical concatenation operators, respectively, such that  $[a_i]_{i \in \{1,2,\dots,n\}}^h = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}$  and  $[a_i]_{i \in \{1,2,\dots,n\}}^v = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}^\top$ . Moreover,  $\sqrt{\mathbf{Q}}$  is the matrix square root of  $\mathbf{Q}$ , which, in the context of Problem F.3, is strictly a diagonal positive semi-definite matrix, and  $\|\bar{\mathbf{u}}_{i,k}\|_{\mathbf{R}} = \left\| \sqrt{\mathbf{R}} \bar{\mathbf{u}}_{i,k} \right\|_2$  with  $\mathbf{R}$  being a diagonal positive semi-definite matrix. Note that the diagonal entries of  $\mathbf{R}$  have to be strictly greater than or equal to 1 so that the acceleration could never exceed the maximum allowable one.

Additionally, since both  $\mathbf{Q}$  and  $\mathbf{R}$  are diagonal matrices, their matrix square roots are equal to their element-wise square roots.

## F.4 Softening of the LP formulation

Softening the LP formulation is also somewhat challenging, as both the cost function and the constraints must remain linear. This requirement makes the separable epigraph form unusable. In order to cast the guidance scheme as a softened LP problem, the cost function terms are treated separately. Namely, the term responsible for minimizing the collision avoidance violations,  $b \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I} \setminus \{i\} \cup \{0\}} \sum_{k \in \mathcal{K} \setminus \{0\}} \beta_{ij,k}$ , remains unchanged, while the term that minimizes the Delta-V effort is taken as the one adopted in Problem F.3, and the SOC constraints in Eq. (F.27) are linearized using the same piece-wise linearization discussed for Problem 4.6. Moreover, the term which is meant to drive the deviations between the optimized states and the reference ones to zero is written as the summation of the  $L_1$  norms of these deviations. Concretely, it is written as  $\sum_{i \in \mathcal{I}} \sum_{k \in \bar{\mathcal{K}}} \|\sqrt{\mathbf{Q}}(\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k})\|_1$ . Adopting the summation of the  $L_1$  norms allows for the problem to be cast as an LP one [129], as follows:

**Problem F.4** (Softened LP formulation).

$$\begin{aligned} \min_{\mathbf{Y}, \bar{\mathbf{U}}, \mathbf{\Gamma}, \mathbf{W}, \mathbf{B}, \Upsilon} \quad & \frac{1}{a_c} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} (\Delta t_k \Gamma_{i,k}) + \sum_{i \in \mathcal{I}} \sum_{k \in \bar{\mathcal{K}}} \mathbf{1}_{6 \times 1}^\top \mathbf{w}_{i,k} + \\ & q_{umin} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_f \setminus \mathcal{K}_i} v_{i,k} + q_{ca} \sum_{i \in \mathcal{I}} \sum_{\substack{j \in \mathcal{I} \setminus \{i\} \\ \cup \{0\}}} \sum_{k \in \mathcal{K} \setminus \{0\}} \beta_{ij,k} \end{aligned}$$

subject to,

$$\mathbf{y}_{i,0} = \mathbf{y}_{i,0} \quad \forall i \in \mathcal{I}, \tag{F.32}$$

$$-\mathbf{w}_{i,k} \leq \sqrt{\mathbf{Q}}(\mathbf{y}_{i,k} - \bar{\mathbf{y}}_{i,k}) \leq \mathbf{w}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \bar{\mathcal{K}} \tag{F.33}$$

$$\mathbf{y}_{i,k+1} = \mathbf{\Phi}_k \mathbf{y}_{i,k} + \mathbf{\Psi}_k \bar{\mathbf{u}}_{i,k} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \tag{F.34}$$

$$\bar{\mathbf{u}}_{i,k} = \mathbf{0} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_n \cup \mathcal{K}_i), \tag{F.35}$$

$$\check{\mathbf{u}}_{i,k}^\top \bar{\mathbf{u}}_{i,k} \geq \left\| \check{\mathbf{u}}_{i,k} \right\|_2 (a_c u_{i,min} - v_{i,k}) \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i) \tag{F.36}$$

$$0 \leq v_{i,k} \leq v_{max} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (\text{F.37})$$

$$\begin{bmatrix} 0 & \cos(\gamma_d) & \sin(\gamma_d) \end{bmatrix} \sqrt{\mathbf{R}} \mathbf{u}_{i,k} \leq \Gamma_{i,k} \cos(\gamma_{max}) \quad (\text{F.38})$$

$$\forall d \in \mathcal{D}, \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i),$$

$$\begin{bmatrix} \cos(\bar{\gamma}_d) & \sin(\bar{\gamma}_d) & 0 \\ \cos(\bar{\gamma}_d) & 0 & \sin(\bar{\gamma}_d) \end{bmatrix} \sqrt{\mathbf{R}} \mathbf{u}_{i,k} \leq \Gamma_{i,k} \cos(\bar{\gamma}_{max}) \quad (\text{F.39})$$

$$\forall d \in \bar{\mathcal{D}}, \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i),$$

$$\Gamma_{i,k} \leq a_c u_{i,max} \quad \forall i \in \mathcal{I}, \forall (i, k) \in (i, \mathcal{K}_f \setminus \mathcal{K}_i), \quad (\text{F.40})$$

$$\begin{aligned} (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})^\top \mathbf{T}_k^\top \mathbf{T}_k (\mathbf{y}_{i,k} - \mathbf{y}_{j,k}) &\geq \|\mathbf{T}_k (\check{\mathbf{y}}_{i,k} - \check{\mathbf{y}}_{j,k})\|_2 (R_{CA} - \beta_{ij,k}) \\ &\forall i, j \in \mathcal{I}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\}, \end{aligned} \quad (\text{F.41})$$

$$\check{\mathbf{y}}_{i,k}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{y}_{i,k} \geq \|\mathbf{T}_k \check{\mathbf{y}}_{i,k}\|_2 (R_{CA} - \beta_{i0,k}) \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.42})$$

$$0 \leq \beta_{ij,k} \leq \beta_{max} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{I} \cup \{0\}, i \neq j, \forall k \in \mathcal{K} \setminus \{0\}, \quad (\text{F.43})$$

where  $\mathbf{1}_{6 \times 1}$  is a matrix of ones with dimensions  $6 \times 1$ , and  $\mathbf{w}_{i,k} \in \mathbb{R}^6$  are vectors of slack variables which are collated in the set  $\mathcal{W}$  as,

$$\mathcal{W} = \{\mathbf{w}_{i,k} : i \in \mathcal{I}, k \in \bar{\mathcal{K}}\}. \quad (\text{F.44})$$

Similar to the softened SOCP case in Problem F.3, the two matrices  $\mathbf{Q}$  and  $\mathbf{R}$  must be diagonal positive semi-definite matrices, with the diagonal entries of  $\mathbf{R}$  being strictly greater than or equal to unity.

