



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH



Universität
Bremen

CHARACTERIZATION AND CAPTURE OF SPACE DEBRIS OBJECTS USING DOMAIN ONTOLOGY AND OPTIMAL CONTROL

MARKO JANKOVIC

Dissertation
zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften
- Dr.-Ing. -

Vorgelegt im Fachbereich 3
Mathematik und Informatik
der Universität Bremen

28. März 2022

Marko Jankovic: *Characterization and capture of space debris objects using domain ontology and optimal control*, © 28. März 2022

GUTACHTER:

Prof. Dr. Dr. h.c. Frank Kirchner (Universität Bremen)

Dr. Edmondo Minisci (University of Strathclyde)

ORT UND DATUM DES KOLLOQUIUMS:

Bremen, 22. März 2022

One, remember to look up at the stars and not down at your feet.
Two, never give up work. Work gives you meaning and purpose and
life is empty without it. Three, if you are lucky enough to find love,
remember it is there and don't throw it away.

— Stephen Hawking

Dedicated to my family.

ABSTRACT

An ever-increasing number of in-orbit objects coupled with inadequate post-mission mitigation activities in certain orbital regions is causing a reaction from the space community more than ever. On one hand, a more sustainable usage of orbital environment is being explored while on the other, methods to actively reduce the in-orbit mass are being researched. Selecting one method over another, even for a specific object, is nonetheless not an easy task. This is mainly due to the amount of information associated with each object. At the same time, the lack of a common, standardized, machine-interpretable framework able to store that information does not facilitate matters. Among the currently investigated active debris removal (ADR) methods, robotic-based systems are perceived as the most mature and versatile ones. But they introduce challenges of their own especially in case of uncooperative objects characterized by high angular momenta (i. e. high angular rates and/or inertia). The reason for this can be found in the strict safety requirements associated with such an ADR method and its ability to manage the angular momentum post-capture. Using a domain-ontology, specifically developed for ADR, the first objective of the thesis is to provide a standardized, machine-interpretable knowledge representation framework. This framework is capable not only of efficiently storing complex information but also inferring the most suited ADR capture method(s) for an object from its minimal set of parameters. The second objective of the thesis is to provide an optimization-based motion planner of a spacecraft equipped with a manipulator, a space robot. The purpose of the planner is to facilitate an autonomous capture of an uncooperative, rapidly tumbling target by means of a space robot. This is achieved by taking the advantage of the particular dynamics of the space robot and an appropriate partial attitude synchronization strategy. By doing so the planner is able to find an optimal coordinated motion of the whole spacecraft to approach a rapidly tumbling object while minimizing its overall control effort. The practicality and validity of the ontology are demonstrated by applying it onto a database of representative objects. The latter was built by combining structured and unstructured data from publicly available sources. The analysis of results proves the ontology capable of inferring the most suited ADR capture methods for considered objects. Furthermore, it confirms its ability to handle the input data from different sources transparently, minimizing user input. This should make the initial planning of future ADR missions simpler, yet more systematic. The performance of the planner is evaluated in 3D simulation environment using different test cases. Each test case is

characterized by different kinematic configuration of the manipulator and angular velocities of the object. The analysis of results confirms the ability of the planner to provide optimal, well-behaved state trajectories of the spacecraft under a range of different conditions. In this manner, a future application of the planner should facilitate close-proximity operations to objects with high angular momenta using a space robot.

ZUSAMMENFASSUNG

Die ständig wachsende Zahl von Objekten in der Umlaufbahn und die unzureichenden Maßnahmen zur Minderung der Auswirkungen nach der Mission in bestimmten Orbitalregionen rufen mehr denn je Reaktionen in der Weltraumgemeinschaft hervor. Einerseits wird an nachhaltigen Nutzungen in der Orbitalumgebung geforscht, andererseits an Methoden zur aktiven Reduzierung der Masse im Orbit. Die Auswahl einer Methode gegenüber einer anderen, selbst nur für ein bestimmtes Objekt, ist jedoch keine leichte Aufgabe. Das liegt vor allem an der Fülle der Informationen, die mit jedem Objekt verbunden sind. Das Fehlen eines gemeinsamen, standardisierten und maschineninterpretierbaren Rahmens für die Speicherung dieser Informationen macht die Sache nicht einfacher. Unter den derzeit untersuchten Methoden der aktiven Weltraumtrümmerbeseitigung (active debris removal (ADR) auf Englisch) gelten roboterbasierte Systeme als die ausgereiftesten und vielseitigsten. Sie bringen jedoch eigene Herausforderungen mit sich, insbesondere bei unkooperativen Objekten, die sich durch hohe Drehimpulse (d. h. hohe Drehraten und/oder Trägheit) auszeichnen. Der Grund dafür liegt in den strengen Sicherheitsanforderungen, die mit einer solchen ADR-Methode verbunden sind, und in ihrer Fähigkeit, den Drehimpuls nach dem Einfangen zu verwalten. Unter Verwendung einer speziell für ADR entwickelten Domänenontologie besteht das erste Ziel der vorliegenden Arbeit darin, einen standardisierten, maschineninterpretierbaren Wissensrepräsentationsrahmen bereitzustellen. Dieser Rahmen ist nicht nur in der Lage, komplexe Informationen effizient zu speichern, sondern auch die am besten geeignete(n) ADR-Erfassungsmethode(n) für ein Objekt aus seinem minimalen Satz von Parametern abzuleiten. Das zweite Ziel dieser Arbeit ist es, einen optimierungsbasierten Bewegungsplaner für ein mit einem Manipulator, einem Weltraumroboter, ausgestattetes Raumfahrzeug zu erstellen. Der Zweck des Planers besteht darin, das autonome Einfangen eines unkooperativen, schnell taumelnden Ziels mit Hilfe eines Weltraumroboters zu erleichtern. Dies wird erreicht, indem die besondere Dynamik des Weltraumroboters und eine geeignete Strategie zur partiellen Lagesynchronisation ausgenutzt werden. Auf diese Weise ist der Planer in der Lage, eine optimale koordinierte Bewegung des gesamten Raumfahrzeugs zu finden, um sich einem schnell taumelnden Objekt zu nähern und dabei den gesamten Steuerungsaufwand zu minimieren. Die Praxistauglichkeit und die Validität der Ontologie werden durch ihre Anwendung auf eine Datenbank mit repräsentativen Objekten demonstriert. Letztere wurde durch die Kombination strukturierter und unstrukturierter

Daten aus öffentlich zugänglichen Quellen erstellt. Die Analyse der Ergebnisse zeigt, dass die Ontologie in der Lage ist, die am besten geeigneten ADR-Erfassungsmethoden für die betrachteten Objekte abzuleiten. Darüber hinaus wird die Fähigkeit der Ontologie bestätigt, die Eingabedaten aus verschiedenen Quellen transparent zu verarbeiten und so die Benutzereingaben zu minimieren. Dies sollte die anfängliche Planung zukünftiger ADR-Missionen einfacher, aber auch systematischer machen. Die Leistung des Planers wird in einer 3D-Simulationsumgebung anhand verschiedener Testfälle bewertet. Jeder Testfall ist durch unterschiedliche kinematische Konfigurationen des Manipulators und Winkelgeschwindigkeiten des Objekts gekennzeichnet. Die Analyse der Ergebnisse bestätigt, dass der Planer in der Lage ist, optimale, gut funktionierende Flugbahnen des Raumfahrzeugs unter einer Reihe von unterschiedlichen Bedingungen zu erstellen. Auf diese Weise sollte eine zukünftige Anwendung des Planers Operationen, in unmittelbarer Nähe von Objekten mit hohen Drehimpulsen, mit Hilfe eines Weltraumroboters erleichtern.

PUBLICATIONS

The publications containing previously published parts of the thesis and related to its overall topic are the following:

JOURNAL ARTICLES

- [1] M. Jankovic, J. Paul, and F. Kirchner. "GNC Architecture for Autonomous Robotic Capture of a Non-Cooperative Target: Preliminary Concept Design." In: *Advances in Space Research* 57.8 (Apr. 15, 2016), pp. 1715–1736. ISSN: 02731177. DOI: 10/f8j2gw.
- [2] M. Jankovic et al. "Space Debris Ontology for ADR Capture Methods Selection." In: *Acta Astronautica* 173 (Aug. 1, 2020), pp. 56–68. ISSN: 0094-5765. DOI: 10/ggsd2b.

BOOK CHAPTERS

- [1] M. Jankovic and J. Paul. "Robotic Active Debris Removal and On-Orbit Servicing." In: *Asteroid and Space Debris Manipulation: Advances from the Stardust Research Network*. Progress in Astronautics and Aeronautics. Reston, VA, USA: AIAA, Jan. 2016, pp. 329–372. ISBN: 978-1-62410-323-0. URL: <https://bit.ly/2qeUGzw> (visited on 10/17/2019).
- [2] M. Jankovic, N. Thiry, and P. Bourdon. "Lasers for Asteroid and Debris Deflection." In: *Asteroid and Space Debris Manipulation: Advances from the Stardust Research Network*. Progress in Astronautics and Aeronautics. Reston, VA, USA: AIAA, Jan. 2016, pp. 291–328. ISBN: 978-1-62410-323-0. URL: <https://bit.ly/2IXYHit> (visited on 10/17/2019).

CONFERENCE PAPERS

- [1] M. Jankovic et al. "Robotic System for Active Debris Removal: Requirements, State-of-the-Art and Concept Architecture of the Rendezvous and Capture (RVC) Control System." In: *5th CEAS Air & Space Conference Proceedings*. Delft, The Netherlands: Council of European Aerospace Societies (CEAS), 2015, pp. 1–15. URL: <https://tinyurl.com/yjz3a86c> (visited on 10/11/2019).

- [2] M. Jankovic et al. "Spacecraft Concept for Active De-Tumbling and Robotic Capture of Ariane Rocket Bodies (Poster)." In: *13th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*. Noordwijk, The Netherlands: ESA Publications Division-ESTEC, 2015, p. 1. URL: <https://tinyurl.com/yzt4no9y> (visited on 10/11/2019).
- [3] M. Jankovic and F. Kirchner. "Taxonomy of LEO Space Debris Population for ADR Selection." In: *67th International Astronautical Congress (IAC)*. Guadalajara, Mexico: International Astronautical Federation (IAF), Oct. 2016, pp. 1–15. URL: <https://bit.ly/2V0jnPb> (visited on 10/17/2019).
- [4] M. Jankovic and F. Kirchner. "Taxonomy of LEO Space Debris Population for ADR Capture Methods Selection." In: *Stardust Final Conference*. Ed. by M. Vasile et al. Astrophysics and Space Science Proceedings. Springer International Publishing, 2018, pp. 129–144. ISBN: 978-3-319-69956-1. DOI: 10/gf93mr.
- [5] M. Jankovic and F. Kirchner. "Trajectory Generation Method for Robotic Free-Floating Capture of a Non-cooperative, Tumbling Target." In: *Stardust Final Conference*. Ed. by M. Vasile et al. Astrophysics and Space Science Proceedings. Springer International Publishing, 2018, pp. 111–127. ISBN: 978-3-319-69956-1. DOI: 10/gf93ms.
- [6] K. Kumar et al. "Agora: Mission to Demonstrate Technologies to Actively Remove Ariane Rocket Bodies." In: *66th International Astronautical Congress (IAC)*. Jerusalem, Israel: International Astronautical Federation (IAF), Oct. 2015, pp. 1–16. URL: <https://iafastro.directory/iac/archive/browse/IAC-15/A6/6/28851/>.
- [7] S. Vyas, M. Jankovic, and F. Kirchner. "Momentum Based Classification for Robotic Active Debris Removal." In: *71st International Astronautical Congress (IAC)*. Online: International Astronautical Federation (IAF), Oct. 2020, pp. 1–11. DOI: 10/gkptq2.

Disclaimer on content reuse: this thesis expands on the already published work. Therefore, it incorporates single sentences, entire sections as well as figures and tables from the listed publications without explicit quotation. However, a full acknowledgment of the original publications from which the contents was borrowed is clearly made at the beginning of each chapter. Furthermore, sources of the reprinted/adapted figures as well as tables from prior publications are credited in captions.

ACKNOWLEDGMENTS

The work presented in this thesis is the result of an academic and personal journey that started at DFKI-Robotics Innovation Center (RIC) in Bremen with the European Union's Seventh Framework Programme FP7/2007-2013/ under the People Programme (Marie Curie Actions), Stardust, grant agreement No. 317185. Therefore, first and foremost, I would like to thank the founder of RIC and my PhD supervisor Prof. Dr. Dr. h.c. Frank Kirchner for giving me the unique opportunity to pursue my research on space debris and ADR at the institute. Without his support and guidance over the past years my research and this thesis would have never been possible.

I would also like to express my gratitude to Dipl.-Inf. Veit Briken and Dr. rer. nat. Jan Paul, project leaders of the Stardust project at RIC, for their constructive feedback, continuous encouragements and patience during the countless meetings we had within the project.

My sincere thanks also goes to Dr.-Ing. José de Gea Fernández for his unwavering support and expert advice that helped me discern the "right path" in more than one occasion.

A PhD journey requires more than academic support. For this I would like to thank brilliant colleagues and peers with whom I had a chance to interact during the years and who have influenced my research in one way or another. In particular, I would like to thank a PhD adventure companion Mehmed Yüksel for his friendship and continuous support both professional as well as personal. A special thanks I also owe to Dipl.-Ing. (FH) Wiebke Brinkmann for her enduring encouragements and relentless reviewing of the thesis that helped shape it to this final form. My thanks also goes to Dr.-Ing. Florian Cordes for his time and thorough reviews of my articles and parts of the thesis that pushed me to improve my writing style and bring it another level. I would also like to extend my gratitude to Stardust colleagues from various institutions across the globe for making the initial years of this journey memorable.

Finally, none of this would have been possible without the most important people in my life, my family. It is thanks to my family that I am the person who I am today. To my wife, Milena, I owe my deepest gratefulness. This thesis is a testament of her unconditional support, unselfish love and continuous motivation to pursue my passions even in times when all seemed lost and I was ready to quit. Lastly, my heartfelt thanks goes to my parents, Ljubisa and Divna, as well as my brother, Dimitrije. I owe my accomplishments to them because they never stopped believing in me despite all the hardships we endured together.

CONTENTS

1	INTRODUCTION	1
1.1	Motivation	1
1.2	Objectives	3
1.3	Structure	4
2	BACKGROUND AND STATE-OF-THE-ART	7
2.1	Space debris environment and risk	7
2.2	Space debris risk management	12
2.2.1	Space debris mitigation	12
2.2.2	Space debris remediation	16
2.3	Space debris information paradox	20
2.3.1	Active debris removal methods assessment	20
2.3.2	Knowledge representation of space debris	25
2.4	Space robotics	28
2.4.1	Past and future missions	28
2.4.2	Robotic capture of uncooperative objects	33
3	SPACE DEBRIS ONTOLOGY FOR ADR	41
3.1	Statistical analysis of breakups	41
3.1.1	Database and data description	43
3.1.2	Survival analysis of breakups	44
3.1.3	Rocket bodies related breakups	51
3.1.4	Payloads related breakups	55
3.2	Core terminology and methodology	59
3.2.1	Introduction into knowledge representation	59
3.2.2	Domain of interest and scope	61
3.2.3	Main classes and class hierarchy	62
3.2.4	Breakup criticality	65
3.2.5	ADR capture method	69
3.3	Ontology development	73
3.3.1	Database generation and import	74
3.3.2	Ontology implementation	75
3.4	Ontology application	78
3.4.1	Example database	79
3.4.2	Classification results	81
3.5	Analysis and discussion	84
3.5.1	Results analysis	84
3.5.2	Discussion	85
4	ROBOTIC CAPTURE OF AN UNCOOPERATIVE TARGET	87
4.1	Mathematical modeling of a space robot	87
4.1.1	Kinematic equations	88
4.1.2	Dynamic equations	93
4.2	Introduction to optimal control	96
4.2.1	Optimal control problem	97

4.2.2	Nonlinear programming problem	98
4.2.3	Direct transcription	99
4.3	Semisynchronizing approach problem	103
4.3.1	Assumptions and reference mission	104
4.3.2	Optimal control problem formulation	110
4.3.3	Optimal control problem implementation	115
4.4	Numerical simulations	119
4.4.1	Simulation test cases	120
4.4.2	Results	128
4.5	Analysis and discussion	142
4.5.1	Results analysis	142
4.5.2	Discussion	143
5	CONCLUSIONS AND FUTURE WORK	147
5.1	Conclusions	147
5.2	Future work	149
	BIBLIOGRAPHY	153

LIST OF FIGURES

Figure 1.1	Thesis structure	4
Figure 2.1	Evolution of objects per objects type	9
Figure 2.2	Evolution of objects per orbit type	11
Figure 2.3	Projected evolution of objects in orbit	13
Figure 2.4	Projections of LEO environment	15
Figure 2.5	Classification of ADR devices	23
Figure 2.6	Space Shuttle Manipulator System	29
Figure 2.7	Special Purpose Dextrous Manipulator	30
Figure 2.8	Artist's concept of e.Deorbit mission	32
Figure 2.9	Artist's concept of OSAM-1 mission	33
Figure 3.1	Evolution of breakups per event cause	42
Figure 3.2	Example of right-censored data	46
Figure 3.3	Spacecraft survival probability	49
Figure 3.4	Spacecraft cumulative hazard	50
Figure 3.5	Rocket body survival probability	52
Figure 3.6	Rocket body breakup hazard estimate	53
Figure 3.7	Pareto chart of rocket bodies breakup events	54
Figure 3.8	Box plot of rocket bodies propulsion events	55
Figure 3.9	Payload survival probability	56
Figure 3.10	Payload breakup hazard estimate	57
Figure 3.11	Pareto chart of payloads breakup events	58
Figure 3.12	Box plot of payloads breakup events	58
Figure 3.13	Example diagram of an ontology	61
Figure 3.14	Extract from class hierarchy of TRACER	64
Figure 3.15	Development workflow of TRACER library	73
Figure 3.16	TRACER within Protégé Desktop	76
Figure 3.17	Usage workflow of TRACER library	79
Figure 3.18	Box plot of example breakup probabilities	80
Figure 3.19	Classification results per target class	82
Figure 3.20	Classification results per target criticality	83
Figure 3.21	Classification results per target attitude	83
Figure 4.1	Schematic model of a single-arm space robot	88
Figure 4.2	Mission phases of a robotic capture	106
Figure 4.3	Geometry of a torque-free motion	109
Figure 4.4	Semisynchronizing approach illustration	111
Figure 4.5	Implementation example of POSE	112
Figure 4.6	POSE algorithm illustration	118
Figure 4.7	Model of the chaser spacecraft	120
Figure 4.8	Mock-up of a Kosmos-3M second stage	121
Figure 4.9	3 DOF chaser and target initial configurations	129
Figure 4.10	q_0 and q_m optimal trajectories of 3 DOF test	130