## Surrogate models of navigation and pointing errors

The relative navigation system introduced in Chapter 3 addresses the problem of relative navigation between satellites with large inter-satellite distances. In contrast, the guidance and control schemes developed in this thesis are tailored for close-formation satellite operations. As a result, the absolute and relative navigation systems fall broadly outside the scope of this thesis. Similarly, this exclusion applies to the [Attitude Determination and Control System \(ADCS\)](#). To validate the control strategies presented in this work, surrogate models are incorporated into the validation simulations to account for navigation and pointing errors. This appendix introduces these surrogate models, which simulate the effects of such errors on the performance of the control schemes.

It is important to remind the reader that, in the subsequent discussion and throughout the thesis, specific notation conventions are used to distinguish between different types of orbital elements and their affected states. An over-tilde signifies osculating absolute or relative orbital elements. For example,  $\tilde{\mathbf{a}}$  represents the vector of osculating absolute orbital elements of a satellite, while  $\delta\tilde{\mathbf{a}}$  denotes the vector of osculating relative orbital elements. Additionally, an over-breve sign is used to indicate quantities affected by errors or physical

limitations, such as navigation errors, pointing errors, or constraints like saturation. For instance,  $\check{\mathbf{a}}$  is the vector of mean orbital elements that the satellite believes to be true but are actually impacted by navigation errors. Similarly,  $\breve{\mathbf{a}}$  represents the vector of osculating orbital elements influenced by navigation inaccuracies.

The following sections detail the surrogate models for navigation and pointing errors, explaining their integration into the validation framework.

## G.1 Model of the absolute navigation system

The simulation experiments validating the proposed control strategies rely on propagating the absolute Cartesian states of the chief and deputy spacecraft in the [ECI](#) frame. From these Cartesian states, the absolute and relative orbital elements are derived. The propagated states are treated as the ground truth for the surrogate absolute navigation model. The Cartesian state vector of the chief at time  $t_k$  is modeled as follows:

$$\breve{\mathbf{x}}_{c,k}^i = \mathbf{x}_{c,k}^i + \boldsymbol{\epsilon}_{\mathbf{x}_c}, \quad (\text{G.1})$$

where  $\boldsymbol{\epsilon}_{\mathbf{x}_c} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{x}_c})$ . The covariance matrix  $\boldsymbol{\Sigma}_{\mathbf{x}_c}$  models the noise affecting the estimation of the Cartesian state of the chief satellite, defined as,

$$\boldsymbol{\Sigma}_{\mathbf{x}_c} = \text{diag}(\sigma_{\mathbf{r}_c}^2, \sigma_{\mathbf{r}_c}^2, \sigma_{\mathbf{r}_c}^2, \sigma_{\mathbf{v}_c}^2, \sigma_{\mathbf{v}_c}^2, \sigma_{\mathbf{v}_c}^2), \quad (\text{G.2})$$

where  $\text{diag}(\cdot, \cdot, \dots)$  creates a diagonal matrix with the specified inputs as its diagonal elements. Here,  $\sigma_{\mathbf{r}_c}^2$  and  $\sigma_{\mathbf{v}_c}^2$  denote the variances of the position and velocity estimation errors in one dimension (x, y, or z), respectively.

The absolute osculating and mean orbital elements are derived from the Cartesian state

vector as,

$$\begin{aligned}\check{\check{\boldsymbol{\alpha}}}_{c,k} &= \text{Cart2OE}(\check{\check{\boldsymbol{x}}}_{c,k}^i), \\ \check{\boldsymbol{\alpha}}_{c,k} &= \text{Osc2Mean}(\check{\check{\boldsymbol{\alpha}}}_{c,k}),\end{aligned}\tag{G.3}$$

where the method `Cart2OE` transforms the Cartesian state vector into orbital elements [16], and `Osc2Mean` converts osculating orbital elements to mean orbital elements [17].

For the  $i^{\text{th}}$  deputy, the mean orbital elements can be similarly disturbed using the model in Eq. (G.1), but with distinct variances for the position and velocity estimation errors, denoted as  $\sigma_{\mathbf{r}_i}^2$  and  $\sigma_{\mathbf{v}_i}^2$ . The absolute osculating and mean orbital elements for the  $i^{\text{th}}$  deputy are obtained using the same approach described in Eq. (G.3).

## G.2 Model of the relative navigation system

Relative navigation is generally more accurate than absolute navigation, as demonstrated in [8, 11]. To reflect this improved accuracy, the disturbed relative states are not derived from the disturbed absolute states. Instead, a separate surrogate model tailored for relative states is developed.

The ground truth of the `ROE` vector for the  $i^{\text{th}}$  deputy is defined as,

$$\delta\boldsymbol{\alpha}_{i,k} = \text{OE2ROE}(\boldsymbol{\alpha}_{c,k}, \boldsymbol{\alpha}_{i,k}),\tag{G.4}$$

where `OE2ROE` is the method that transforms the absolute orbital elements of the chief and the deputy into a `ROE` vector, as defined in Eq. (2.34). The absolute mean orbital elements of the chief and a deputy, denoted as  $\boldsymbol{\alpha}_{c,k}$  and  $\boldsymbol{\alpha}_{i,k}$ , respectively, are obtained by first transforming the undisturbed Cartesian state vector into osculating orbital elements and subsequently

converting them to mean orbital elements:

$$\begin{aligned}\tilde{\boldsymbol{\alpha}}_{c,k} &= \text{Cart2OE}(\mathbf{x}_{c,k}^i), \\ \boldsymbol{\alpha}_{c,k} &= \text{Osc2Mean}(\tilde{\boldsymbol{\alpha}}_{c,k}).\end{aligned}\tag{G.5}$$

Two surrogate models are introduced to simulate the relative navigation system with realistic error conditions. The first model assumes uniform disturbances across all components of the ROE vector:

$$\check{\mathbf{y}}_{i,k} = a_c \delta \boldsymbol{\alpha}_{i,k} + \boldsymbol{\epsilon}_{\mathbf{y}_i},\tag{G.6}$$

where  $a_c$  is the semi-major axis of the chief satellite,  $\boldsymbol{\epsilon}_{\mathbf{y}_i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{y}_i})$ , and  $\boldsymbol{\Sigma}_{\mathbf{y}_i}$  is the covariance matrix of the zero-mean normally distributed random error vector affecting the ROE vector:

$$\boldsymbol{\Sigma}_{\mathbf{y}_i} = \text{diag}(\sigma_{\mathbf{y}_i}^2, \sigma_{\mathbf{y}_i}^2, \sigma_{\mathbf{y}_i}^2, \sigma_{\mathbf{y}_i}^2, \sigma_{\mathbf{y}_i}^2, \sigma_{\mathbf{y}_i}^2),\tag{G.7}$$

where  $\sigma_{\mathbf{y}_i}^2$  is the variance of the estimation error for each component of the ROE vector.