Figure 4.11	\dot{q}_0 and \dot{q}_m optimal trajectories of 3 DOF test	131
Figure 4.12	τ_0 and τ_m optimal trajectories of 3 DOF test	132
Figure 4.13	p_e and v_e trajectories of 3 DOF test	133
Figure 4.14	Collision distances of 3 DOF test	133
Figure 4.15	\dot{q}_m optimal trajectories of 3 DOF tests	134
Figure 4.16	7 DOF chaser and target initial configurations	135
Figure 4.17	q_0 and q_m optimal trajectories of 7 DOF test	136
Figure 4.18	\dot{q}_0 and \dot{q}_m optimal trajectories of 7 DOF test	137
Figure 4.19	τ_0 and τ_m optimal trajectories of 7 DOF test	138
Figure 4.20	p_e , \dot{Q}_e , v_e and ω_e trajectories of 7 DOF test	139
Figure 4.21	Collision distances of 7 DOF test	140
Figure 4.22	\dot{q}_m trajectories of 7 DOF tests	141

LIST OF TABLES

Table 3.1	Severity levels definition	65
Table 3.2	Probability levels definition	66
Table 3.3	Severity numbers of rocket bodies breakups	67
Table 3.4	Severity numbers of payloads breakups	67
Table 3.5	Criticality matrix	68
Table 3.6	Uncooperativeness levels characteristics	70
Table 3.7	Capture methods classification axioms	72
Table 3.8	TRACER metrics	77
Table 3.9	Objects characteristics of the example database	80
Table 3.10	Classification results per capture method class	81
Table 4.1	Inertial parameters of the chaser and target	121
Table 4.2	State bounds of 3 DOF tests	124
Table 4.3	State bounds of 7 DOF tests	127
Table 4.4	Performance summary of 3 DOF tests	129
Table 4.5	Performance summary of 7 DOF tests	139

LISTINGS

Listing 3.1	Definition of the Medium_Tumbling_Regime class	63
Listing 3.2	Definition of the High_Criticality class	69
Listing 3.3	Definition of the Manipulator_Based class	71

ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
ACS	Attitude Control System
ADR	Active Debris Removal
AMR	Area-to-Mass Ratio
API	Application Programming Interface
ASAP	Ariane Structure for Auxiliary Payloads
CAM	Collision Avoidance Maneuver
CF	Cataloged Fragment
CN	Criticality Number
CNES	Centre National d'Études Spatiales (France)
COM	Center of Mass
COSPAR	Committee on Space Research
COSPAR ID	COSPAR International Designator
CPU	Central Processing Unit
CQ	Competency Question
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz (Germany)
DISCOS	Database and Information System Characterising Objects in Space
DLR	Deutsches Zentrum für Luft- und Raumfahrt (Germany)
DOF	Degrees of Freedom
ENVISAT	Environment Satellite (ESA)
EOL	End of Life
ESA	European Space Agency
ETS-VII	Engineering Test Satellite VII "KIKU-7" (JAXA)
EVA	Extra-Vehicular Activity
FMEA/FMECA	Failure Modes, Effects (and Criticality) Analysis
GA	Genetic Algorithm
GEO	Geostationary Earth Orbit
GNC	Guidance, Navigation and Control
GTO	GEO Transfer Orbit
GUI	Graphical User Interface
GmbH	Gesellschaft mit beschränkter Haftung
HEO	Highly Eccentric Earth Orbit

HIL	Hardware-in-the-Loop
IAA	International Academy of Astronautics
IADC	Inter-Agency Space Debris Coordination Committee
ICLOCS	Imperial College London Optimal Control Software
IDO	Intact Derelict Object
IPOPT	Interior Point Optimizer
ISO	International Organization for Standardization
ISS	International Space Station
JAXA	Japan Aerospace Exploration Agency (Japan)
JCA	Just-in-time Collision Avoidance
LAR	Launch Adapter Ring
LEGEND	LEO-to-GEO Environment Debris Model (NASA)
LEO	Low Earth Orbit
LNT	Lethal Non-Trackable Debris
LOS	Line-of-Sight
LVA	Launch Vehicle Adapter
MEO	Medium Earth Orbit
MRAVS	Minimum Relative Angular Velocity Strategy
NASA	National Aeronautics and Space Administration (US)
NLP	Nonlinear Programming
OC	Optimal Control Problem
ODE	Ordinary Differential Equation
ODP	Ontology Design Pattern
OOS	On-Orbit Servicing
OSAM-1	On-orbit Servicing, Assembly, and Manufacturing 1
OWL	Web Ontology Language
PC	Personal Computer
PMD	Post-Mission Disposal
PN	Probability Number
POSE	Planner for Semisynchronizing Capture
PSO	Particle Swarm Optimization
RAM	Random-Access Memory
RCS	Radar Cross-Section
RDF	Resource Description Framework
RIC	Robotics Innovation Center
ROTEX	Robot Technology Experiment
RSO	Resident Space Object
RVC	Rendezvous and Capture
SN	Severity Number
SPART	Spacecraft Robotics Toolkit

SQP	Sequential Quadratic Programming
SSA	Space Situational Awareness
STS	Space Transportation System
SWRL	Semantic Web Rule Language
TRACER	Ontology for Active Debris Removal
TRL	Technology Readiness Level
UK	United Kingdom of Great Britain and Northern Ireland
UN	United Nations
URDF	Unified Robot Description Format
US	United States (of America)
USA	United States of America
W ₃ C	World Wide Web Consortium
R. H. S.	right hand side
W. R. T.	with respect to

SYMBOLS

LATIN SYMBOLS

$\mathbf{0}_{m \times n}$	$m \times n$ matrix of zeros
$\mathbf{1}_{m \times n}$	$m \times n$ matrix of ones
A	area (m^2)
\mathcal{C}	configuration space of a robot
\mathbf{c}	nonlinear centripetal and Coriolis forces ($\text{kg m}^2 \text{s}^{-1}$)
\mathbf{C}	Coriolis matrix ($\text{kg m}^2 \text{s}^{-1}$)
\mathbf{f}	vector of forces (N)
$F(t)$	cumulative distribution function
$\hat{F}(t)$	estimate of $F(t)$
$f(t)$	probability density function
\mathbf{H}	inertia matrix (kg m^2)
\mathbb{H}	set of all quaternions
$H(t)$	cumulative hazard function
$\hat{H}(t)$	estimate of $H(t)$
$h(t)$	hazard function
\mathbf{I}	inertia tensor (kg m^2)
$\mathbf{I}_{n \times n}$	$n \times n$ identity matrix
\mathbf{J}	Jacobian matrix

L	angular momentum ($\text{kg m}^2 \text{s}^{-1}$)
\mathcal{L}	Lagrangian of a dynamical system
m	mass (kg)
\mathbf{n}	vector of torques (N m)
P	linear momentum (kg m s^{-1})
\mathbf{p}	position vector (m)
P	probability of an event
\mathbf{q}	vector of generalized coordinates (m, rad)
$\dot{\mathbf{q}}$	vector of generalized velocities (m s^{-1} , rad s^{-1})
\mathcal{Q}	quaternion
$\dot{\mathcal{Q}}$	quaternion derivative
$\hat{\mathcal{Q}}$	unit quaternion
\mathbf{r}	position vector of a mass center of a body (m)
\mathbb{R}	set of real numbers
$R(t)$	reliability function
$\hat{R}(t)$	estimate of $R(t)$
\mathbb{R}^3	set of 3-D points
$S(t)$	survivor function
\mathbb{S}^3	three-dimensional unit sphere in \mathbb{R}^4
$\text{SE}(3)$	special Euclidean group representing the set of all poses in 3 dimensions
T	time to failure (s)
\mathcal{T}	kinetic energy of a system (J)
\mathbf{T}	homogeneous transformation matrix
t	time (s)
\mathcal{U}	potential energy of a system (J)
$\mathbf{u}(t)$	control variables (of an OCP)
\mathcal{V}	spatial velocity vector (m s^{-1} , rad s^{-1})
\mathbf{v}	velocity vector (m s^{-1})
$\mathbf{y}(t)$	state variables (of an OCP)
\mathbb{Z}	set of all integers

GREEK SYMBOLS

Γ	3-angle (XYZ roll-pitch-yaw) representation of a rotation (rad)
Σ	coordinate frame
$\boldsymbol{\varepsilon}$	vector of performance values (of an OCP variable)
η	scalar part of a quaternion
θ	angle of rotation about an axis (rad)
σ^2	variance
$\hat{\sigma}^2$	estimate of σ^2
$\boldsymbol{\tau}$	vector of generalized forces (N, N m)
ω	angular, or rotational, rate (rad s^{-1})
$\boldsymbol{\omega}$	angular velocity vector (rad s^{-1})
$\boldsymbol{\varepsilon}$	vector part of a quaternion

$\dot{\phi}$ vector of angle velocities, generally joint velocities
(rad s⁻¹)

OPERATORS

$[\cdot]_{\times}$ skew-symmetric matrix
 $[\cdot]^{-1}$ inverse of a matrix
 $[\cdot]^T$ transpose of a matrix
 \times cross or vector product
 \hat{v} pure quaternion
 \cdot dot or scalar product
 $\|\cdot\|$ norm, magnitude or length of a vector
 \otimes quaternion (Hamiltonian) product operator

CONSTANTS

π pi (approximately 3.14159): the ratio of the length of the circumference of a circle to its diameter

1

INTRODUCTION

As an entry point into the thesis, this chapter introduces the overall field of research, the underlying problem and points out unsolved challenges related to this work. Then, it defines the objectives of the thesis and outlines the main contributions in the context of those objectives. Finally, it describes the structure of the document and highlights the main publications related to each chapter.

1.1 MOTIVATION

The launch of the first artificial satellite, Sputnik-1 on 4 October 1957, marked the beginning of the human space exploration. It also marked the birth of non-functional, man-made, Earth-orbiting objects, denoted as *space debris*. Even at the very start of the human space exploration and before the first ever recorded fragmentation event, almost half of the cataloged objects were space debris (i. e. mission related objects and rocket bodies), generated as a result of previous launch activities. Since then this trend has continued and there has been a steady increase in the total number of objects tracked and cataloged by space surveillance networks, whose origins are either intentional, as a result of normal launch activities, or unintentional, owing their existence to unplanned releases of equipment, burns of solid rocket motors, etc [1]. It is therefore understandable that there is a growing concern of the society at large over the space debris issue, that places vital services, that we have come to depend on a daily basis, in a constant danger of collision. However, dealing with this issue is at present a challenging task, considering the amount of uncertainty surrounding it [2]. In fact, currently, we can only reliably detect and catalog objects that are larger than 0.05 m to 0.1 m in low Earth orbit (LEO) and 0.3 m to 1 m in geostationary Earth orbit (GEO) [3]. The population of smaller objects can be estimated only by dedicated models and validated through retrieved surfaces which were exposed to the space debris environment. In addition, although certain orbital regions, such as LEO, have theoretically exceeded the critical density of objects that would ensure the onset of a self-sustaining collision process, named the *Kessler syndrome* [4], it is uncertain when and in which measure exactly will this phenomenon appear on-orbit [2]. The currently implemented space debris mitigation guidelines have proven to be necessary but insufficient activities to manage the risk posed by the space debris and maintain a stable space debris environment. The only way to

permanently stabilize the current orbital environment and prevent the onset of the Kessler syndrome consists of actively removing the existing large sources of debris, via remediation activities. This way future collisions between large objects can be prevented and with them the generation of fragmentation debris that are more difficult to track and, currently, neither practical nor economically feasible to remove actively [2, 5].

Among the currently investigated remediation activities, ADR is perceived as the only one able to permanently reduce the number of objects in orbit, at the expense of difficult mission planning, execution and necessity to act over a long period. Among all the phases of a generic ADR mission, the *capture* emerges as one of the most challenging ones (along with the *close-range rendezvous* and *removal*). It consists of actions, performed by a chaser spacecraft, to capture a target, stabilize the compound and prepare it for disposal (i. e. de- or re-orbit). Methods able to achieve such a task that are currently being researched are numerous and span from more familiar ones, involving robotic systems requiring close-range operations and a physical contact with a target, to more innovative ones using electromagnetic or electrostatic forces to achieve the “capture” by actively controlling the attitude of a target from a stand-off distance. Every method has its own advantages and disadvantages, but there is unfortunately not one that can tackle all possible targets. Even considering one specific target, it is not guaranteed to be able to easily identify its most suited capture method(s), as demonstrated by the e.Deorbit study [6]. Furthermore, the available data about cataloged objects and their physical as well as dynamical characteristics often lacks coherence and structure, thus hindering data sharing, collaboration and ultimately decision-making.

Selecting the most suited capture method(s) for a specific target solves only one part of the problem as achieving the capture itself presents a range of additional challenges on its own. This is especially true in case of targets characterized by a tumbling motion¹ and capture methods requiring physical contact, such as robotics-based systems. Under these circumstances, a capture maneuver needs to comply with strict safety requirements to avoid unintentional contacts with a target that might generate even more debris. Additionally, the maneuver will also need to consider any residual angular momentum that the target might have at the moment of contact. Not doing so will most likely result in a failed maneuver due to the limited amount of momentum that can be transferred from the target to the chaser and subsequently be absorbed by its attitude control system (ACS). Finally, depending on the type of mission, i. e. single-target or multi-target, the capture maneuver might also need to account for limited on-board resources, such as propellant, to increase the mission duration as much as possible. For these reasons and despite years of studies, a capture of a truly

¹ An attitude motion of an object characterized by an attitude error interval $[0,180]$ deg and an attitude rate error interval $[1,18]$ deg s⁻¹ [7].

uncooperative object without humans-in-the-loop remains an open challenge and has yet to be demonstrated in-orbit [8].

1.2 OBJECTIVES

The objective of this thesis is to provide solutions to the previously raised issues affecting the current ADR research in the form of:

1. a *domain-ontology* of large, intact objects, i. e. payloads and rocket bodies, to enable automatic ADR capture method(s) selection, to ease sharing and reuse of the domain knowledge;
2. an optimization-based *motion planner* to facilitate a space robot autonomously capturing an uncooperative, rapidly tumbling target with a multi-target mission in mind.

The term *ontology* in this work defines a method to model a field of discourse by explicitly defining the domain concepts, relationships among them, their properties and restrictions [9]. Likewise, the expression *space robot* identifies a spacecraft equipped with a manipulator.

The developed ontology for active debris removal (TRACER) enables ADR researchers to use standardized, machine-interpretable vocabulary of characteristics of cataloged objects, and relations among them, to analyze the domain and infer new knowledge, making the initial ADR mission planning easier and more systematic. The nature of the associated results presented hereafter is both theoretical and applied, since not only the methodology used to develop TRACER is described, but also workflows used for its software implementation and usage. With respect to the existing state-of-the-art, focused mainly on the space situational awareness (SSA), TRACER covers specifically the domain of ADR capture methods and establishes the minimal number of parameters needed to identify the most suited capture method(s) for a specific target. Furthermore, it describes a way to handle the input of data from an existing database of cataloged objects, an aspect often overlooked by the current state-of-the-art.

The developed motion planner for semisynchronizing capture (POSE), provides a tool to generate locally optimal, coordinated motions of both the spacecraft and manipulator to approach a rapidly tumbling target that minimize the overall control effort and satisfy other constraints, such as collision avoidance, compliance to final conditions, kinematic and dynamic limitations of the chaser spacecraft. To achieve such motions the planner takes the advantage of the dynamics of the space robot and the minimum relative angular velocity strategy (MRAVS) [10]. The latter exploits the geometric properties of the rotational dynamics of tumbling objects to achieve a partial motion synchronization between the chaser and target while minimizing the relative angular velocity, unlike the main stream solutions consisting

of a complete relative motion synchronization preceding the capture maneuver. With respect to the existing state-of-the-art, confined mainly to single-target missions and capture of targets with relatively low tumbling rates, POSE specifically addresses the issue of capturing a rapidly tumbling target with multi-target missions in mind.

1.3 STRUCTURE

The structure of the thesis consists of five chapters. Fig. 1.1 shows the overall outline of the document and lists the publications supporting each chapter.