The second model accounts for increased errors in the transversal direction, typically observed in relative navigation systems [6]. It is expressed as,

$$\check{\mathbf{y}}_{i,k} = a_c \delta \boldsymbol{\alpha}_{i,k} + \overline{\mathbf{T}}_k^{-1} \mathbf{w}_{\Delta \mathbf{x}_i},\tag{G.8}$$

where  $\overline{\mathbf{T}}_k^{-1}$  is the transformation matrix that converts the relative Cartesian state to the ROE vector at time  $t_k$  (explicitly defined in Eq. (2.59)), and  $\mathbf{w}_{\Delta \mathbf{x}_i}$  is the zero-mean normally distributed random error vector affecting the relative Cartesian state:

$$\mathbf{w}_{\Delta \mathbf{x}_i} = \boldsymbol{\epsilon}_{\Delta \mathbf{x}_i},\tag{G.9}$$

where  $\boldsymbol{\epsilon}_{\Delta \mathbf{x}_i} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\Delta \mathbf{x}_i})$  and  $\boldsymbol{\Sigma}_{\Delta \mathbf{x}_i}$  is the covariance matrix of  $\mathbf{w}_{\Delta \mathbf{x}_i}$ , given as,

$$\boldsymbol{\Sigma}_{\Delta \mathbf{x}_i} = \text{diag}(\sigma_{\Delta \mathbf{r}_i}^2, 4\sigma_{\Delta \mathbf{r}_i}^2, \sigma_{\Delta \mathbf{r}_i}^2, \sigma_{\Delta \mathbf{v}_i}^2, 4\sigma_{\Delta \mathbf{v}_i}^2, \sigma_{\Delta \mathbf{v}_i}^2),\tag{G.10}$$

where  $\sigma_{\Delta \mathbf{r}_i}^2$  and  $\sigma_{\Delta \mathbf{v}_i}^2$  are the variances of the estimation errors for the radial and normal components of the relative position and velocity vectors, respectively. The covariance matrix,  $\Sigma_{\Delta \mathbf{x}_i}$ , indicates that the relative position/velocity errors in the transversal direction are twice as large as the errors in the radial or normal directions.

Note that the models in Eqs. (G.6) and (G.8) consider constant covariance matrices over the whole maneuver horizon and do not account for potential improvements in the relative estimation accuracy.

### G.3 Effect of pointing errors on the control acceleration vector

The effect of the pointing errors affecting the direction of the thrust vector is modeled according to the following equation:

$$\begin{aligned} \mathbf{q}_{\text{pe}} &= \begin{bmatrix} \cos(\zeta_{\text{pe}}/2) & \sin(\zeta_{\text{pe}}/2) \hat{\mathbf{q}}_{\text{pe}}^T \end{bmatrix}^T, \\ \check{\mathbf{u}}_k &= \mathbf{q}_{\text{pe}} \circ \mathbf{u}_k \circ \tilde{\mathbf{q}}_{\text{pe}}, \end{aligned} \tag{G.11}$$

where  $\mathbf{q}_{\text{pe}}$  is the thruster misalignment unit quaternion, with  $\tilde{\mathbf{q}}_{\text{pe}}$  being its quaternion conjugate,  $\zeta_{\text{pe}}$  is the pointing error angle,  $\hat{\mathbf{q}}_{\text{pe}}$  is a random 3-element unit vector,  $\circ$  is the quaternion multiplication operator, and  $\mathbf{u}_k$  is the acceleration vector commanded by the controller at time  $t_k$ .



# Appendix H

## Stability of the surrogate model of the attitude control system

Letting  $\mathbf{q}_e$  be the unit error quaternion between the desired and the actual attitudes as defined in Eq. (6.5), and  $\boldsymbol{\omega}_e := \boldsymbol{\omega} - \boldsymbol{\omega}_r$  be the error angular velocity vector, with  $\boldsymbol{\omega}$  being the angular velocity vector and  $\boldsymbol{\omega}_r$  being the reference/set point of  $\boldsymbol{\omega}$ , the dynamics of the error unit quaternion as given by [124] could be written as

$$\dot{\mathbf{q}}_e = \frac{1}{2} \mathbf{q}_e \circ \boldsymbol{\omega}_e. \quad (\text{H.1})$$

Note that, for slew maneuvers, the reference angular velocity vector is always zero, i.e.,  $\boldsymbol{\omega}_r = \mathbf{0}$ , which enables (H.1) to be rewritten as

$$\dot{\mathbf{q}}_e = \frac{1}{2} \mathbf{q}_e \circ \boldsymbol{\omega}. \quad (\text{H.2})$$

If the angular velocity profile could be forced, via the input torques, to follow (6.6), i.e.,  $\boldsymbol{\omega} = -K q_{e,0} \mathbf{q}_e$ , the error quaternion kinematics could be expressed as

$$\dot{\mathbf{q}}_e = -\frac{K q_{e,0}}{2} \mathbf{q}_e \circ \mathbf{q}_e = -\frac{K q_{e,0}}{2} \begin{bmatrix} -\|\mathbf{q}_e\|_2^2 \\ q_{e,0} \mathbf{q}_e \end{bmatrix} = \frac{K}{2} \begin{bmatrix} \|\mathbf{q}_e\|_2^2 & q_{e,0} \\ -q_{e,0}^2 & \mathbf{q}_e \end{bmatrix}. \quad (\text{H.3})$$

Noting that  $K$  is a positive gain, (H.3) could be rewritten as

$$\begin{bmatrix} \dot{q}_{e,0} \\ \dot{q}_{e,1} \\ \dot{q}_{e,2} \\ \dot{q}_{e,3} \end{bmatrix} = \begin{bmatrix} K_0 q_{e,0} \\ -K_1 q_{e,1} \\ -K_2 q_{e,2} \\ -K_3 q_{e,3} \end{bmatrix}, \quad (\text{H.4})$$

where  $K_0$ ,  $K_1$ ,  $K_2$ , and  $K_3$ , are all positive, and  $q_{e,1}$ ,  $q_{e,2}$ , and  $q_{e,3}$  are the three components of  $\mathbf{q}_e$ . Equation (H.4) suggests that, under the constraint  $\|\mathbf{q}_e\|_2 = 1$ , the error quaternion asymptotically approaches  $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\top$  regardless of the initial condition of  $\mathbf{q}_e$ .

# Bibliography

- [1] Joseph N. Pelton and Scott Madry. “Introduction to the Small Satellite Revolution and Its Many Implications”. In: *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation*. Cham: Springer International Publishing, 2020. Chap. 1, pp. 3–31. ISBN: 978-3-030-36308-6. DOI: [10.1007/978-3-030-36308-6\\_1](https://doi.org/10.1007/978-3-030-36308-6_1). URL: [https://doi.org/10.1007/978-3-030-36308-6\\_1](https://doi.org/10.1007/978-3-030-36308-6_1).
- [2] Gerhard Krieger et al. “Interferometric Synthetic Aperture Radar (SAR) Missions Employing Formation Flying”. In: *Proceedings of the IEEE* 98.5 (2010), pp. 816–843. DOI: [10.1109/JPROC.2009.2038948](https://doi.org/10.1109/JPROC.2009.2038948).
- [3] Giancarmine Fasano, Alfredo Renga, and Marco D’Errico. “Formation geometries for multistatic SAR tomography”. In: *Acta Astronautica* 96 (2014), pp. 11–22. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2013.11.024>. URL: <https://www.sciencedirect.com/science/article/pii/S009457651300427X>.
- [4] Franz D Busse, Jonathan P How, and James Simpson. “Demonstration of adaptive extended Kalman filter for low-earth-orbit formation estimation using CDGPS”. In: *Journal of the Institute of Navigation* 50.2 (2003), pp. 79–94.
- [5] Remco Kroes et al. “Precise GRACE baseline determination using GPS”. In: *Gps Solutions* 9.1 (2005), pp. 21–31.
- [6] Simone D’Amico. “Autonomous formation flying in low earth orbit”. PhD thesis. TU Delft, 2010.

- [7] Sunny Leung and Oliver Montenbruck. “Real-time navigation of formation-flying spacecraft using global-positioning-system measurements”. In: *Journal of Guidance, Control, and Dynamics* 28.2 (2005), pp. 226–235.
- [8] Simone D’Amico, Jean-Sebastien Ardaens, and Oliver Montenbruck. “Navigation of formation flying spacecraft using GPS: the PRISMA technology demonstration”. In: *Proceedings of the 22nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2009)*. 2009, pp. 1427–1441.
- [9] I. Michael Ross. “Space Trajectory Optimization and L1-Optimal Control Problems”. In: *Modern Astrodynamics*. Ed. by Pini Gurfil. Vol. 1. Elsevier Astrodynamics Series. Butterworth-Heinemann, 2006. Chap. 6, pp. 155–VIII. DOI: [https://doi.org/10.1016/S1874-9305\(07\)80008-2](https://doi.org/10.1016/S1874-9305(07)80008-2).
- [10] European Space Agency. *SWARM mission data*. <https://earth.esa.int/eogateway/missions/swarm/data>. 2022.
- [11] Ahmed Mahfouz et al. “GNSS-based baseline vector determination for widely separated cooperative satellites using L1-only receivers”. In: *Advances in Space Research* (2023). ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2023.06.037>.
- [12] Ahmed Mahfouz et al. “Autonomous Optimal Absolute Orbit Keeping through Formation Flying Techniques”. In: *Aerospace* 10.11 (Oct. 2023). ISSN: 2226-4310. DOI: [10.3390/aerospace10110959](https://doi.org/10.3390/aerospace10110959). URL: <https://www.mdpi.com/2226-4310/10/11/959>.
- [13] Ahmed Mahfouz et al. “Fuel-Optimal Formation Reconfiguration by Means of Unidirectional Low-Thrust Propulsion System”. In: *Journal of Guidance, Control, and Dynamics* 48.2 (2025), pp. 404–413. DOI: [10.2514/1.G008214](https://doi.org/10.2514/1.G008214).
- [14] Ahmed Mahfouz et al. “Delta-V-optimal centralized guidance strategy for under-actuated N-satellite formations”. In: *Acta Astronautica* 228 (2025), pp. 69–87. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2024.11.048>. URL: <https://www.sciencedirect.com/science/article/pii/S0094576524007021>.

- [15] Ahmed Mahfouz et al. “Low-Thrust Under-Actuated Satellite Formation Guidance and Control Strategies”. In: Dec. 2024. DOI: [10.48550/arXiv.2412.20489](https://arxiv.org/abs/2412.20489). URL: <https://arxiv.org/abs/2412.20489>.
- [16] David A Vallado. “Fundamentals of astrodynamics and applications”. In: 4th ed. Hawthorne, CA, U.S.A.: Microcosm Press & Springer, 2013. Chap. 2, pp. 95–105.
- [17] Gabriella Gaias, Camilla Colombo, and Martin Lara. “Analytical Framework for Precise Relative Motion in Low Earth Orbits”. In: *Journal of Guidance, Control, and Dynamics* 43.5 (2020), pp. 915–927. DOI: [10.2514/1.G004716](https://doi.org/10.2514/1.G004716).
- [18] Roger R Bate et al. *Fundamentals of astrodynamics*. Courier Dover Publications, 2020, pp. 419–425.
- [19] Hanspeter Schaub and John Junkins. “Transport Theorem Derivation Using Linear Algebra”. In: *Analytical Mechanics Of Space Systems*. 4th. AIAA Education Series, 2018. Chap. Appendix A, pp. 863–866. DOI: [10.2514/5.9781624105210.0863.0866](https://doi.org/10.2514/5.9781624105210.0863.0866).
- [20] Richard H. Battin. “Two-Body Orbits and the Initial-Value Problem”. In: *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. AIAA Education Series, 1999. Chap. 4, pp. 141–190. DOI: [10.2514/5.9781600861543.0141.0190](https://doi.org/10.2514/5.9781600861543.0141.0190).
- [21] Bradley Kuiack and Steve Ulrich. “Nonlinear Analytical Equations of Relative Motion on J2-Perturbed Eccentric Orbits”. In: *Journal of Guidance, Control, and Dynamics* 41.12 (2018), pp. 2666–2677. DOI: [10.2514/1.G003723](https://doi.org/10.2514/1.G003723).
- [22] Gabriella Gaias, Jean-Sébastien Ardaens, and Oliver Montenbruck. “Model of J<sub>2</sub>-perturbed satellite relative motion with time-varying differential drag”. In: *Celestial Mechanics and Dynamical Astronomy* 123.4 (Dec. 2015), pp. 411–433. ISSN: 1572-9478. URL: <https://doi.org/10.1007/s10569-015-9643-2>.
- [23] WH Clohessy and RS Wiltshire. “Terminal guidance system for satellite rendezvous”. In: *Journal of the Aerospace Sciences* 27.9 (1960), pp. 653–658.