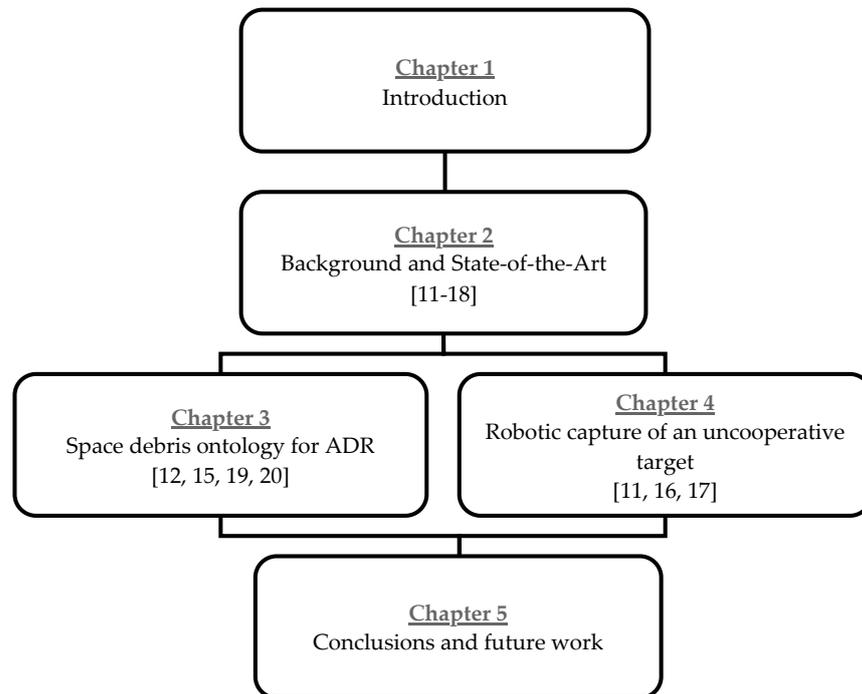


Figure 1.1: Thesis structure and supporting publications.

Chapter 1 introduces the motivation and objective of the thesis and presents its structure.

Chapter 2 describes in detail the background and motivation of the thesis consisting of the current space debris environment, future projections and methods to manage the risk posed by space debris. It also presents the problem of the *information paradox of space debris* and reviews the state-of-the-art in the field of space debris *knowledge representation*. It concludes with a review of most influential space robotics missions and a survey of the state-of-the-art studies of the robotic capture of uncooperative objects. The publications supporting this chapter are [11–18].

Chapter 3 presents the developed domain-ontology for ADR capture method(s) selection, TRACER. First, a brief overview of the statistical analysis of fragmentation events (i. e. breakups) of rocket bodies and payloads, needed by the ontology, is provided. Then, the core terminology and methodology of TRACER are described, followed by its software implementation and development workflows. Next, results of the application of TRACER onto a database of representative objects are presented. Finally, the analysis of those results, along with the strengths and weaknesses of TRACER are discussed. The publications supporting this chapter are [12, 15, 19, 20].

Chapter 4 focuses on the description of the developed motion planner for a capture of a rapidly tumbling target, POSE. The chapter begins with a brief overview of the main mathematical notation and equations of motion of a space robot. This is followed by an introduction to nonlinear optimization and methods to solve an optimal control problem (OCP). This leads to a formulation of the *semisynchronizing approach problem*, which is at the heart of the developed planner, and the description of the software implementation logic of POSE. Its evaluation in 3D simulation is described next and results are illustrated. The chapter concludes with the analysis of the performance of POSE in light of the simulation results and subsequent discussion. The publications supporting this chapter are [11, 16, 17].

Chapter 5 draws conclusions of the presented research and identifies future direction that might further it based on the presented results.

2

BACKGROUND AND STATE-OF-THE-ART

Disclaimer on content reuse: this chapter is based on the following publications: [11–13, 15, 16]ⁱ.

Space debris, a byproduct of more than 60 years of space exploration and exploitation, has rapidly evolved from theoretical studies to a serious threat that needs to be considered nowadays even in early stages of a mission development. The only way to permanently stabilize the current orbital environment is to actively remove existing in-orbit mass using remediation methods such as ADR. How to achieve such a task is still an open research question both in terms of selecting a suitable ADR method for a specific target as well as enabling a capture of an uncooperative, tumbling target using a space robot.

In this chapter an in depth description of the background and state-of-the-art of this work is provided. The space debris environment and the associated risk are first presented. This is followed by a description of methods that could be used to reduce the current risk due to the ever-increasing in-orbit objects and decades of space activity. Next the space debris information paradox and the literature in the field of knowledge representation of space debris are analyzed pointing out the existing challenges. The chapter continues with the description of the state-of-the-art space robotics missions and concludes with the review of the literature on the motion planning of space manipulators to capture an uncooperative, tumbling target.

2.1 SPACE DEBRIS ENVIRONMENT AND RISK

As of 2020, there have been a total of 5567 successful launches, that deployed 9364 payloads, 6290 rocket bodies and 7612 mission related objects. Of those 4707, 1871 and 1241, respectively, are still in orbit and only ~ 1950 [21] is attributable to functioning spacecraft. These intact objects account for most of the total in-orbit mass, amounting to 8447.5 t, while representing only 34.4 % of the 22756 objects routinely tracked and cataloged by space surveillance networks. The number of debris having sub-catalog dimensions (i. e. < 0.1 meter), but

ⁱ Some of the text in this chapter is reprinted/adapted by permission from Springer Nature Customer Service Centre GmbH: Springer International Publishing, *Stardust Final Conference. Astrophysics and Space Science Proceedings. "Taxonomy of LEO Space Debris Population for ADR Capture Methods Selection"*, DOI: 10/gf93mr and "Trajectory Generation Method for Robotic Free-Floating Capture of a Non-cooperative, Tumbling Target", DOI: 10/gf93ms, M. Jankovic and F. Kirchner, © 2018.

nonetheless capable of degrading a mission or even terminating it, is even grater. Their numbers can be estimated only statistically and are determined to be around 900 000 for objects within 0.01 m to 0.1 m range and around 128 million for objects within 0.001 m to 0.01 m range [22]. For reference, space debris objects of up to 0.01 m represent the shielding limit of the International Space Station (ISS) modules.

The historic evolution of the number and mass of trackable on-orbit objects (e. g. larger than 0.05 m in LEO and 0.3 m GEO) is illustrated in Fig. 2.1. The object types identified in the figure determine the source categories of cataloged objects and are defined as follows [21]:

PAYLOADS space objects designated to perform a certain function, e. g. satellites, calibration objects, etc.

PAYLOAD MISSION RELATED OBJECTS space objects released as space debris during the normal functioning of a payload, e. g. a cover of an optical instrument.

PAYLOAD FRAGMENTATION DEBRIS space objects unintentionally released by one or more identifiable parent payloads as a result of a unique event (e. g. a collision, a breakup, etc.).

PAYLOAD DEBRIS space objects unintentionally released by one or more identifiable parent payloads as a result of an unclear event.

ROCKET BODIES space objects performing orbit insertion of payloads (excluding those releasing smaller spacecraft themselves), e. g. launch vehicle stages.

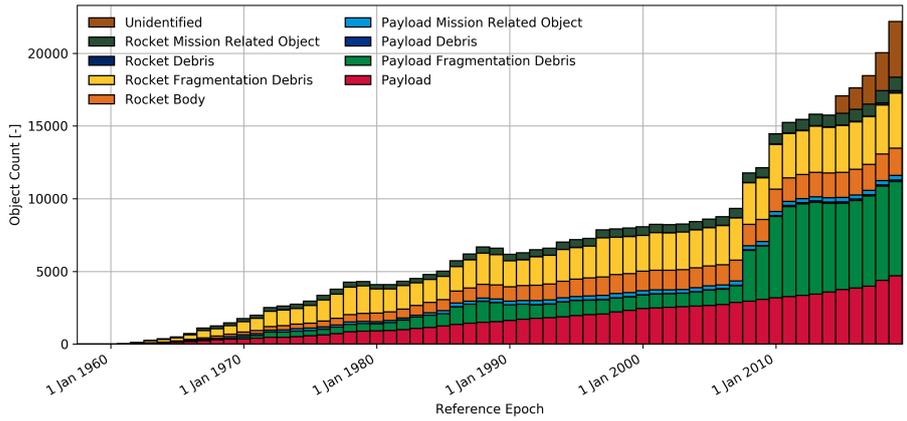
ROCKET MISSION RELATED OBJECTS space objects released as space debris during the normal functioning of a launcher stage, e. g. a shroud, an engine, etc.

ROCKET FRAGMENTATION DEBRIS space objects unintentionally released by one or more identifiable parent rocket bodies as a result of a unique event (e. g. a collision, a breakup, etc.).

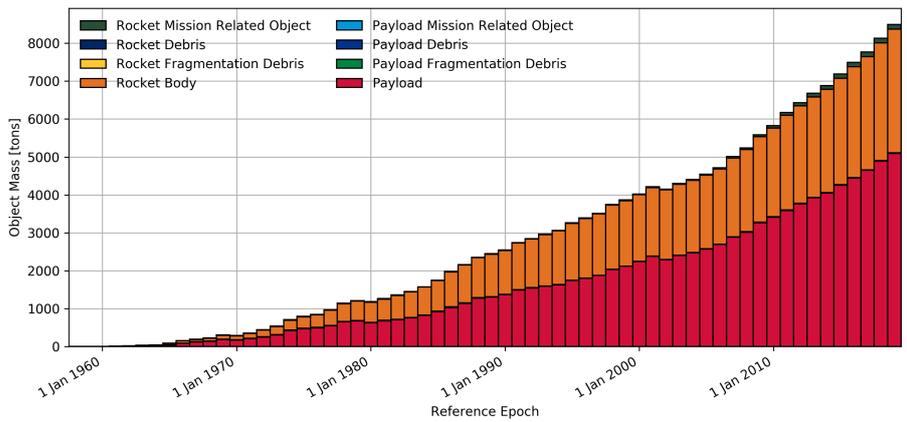
ROCKET DEBRIS space objects unintentionally released by one or more identifiable parent rocket bodies as a result of an unclear event.

UNIDENTIFIED space objects whose launch event and type cannot be determined at the moment.

From Fig. 2.1 it is evident that despite the natural cleansing effects, due to the atmospheric drag at low orbits, such as LEO, and luni-solar perturbations (in combinations with the atmospheric drag) at highly eccentric orbits, the number and mass of cataloged objects have seen an almost linear increase till the first decade of the 21st century [1]. Since 2007 two significant step increases in the number of objects have characterized the space debris environment [2]:



(a) Evolution of the number of objects per object type.



(b) Evolution of the mass of objects per object type.

Figure 2.1: Historic evolution of the number and mass of trackable objects in Earth orbit divided by object type (credit: [21]).

- on January 11, 2007, the Chinese weather satellite Fengyun-1C (FY-1C) was intentionally destroyed, at an altitude of 862 km, as part of an anti-satellite weapon test, generating 3431 catalog objects, of which 2809 were still in orbit 12 years later [23];
- on February 10, 2009, the first-ever unintentional in-orbit collision between two intact spacecraft, the defunct Russian satellite Cosmos-2251 and the operational US satellite Iridium 33, occurred at an altitude of 789 km generating 2294 cataloged fragments, of which 1396 are still in orbit 10 years later [23].

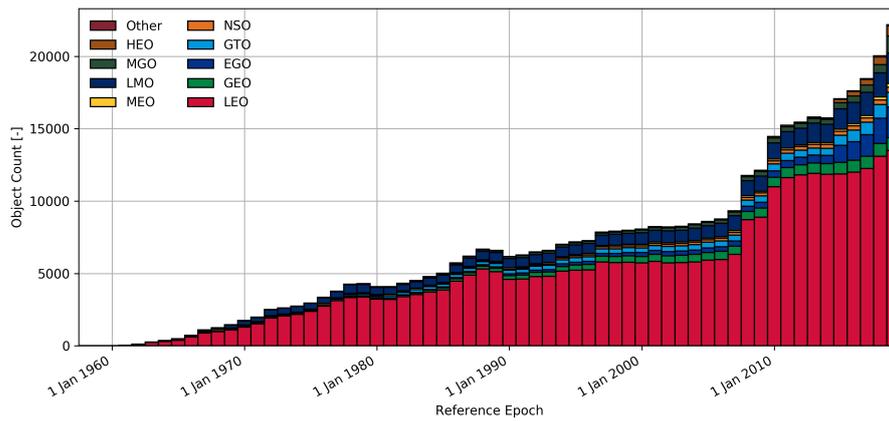
Both of these events have contributed, on their own, to more than half of the cataloged objects in the region below 1000 km and generated long-lasting fragments that increased significantly on-orbit spatial densities and collision risk for objects residing at altitudes between 750 km to 900 km [2].

The sudden increase in the number of *Unidentified* objects from 2015 onward, visible in Fig. 2.1a, is to be attributed to the improved sensor capabilities of space surveillance networks.

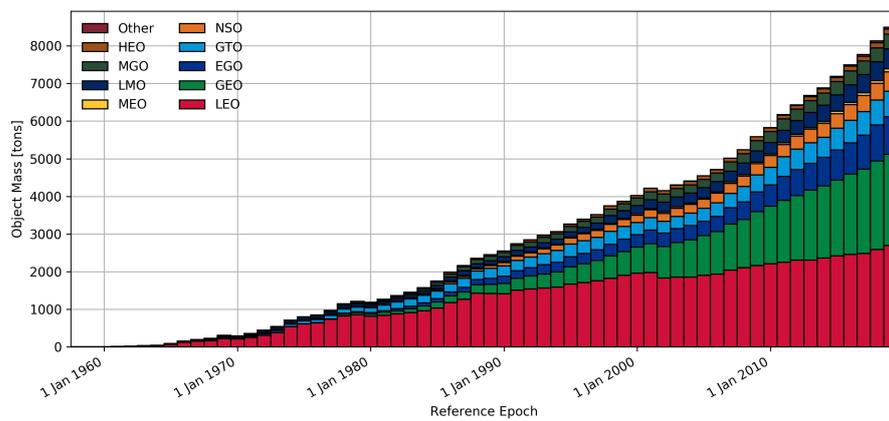
Fig. 2.1 also pinpoints the origins of the cataloged objects and possible future sources. Indeed, as of 2020, from Fig. 2.1a can be seen that the majority of cataloged objects can be classified as debris and are attributable to some 500 on-orbit fragmentation events. More in detail, of all cataloged objects, 45.9% are debris, 20.7% are payloads, 19.8% are unidentified objects, 8.2% are rocket bodies and 5.5% are mission related objects [21]. On contrary, Fig. 2.1b illustrates that the majority of on-orbit mass lies within the intact payloads (60.1%) and rocket bodies (38.3%), suggesting those two categories of objects as potential future sources of tens of thousands of fragments. In fact, based on the previous two recorded catastrophic collisions² and the NASA breakup model [25], a collision between only two 1000 kg rocket bodies in LEO would produce about 4000 trackable objects and over 100 000 lethal, non-trackable fragments capable of disabling a functional spacecraft [2].

² Collisions which energy-to-mass-ratios exceed values ranging from 35 J g^{-1} to 45 J g^{-1} [24].

The evolution of cataloged objects according to their orbital regimes is illustrated in Fig. 2.2. It shows, as in case of Fig. 2.1, a steady increase in the number of objects and mass across all orbital regions since 1957, altered by two sharp step increases in 2007 and 2009, due to the two previously mentioned catastrophic events. If this trend continues, more accidental collisions, similar to the one occurred in 2009, are to be expected in the future up until a critical point is reached that will trigger the Kessler syndrome even if all the future launches are suspended [4]. The result would be the creation of debris shells at certain orbital altitudes which would make them unsafe for a long time [1]. In fact, at typical LEO collision velocities of 10 km s^{-1} to 14 km s^{-1} , a millimeter-size object can penetrate a satellite and disable sensitive sub-systems. A centimeter-size object can generate mission-



(a) Evolution of the number of objects per orbit type.



(b) Evolution of the mass of objects per orbit type.

Figure 2.2: Historic evolution of the number and mass of trackable objects in Earth orbit divided by orbit type (credit: [21]).

critical damage and objects larger than 10 cm will most likely result in a catastrophic disintegration of a target [2].

The LEO region is particularly susceptible to this phenomenon since it contains $\sim 60\%$ of the total number of cataloged objects and $\sim 30\%$ of the in-orbit mass. Its certain altitudes have already reached the critical density of objects necessary to trigger the Kessler syndrome, but it is presently unclear in which manner and how fast will this long-term phenomenon manifest itself [2]. This situation is being even more exacerbated with the rise of mega-constellations, from commercial companies such as OneWeb, SpaceX and Amazon, that plan to deploy more than 53 000 new satellites in LEO.

The situation in the GEO region is less severe as it presents around two orders of magnitude smaller spatial density of objects, but its population is steadily increasing at a rate of 30 objects per year. Furthermore, while LEO benefits from the non-conservative aerodynamic perturbations, acting as sinks for cataloged objects, the GEO region is unaffected by them. Therefore, any object deployed or generated in that region will remain there indefinitely, which makes it highly sensitive to permanent overcrowding [1]. Additionally, GEO has peculiar orbital characteristics, guaranteed within a very limited band surrounding the Earth, and is of great commercial interest. Therefore, despite the current modest spatial density, the action against overpopulation of GEO should not be neglected, especially considering that it currently contains $\sim 27\%$ of the total in-orbit mass and the risk assessment is even more complicated at these altitudes due to the current inability to measure objects smaller than 0.3 m [3].

2.2 SPACE DEBRIS RISK MANAGEMENT

The risk management of space debris designates all activities aimed toward reducing the harmful impact of the ever growing population of space debris and can be divided essentially into two categories: the *space debris mitigation* and *space debris remediation*.

2.2.1 Space debris mitigation

The evolution of the space debris population depends simply on the balance between the creation and removal rates. With the current launch rates of 70 to 90 launches per year and an increasing number of spacecrafts being inserted into the orbit with each launch, the creation rate has outpaced the natural removal rate, leading toward a nonlinear increase in the number of objects in the next 200 years in LEO [2]. This future evolution is illustrated in Fig. 2.3 and represents the worst-case scenario, i. e. the “business-as-usual” scenario at the time of the study (i. e. year 2011), where the past seven to ten years of the traffic are

projected into the future using the NASA's LEO-to-GEO Environment Debris Model (NASA) (LEGEND) [2, 26]. The three curves represent averages of 100 Monte Carlo simulations and include the one sigma uncertainty error bars of the averages.