- [24] Adam W. Koenig, Tommaso Guffanti, and Simone D’Amico. “New State Transition Matrices for Spacecraft Relative Motion in Perturbed Orbits”. In: *Journal of Guidance, Control, and Dynamics* 40.7 (2017), pp. 1749–1768. DOI: [10.2514/1.G002409](https://doi.org/10.2514/1.G002409).
- [25] MC Eckstein, CK Rajasingh, and P Blumer. “Colocation strategy and collision avoidance for the geostationary satellites at 19 degrees west”. In: *International Symposium on Space Flight Dynamics*. Vol. 6. 1989.
- [26] Simone D’Amico and Oliver Montenbruck. “Proximity Operations of Formation-Flying Spacecraft Using an Eccentricity/Inclination Vector Separation”. In: *Journal of Guidance, Control, and Dynamics* 29.3 (2006), pp. 554–563. DOI: [10.2514/1.15114](https://doi.org/10.2514/1.15114).
- [27] Simone D’Amico. “Relative orbital elements as integration constants of Hill’s equations”. In: *DLR, TN* (2005). Technical report, pp. 05–08.
- [28] G. Di Mauro et al. “Continuous maneuvers for spacecraft formation flying reconfiguration using relative orbit elements”. In: *Acta Astronautica* 153 (2018), pp. 311–326. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2018.01.043>.
- [29] Christopher W. T. Roscoe et al. “Formation Establishment and Reconfiguration Using Differential Elements in J2-Perturbed Orbits”. In: *Journal of Guidance, Control, and Dynamics* 38.9 (2015), pp. 1725–1740. DOI: [10.2514/1.G000999](https://doi.org/10.2514/1.G000999).
- [30] Gabriella Gaias and Marco Lovera. “Trajectory design for proximity operations: The relative orbital elements’ perspective”. In: *Journal of Guidance, Control, and Dynamics* 44.12 (2021), pp. 2294–2302. DOI: [10.2514/1.G006175](https://doi.org/10.2514/1.G006175).
- [31] G. Gaias, J.-S. Ardaens, and C. Colombo. “Precise line-of-sight modelling for angles-only relative navigation”. In: *Advances in Space Research* 67.11 (2021). Satellite Constellations and Formation Flying, pp. 3515–3526. ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2020.05.048>.
- [32] Hanspeter Schaub and John L. Junkins. “Spacecraft Formation Flying”. In: *Analytical Mechanics of Space Systems*. 4th ed. American Institute of Aeronautics and Astronautics (AIAA), 2018. Chap. 14, pp. 767–862. DOI: [10.2514/5.9781624105210.0767](https://doi.org/10.2514/5.9781624105210.0767).

0862. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/5.9781624105210.0767.0862>. URL: <https://arc.aiaa.org/doi/abs/10.2514/5.9781624105210.0767.0862>.
- [33] Nixon A Correa-Muños and Liliana A Cerón-Calderón. “Precision and accuracy of the static GNSS method for surveying networks used in Civil Engineering”. In: *Ingeniería e Investigación* 38.1 (2018), pp. 52–59.
  - [34] AQ Le and CCJM Tiberius. “GPS standard positioning service: how good is it?” In: *European Journal of Navigation* 1.2 (2003), pp. 21–27.
  - [35] Marco D&’Errico. *Distributed space missions for earth system monitoring*. Vol. 31. Springer Science & Business Media, 2012.
  - [36] S De Florio et al. “Flight results from the autonomous navigation and control of formation flying spacecraft on the prisma mission”. In: (2010).
  - [37] Jean-Sébastien Ardaens and Gabriella Gaias. “Angles-only relative orbit determination in low earth orbit”. In: *Advances in Space Research* 61.11 (2018), pp. 2740–2760.
  - [38] Jean-Sébastien Ardaens and Gabriella Gaias. “A numerical approach to the problem of angles-only initial relative orbit determination in low earth orbit”. In: *Advances in Space Research* 63.12 (2019), pp. 3884–3899.
  - [39] John O Woods and John A Christian. “Lidar-based relative navigation with respect to non-cooperative objects”. In: *Acta Astronautica* 126 (2016), pp. 298–311.
  - [40] Oliver Montenbruck et al. “A real-time kinematic GPS sensor for spacecraft relative navigation”. In: *Aerospace Science and Technology* 6.6 (2002), pp. 435–449.
  - [41] Jae-Ik Park et al. “Hardware-in-the-loop simulations of GPS-based navigation and control for satellite formation flying”. In: *Advances in Space Research* 46.11 (2010), pp. 1451–1465.
  - [42] Han-Earl Park et al. “Integrated orbit and attitude hardware-in-the-loop simulations for autonomous satellite formation flying”. In: *Advances in Space Research* 52.12 (2013), pp. 2052–2066.

- [43] Vincenzo Capuano, Alexei Harvard, and Soon-Jo Chung. “On-board cooperative spacecraft relative navigation fusing GNSS with vision”. In: *Progress in Aerospace Sciences* 128 (2022), p. 100761. ISSN: 0376-0421. DOI: <https://doi.org/10.1016/j.paerosci.2021.100761>. URL: <https://www.sciencedirect.com/science/article/pii/S0376042121000646>.
- [44] Urbano Tancredi, Alfredo Renga, and Michele Grassi. “Real-time relative positioning of spacecraft over long baselines”. In: *Journal of Guidance, Control, and Dynamics* 37.1 (2014), pp. 47–58.
- [45] Vincent Giraldo and Simone D’Amico. “Precise Real-Time Relative Orbit Determination for Large-Baseline Formations Using GNSS”. In: *Proceedings of the 2021 International Technical Meeting of The Institute of Navigation*. 2021, pp. 366–384.
- [46] Urbano Tancredi et al. “Relative positioning of spacecraft in intense ionospheric conditions by GPS”. In: *Aerospace Science and Technology* 43 (2015), pp. 191–198.
- [47] Luxspace. *Triton-X, The high-performance microsatellite platform*. Brochure. 2021. URL: [https://luxspace.lu/wp-content/uploads/2021/08/Triton-X\\_Heavy\\_Brochure.pdf](https://luxspace.lu/wp-content/uploads/2021/08/Triton-X_Heavy_Brochure.pdf).
- [48] Alfred Leick, Lev Rapoport, and Dmitry Tatarnikov. “GNSS Positioning Approaches”. In: *GPS Satellite Surveying*. John Wiley & Sons, Ltd, 2015. Chap. 6, pp. 257–399. ISBN: 9781119018612. DOI: <https://doi.org/10.1002/9781119018612.ch6>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119018612.ch6>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119018612.ch6>.
- [49] J. T. Wu et al. “Effects of antenna orientation on GPS carrier phase”. In: *manuscripta geodaetica* 18.2 (Apr. 1993), pp. 91–98. ISSN: 0340-8825. DOI: [10.1007/BF03655303](https://doi.org/10.1007/BF03655303). URL: <https://doi.org/10.1007/BF03655303>.
- [50] P Teunissen. “The Least-Squares Ambiguity Decorrelation Adjustment: A Method for Fast GPS Ambiguity Estimation”. In: *Journal of Geodesy* 73 (1999), pp. 587–593.
- [51] X-W Chang, X Yang, and T Zhou. “MLAMBDA: A modified LAMBDA method for integer least-squares estimation”. In: *Journal of Geodesy* 79.9 (2005), pp. 552–565.