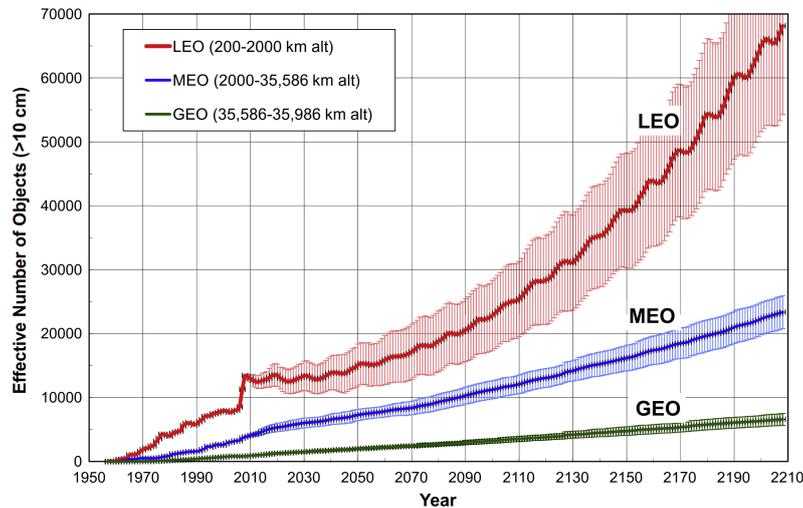


Figure 2.3: Projected evolution of the number of objects >10 cm with one sigma uncertainty error bars in LEO, MEO and GEO, in case of a "business-as-usual", i. e. non-mitigation, scenario (credit: [26]ⁱⁱ).

To mitigate this phenomenon various space agencies and other national and international organizations strived since mid-90s towards developing and implementing a set of guidelines aimed to moderate the proliferation of space debris in the orbital environment. These efforts culminated in 2002 with the release by the Inter-Agency Space Debris Coordination Committee (IADC) of the *IADC Space Debris Mitigation Guidelines*. These were later used as a model for national legislations and international standardization (see ISO24113:2019 [28]) as well as a baseline for the *UN Space Debris Mitigation Guidelines of the Committee on the Peaceful Uses of Outer Space* [29].

Within the mentioned guidelines a particular attention is brought towards the protection against the proliferation of space debris within the two unique orbital regions: the LEO and GEO protected regions. The former is defined as a spherical region spanning from 0 km to 2000 km in altitude. The latter, instead includes the portion of the volume of the GEO bound in altitude at $35\,786 \pm 200$ km and in inclination at ± 15 deg. The reason for defining those two regions lies in their uniqueness when it comes to Earth observation, remote sensing, telecommunication and human space flight, which is why any activity performed within should ensure their safe and sustainable usage [2].

ii Reprinted from *Advances in Space Research*, Vol. 47/Issue 11, J.-C. Liou, "An active debris removal parametric study for LEO environment remediation", pp. 1865-1876, DOI: 10/btsrbz, © 2011, with permission from Elsevier.

The fundamental principles contained within the commonly accepted space debris mitigation guidelines can be summarized with the following list of actions [2, 27, 30]:

- Eliminate or limit the release of debris during nominal operations in all orbital regimes by proper design of spacecraft and orbital stages.
- Minimize the potential for on-orbit breakups during nominal operations by proper design and operations of spacecraft or by passivating all on-board sources of stored energy (e. g. residual propellants, batteries, self-destructive devices, flywheels, etc.) after the completion of mission operations. Furthermore, all intentional destruction of objects capable of generating long-lived debris should be avoided.
- Perform post-mission disposal (PMD) of spacecraft and stages deployed within the protected regions. In case of LEO, this requires that an object within this orbit is able to leave within 25 years after its end of life (EOL) or if it operates above it, not be able to re-enter it within 100 years after its EOL. In case of GEO, the guideline dictates that an object within this orbit is transferred into a *graveyard orbit* after its EOL so that it does not re-enter the GEO protected region within 100 years.
- Prevent on-orbit collisions by limiting the collision probability of spacecraft and stages via proper orbit design and/or collision avoidance maneuvers (CAMs).
- Limit the on-ground casualty risk to humans to less than 1×10^{-4} in case of an atmospheric re-entry event by either ensuring the design for demise paradigm of spacecraft or enforcing a controlled reentry of large objects likely to have surviving fragments.

Despite the general agreement that having disposal guidelines in place is beneficial and that their implementation has been in the workings since mid-90s, the rate of compliance today is still far from 100%. In fact, in the last ten years, among the payloads reaching the EOL in a non-compliant orbit within the LEO protected region, between 15% and 30% of the spacecraft attempted to comply with the space debris mitigation measures³. However, only 5% to 20% of those spacecraft have done so successfully. In case of rocket bodies in the same region, between 40% and 80% have attempted to comply with the mitigation measures but only 30% to 70% have done so successfully and the trend is linearly rising. Better performance is achieved by payloads reaching EOL in the GEO protected region, i. e. between 85% and 100% have attempted to comply with the space mitigation measures while 60% to 90% have done so successfully and the compliance trend in this region is increasing asymptotically [31].

³ Excluding human spaceflight.

Even if the compliance rate in the LEO protected region was similar to that of the GEO region, studies performed in the last decade, such as the one done by the IADC in 2013 [32], have shown that the mitigation measures alone may not be sufficient to stabilize the future LEO environment. In fact, even assuming a 90 % compliance rate with the PMD measures and a 100 % compliance rate with the passivation measures (i. e. no-future explosions), but without considering collision avoidance maneuvers, the LEO population of debris is expected to grow by an average of 30 % in the next 200 years, while the catastrophic collisions are expected to occur every five to nine years [32]. Furthermore, even in the scenario of “no-future-launches”, Liou [26] demonstrated that the existing LEO population of debris is enough to fuel its growth over time. In reality, the situation is much worse since new spacecraft are expected to be inserted into LEO, especially with the upcoming deployment of mega-constellations, and suggests the need to couple the existing mitigation measures with remediation measures, such as the ADR, to permanently stabilize the LEO environment and bring the number of cataloged objects comparable to that in 2011. This concept is illustrated in Fig. 2.4 representing the simulated growth of the LEO population under three different scenarios assuming the traffic cycle from the previous years.

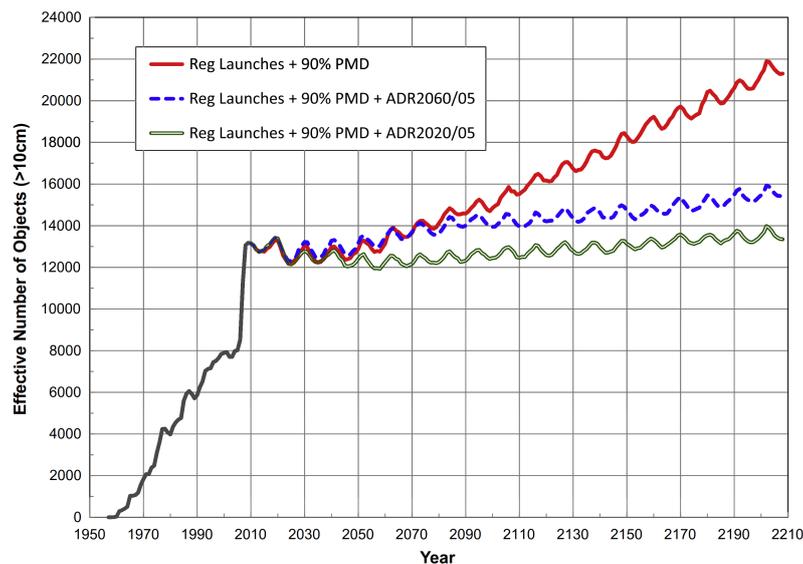


Figure 2.4: LEO environment projections considering different mitigation and remediation scenarios (Reg = regular, PMD = post-mission disposal, ADR_{20xx/05} = active debris removal starting from year 20xx with removal rate of five objects per year) (credit: [26]ⁱⁱⁱ).

The three scenarios consist of: a) regular launches coupled with a 90 % success rate in applying PMD measures (red line), b) regular

iii Reprinted from *Advances in Space Research*, Vol. 47/Issue 11, J.-C. Liou, "An active debris removal parametric study for LEO environment remediation", pp. 1865-1876, DOI: 10/btsrbz, © 2011, with permission from Elsevier.

launches coupled with a combination of 90 % success rate in applying PMD measures and ADR removal of five objects per year starting from year 2060 (blue dashed line), c) regular launches coupled with a combination of 90 % success rate in applying PMD measures and ADR removal of five objects per year starting from year 2020 (green hollow line).

The figure clearly shows that to maintain the LEO population comparable to that in 2011, the PMD measures in LEO need to be successfully enforced and the active removal of at least five objects per year starting as soon as possible should be implemented. An application of this latter requirement at a later date, e. g. in 2060, will result in additional 2000 objects and seven more catastrophic collisions in the next 200 years [26]. The acceptability of this outcome depends on many factors but it confirms the necessity to seriously consider an active removal of the existing on-orbit mass in order to preserve the orbital environment for future generations.

2.2.2 Space debris remediation

The debris remediation indicates activities to manage the risk posed by the space debris that go beyond the currently-adopted mitigation measures [2], described in Section 2.2.1. It does not represent a new concept but up until now it has never been widely accepted due to the enormous technical, legal and economical challenges that it entails. Furthermore, up until recently (see [26, 32]) there were no tools nor studies that could quantify the benefits of those activities on the growing population of debris [26].

The categories of space debris that should be considered for potential remediation activities can be divided into [2]:

LETHAL NON-TRACKABLE DEBRIS (LNT) objects being in the size range of 5 mm to 100 mm in LEO and 0.1 m to 1 m in GEO.

CATALOGED FRAGMENTS (CFs) objects being in the size range of 0.1 m to 1 m in LEO and 1 m to 2 m in GEO.

INTACT DERELICT OBJECTS (IDOs) objects having a size >1 m in LEO and >2 m in GEO.

Removing LNTs and CFs is considered, as of today, neither practical nor economically feasible due to their size, mass and ability to reliably track those objects from ground [2, 5]. In addition, as described in Section 2.1, the majority of the on-orbit mass (i. e. 98.4 %) is concentrated within the intact payloads and rocket bodies. For these reasons, remediation activities should at present strive to eliminate sources of LNTs and CFs which is best done by actively preventing future collisions between existing IDOs [2]. Possible means to manage the risk posed by the IDOs can be divided into two broad categories: those

that reduce the growth of the space debris population in the short-term (i. e. by preventing an imminent collision between two inactive spacecraft) and those that limit their growth over the longer term by acting years to decades in advance of the expected collision event. The former involves activities that move one of the two objects expected to collide in the imminent future, via the usage of a just-in-time collision avoidance (JCA) method, without the need for “complex” or costly maneuvers. The latter instead entails a “rescue” of a derelict object via ADR which involves its identification, rendezvous, capture, detumble and de- or re-orbit [2].

CAMs are a common practice among spacecraft able to perform orbital maneuvers to decrease the probability of collision with a space debris or a non-maneuvering spacecraft (e. g. a CubeSat). However, when both of the spacecraft involved in a possible imminent collision are not able to maneuver or are both debris, which is by far the most frequent scenario, there is currently no possibility to act on one or both of them to decrease the collision probability to an acceptable level [33]. JCA has been conceived to fill this gap and prevent imminent orbital collision when objects are unable to maneuver themselves. This might be accomplished via a small rocket that could be launched from an air-launch system to intercept one of the two objects expected to collide and “nudge” it slightly by introducing a gaseous cloud into its path [33]. In this regard JCA might be considered a more deterministic remediation measure, i. e. to be employed only when an impending collision is expected, while ADR can be considered a more statistical method since it requires years if not decades of adoption to prevent any future event [2]. Nevertheless, even JCA can only provide a necessary but not sufficient condition for the stabilization of the space debris environment, on pair with the debris mitigation measures. The only way of achieving the necessary and sufficient conditions consists of pairing the mitigation measures with the active removal of IDOs from their operational orbits via ADR. The selection of objects may be made based upon several criteria such as mass, cross-sectional area, probability of collision, orbital lifetime of generated debris, etc., or any of their combinations. In fact, if the objective of ADR activities is to address the long-term debris issue by eliminating the sources of fragmentation debris than the priority should be made to remove objects having the highest collision probabilities at certain time and greatest mass [26].

A generic ADR mission requires the following phases [34]: a) launch and orbit insertion, b) phasing and transfer to an orbit near a target, c) far-range rendezvous or homing, d) close-range rendezvous, e) capture of a target, f) removal (de- or re-orbit depending on the altitude of a target object) of a target. Among them, the last three emerge as especially challenging and safety-critical as they are to be performed in close-proximity to an uncooperative target. Indeed, while the first

three phases of an ADR mission are characterized by a significant flight heritage, dating back to the first ever in-orbit rendezvous performed by the crews of the spacecraft Gemini VI and VII on December 15, 1965, “no spacecraft has ever performed autonomous capture of an uncooperative vehicle, and a full six degrees of freedom (DOF) relative navigation sensing to uncooperative vehicle has only been shown to a limited extent” [8]. This can be traced to the fact that a typical target is not sufficiently equipped for a capture. Therefore, a typical target does not have reflectors, markers or radio beacons that could ease the determination of its relative position and attitude. Moreover, no grappling features are usually available, making its capture even more complicated. Finally, the target might have some sort of tumbling motion which poses strict requirements on the approach trajectory safety, due to the increased possibility of collision of the chaser with rotating appendages of the target.

The objectives of the close-range rendezvous are to reduce the range to the target and achieve capture conditions defined in terms of the relative pose⁴ and pose rates. At the end of this phase, the target object must remain within a pre-defined volume of space (defined by the capture mechanism) for a certain amount of time with respect to (w.r.t.) the chaser spacecraft which must achieve very narrow boundaries of relative position, velocity, attitude and angular rate despite external perturbations and/or uncertainties.

⁴ Intended as position and attitude of an object.

The capture phase starts when the guidance, navigation and control (GNC) system of the chaser has delivered the target within the range of the capture mechanism which is then responsible to [34]: a) achieve capture conditions, i. e. conditions of no-escape, b) stabilize the compound by attenuating residual relative velocities, c) achieve latching conditions, d) achieve rigid structural connection with the target. Depending on the type of “contact” needed to successfully accomplish the capture and stabilization conditions, the most promising capture methods currently being studied can be grouped into two categories: contact-based and contactless. The former require a physical connection with a target, while the latter do not and are able to achieve the capture by actively controlling the attitude of a target from a stand-off distance. Contact-based methods can be further grouped into:

ROBOTICS-BASED devices employing a stiff type of connection, such as a robotic system (e. g. a manipulator or a clamp/tentacle), to capture a target and stabilize the compound [6].

TETHER-BASED devices using a flexible type of connection, such as a tethered net or harpoon, to capture a target from a stand-off distance by either enveloping it or perforating its surface, respectively [35].

Contactless methods can be likewise further grouped into:

PLUME IMPINGEMENT-BASED devices employing an electric- or chemical thruster to create a plume of particles in front of a target to reduce its momentum and achieve a “capture” within a predefined volume of space, from a stand-off distance [36, 37].

ABLATION-BASED devices applying a concentrated source of electromagnetic radiation (e. g. visible light) onto a small portion of the target’s surface to ablate it, thus generating a small, but constant thrust opposite to the direction of the applied radiation [38, 39].

ELECTROMAGNETIC-BASED devices exploiting electromagnetic (e. g. eddy brake) or electrostatic forces (e. g. electrostatic tractor) to envelop a target in a magnetic or electric vector field, respectively, and generate necessary dissipative forces to “capture” it within a desired volume of space [40, 41].

After the capture phase, the target will need to be removed from its current orbit by either ensuring its de-orbit, a preferred option for objects within the LEO protected region, or re-orbit, a preferred option for objects within the GEO protected region. Both of those maneuvers can be achieved either directly by a chaser or a device attached to a target by a chaser. In case of a de-orbit maneuver, the disposal can be either controlled or uncontrolled. Considering that 10 % to 20 % of the mass of the object being removed is likely to reach the surface of the Earth, a controlled re-entry should be applied in case of a maximum casualty risk exceeding the threshold of 1×10^{-4} , which roughly translates to an object of a mass higher than 500 kg [42].

The most prominent de- and/or re-orbit methods can be grouped into two categories, active and passive, based on the type of force exerted onto the target object. Active methods can be further grouped into:

PROPULSION-BASED devices employing a chemical or electric thruster, in a form of a kit to be attached to a target prior to a de-orbit maneuver, that can execute controlled orbital maneuvers to either de- or re-orbit a target in a controlled manner.

PLUME IMPINGEMENT-BASED devices using, as in case of capture devices, an ion- or inert gas-based engine to create a plume of particles in front of a target. This way a constant force acting opposite to its direction of motion is generated, similar to aerodynamic forces, that will lead to a time-proportional change of its altitude and eventually to an uncontrolled re-entry.

ABLATION-BASED devices applying, as in case of capture devices, a concentrated source of electromagnetic radiation to ablate the surface of a target and create a constant thrust opposite to the direction of motion that will also eventually lead to an uncontrolled re-entry.

Passive removal methods consist of devices leveraging perturbation forces, present in the Earth orbital environment, to cause an uncontrolled re-entry or in some cases a re-orbit of a target without the need for consumables, such as propellant. The most promising ones can be grouped into:

TETHER-BASED devices employing tethers to actively reduce the semi-major axis of target's orbit by interacting with the Earth magnetic field in case of electro-dynamic tethers or by momentum transfer between the target and chaser via a tether, in case of momentum exchange tethers.

DEPLOYABLES devices leveraging the increase of the area-to-mass ratio (AMR) of a target to lower the semi-major axis of its orbit, in case of drag augmentation devices or increase its semi-major axis, in case of solar sails by exploiting the increased solar radiation pressure.