- [52] Tomoji Takasu and Akio Yasuda. “Kalman-filter-based integer ambiguity resolution strategy for long-baseline RTK with ionosphere and troposphere estimation”. In: *Proceedings of the 23rd international technical meeting of the satellite division of the institute of navigation (ION GNSS 2010)*. 2010, pp. 161–171.
- [53] T Ebinuma, O Montenbruck, and EG Lightsey. “Precise Spacecraft Relative Navigation Using Kinematic Inter-Spacecraft State Estimates”. In: *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*. 2002, pp. 2038–2046.
- [54] Thomas P Yunck. “Coping with the atmosphere and ionosphere in precise satellite and ground positioning”. In: *Washington DC American Geophysical Union Geophysical Monograph Series 73* (1993), pp. 1–16.
- [55] Peter F De Bakker, Hans van der Marel, and Christian CJM Tiberius. “Geometry-free undifferenced, single and double differenced analysis of single frequency GPS, EGNOS and GIOVE-A/B measurements”. In: *GPS solutions* 13 (2009), pp. 305–314. DOI: <https://doi.org/10.1007/s10291-009-0123-6>. URL: <https://link.springer.com/article/10.1007/s10291-009-0123-6>.
- [56] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.
- [57] Ahmed Mahfouz et al. “Relative State Estimation for LEO Formations with Large Inter-satellite Distances Using Single-Frequency GNSS Receivers”. In: *Proceedings of the 11th International Workshop on Satellite Constellations & Formation Flying*. Milan-Italy, May 2022.
- [58] Alexandru Mihai Crisan et al. “Distance estimation in OFDM inter-satellite links”. In: *Measurement* 154 (2020), p. 107479.
- [59] John A Klobuchar. “Ionospheric time-delay algorithm for single-frequency GPS users”. In: *IEEE Transactions on aerospace and electronic systems* 3 (1987), pp. 325–331.

- [60] G Sasi Bhushana Rao. “Ionospheric delay estimation for improving the global positioning system position accuracy”. In: *IETE Journal of Research* 54.1 (2008), pp. 23–29.
- [61] Alfredo Renga et al. “Accurate ionospheric delay model for real-time GPS-based positioning of LEO satellites using horizontal VTEC gradient estimation”. In: *Gps Solutions* 22.2 (2018), pp. 1–14.
- [62] Attila Komjathy and RB Langley. “An assessment of predicted and measured ionospheric total electron content using a regional GPS network”. In: *Proceedings of the national technical meeting of the Institute of Navigation*. Citeseer. 1996, pp. 615–624.
- [63] Centre National d’Études Spatiales, European Commission, and European Space Agency. *User guide for EGNOS application developers : ED 2.0, 15/12/2011*. Publications Office, 2011. DOI: [doi/10.2769/22157](https://doi.org/10.2769/22157).
- [64] M. M. Hoque and N. Jakowski. “A new global model for the ionospheric F2 peak height for radio wave propagation”. In: *Annales Geophysicae* 30.5 (2012), pp. 797–809. DOI: [10.5194/angeo-30-797-2012](https://doi.org/10.5194/angeo-30-797-2012). URL: <https://angeo.copernicus.org/articles/30/797/2012/>.
- [65] WM Lear. *GPS navigation for low-earth orbiting vehicles, 1st revision*. Tech. rep. 87-FM-2, JSC-32,031. NASA Lyndon B. Johnson Space Center, Mission planning and analysis division, 1988.
- [66] Fei Guo and Xiaohong Zhang. “Real-time clock jump detection and repair for precise point positioning”. In: *Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*. 2012, pp. 3077–3088.
- [67] Thomas E Carter. “State transition matrices for terminal rendezvous studies: brief survey and new example”. In: *Journal of Guidance, Control, and Dynamics* 21.1 (1998), pp. 148–155.

- [68] Juan G Ceva and Bradford W Parkinson. “Multipath Interference in Orbiting Receivers Due to Earth Surface Reflections”. In: *Proceedings of the 6th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1993)*. 1993, pp. 1557–1563.
- [69] Alan Julian Izenman. *Modern multivariate statistical techniques: Regression, classification and manifold learning*. Springer, 2008, pp. 978–. DOI: [10.1007/978-0-387-78189-1](https://doi.org/10.1007/978-0-387-78189-1).
- [70] José Luis Sanchez Lopez. *A general architecture for autonomous navigation of unmanned aerial systems*. 2017.
- [71] NM Ansor et al. “Solar activity-climate relations during solar cycle 24”. In: *IOP Conference Series: Earth and Environmental Science*. Vol. 1151. 1. IOP Publishing. 2023, p. 012022.
- [72] Sara Miller et al. “Survey and Performance Evaluation of Small-Satellite Propulsion Technologies”. In: *Journal of Spacecraft and Rockets* 58.1 (2021), pp. 222–231. DOI: [10.2514/1.A34774](https://doi.org/10.2514/1.A34774).
- [73] Dillon O’Reilly, Georg Herdrich, and Darren F. Kavanagh. “Electric Propulsion Methods for Small Satellites: A Review”. In: *Aerospace* 8.1 (2021). ISSN: 2226-4310. DOI: [10.3390/aerospace8010022](https://doi.org/10.3390/aerospace8010022). URL: <https://www.mdpi.com/2226-4310/8/1/22>.
- [74] Vincenzo Stanzione and Beatrice Sabbatinelli. “PLATiNO Project: a new Italian multi-application small satellite platform for highly competitive missions.” In: *69th International Astronautical Congress (IAC)*. Bremen, Germany: International Astronautical Federation, 2018.
- [75] Evan Helmeid et al. *The Integrated Avionics Unit - performance, innovation and application*. Poster presentation at The 4S Symposium. Vilamoura, Portugal, 2022. 2022.
- [76] R Larsson et al. “Autonomous Formation Flying in LEO - Seven months of routine formation flying with frequent reconfigurations”. In: *4th International Conference on Spacecraft Formation Flying Missions & Technologies*. 2011. URL: <https://elib.dlr.de/74364/>.

- [77] G. Gaias and S. D’Amico. “Impulsive Maneuvers for Formation Reconfiguration Using Relative Orbital Elements”. In: *Journal of Guidance, Control, and Dynamics* 38.6 (2015), pp. 1036–1049. DOI: [10.2514/1.G000189](https://doi.org/10.2514/1.G000189).
- [78] Michelle Chernick and Simone D’Amico. “New Closed-Form Solutions for Optimal Impulsive Control of Spacecraft Relative Motion”. In: *Journal of Guidance, Control, and Dynamics* 41.2 (2018), pp. 301–319. DOI: [10.2514/1.G002848](https://doi.org/10.2514/1.G002848).
- [79] Francesca Scala et al. “Design of optimal low-thrust manoeuvres for remote sensing multi-satellite formation flying in low Earth orbit”. In: *Advances in Space Research* 68.11 (2021), pp. 4359–4378. ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2021.09.030>.
- [80] Andrea De Vittori et al. “Low-Thrust Collision Avoidance Maneuver Optimization”. In: *Journal of Guidance, Control, and Dynamics* 45.10 (2022), pp. 1815–1829. DOI: [10.2514/1.G006630](https://doi.org/10.2514/1.G006630).
- [81] Daniel Morgan, Soon-Jo Chung, and Fred Y. Hadaegh. “Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming”. In: *Journal of Guidance, Control, and Dynamics* 37.6 (2014), pp. 1725–1740. DOI: [10.2514/1.G000218](https://doi.org/10.2514/1.G000218).
- [82] S. Sarno et al. “Path planning and guidance algorithms for SAR formation reconfiguration: Comparison between centralized and decentralized approaches”. In: *Acta Astronautica* 167 (2020), pp. 404–417. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2019.11.016>. URL: <https://www.sciencedirect.com/science/article/pii/S0094576519313955>.
- [83] Gabriella Gaias and Jean-Sébastien Ardaens. “Flight Demonstration of Autonomous Noncooperative Rendezvous in Low Earth Orbit”. In: *Journal of Guidance, Control, and Dynamics* 41.6 (2018), pp. 1337–1354. DOI: [10.2514/1.G003239](https://doi.org/10.2514/1.G003239).
- [84] Gabriella Gaias, Simone D’Amico, and Jean-Sébastien Ardaens. “Generalised multi-impulsive manoeuvres for optimum spacecraft rendezvous in near-circular orbit”. In: *International Journal of Space Science and Engineering* 3.1 (2015), pp. 68–88. DOI: [10.1504/IJSPACESE.2015.069361](https://doi.org/10.1504/IJSPACESE.2015.069361).