2.3 SPACE DEBRIS INFORMATION PARADOX

As shown previously, the remediation activities or more specifically ADR is an essential tool to stabilize the existing space debris population. Nevertheless, choosing one method over another is currently a difficult task not only due to the number of parameters characterizing targets and potential solutions but also due to the lack of coherence and structure of the available data. This has led up until now to confusion and general hesitation on how and when to best perform ADR. In this context, this section illustrates the problem of the information paradox within the domain of space debris with results from several state-of-the-art studies evaluating existing ADR methods and mission architectures. In addition, it analyzes state-of-the-art research in the field of knowledge representation of space debris meant to solve the existing information paradox.

2.3.1 Active debris removal methods assessment

Considering the variety of space debris and their orbital regimes, choosing an ADR method over another is a challenging task due to the dimensions of the parameter space characterizing targets, possible capture and removal devices, as well as related mission profiles (e. g. single- or multi-target missions, etc.). In fact, when examining possible ADR options several metrics should be considered, such as the technological availability (i. e. the technology readiness level (TRL)), safety, rendezvous and capture simplicity, cost benefits, reusability, etc [2]. In addition, the physical and dynamical characteristics of a target should also not be neglected, as they might reflect the measure of difficulty

that the capture and/or removal maneuvers are going to be subject to [43].

The state-of-the-art research on the topic can be broadly divided into qualitative and quantitative studies performed to assess: a) ADR capture technologies [43, 44], b) ADR de-orbit technologies [43, 44], c) ADR mission architectures [45–48], d) some combination of the previous three points (e. g. mission architectures coupled with de-orbit technologies) [6, 49]. In fact, assessing the entire ADR mission, or more specifically its last three phases, is an arduous task, especially considering the current lack of a common framework for data collection and storage of results that could ease the reuse of the already acquired knowledge.

Starting with qualitative studies, one way of tackling this issue could consist of a survey among ADR experts to rank, based on their professional experience, most prominent ADR capture and removal methods, using predefined classification parameters. This methodology was employed in late 2016, assuming a mission scenario to remove a large LEO space debris object⁵, e. g. a Soviet/Russian Cosmos satellite, or an SL-16 rocket body, using (some of) the most prominent state-of-the-art of ADR technologies listed in the previous subsection. The selection of experts that were asked to participate in the survey was carefully performed in advance to avoid professional biases as much as possible. Additionally, all participants were given the option not to evaluate technologies they were not familiar with in order not to skew the results.

⁵ An object having an average size >1 m and mass >1000 kg.

The classification parameters and their definitions, as delineated in the survey, are as follows:

TECHNOLOGY AVAILABILITY a parameter describing the current TRL of a technology.

SAFETY a parameter representing the assessment of the safety of a technology, w. r. t. the generation of further debris (in case of capture technologies) and ground casualties (in case of de-orbiting technologies).

RENDEZVOUS AND CAPTURE (RVC) SIMPLICITY a parameter defining the amount of energy (e. g. propellant) needed to complete a capture maneuver (defined only for capture technologies).

CONTROL CAPABILITY a parameter detailing the ability to control the DOF of an object;

COST BENEFITS a parameter expressing the assessment of the recurring costs of future missions using the same type of technology.

VERSATILITY a parameter specifying the capability of a technology to be used on different types of targets during the same mission.

REUSABILITY a parameter defining the capacity of a technology to be reused during the same mission.

MISSION CONCISENESS a parameter detailing the assessment of the length of a de-orbiting maneuver (defined only for de-orbiting technologies).

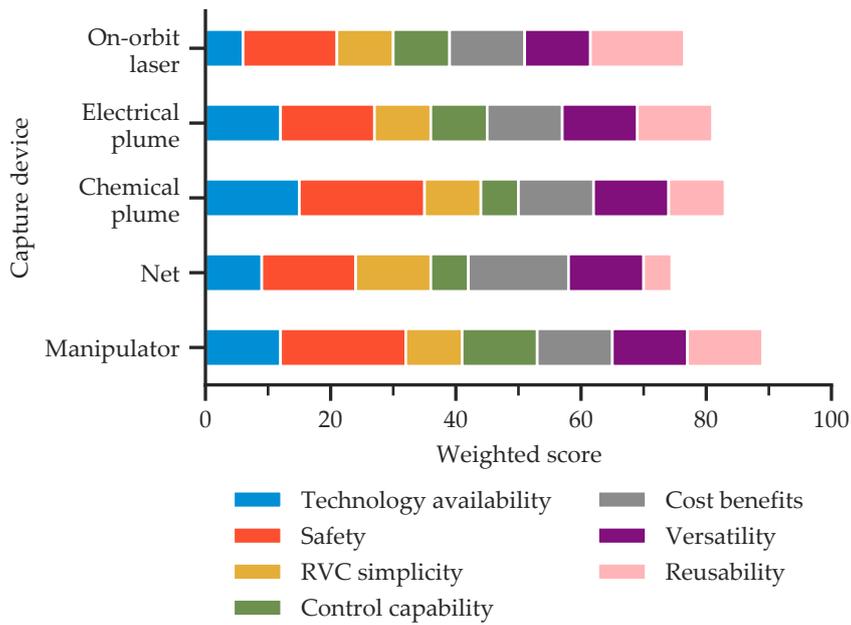
The results of the survey, completed successfully by 35 experts, are visualized in Fig. 2.5 for the five highest ranking capture and de-orbit devices. The professional background of the experts ranges from university professors and senior researchers in leading European and North American academic research institutions (e. g. Sapienza Università di Roma (Italy), University of Strathclyde (UK), McGill University (Canada), University of Colorado Boulder (USA)) to project managers in leading European aerospace agencies and companies (e. g. ESA, DLR, CNES, Airbus Defence and Space (Germany), Deimos Space S.L.U. (Spain)).

Analyzing Fig. 2.5a, it can be seen that manipulator-based devices exhibit the highest overall score among all the considered capture devices, although not by a large margin. This result is not surprising as among the considered technologies manipulator-based devices have currently one of the highest TRLs, second only to plume impingement-based devices using chemical thrusters. They are also characterized by a high safety (i. e. low risk of generating further debris) and high control capability parameters, limited only by the number of joints of a manipulator.

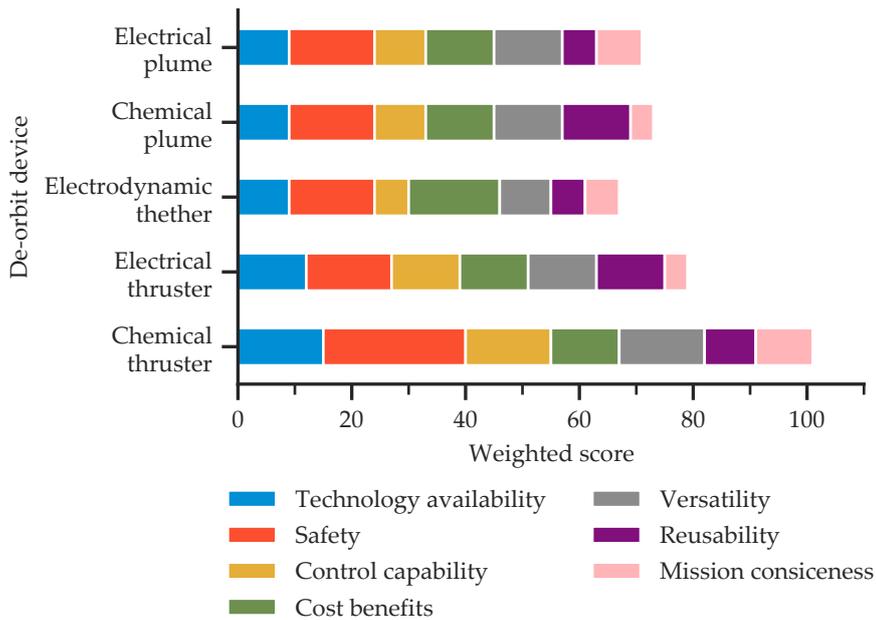
As to de-orbit devices, Fig. 2.5b shows that those having a significant flight heritage and capable of guaranteeing a controlled re-entry (i. e. chemical and electric thruster kit devices) emerge as clear winners of the survey. In fact, when compared to other devices, they have obtained the highest TRLs and safety scores.

Another qualitative study of common ADR capture and removal methods was performed by Shan, Guo, and Gill [43]. Within the research, the authors reviewed and compared both capture and removal methods by analyzing their major advantages and disadvantages collected from the state-of-the-art literature. Additionally, the authors sought to associate the considered ADR methods with space debris categorized in four groups based on their level of uncooperativeness. These levels were defined based on two parameters only: the “a priori” knowledge of physical properties of a target and existence of a docking interface. The influence of other parameters related to a target object (e. g. its attitude state) was not explored, thus leading in the majority of cases to a rough association between ADR methods and space debris categories.

A quantitative assessment of ADR methods for the removal of large LEO objects was instead performed by Hakima and Emami [44]. In this study an object was identified as large if having an radar cross-section (RCS) $>0.1 \text{ m}^2$ and mass $>1 \text{ t}$. The authors used a Monte Carlo



(a) Classification of five highest ranking capture devices.



(b) Classification of five highest ranking de-orbit devices.

Figure 2.5: Classifications of five highest ranking ADR devices based on answers from 35 experts. Color bars represent median weighted score values of classification parameters. The higher the bar the better is the overall weighted score (RVC = rendezvous and capture).

simulation in combination with an analytical hierarchy process and utility-based approaches to analyze and compare the performance of a net, an on-orbit laser, an electro-dynamic tether, an ion beam shepherd and a robotic arm. The assessment parameters used in the research were: a) removal time, b) performance robustness, c) controlled re-entry, d) propellant mass, e) total mission cost, f) average required power, g) TRL, h) mission risk. The outcome of the study revealed net methods as the overall best, followed closely by on-orbit lasers and robotic arm methods [44]. Nevertheless, the authors highlight that while robotic methods have been already extensively employed on-orbit (albeit not yet in the context of an ADR mission), nets have been tested on-orbit only minimally (see [35]) and laser-based methods lack the ability to guarantee a controlled re-entry. Furthermore, nets present the risk of generating further debris, in case of targets with large appendages, and the possibility of missing a target entirely. Robotic arms on the other hand present the risk of generating debris due to contact and are unsuitable for targets having high attitude rates. These results support in a more systematic way the notion that there is not a single method that clearly stands out from the rest as an optimal, yet generic solution.

With respect to ADR mission architectures, a number of recent quantitative studies [45, 47, 48] have examined the suitability of different mission concepts (e. g. single-target, multi-target or multi-target with de-orbit kits) to remove a number of large LEO objects by analyzing the overall mission duration, different chaser designs, launch vehicles and overall mission cost. The capture phase was not included in these studies as its impact on the overall mission trade-offs was deemed very low. The analysis of their results showed that in all of them the usage of a large, shared launcher (such as Ariane 5) is to be preferred in terms of cost for missions requiring a controlled re-entry. When it comes to the efficiency of single-target missions over multi-target ones, the results are mixed. In fact, while [45, 48] suggest that a multi-target mission with de-orbit kits would result in a more efficient mission cost per kilogram of removed debris, [47] concludes that the most efficient concept should instead be a single-target mission. The reason for these contradictory results is to be attributed to different assumptions made regarding chaser designs and employed cost models.

From the perspective of robustness to failure and consequences for the space debris environment, [50] compared two ADR mission concepts, i. e. a single- and multi-target missions, thus analyzing the potential for ADR missions to increase the space debris hazard due to collisions involving failed chaser spacecraft. The conclusions of the study were that multi-target missions offer substantial benefits in terms of robustness to failure when compared to single-target missions. In fact, in case of single-target missions, up to 35 % of all ADR missions were needed to remove failed chaser vehicles and 27 % of all collisions

did involve failed ones. However, multi-target missions are inherently more complex and could result in greater failure risk since one chaser will be involved in multiple proximity operations. Therefore, even in case of mission architectures once again it can be concluded that there is no unique answer when it comes to an optimal, yet generic solution, although some high-level conclusions might be drawn.

When it comes to the assessment of the ADR capture methods, it is worth mentioning the influential studies performed by ESA [6, 49] and related system studies performed by the industry [51]. Their results contain a wealth of information regarding possible ADR capture methods and approaches, with mixed results when it comes to the best one to actively remove a large LEO object, such as ENVISAT. Among the considered capture methods, the final preference went to the net and robotic arm, considering their existing heritage and versatility. Yet, the results also point out that any final decision on the most suited capture method may not be made on the system engineering results alone. Instead, it must also consider potential preferences of funding bodies [51] which adds a yet another criterion to the already complex parameter space of ADR capture methods.

In conclusion, the current literature on the topic suggests that the quest for an optimal ADR solution does not have a unique answer since it strongly depends on the characteristics of a target, its orbit, the required de-orbiting strategy and a wealth of other parameters. In addition, it also points toward a more fundamental problem currently afflicting the ADR domain being the absence of a common method to efficiently capture and store information about ADR technologies and space debris that can be used to “more readily” infer new knowledge when needed. Therefore, it might be argued that the current space debris domain is plagued by the *information paradox* [52] where we are “... drowning in information but are starved for knowledge” [53]. Just as the society at large, the space debris domain is going through an information age. Vast quantities of data are systematically collected, stored and disseminated. Yet, our understanding of the existing data and the ability to generate new knowledge out of it is limited. A variety of causes are culpable for this paradox, some of which are difficult to identify and more so to remedy. One of them is the ability to adequately store the accumulated information about a domain. Failing to do so hinders not only the reuse of the existing knowledge but also the generation of new one, considering the lack of the explicit information structure and domain assumptions.

2.3.2 Knowledge representation of space debris

To counter the information paradox of the space debris domain, methods that have been shaping the World Wide Web, medicine and pharmacy domains for some time now, are finding their way into the space

debris domain as well. These methods are capable of not only systematically storing and categorizing data, but also allowing creation of new knowledge, which would otherwise be lost or at least obscured. The methods in question and their definitions, as intended in this research, are:

TAXONOMY “a hierarchically organized set of terms (describing concepts or things) with defined broader and narrower relationship types” [54].

ONTOLOGY a formal and explicit description of domain concepts, their properties and relationships [9], in a way that the data and its meaning are encoded alongside each other [55].

KNOWLEDGE GRAPH a method to describe “real world entities and their interrelations, organized in a graph” [56] and “integrated into an ontology” [55, 57].

Prior efforts to use those methods in the space debris domain have focused almost exclusively onto the domain of SSA. The ADR domain was neglected and is therefore missing from the existing knowledge representation techniques. Furthermore, most implementations appear to overlook the handling of input data from potential databases, although this issue often represents the crux of a method, given the sheer size of existing space debris catalogs such as the ESA’s Database and Information System Characterising Objects in Space (DISCOS) [23] or US Space Force’s Space-Track database [58].

⁶ Defined as either a natural or man-made object orbiting another body, e. g. Earth, Sun, etc.

Wilkins et al. [59] define a basis for a resident space object (RSO)⁶ taxonomy. The goal of the method is to provide the basis for a holistic framework needed to enable an automated agent to classify and identify RSOs without ambiguity based on observation data, while providing the probability of the correct association. The research does not include ADR domain knowledge, thus precluding any possibility of further related knowledge inference. The issue of the data import was not explicitly tackled, thus suggesting at a manual entry of objects’ characteristics.

Frueh et al. [60] detailed a phylogenetic taxonomy of LEO objects with the goal of identifying their main classes and sources of origin based on specific physical and dynamic traits of objects. The outcome of the classification is an acronym providing a concise way of visualizing main traits of an object (and thus its classification). In addition, the research provides a method of disclosing how dangerous an object is for the surrounding population via a hazard scale of objects based on their size, velocity and AMR. The framework was not explicitly developed to aid future ADR mission planning, therefore some discerning traits are missing, while others were not defined rigorously enough (e. g. the material parameter), thus leaving space for individual interpretation.

Cox et al. [61] present a “Space Object Ontology” designed to support the space domain awareness by enabling improved characterization of objects and related events. This facilitates more robust entity tracking, classification and analysis of space objects. To enable a cross-domain integration, the ontology was built as an extension of the Basic Formal Ontology and the suite of Common Core Ontologies, providing the upper-level and mid-level structures, respectively. The novelty of the study lies in its ability to represent spacecraft capabilities and vulnerabilities which enables a semi-automated “threat assessment” of an object (e. g. based on the band type used by its communication subsystem). The authors mention integration of data from multiple data sources but do not provide additional implementation details.

Similarly, Rovetto [62] illustrates an ontological architecture of the orbital debris domain. Its goal is to formally model the domain of space debris by creating a machine-readable lexicon capable not only to improve the ability to capture general scientific knowledge of the domain in a more systematic manner but also to advance data-exchange among space object catalogs and information systems. Limited details regarding the implementation method are provided within the research and the methodology used for the input of the necessary data into the ontology is not mentioned, thus hinting at a manual process.

Furfaro et al. [63] describe an approach to characterize the behavior of RSOs (starting from sensor measurements), classify them and execute probabilistic reasoning. In this context, ontologies are used to construct, in a semi-automatic manner, Bayesian networks capable of providing a probabilistic assessment of an event of relevance for SSA, such as a satellite collision threat evaluation. The developed ontology does not include any knowledge about the ADR domain and methodology used for the input of the necessary data is not mentioned, thus hinting at a manual process.

Alike, Liu, Yao, and Han [64] outline an ontology for RSOs developed using expert domain knowledge and unordered machine learning rules. The aim of the work was to provide a classification of RSOs even in case of missing data properties. The developed ontology is compared to classical machine learning classifiers (e. g. C4.5, Bayesian network, Ripper) and its advantage, in accuracy and precision is demonstrated albeit at a cost of higher computation time. The ADR domain was once again not within the scope of the developed ontology and the methodology concerning the input of the data into the ontology was not mentioned.