- [85] Enrico Belloni et al. “Relative and absolute on-board optimal formation acquisition and keeping for scientific activities in high-drag low-orbit environment”. In: *Advances in Space Research* 73.11 (2024). Recent Advances in Satellite Constellations and Formation Flying, pp. 5595–5613. ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2023.07.051>. URL: <https://www.sciencedirect.com/science/article/pii/S0273117723006002>.
- [86] Tomas Pippia et al. “Reconfiguration of a satellite constellation in circular formation orbit with decentralized model predictive control”. In: *Proceedings of the 2022 CEAS EuroGNC conference*. CEAS-GNC-2022-027. Berlin, Germany, May 2022. DOI: [10.48550/ARXIV.2201.10399](https://doi.org/10.48550/ARXIV.2201.10399).
- [87] Stephen P Boyd and Lieven Vandenberghe. “Convex optimization”. In: Cambridge university press, 2004. Chap. 4, pp. 136–137.
- [88] Jean-Thomas Camino et al. “Linearization of Euclidean norm dependent inequalities applied to multibeam satellites design”. In: *Computational Optimization and Applications* 73 (2019), pp. 679–705. DOI: [10.1007/s10589-019-00083-z](https://doi.org/10.1007/s10589-019-00083-z).
- [89] Anas Abdelkarim, Yanzhao Jia, and Daniel Gorges. “Optimization of Vehicle-to-Grid Profiles for Peak Shaving in Microgrids Considering Battery Health”. In: *IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2023, pp. 1–6. DOI: [10.1109/IECON51785.2023.10312094](https://doi.org/10.1109/IECON51785.2023.10312094).
- [90] Miguel Sousa Lobo et al. “Applications of second-order cone programming”. In: *Linear Algebra and its Applications* 284.1 (1998). International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing, pp. 193–228. ISSN: 0024-3795. DOI: [https://doi.org/10.1016/S0024-3795\(98\)10032-0](https://doi.org/10.1016/S0024-3795(98)10032-0). URL: <https://www.sciencedirect.com/science/article/pii/S0024379598100320>.
- [91] Joseph M. Elble and Nikolaos V. Sahinidis. “Scaling linear optimization problems prior to application of the simplex method”. In: *Computational Optimization and Applications* 52.2 (June 2012), pp. 345–371. ISSN: 1573-2894. DOI: [10.1007/s10589-011-9420-4](https://doi.org/10.1007/s10589-011-9420-4).

- [92] Wei Wang et al. “Optimal Projected Circular Orbit for Spacecraft Formation Flying Near a Slowly Rotating Asteroid”. In: *IEEE Transactions on Aerospace and Electronic Systems* 58.6 (2022), pp. 5483–5493. DOI: [10.1109/TAES.2022.3173247](https://doi.org/10.1109/TAES.2022.3173247).
- [93] Andrew Makhorin. *GNU Linear Programming Kit (GLPK)*. Accessed: 2024-05-21. 2012. URL: <https://www.gnu.org/software/glpk/>.
- [94] John Forrest et al. *coin-or/Clp: Release releases/1.17.9*. 2023. DOI: [10.5281/zenodo.10041272](https://doi.org/10.5281/zenodo.10041272).
- [95] B. Stellato et al. “OSQP: an operator splitting solver for quadratic programs”. In: *Mathematical Programming Computation* 12.4 (2020), pp. 637–672. DOI: [10.1007/s12532-020-00179-2](https://doi.org/10.1007/s12532-020-00179-2). URL: <https://doi.org/10.1007/s12532-020-00179-2>.
- [96] E. Michael Gertz and Stephen J. Wright. “Object-oriented software for quadratic programming”. In: *ACM Trans. Math. Softw.* 29.1 (Mar. 2003), pp. 58–81. ISSN: 0098-3500. DOI: [10.1145/641876.641880](https://doi.org/10.1145/641876.641880). URL: <https://doi.org/10.1145/641876.641880>.
- [97] Brendan O’Donoghue et al. “Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding”. In: *Journal of Optimization Theory and Applications* 169.3 (June 2016), pp. 1042–1068. URL: <http://stanford.edu/~boyd/papers/scs.html>.
- [98] A. Domahidi, E. Chu, and S. Boyd. “ECOS: An SOCP solver for embedded systems”. In: *European Control Conference (ECC)*. 2013, pp. 3071–3076.
- [99] Andreas Wächter and Lorenz T. Biegler. “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming”. In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. ISSN: 1436-4646. DOI: [10.1007/s10107-004-0559-y](https://doi.org/10.1007/s10107-004-0559-y). URL: <https://doi.org/10.1007/s10107-004-0559-y>.
- [100] Ambros Gleixner et al. *The SCIP Optimization Suite 5.0*. Technical Report. Optimization Online, Dec. 2017. URL: [http://www.optimization-online.org/DB\\_HTML/2017/12/6385.html](http://www.optimization-online.org/DB_HTML/2017/12/6385.html).