Le May et al. [65], on the other hand, illustrate a knowledge graph-based method to represent RSOs and support early SSA operations and observation planning, characterized by a semi-automated data input. The graph database in question has been specifically developed with two data sources in mind: structured and unstructured. The structured data sources considered are the satellite databases of the Union of

Concerned Scientists [66] and Space-Track [58]. The unstructured input data is instead obtained via an information extraction technique (e. g. web-scraping), thus minimizing user input. However, even this method does not model the ADR domain, thus precluding any possibility to represent it or infer knowledge from it.

2.4 SPACE ROBOTICS

Space robotics represents one of the key technologies that enabled our species to safely explore and operate in harsh and extreme conditions of the space environment. Obviously, present-day robots appear to be clumsy in comparison to human beings. But if the analogy is made between the skills of an astronaut during an extra-vehicular activity (EVA) and those of the best available space robot, then the difference is almost negligible [67]. Additionally, robotic systems can perform certain in-orbit tasks with less risk, work restlessly, and without the need for costly transport and life support needed by astronauts. Nevertheless, at the current state of research the human intuition and intelligence in certain, unpredictable situations make astronauts irreplaceable [68].

This section describes the state-of-the-art in space robotics and in particular in the field of robotic capture of uncooperative objects.

2.4.1 Past and future missions

The concept of on-orbit servicing and assembly by a robotic *free-flyer* dates back to early 1980s when NASA published a series of “Space Applications of Automation, Robotics and Machine Intelligence System” reports (e. g. [69]). This idea was inspired by the successful use of a manipulator in an orbital environment in 1981 during the second Space Transportation System (STS) mission⁷. Later on, several manned missions followed in order to capture, repair and re-deploy malfunctioning satellites (such as Anik-B, Intel-sat 6 and Hubble telescope). Despite these successes, such unmanned and completely autonomous missions have yet to become operational, despite several demonstration missions, such as the Engineering Test Satellite VII “KIKU-7” (JAXA) (ETS-VII) and Orbital Express [70].

⁷ During which the Space Shuttle remote manipulator system was used for the first time [70].

2.4.1.1 Space Shuttle Remote Manipulator Systems

The *Shuttle Remote Manipulator System*, also known as *Canadarm*, illustrated in Fig. 2.6, was the first robotic manipulator arm used in Earth’s orbital environment. It made its space debut on the Space Shuttle Columbia (STS-2) on November 13, 1981 and was successfully used for 30 years, retiring along with the Space Shuttle program, on July 21,

2011. It was a mechanical arm, 15 m long with 6 DOF. Its purpose was mainly to maneuver payloads from the payload bay of a Space Shuttle orbiter to its final position and then release them. Canadarm was also used to catch an in-orbit spacecraft and berth it to the payload bay of the orbiter as well as to assist human EVAs by means of a foothold that could be attached at the end of the manipulator [70].



Figure 2.6: Space Shuttle Manipulator System or Canadarm (right) during the Space Shuttle mission STS-72 (credit: NASA).

The *Robot Technology Experiment (ROTEX)* was a first remotely controlled (i. e. teleoperated) multi-sensory robotic arm able to cope with a signal delay of up to 6 s using GEO relay satellites. It represents not only the “starting point” for Germany’s space automation and robotics but also one of the important milestones of space robotics in general. The ROTEX manipulator was flown on the Space Shuttle Columbia (STS-55) in 1993 with the Spacelab D2 mission and consisted of a small, multi-sensory manipulator with 6 DOF, mounted inside a space-lab rack. The most challenging experiment conducted with the manipulator was the autonomous capture of a free-floating cube with flattened edges. This was achieved by processing on-ground the stereo images captured by the eye-in-hand manipulator configuration, estimating the pose of the cube in the next 6 s and generating appropriate commands for the cube’s capture [70].

2.4.1.2 Manipulator Systems of the International Space Station

The *Space Station Remote Manipulator System*, also known as *Canadarm2*, launched in April 2001 during the STS-100 mission, is the next generation of the Space Shuttle’s original manipulator, used on the ISS. It

is part of the mobile servicing system of the ISS and is a 17.6 m long manipulator equipped with seven joints and latching end-effectors at both ends that enable it to self-reallocate using an inch-worm like movement. Since its installation, it has played a major role, alongside the Canadarm, in all the phases of construction and maintenance of the ISS both by assisting EVAs of astronauts and taking over the payload from a Space Shuttle orbiter [70, 71].

The *Special Purpose Dexterous Manipulator*, also known as *Dextre*, is another manipulator of the ISS (see Fig. 2.7) and part of the mobile servicing system. It was launched in March 2008 and first used on February 4, 2011 to unpack equipment delivered by the Kounotori 2 spacecraft [72]. It resembles a headless torso equipped with two extremely agile, 3.5 m arms, each of which has 7 DOF and special end-effectors capable of performing delicate maintenance work and repairs during EVAs [73].



Figure 2.7: Special Purpose Dexterous Manipulator or Dextre, photographed by an Expedition 27 crew member on the ISS (credit: NASA).

The *Japanese Experiment Module Remote Manipulator System* is yet another manipulator installed on the ISS. The manipulator system is composed of two arms having both 6 DOF: the main 9.9 m arm, fixed to the Japanese module, and the small fine 1.9 m arm. It is a robotic manipulator system intended for supporting experiments conducted on the exposed facility of the Japanese Experiment Module as well as its maintenance. The main arm was launched on May 2008, alongside the Kibo Pressurized Module, while the small arm was launched a year after on September 2009 aboard the unmanned H-II Transfer Vehicle [70].

2.4.1.3 Engineering Test Satellite VII "KIKU-7"

The *Engineering Test Satellite VII "KIKU-7"* (JAXA) (ETS-VII) was the first mission to include a satellite equipped with a 6 DOF robotic ma-

nipulator. It represents a milestone of the space robotics technology in the area of unmanned satellite servicing. Launched in November 1997 and used till December 1999, the spacecraft consisted of two main parts that could be separated for mission purposes. The main spacecraft, named Hikoboshi, was equipped with a 2 m long, 6 DOF manipulator arm, while the smaller spacecraft, named Orihime acted as a target spacecraft [70, 74]. The main mission goals were to test free-flying robotics technology and demonstrate the feasibility of unmanned orbital operations, which were achieved despite few mission hiccups [75]. To reach those objectives the mission needed to successfully complete autonomous rendezvous and docking operations as well as a series of robot experiments. The latter included: a) teleoperation with large time delay (5 s to 7 s), b) exchange of orbital replacement units and deployment of a structure, c) coordinated control of the base spacecraft and manipulator, d) capture and berthing of cooperative target [70].

2.4.1.4 *Orbital Express*

The *Orbital Express* mission represents an important milestone in unmanned space robotics. The objective of the mission was to validate the readiness, cost-effectiveness and utility of autonomous on-orbit servicing (OOS) technologies. The system was launched in March 2007 and was used till July of the same year. The space infrastructure of the mission consisted of two spacecrafts: a servicer and a satellite being serviced, a prototype of a modular serviceable satellite. The servicing spacecraft was equipped with a 3 m long robotic manipulator arm with 6 DOF which purpose was to capture, video survey, correctly position and service the target satellite [74]. During the mission, objectives similar to those of the ETS-VII were successfully accomplished, but with a higher degree of autonomy and without any help from the ground control. The mission was completed successfully despite an anomaly in the flight software that almost ended it [74].

It is important to notice that although both ETS-VII and *Orbital Express* missions performed successful autonomous capture/berthing and docking of the target spacecraft, the latter was always stabilized by an on-board ACS and it also presented dedicated grappling fixtures and optical markers [74]. In reality this will not always be the case. Therefore, while past missions represent an important achievement of space robotics, they present a gap that still needs to be filled by future missions if an autonomous capture of a truly uncooperative target is to become a reality. Two example missions of such kind are detailed in what follows.

2.4.1.5 *e.Deorbit*

The *e.Deorbit* mission illustrated in Fig. 2.8, was planned to be the first ADR mission to remove an ESA-owned derelict satellite, from

⁸ Defined during the last Phase B mission study [77].

LEO, as part of the ESA's Clean Space initiative [76]. The reason for removing ENVISAT was based on its high collision risk, tumbling attitude, large inertia and complicated capture access, which made it a representative, albeit difficult target. The chaser spacecraft⁸, was based on the Airbus' SpaceTug platform equipped with a robotics-based capture system consisting of a robotic manipulator, a launch adapter ring (LAR) gripper, a LAR clamping mechanism and a vision system. The manipulator was assumed to have 7 DOF, a length of 4.39 m, a mass of 52.43 kg and be able to provide a peak torque of 314 N·m. The capture of a tumbling spacecraft was identified to take place after a contactless de-tumbling maneuver, assumed to be achieved via a plume impingement-based method using chemical thrusters. Any residual relative attitude motion of the target was assumed to be compensated by a synchronizing maneuver [77]. The mission was scheduled to be launched in early 2023 and last for 90 days, after which both the e.Deorbit spacecraft and the captured ENVISAT would have reentered into the atmosphere in a controlled manner. Due to the lack of budget, the mission and its related studies, that began in 2013, were concluded in November 2018 in favor of a smaller removal mission of an ESA-owned object able to demonstrate in-orbit servicing capabilities as well, the ClearSpace-1. The legacy of e.Deorbit lies in many innovative solutions identified during the years of study consisting of: a) in-orbit characterization of and navigation to a tumbling target, b) capture and stabilization of a tumbling target using different capture methods (e. g. a net, a robotic arm, a harpoon, etc.), c) advanced GNC algorithms for safe automatic operations. These solutions will hopefully be reused in upcoming ESA missions, such as the ClearSpace-1 commissioned to remove a 100 kg secondary payload adapter of a Vega upper stage by 2025 [78].

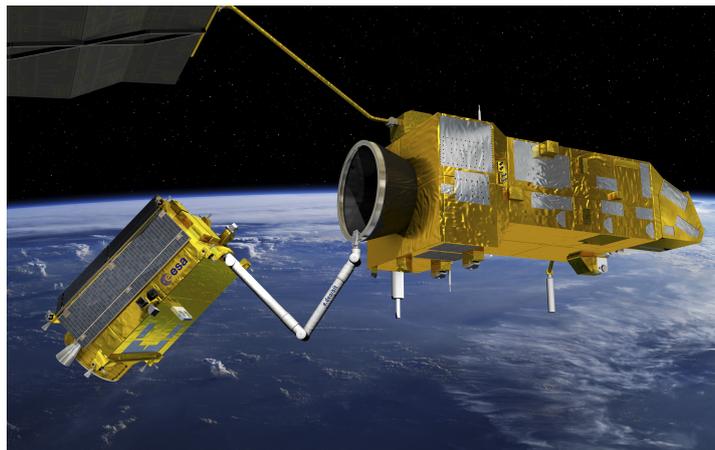


Figure 2.8: Artist's concept of the e.Deorbit chaser (left) capturing ENVISAT (right) (credit: ESA–David Ducros, 2016).

2.4.1.6 OSAM-1

The On-orbit Servicing, Assembly, and Manufacturing 1 (OSAM-1) mission, visible in Fig. 2.9, is a proof of concept OOS mission from NASA aimed to be launched in 2023 to perform autonomous rendezvous, robotic capture, refuel and reallocation of a cooperative LEO satellite owned by the US government, Landsat 7. The reason for choosing Landsat 7 was made due to its fairly conventional characteristics, such as the presence of an LAR, and the fact that it was not designed for servicing. The spacecraft will be equipped with two 7 DOF arms, that will be able to capture and service Landsat 7 from about 1 m away. The spacecraft will also include a Space Infrastructure Dexterous Robot payload that will demonstrate an on-orbit assembly of a modular 3 m antenna and construction of large on-orbit structures. If successful, OSAM-1 will extend the service life of Landsat 7 potentially by several years and will demonstrate the maturity of the OOS technology to be incorporated in other NASA missions. In addition, the mission will also help mature technologies for on-orbit assembly of modular, large structures that could enable new type of missions and replace the need for astronaut EVAs with agile on-orbit robots [79, 80].



Figure 2.9: Artist's concept of the OSAM-1 chaser (left) capturing Landsat 7 (right) (credit: NASA).

2.4.2 Robotic capture of uncooperative objects

Despite the previously mentioned accomplishments and a great deal of research performed on the topic in the span of 30 years [81], the autonomous capture of an uncooperative object remains an open challenge [8], especially in case of objects characterized by a high tumbling rate and/or inertia. The autonomy is requested in particular in the

final phases of the approach of the chaser vehicle to the target object due to the limited reaction time available to face anomalies or communication problems [75]. The target might also be characterized by a tumbling motion which poses strict requirements onto the motion safety and capture maneuver duration, due to an increased possibility of collision of the chaser with the rotating appendages of the target, if present. In addition, space robotic systems exhibit a dynamical coupling between the manipulator and the base spacecraft during the activity of the former, which complicates the capture maneuver even further due to the resulting attitude disturbance of the base. This phenomenon is especially prominent on *free-floating* robotic systems in which both the position and orientation (i. e. the pose) of the base spacecraft are not actively controlled during the activity of the manipulator, thus causing a path dependent reaction of the base spacecraft around the center of mass of the overall chaser system. In contrast, on *free-flying* robotic systems the attitude (and in some cases even the position) of the spacecraft is actively controlled during the motion of the manipulator, thus providing a highly redundant system with nearly unlimited workspace, at a cost of the propellant usage [82, 83].

Studying the entire capture maneuver of an uncooperative, tumbling target is a difficult task due to the nature of problems that might occur during different phases of the maneuver. Therefore, to date the research in the field has been divided into different phases: a) spacecraft approach, b) manipulator approach, c) impact/capture of the target, d) post-impact motion/stabilization of the compound (chaser plus target) [84]. To successfully execute any of the mentioned phases the robotics module of a chaser spacecraft will need to provide a feasible guidance and control of both the spacecraft and its manipulator. Past studies have shown that nearly any control scheme developed for terrestrial robots can also be implemented on free-floating/free-flying robotic systems (assuming that correct dynamic models are used and dynamic singularities⁹ are avoided) [82, 83, 86]. However, computing an optimal, feasible and safe motion of the manipulator, in a timely manner, remains still a difficult and challenging task [87]. The literature on the topic can be mainly classified in two broad categories, based on the assumed control authority of the base spacecraft during the activity of the manipulator, i. e. the free-floating or free-flying.

⁹ Path dependent positions in the Cartesian space at which the end-effector of a free-floating manipulator system cannot be moved in some inertial direction [85].

2.4.2.1 Free-floating robots

In the free-floating category, the nonholonomic behavior of the system complicates the motion planning of the manipulator. At the same time, the free-floating nature of the system offers an opportunity to develop interesting planning solutions for correcting or minimizing attitude disturbances of the base using only the joint motions of a manipulator.

Vafa and Dubowsky [88] proposed a self-correcting motion planning technique that uses cyclic motions of manipulator joints to correct the

attitude of the base spacecraft¹⁰ while executing a planned Cartesian trajectory. The limiting factor of the method are the many joint cycles required to achieve even a small correction of the attitude of the base spacecraft [89]. Moreover, it was found that in some system configurations the joint motions are unable to affect the attitude of the spacecraft at all.

¹⁰ Should it deviate from the desired one.

Nakamura and Mukherjee [90] proposed a method, the *bi-directional approach*, based on a Lyapunov function, for path generation of a free-floating manipulator to control both the manipulator configuration and spacecraft orientation. Nevertheless, the technique proved to be affected by dynamic singularities [81, 89] which locations and existence cannot be determined using only the kinematics of the manipulator, differently from the ground-based robots [85].

Papadopoulos [91] tackled the issue of dynamic singularities of a space robot by defining a subset of the reachable workspace of the manipulator, a *path independent workspace* within which no dynamic singularities can occur during the manipulator motion. This concept was applied to a planning method that avoids dynamically singular configurations by using small cyclical motions of the end-effector.

Nanchev, Yoshida, and Umetani [92, 93] introduced the concept of the reactionless motion planing based on the theory of the *reaction null space* that can minimize the dynamical coupling problem of a free-floating robotic system while being able to follow a predefined path. The concept was tested on-orbit during the ETS-VII mission (see Section 2.4.1.3) [94] with the *zero reaction maneuver*. The maneuver requires a kinematically redundant arm and presents a limited range of motion when compared to those not leveraging the reaction null space, especially in case of systems exhibiting small base to manipulator mass ratios [95].

More recently, Xu et al. [89, 96] addressed the issue of path planning of a free-floating robotic system to capture a static target as an optimization problem using evolutionary algorithms. As a result, the planned trajectories are smooth, feasible and the disturbance of the base is constrained.

Similarly, Lampariello and Hirzinger [97] formulated the problem of capturing an uncooperative tumbling target in a flat spin motion of $\pm 4 \text{ deg s}^{-1}$, as an optimization problem and solved it via a direct shooting method. The solution of the optimization is then stored in a look-up table along with the four input parameters describing the target attitude so that it can be used to initialize on-line motion planner onboard the spacecraft. The angular momentum management of the stack is relegated entirely to the ACS of the base spacecraft during the post-capture phase.

Flores-Abad et al. [98] developed an optimal capture strategy of a tumbling target by first deriving the optimal time and capture configuration of the target so that the contact force is directed through

the center of mass of the entire system (chaser plus target). Then, the optimal trajectory of the manipulator to reach those conditions, while minimizing the control effort, is found by solving the associated OCP. The method formulated in this way is found hard to be accomplished, due to the generally unpredictable contact dynamics. The management of the angular momentum of the target has not been addressed and the collision avoidance has not been mentioned. Furthermore, the method was applied only in a planar (i. e. 2D) case.

Mentioned studies of free-floating robots all propose solutions to the path planning problem of a system which is allowed to translate and rotate freely in response to manipulator motions. In absence of external forces and torques they could be used to capture a tumbling target without the need for the propellant usage. However, due to the nonholonomic nature of the overall system, a robotic chaser will need to execute the capture maneuver from a safe, close-enough holding position which might not exist in case of tumbling targets having large appendages. In addition, capturing a tumbling target without considering its angular momentum, especially if it is large, either due to the target's mass or angular rate, will result in a difficult stabilization phase and will most probably result in a failed maneuver [84]. For this reason, free-flying robots might be a better choice when the aim is to capture a tumbling target.