- [101] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 10.0*. 2024. URL: <https://docs.mosek.com/10.0/toolbox/index.html>.
- [102] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual. Version 11.0*. 2024. URL: <https://www.gurobi.com/documentation/current/refman/index.html>.
- [103] CPLEX, IBM ILOG. “V12. 1: User’s Manual for CPLEX”. In: *International Business Machines Corporation* 46.53 (2009), p. 157.
- [104] Dongdong Ge et al. *Cardinal Optimizer (COPT) user guide*. <https://guide.coap.online/copt/eng/doc>. 2023.
- [105] Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. “Knitro: An Integrated Package for Nonlinear Optimization”. In: *Large-Scale Nonlinear Optimization*. Boston, MA: Springer US, 2006. Chap. 4, pp. 35–59. ISBN: 978-0-387-30065-8. DOI: [10.1007/0-387-30065-1\\_4](https://doi.org/10.1007/0-387-30065-1_4). URL: [https://doi.org/10.1007/0-387-30065-1\\_4](https://doi.org/10.1007/0-387-30065-1_4).
- [106] Fair Isaac Corporation, FICO. *FICO Xpress Optimizer Reference Manual. Version 9.4*. 2024. URL: <https://www.fico.com/fico-xpress-optimization/docs/latest/solver/optimizer/HTML/GUID-3BEAAE64-B07F-302C-B880-A11C2C4AF4F6.html>.
- [107] The MathWorks Inc. *Optimization Toolbox version: 9.2 (R2021b)*. Natick, Massachusetts, United States, 2021. URL: <https://www.mathworks.com/products/optimization.html>.
- [108] Ahmed Mahfouz. *Log files for the benchmark experiment over the formation reconfiguration guidance formulations*. 2024. DOI: [10.5281/zenodo.14545008](https://doi.org/10.5281/zenodo.14545008). URL: <https://doi.org/10.5281/zenodo.14545008>.
- [109] Behçet Açıkmese and Lars Blackmore. “Lossless convexification of a class of optimal control problems with non-convex control constraints”. In: *Automatica* 47.2 (2011), pp. 341–347. ISSN: 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2010.10.037>. URL: <https://www.sciencedirect.com/science/article/pii/S0005109810004516>.

- [110] Lars Blackmore, Behçet Açıkmeşe, and John M. Carson. “Lossless convexification of control constraints for a class of nonlinear optimal control problems”. In: *Systems & Control Letters* 61.8 (2012), pp. 863–870. ISSN: 0167-6911. DOI: <https://doi.org/10.1016/j.sysconle.2012.04.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0167691112000874>.
- [111] Takahiro Sasaki. “Linear Time/Parameter-Varying Models with Relative Orbit Elements Considering Continuous Controllability”. In: *Journal of Guidance, Control, and Dynamics* 0.0 (0), pp. 1–15. DOI: [10.2514/1.G008297](https://doi.org/10.2514/1.G008297).
- [112] Julian Scharnagl, Panayiotis Kremmydas, and Klaus Schilling. “Model predictive control for continuous low thrust satellite formation flying”. In: *IFAC-PapersOnLine* 51.12 (2018), pp. 12–17.
- [113] Mohd Bilal, Ria Vijayan, and Klaus Schilling. “SDRE Control with Nonlinear J2 Perturbations for Nanosatellite Formation Flying”. In: *IFAC-PapersOnLine* 52.12 (2019). 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019, pp. 448–453. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2019.11.284>.
- [114] Lukas M Steindorf et al. “Constrained low-thrust satellite formation-flying using relative orbit elements”. In: *27th AAS/AIAA Space Flight Mechanics Meeting*. Vol. 160. 2017, pp. 3563–3583.
- [115] Stefano Silvestrini and Michèle Lavagna. “Neural-aided GNC reconfiguration algorithm for distributed space system: development and PIL test”. In: *Advances in Space Research* 67.5 (2021), pp. 1490–1505. ISSN: 0273-1177. DOI: <https://doi.org/10.1016/j.asr.2020.12.014>. URL: <https://www.sciencedirect.com/science/article/pii/S027311772030870X>.
- [116] G.J. Holzmann. “The power of 10: rules for developing safety-critical code”. In: *IEEE Computer* 39.6 (2006), pp. 95–99. DOI: [10.1109/MC.2006.212](https://doi.org/10.1109/MC.2006.212).
- [117] Hans J. Königsmann et al. “Autonomous orbit maintenance system”. In: *Acta Astronautica* 39.9 (1996). IAA International Symposium on Small Satellites for Earth



- Observation, pp. 977–985. ISSN: 0094-5765. DOI: [https://doi.org/10.1016/S0094-5765\(97\)00084-2](https://doi.org/10.1016/S0094-5765(97)00084-2).
- [118] Sergio De Florio and Simone D’Amico. “The precise autonomous orbit keeping experiment on the PRISMA mission”. In: *The Journal of the Astronautical Sciences* 56.4 (2008), pp. 477–494. DOI: [10.1007/BF03256562](https://doi.org/10.1007/BF03256562).
  - [119] Andrea Garulli et al. “Autonomous Low-Earth-Orbit Station-Keeping with Electric Propulsion”. In: *Journal of Guidance, Control, and Dynamics* 34.6 (2011), pp. 1683–1693. DOI: [10.2514/1.52985](https://doi.org/10.2514/1.52985).
  - [120] Francois Bonaventure et al. “Autonomous Orbit Control for Routine Station-Keeping on a LEO Mission”. In: *Proceedings of the 23rd International Symposium on Space Flight Dynamics*. 2012. URL: <https://api.semanticscholar.org/CorpusID:202695952>.
  - [121] S. De Florio, S. D’Amico, and G. Radice. “Virtual Formation Method for Precise Autonomous Absolute Orbit Control”. In: *Journal of Guidance, Control, and Dynamics* 37.2 (2014), pp. 425–438. DOI: [10.2514/1.61575](https://doi.org/10.2514/1.61575).
  - [122] Michael Ovchinnikov, Georgi V. Smirnov, and Irina Zaramenskikh. “Orbital corrections by a single-input impulsive control applied along the geomagnetic field”. In: *Acta Astronautica* 65.11 (2009), pp. 1826–1830. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2009.04.017>.
  - [123] Marcel J Sidi. *Spacecraft dynamics and control: a practical engineering approach*. Vol. 7. Cambridge university press, 1997. ISBN: 9780511815652. DOI: <https://doi.org/10.1017/CB09780511815652>.
  - [124] Ahmed Mahfouz, Dmitry Pritykin, and James Biggs. “Hybrid Attitude Control for Nano-Spacecraft: Reaction Wheel Failure and Singularity Handling”. In: *Journal of Guidance, Control, and Dynamics* 44.3 (2021), pp. 548–558. DOI: [10.2514/1.G005525](https://doi.org/10.2514/1.G005525).
  - [125] Bong Wie, H Weiss, and Aristotle Arapostathis. “Quaternion feedback regulator for spacecraft eigenaxis rotations”. In: *Journal of Guidance, Control, and Dynamics* 12.3 (1989), pp. 375–380. DOI: [10.2514/3.20418](https://doi.org/10.2514/3.20418).

- [126] Mohammed Alhajeri and Masoud Soroush. “Tuning Guidelines for Model-Predictive Control”. In: *Industrial & Engineering Chemistry Research* 59.10 (2020), pp. 4177–4191. DOI: [10.1021/acs.iecr.9b05931](https://doi.org/10.1021/acs.iecr.9b05931).
- [127] S. Varrette et al. “Management of an Academic HPC Cluster: The UL Experience”. In: *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*. Bologna, Italy: IEEE, July 2014, pp. 959–967.
- [128] Jonathan Currie and David I. Wilson. “OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User”. In: *Foundations of Computer-Aided Process Operations*. Ed. by Nick Sahinidis and Jose Pinto. Savannah, Georgia, USA, Jan. 2012.
- [129] Stephen P Boyd and Lieven Vandenberghe. “Convex optimization”. In: Cambridge university press, 2004. Chap. 6, p. 294.