2.4.2.2 *Free-flying robots*

In the free-flying category, the control of the attitude and/or position of the robotic spacecraft are taken into consideration during the motion planing of the manipulator thus, enabling a virtually unlimited workspace at the expense of propellant usage.

Dubowsky, Vance, and Torres [99] presented an optimal path planning technique of a free-flying space robot that accomplishes the manipulation task as quickly as possible without saturating the attitude control thrusters. The overall consumption of the propellant was not among the objectives of the optimization, highlighting a possible limitation of the method as it might limit the useful on-orbit life of the system [83].

Torres and Dubowsky [100] addressed this issue via a heuristic path planning method, the *enhanced disturbance map*, that can provide paths of a manipulator able to reduce or even eliminate the base disturbance. The method was an improvement of the previously developed *disturbance map* [101], since it allowed a representation of a disturbance map of a 3 DOF manipulator. Although the method is valid for a generic space manipulator (i. e. with more than 6 DOF), its application in this case was found to be challenging [89, 102]. Furthermore, the capture of a tumbling target and management of its angular momentum by the chaser was out of the scope of the method.

Dimitrov [102] studied the optimal pre-capture distribution of the angular momentum within a free-flying robotic system, as a way to facilitate the management of the angular momentum of the target in the post-capture phase. To this aim the author developed a trajectory generation method of a manipulator, the *bias momentum approach*, that pre-loads the angular momentum of the target, equal in magnitude but opposite in direction, in the chaser manipulator. The base disturbance is kept to a minimum via the usage of reaction wheels which are pre-loaded with the angular momentum of the target. This way upon capture no transfer of angular momentum between the target and chaser is expected and only locking the joints of the manipulator could complete the maneuver. However, the method appears not suitable for targets characterized by large angular momenta, as it would require either massive manipulators or joints capable of achieving high angular velocities. In addition, it would require a very stiff manipulator capable of transferring at the time of contact a large amount of kinetic energy.

Using optimal control Aghili [103] addressed the pre- and post-capture phases of a space robot capturing a tumbling target. In the pre-grasping phase, an optimal trajectory is planned to intercept a grasping point on the target with zero relative velocity, subject to acceleration limit and adequate target alignment. In the post-grasping phase, the manipulator is used to damp out the angular and linear momenta of a target as quickly as possible, subject to the constraints of the manipulator. The management of the angular momentum of the target by the free-flying robotic system was not discussed in the study, thus pointing towards a possible limitation of the method to deal with targets with high angular momenta.

More recently, Jaekel et al. [104] presented the results of a robotic concept to capture the European uncooperative satellite ENVISAT, as part of the planned ADR mission e.Deorbit (see Section 2.4.1.5). Considering the large momentum of the target, the concept assumed that the capture is preceded by a synchronization maneuver with the robotic arm in a stowed configuration. At the delivery point, the manipulator is unfolded and a pre-planned approach trajectory is executed via the coupled control between the manipulator and base spacecraft while assuming the spacecraft to be fixed in target body frame. This strategy assures that the target can be safely captured despite the tumbling rate it might have at the disadvantage of propellant consumption, a limitation that might become important in case this strategy needs to be applied to a multi-target mission.

Assuming a difficulty in achieving a safe holding position in the immediate vicinity of a target¹¹ Virgili-Llop et al. [105] proposed a guidance strategy of a free-flying manipulator to capture a tumbling target from a sufficiently faraway position. The guidance strategy was solved as a collection of sequential convex programming prob-

¹¹ From which the manipulator deployment and approach can be accomplished.

lems: a system-wide translation and an internal re-configuration of the manipulator. The proposed algorithm is able to handle the collision avoidance and line-of-sight (LOS) constraints, as well as satisfy the control limitations of both spacecraft and manipulator, while minimizing the control effort. The issue of the momentum management of the target, once captured, was outside of the scope of the research, thus revealing a possible limitation of the developed strategy.

Virgili-Llop and Romano [87] addressed this issue by including in the previously mentioned guidance strategy a terminal constraint on the angular momentum and linear velocity of the chaser spacecraft in order to perform a simultaneous capture and de-tumbling of a target. The base spacecraft during this maneuver is in a translation-flying/rotation-floating mode, where only the position of the base is controlled to reduce the control effort. Numerical simulations of the method revealed its decrease in applicability with the increase of the angular momentum of a target.

A conclusion that might be drawn from the mentioned studies on free-flying robots is that they appear to be better suited for capturing a tumbling target from a safety perspective, as they are characterized by a virtually unlimited workspace. However, when it comes to capturing a target with a large angular momentum, they present similar limitations of the studies cited in Section 2.4.2.1, due to physical limitations of a robotic spacecraft to cancel-out, transfer and store a large angular momentum of a target. In such cases, excluding the mass reduction of targets, by means of an ablation for example, the only feasible way to achieve safe capture conditions consists in reducing the relative attitude motion between the chaser and target so that the attitude of the latter is within the physical constraints of the manipulator.

2.4.2.3 *Relative angular rate reduction*

The active reduction of the relative angular rate between the chaser and target can be accomplished essentially using two different approaches: the relative motion synchronization and the active reduction of the angular rate of a target. Employing the relative motion synchronization, the relative angular rate between the chaser and target is reduced, prior to the capture maneuver, via a relative motion synchronization which is accomplished using the ACS of the base spacecraft. The advantage of such an approach lies in its similarity with the autonomous rendezvous and docking maneuver and increased safety. The disadvantage consists of the need for strict requirements on the performance of the spacecraft's GNC and sizable propellant usage, in case of full motion synchronization methods.

The seminal work of Nagamatsu, Kubota, and Nakatani [106] presented a capture strategy based on the decomposition of the attitude motion of the target into a superposition of three rotational motions with constant angular velocities. This way both the motion synchro-

nization of the spacecraft and manipulator are simplified. The method appears suitable for the capture of targets with fast and/or complicated attitude motions, but did not explore optimizing the maneuver and has not addressed the issue of the safety that might become crucial in case of large targets.

Tsuda and Nakasuka [107] proposed a capture procedure based on a shape adjusting spacecraft and an attitude control method, the *free motion path method*. The latter is able to circumvent the nonlinear dynamics of the system while limiting the maximum control torque and required computational load. The limitation of the method stems from the shape adjusting spacecraft which might become impractical in case of large targets.

Ma, Ma, and Shashikanth [108] took a more traditional approach to the capture of a tumbling target and employed optimal control with two criteria, minimum time and propellant consumption, able to obtain a trajectory of the chaser with zero relative rotation w. r. t. a target. The method was only demonstrated on a planar example and has overlooked the possibility of usage of the manipulator to reduce the relative motion while at the same time reducing the propellant consumption.

This idea was explored recently by Ma, Wei, and Yuan [10] who designed a semisynchronizing (i. e. partially synchronizing) strategy for capturing a fast tumbling target. This way propellant consumption can be significantly reduced during the synchronizing maneuver while at the same time reducing the relative angular velocity of the grappling feature within the acceptable ranges for the robotic arm. The developed approach consists of a chaser achieving a partial attitude synchronization by a pure rotation around a fixed axis coincident with the angular momentum vector of the target at a uniform angular rate. The motion of the manipulator to capture the target was outside of the scope of the research and any disturbance that it might cause to the spacecraft attitude during its deployment was simply assumed to be compensated by a feed-forward coordinated control (e. g. [109]) without any indication of magnitudes of required control forces.

Utilizing the active reduction of the angular rate of a target, the relative angular rate is assumed to be actively reduced by means of a contact/contactless method, such as a brush, eddy currents, etc. The advantage of such methods consists mainly of potentially lower propellant requirements (when compared to those from the previous category), while the disadvantages include even stricter GNC performance requirements, reduced safety and no previous flight heritage.

Matunaga, Kanzawa, and Ohkami [110] presented a method for reducing the angular momentum of the target using a flexible, cushion-type, end-effector that is to be placed in contact with the target. However, this method might not be safe in case of a fast tumbling target with appendages, and, since it uses a contact/push-based method,

could cause a momentary separation between the end-effector and target after each contact [84]. A similar method, based on a repetitive impulses applied by the manipulator to a designated contact point was discussed by Yoshikawa and Yamada [111] as a way to damp the rotational motion of a target. The method presents the same limitations as the previous one.

More recently, Nishida and Kawamoto [112] examined a capture strategy of a tumbling target with unknown moment of inertia. To this aim, the authors developed a *joint virtual depth control* algorithm which if applied to a brush-type end-effector is able to reduce the rotational motion of the target while complying with the imposed contact force profile and avoid pushing away the target during the process. The contact with the target is still required and might cause safety concerns should it need to be extended in order to properly reduce the angular rate of the target.

To avoid contact with the target, Sugai et al. [113] proposed a de-tumbling method using eddy currents generated by two magnetic coils attached to the end-effectors of a dual-manipulator spacecraft. The challenges afflicting this method consist mainly out of the need to maintain a constant and a very small distance¹² between the coils and the structure. This entails that during the maneuver manipulators would need to continuously track a complex motion of a target and be very long, in case of large targets. Furthermore, considering the small relative distances involved, the requirements on the GNC of the spacecraft will need to be extremely stringent.

In conclusion, despite a significant flight heritage of space robotics and 30 years of research an autonomous robotic capture of an un-cooperative, tumbling target has yet to be demonstrated in-orbit. In addition, although capturing a tumbling object is a frequent problem of the research in space robotics, it is difficult to identify one that could easily solve it [84], especially in case of targets characterized by large angular momenta. This is due to the fact that most of the state-of-the-art studies are confined to target objects with relatively low angular momenta and only a few have considered solving the motion planning problem of the approach phase of the manipulator such that the subsequent capture and stabilization/de-tumbling are feasible. The prevailing approach employs relative motion synchronization to achieve desired capture conditions at the disadvantage of propellant consumption and strict requirements on the performance of the GNC system of a robotic spacecraft.

¹² In the order of few millimeters.

3

SPACE DEBRIS ONTOLOGY FOR ACTIVE DEBRIS REMOVAL

Disclaimer on content reuse: this chapter is based on the following publications: [12, 15, 19]ⁱ.

The quest for an optimal ADR solution does not have a unique answer. Furthermore, the available domain data often lacks coherence, which leads to a general hesitation on how and when ADR should be performed. Finally, the existing state-of-the-art knowledge representation techniques in the domain of space debris have only focused onto SSA, neglecting ADR. To bridge this gap the first objective of this research is to develop a domain ontology, for data collection, storage and sharing of characteristics of IDOs, i. e. payloads and rocket bodies, useful to ADR. The ontology defines the minimal set of physical and dynamical parameters of an object deemed sufficient to infer its most suited ADR capture method(s). This way, not only the management, but also the discovery of new knowledge is facilitated.

The chapter presents the developed ontology starting with a statistical analysis of on-orbit fragmentations necessary to identify the severity and probability of each event. Then, the core terminology and the overall methodology of the underlying ontology are described. Next its software implementation and development workflows are presented. This is followed by the application of the ontology onto a database of representative objects. The chapter concludes with the analysis of results and a discussion about the strengths and weaknesses of the developed ontology in light of the performed analysis.

3.1 STATISTICAL ANALYSIS OF BREAKUPS

A fragmentation (i. e. a breakup) can be defined as a separation of at least one object (or fragment), with no operational function, from a parent spacecraft [23]. Such event can be seen as one of the main hazards to which the chaser spacecraft can be exposed during a capture phase of a generic ADR mission. This fact is especially true in case of objects characterized by stored on-board energy that were not passivated before their end-of-life. Therefore, a study of consequences

ⁱ Some of the text in this chapter is reprinted/adapted by permission from Springer Nature Customer Service Centre GmbH: Springer International Publishing, *Stardust Final Conference. Astrophysics and Space Science Proceedings*. "Taxonomy of LEO Space Debris Population for ADR Capture Methods Selection", DOI: 10/gf93mr, M. Jankovic and F. Kirchner, © 2018.

and probability of occurrence of such failure mode is deemed of utmost importance to limit the exposure of a chaser spacecraft to it and mitigate possible adverse effects. Past studies on the topic (see [114]) have focused exclusively on the statistical analysis of satellite failure data in order to help improve their reliability. To best of my knowledge, no statistical analysis of on-orbit fragmentation events using *survival analysis* has been performed before and therefore represents another contribution of the research illustrated in this thesis. The proof of its validity lies in the usage of the described methodology by the ESA Space Debris Office in the assessment of the impact of a mission on the overall space debris environment [115, 116] and the inclusion of the same methodology within the updated version of the International Academy of Astronautics (IAA) Situation Report on Space Debris [2].

By 2020, there have been a total of 561 confirmed on-orbit fragmentation events [31]. The historic evolution of such events since the beginning of the space age is illustrated in Fig. 3.1, grouped in terms of event causes.

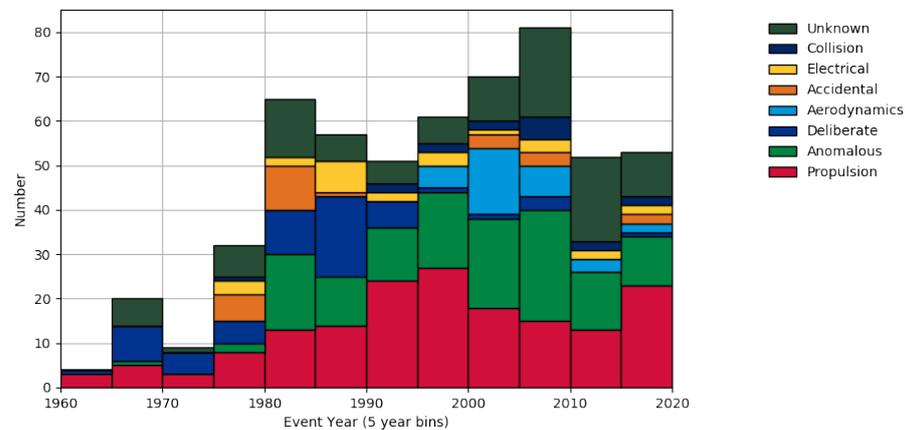


Figure 3.1: Evolution of the number of breakups per event cause (credit: [21]).

The definition of those causes as specified by the ESA Space Debris Office [21, 31] is as follows:

ACCIDENTAL events caused by design flaws of subsystems that ultimately leading to breakups (e.g. Cosmos 862 class explosive charges that proved to be unreliable).

AERODYNAMICS events due to an interaction of spacecraft with the Earth's atmosphere.

ANOMALOUS events characterized by an unplanned separation, usually at low velocity, of one or more detectable objects from the parent that essentially remains intact.

COLLISION events induced by a collision between two or more objects.

DELIBERATE events due to deliberate destruction of satellites.

ELECTRICAL events caused by a sudden release of energy stored within on-board batteries.

PROPULSION events generated by non-passivated energy stored within the propulsion subsystem.

UNCONFIRMED events having an unconfirmed origin and thus being only a provisional classification.

UNKNOWN events unable to be reliably classified in one of the above categories due to missing evidence.

Analyzing in more detail the historic evolution in Fig. 3.1, it can be seen that by 2020 the propulsion events are found to be the overall dominating type of breakups, with a total of 30.2 % of the total number of events. These are followed closely by anomalous and unknown events, each having accumulated over the whole history 23.33 % and 18.63 % of the total number, respectively. The remaining 27.84 % is distributed among the deliberate (10.49 %), aerodynamics (5.79 %), accidental (4.52 %), electrical (4.52 %) and collision (2.53 %) events. However, the distribution of those events is not uniform among the object classes, pointing at their different susceptibility in terms of breakup causes [31]. For this reason, any statistical analysis of on-orbit fragmentation events needs to distinguish between the events related to payloads and rocket bodies as any generalization will lead to skewed results.

3.1.1 Database and data description

The database considered in this research is the ESA's DISCOS. It can be considered one most authoritative databases in the domain of space debris and is used on a daily basis by the ESA's Space Debris Office for its activities and more widely throughout academia and industry. It provides the data about launch information, object registration, launch vehicle description, spacecraft information (as well as fragmentation events) of all trackable, unclassified objects dating back to the launch of the first satellite in 1957. As of 2020, it is estimated to contain more than 40 000 objects [117]. However, as any database of its kind it is far from being "complete" and is constantly being refined and updated.

The sample of DISCOS analyzed hereafter consists of all unmanned, cataloged objects from October 4, 1957 till July 9, 2019 having an average observable AMR within the $(0, 0.1) \text{ m}^2 \text{ kg}^{-1}$ interval¹³ as those objects would more likely represent future ADR targets. Human spaceflight related objects are excluded from the sample as they would tend to skew the probability results in view of more stringent spaceflight safety measures and reliability characterizing manned spacecraft [21]. The results are 11 490 cataloged objects characterized by 360 fragmentation events, out of which 324 are to be attributed to a parent object

¹³ Characterized by an average RCS $>1 \text{ m}^2$ and mass $>10 \text{ kg}$.

having a unique COSPAR International Designator (COSPAR ID), indicating that some cataloged objects within the sample have been subject to more than one fragmentation event. For each object within the sample, following information was collected and processed for the purposes of the statistical data analysis outlined hereafter: a) identification data, b) physical properties, c) launch date, d) fragmentation information, e) reentry date.

3.1.2 Survival analysis of breakups

The survival analysis represents in general a collection of statistical methods for data analysis whose outcome addresses the problem of deriving the *time until the occurrence of an event of interest* [118]. It is a major tool used by health care professionals in the statistical analysis of clinical trials, hence the name “survival analysis”. It is also extensively used in engineering within the context of reliability analysis to determine the time to failure of a mechanical system. In this context the term “reliability” is to be conceived as “the ability of an item to perform a required function, under given environmental and operational conditions and for a stated period of time” [119].

Time in this context refers to days, months, years from the beginning of a study/observation period till the occurrence of an *event* of interest, which can be a death, in case of biological organisms, or a failure, in case of mechanical systems, or any experience of interest that an individual or object of study might be involved in. Therefore, often the time variable is identified with the term *survival time* since it defines the time that an individual has “survived” without experiencing an event over some period of study. The term *event* is often identified with the word *failure* since it generally characterizes an event that has a negative experience for the individual/object [118].

This terminology is most definitely appropriate for the survival analysis of breakups where objects of the analysis are cataloged spacecraft and events of interest are the fragmentation events as defined at the beginning of the section.

Censoring is another important term within the survival analysis and is characteristic of “incomplete” life data, i. e. data containing survival times of some of the objects but not all. By contrast, the life data is said to be “complete”, if it contains records of all failure times of all objects under the study [114, 118]. Censoring generally occurs for three reasons: a) an object does not experience the failure within the time-frame of the study, b) an object is lost to follow-up during the study period, c) an object is removed from the study before any failure is recorded [118]. Different classifications and types of censoring exist but the two most common types are the “right-censoring” and “left-censoring”, even though most of the survival data is right-censored.

The data is said to be right-censored when the survival time is incomplete on the right side of the follow-up period, i. e. the entry of an item into the study is known but not its failure time, due either to an early end of the study or an early removal of the item from the same (see Fig. 3.2). By contrast, the data is said to be left-censored when the survival time of an item is less than or equal to the observed one, i. e. the entry of an item into the study is unknown and thus is incomplete on the left side. This might be the case of an uncertain date of launch of a cataloged object or a delayed report of symptoms of a disease contracted by an individual at an earlier stage of the observation period [114, 118].

Right-censored data can be further detailed into [119, 120]:

TYPE 1 data containing the survival times recorded only until the predetermined observation period, characterized by a well-defined end time.

TYPE 2 data incorporating the survival times recorded only until a predetermined number of events.

TYPE 3 data which is a combination of the types 1 and 2, i. e. the survival times are recorded until either the end of the observation period is reached or the maximum predetermined number of events is recorded, whichever happens first.

TYPE 4 data also known as random censored data where both time to event and time to censoring are stochastic.

The sample data introduced in Section 3.1.1 is right-censored of type 4 with mostly staggered entries, since most of the objects within the sample are characterized by different entry dates¹⁴ and stochastic failure and censoring times. This situation is illustrated in Fig. 3.2, where: a) the entry of an object into the study is identified with its launch, b) the failure is recorded when a breakup occurs (if any), c) the censoring ensues either because an object is removed from the study before a failure, due to its reentry into the atmosphere, or because it did not experience a breakup till the end of the observation period, set to 9 July 2019.

Data of this type poses two main challenges. The first one in the form of staggered entries and the second one in the form of censored data. Staggered entries can be easily tackled by changing the observation period time format from calendar date to in-orbit time, i. e. time after successful orbit insertion. Censoring on the other hand requires a particular attention especially when the desired outcome is the survivor function $S(t)$ or the reliability function $R(t)$ of spacecraft in the observed time period.

Let us now define the main functions considered in any survival analysis. Identifying with T the random variable of the survival time of an item, also known as the *time to failure* of an object, and with t a

¹⁴ Due to their mostly different launch dates.

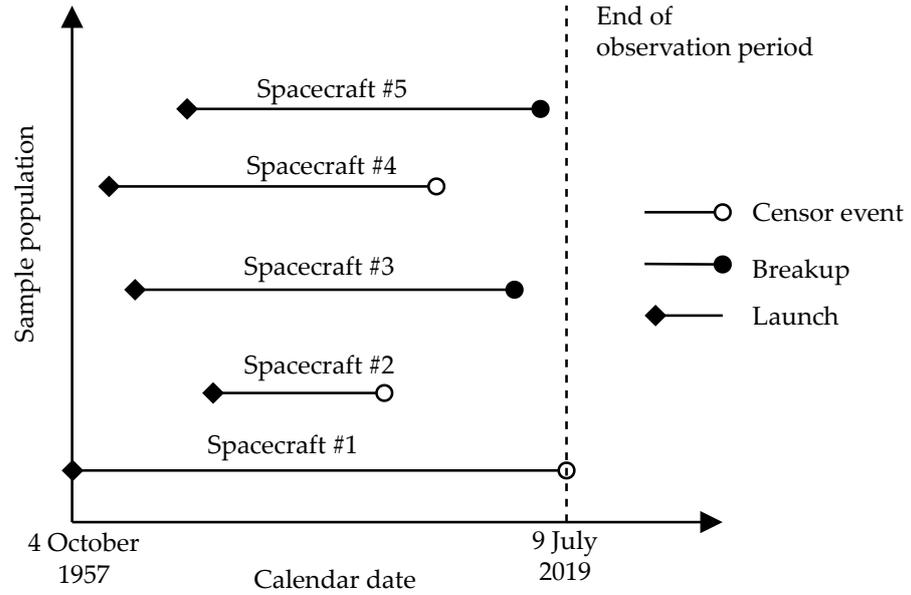


Figure 3.2: Example of right-censored data with staggered entries (based on [114, Fig. 2.1]).

specific value of interest of T , the reliability function $R(t)$ of an object or better the probability that it will not fail in the time interval $(0, t]$ can be expressed as [119]:

$$R(t) = P(T > t) = \int_t^{\infty} f(u)du = 1 - F(t) \quad \text{for } t > 0 \quad (3.1)$$

where P is the probability of an event, $f(t)$ (or $f(u)$) is the probability density function and u is the variable of integration.

The probability density function $f(t)$ can be defined as [119]:

$$\begin{aligned} f(t) &= \frac{d}{dt}F(t) \\ &= \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{P(t < T \leq t + \Delta t)}{\Delta t} \end{aligned} \quad (3.2)$$

The cumulative distribution function $F(t)$, i. e. the probability of failure of an item within the interval $(0, t]$, can be written as [119]:

$$F(t) = P(T \leq t) = \int_0^t f(u)du = 1 - R(t) \quad \text{for } t > 0 \quad (3.3)$$

The failure rate function, also known as the hazard function $h(t)$, can be expressed as [118, 119]:

$$\begin{aligned} h(t) &= \lim_{\Delta t \rightarrow 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t} \\ &= \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} \frac{1}{R(t)} \\ &= \frac{f(t)}{R(t)} \end{aligned} \quad (3.4)$$

$$= -\frac{d}{dt} \ln R(t) \quad (3.5)$$

which reveals the instantaneous potential per unit time for a failure event to occur given that an item has “survived” until time t . Therefore, $h(t)$ contains antagonistic information to $R(t)$, in line with $F(t)$. However, differently from $R(t)$ and $F(t)$, $h(t)$ is not a probability, rather it is a rate since it is a conditional probability divided by Δt . In fact, its value may be greater than one since its magnitude depends on the time units used in the survivor analysis [118].

The relationship between the probability density function $f(t)$, the hazard function $h(t)$ and the reliability function $R(t)$ can be derived by expressing $R(t)$ in terms of $h(t)$ as [119]:

$$R(t) = \exp\left(-\int_0^t h(u) du\right) \quad (3.6)$$

$$= \exp(-H(t)) \quad (3.7)$$

where

$$H(t) = \int_0^t h(u) du \quad (3.8)$$

is the cumulative hazard function.

Using Eqs. (3.4) and (3.6), the expression of the probability density function $f(t)$ can be rewritten in terms of $h(t)$ as [119]:

$$\begin{aligned} f(t) &= h(t) \exp\left(-\int_0^t h(u) du\right) \\ &= h(t)R(t) \end{aligned} \quad (3.9)$$

Defining one of $f(t)$, $R(t)$ or $h(t)$ it is possible to determine the other two variables using the mentioned equations [118].

For the assessment of those functions from incomplete life data this work uses the *nonparametric* estimators, the *Kaplan-Meier* and *Nelson estimators*. The reason for choosing them over other methods lies in their accuracy considering that they are based on the actual data and do not fit any predefined life distribution of $F(t)$ [114].

Given a fixed $t > 0$, the general expression of the Kaplan-Meier estimator of $R(t)$ can be defined as [119]:

$$\hat{R}(t) = \prod_{i \in \mathbb{Z} | t_{(i)} \leq t} \frac{n_i - d_i}{n_i} \quad (3.10)$$

where $\hat{R}(t)$ is the estimate of $R(t)$, $t_{(i)}$ denotes recorded times until failure or censoring, $i \in \mathbb{Z}$ as well as $i = 1, \dots, n$, with n being the initial number of operating units, and

$$\begin{aligned} n_i &= \text{number of items at risk right before } t_{(i)} \\ &= n - \text{number of censored objects right before } t_{(i)} \\ &\quad - \text{number of failed objects right before } t_{(i)}. \end{aligned}$$

Finally, in Eq. (3.10) d_i denotes the number of objects failing exactly at $t_{(i)}$ for $i = 1, \dots, n$ and is referred to as a *tie of multiplicity d* . The result of the estimator is therefore a staircase reliability function with discontinuities at observed failure times that does not change at censored times [114].

In case of objects characterized by exactly the same censoring and failure times, the failure event is assumed to have precedence. Therefore, the event is recorded as observed, since a censored object can always be assumed to be able to survive an infinitely small period past $t_{(i)}$ [114].

The estimator in Eq. (3.10) provides only a maximum likelihood of $\hat{R}(t)$ and does not give any information about the dispersion of values around $\hat{R}(t)$. This quantity is readily captured by the variance or standard deviation of $\hat{R}(t)$ which then can be used to derive the upper and lower bounds of a desired confidence interval around the maximum likelihood value. Assuming that a desired value of the confidence is 95 %, the corresponding interval around $\hat{R}(t)$ can be determined by [114]:

$$R_{95\%}(t) = \hat{R}(t) \pm 1.96\hat{\sigma}_R^2(t) \quad (3.11)$$

where $\hat{\sigma}^2$ is the estimate of the variance of $\hat{R}(t)$ and can be derived in its simplest form using the *Greenwood's formula* [119]:

$$\hat{\sigma}_R^2(t) = \widehat{\text{Var}}[\hat{R}(t)] = \hat{R}(t)^2 \sum_{i \in \mathbb{Z} | t_{(i)} \leq t} \frac{d_i}{n_i(n_i - d_i)} \quad (3.12)$$

Applying Eqs. (3.10) and (3.12) to the sample database described in Section 3.1.1 ¹⁵ results in Fig. 3.3, representing the estimate of spacecraft reliability/survival (to fragmentation events/breakups) as a whole, i. e. without distinction between object nor breakup classes.

In the analyzed data set, fragmentation events, that could be linked to the same parent object (36 in total) were excluded from the analysis in order to eliminate any correlation between multiple, consecutive fragmentation events and decrease the “false” number of objects being subject to the same class of event.

Assuming a successful launch and orbit insertion, from Fig. 3.3 can be seen that the estimate of the spacecraft reliability¹⁶ at $t = 5$ years is expected to drop by 2.33 % from 100 % and have the most likely value of 97.29 % while being within 96.89 % to 97.65 % interval with a 95 %

¹⁵ Consisting of 11 454 spacecraft, 324 unique failure and 11 221 censored times.

¹⁶ Its probability of survival to breakup events.

confidence. After 50 years of being in-orbit, the reliability estimate is expected to drop further and be most likely equal to 92.69%, i. e. be between 91.45% to 93.76% with a 95% likelihood.

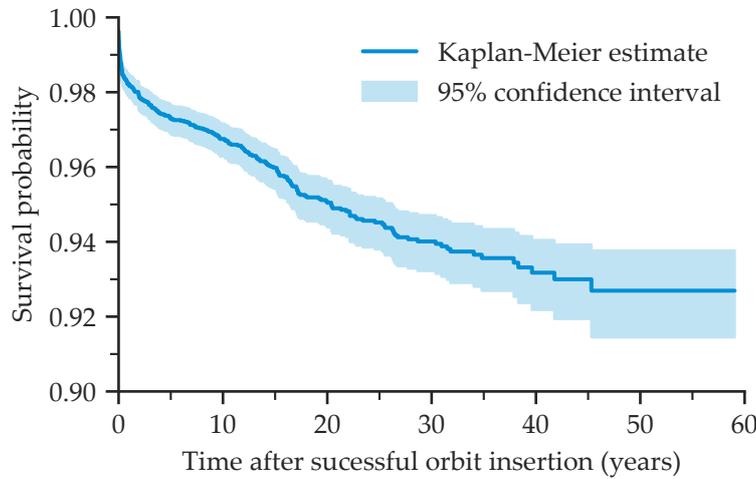


Figure 3.3: Spacecraft survival probability with 95% confidence interval.

Finally, from Fig. 3.3 it can be noticed that the overall variance of $\hat{R}(t)$ (i. e. its 95% confidence interval) increases over time and is a direct consequence of the decreasing sample size as more spacecraft are subject to failure or censoring [114].

The estimate of the cumulative hazard function $H(t)$, and consequently of $h(t)$ (see Eq. (3.8)), from censored data while can be obtained from the Kaplan-Meier estimator is more readily deduced from the nonparametric Nelson estimator having the following generic expression [120]:

$$\hat{H}(t) = \sum_{i \in \mathbb{Z} | t_{(i)} \leq t} \frac{m_i}{n_i} \quad (3.13)$$

Plotting $\hat{H}(t)$ against time it is possible to obtain the *Nelson plot* that indicates (based on the relationship between $h(t)$ and $H(t)$) that:

- an increasing $h(t)$ in case of a convex curve of $\hat{H}(t)$,
- a decreasing $h(t)$ in case of a concave curve of $\hat{H}(t)$,
- a bathtub-shaped $h(t)$ in case of a curve of $\hat{H}(t)$ that is initially concave, then constant and finally convex $h(t)$ ¹⁷ [119].

The Nelson estimator (in Eq. (3.13)) provides only a maximum likelihood estimate of $H(t)$ and does not give any information about the dispersion around $\hat{H}(t)$. However, just as in case of the Kaplan-Meier estimator (see Eq. (3.11)), this dispersion can be readily captured by the variance or standard deviation of $\hat{H}(t)$. The estimate of the

¹⁷ Characteristic shape of $h(t)$ of consumer electronic/mechanic devices.

variance of $\hat{H}(t)$ can be derived, similarly as in case of Eq. (3.12), using the following expression [120]:

$$\hat{\sigma}_H(t)^2 = \widehat{Var}[\hat{H}(t)] = \sum_{i \in \mathcal{Z} | t_{(i)} \leq t} \frac{d_i(n_i - d_i)}{n_i^3} \quad (3.14)$$

or alternatively [120]:

$$\hat{\sigma}_H(t)^2 = \widehat{Var}[\hat{H}(t)] = \sum_{i \in \mathcal{Z} | t_{(i)} \leq t} \frac{d_i}{n_i^2} \quad (3.15)$$

Applying Eqs. (3.13) and (3.14) to the sample database described in Section 3.1.1 results in Fig. 3.4, illustrating $\hat{H}(t)$ with a 95 % confidence interval without distinction between object nor breakup classes. The figure shows an overall decreasing $h(t)$ of spacecraft, i. e. their instantaneous potential at time t for a fragmentation to occur. This trend is interrupted with a rapid increase of $h(t)$ at intervals $[0, 2.5]$ and $[5, 17.5]$.

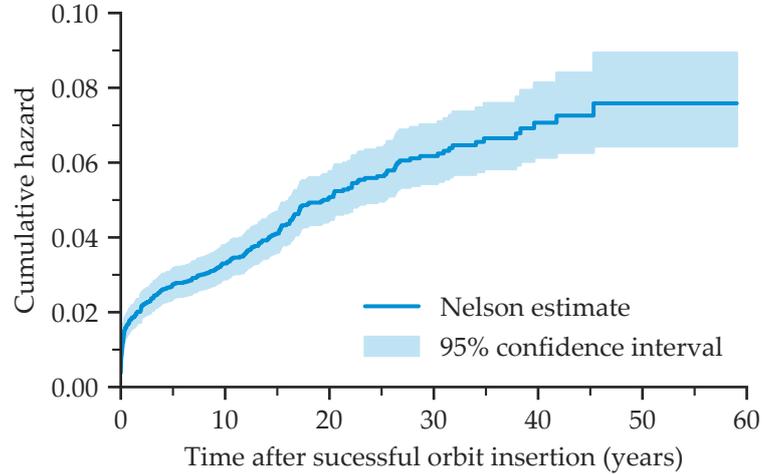


Figure 3.4: Spacecraft cumulative hazard with 95 % confidence interval.

Before proceeding to the survival analysis of the two major object classes considered by this research, i. e. rocket bodies and payloads, a note on the limitation of the approach is necessary. In fact, it can be argued that in the absence of a mass production of spacecraft no two are truly alike nor do they operate under the exactly same environmental conditions, even if they are placed in the same orbit, as part of a larger constellation [114]. Therefore, a survival analysis of spacecraft can follow two distinct approaches.

The first, where the analysis is performed on a “collective”, assuming different breakups independent of each other and equally distributed, without any distinction between spacecraft themselves, in terms of bus, mission type, mass, etc. The advantage of such approach is that the survival analysis can be performed on a larger sample that

would guarantee some precision and a narrow confidence interval of the “collective”. Its disadvantage lies in the fact that the assumptions made on fragmentation events might not be completely realistic and any conclusion on the survivor or hazard functions of the “collective” might not entirely reflect that of individuals of the “collective” [114].

The second approach consists of a survival analysis performed on a smaller sample of individuals, specialized in terms of bus, mission type, mass, etc., and thus draws conclusions on those groups exclusively. The advantage of such approach lies in the data specialization, therefore any reliability conclusion is more focused. Its disadvantage is the sample size reduction of life data which has as a consequence the expansion of the confidence interval and thus of the uncertainty of the more focused analysis [114]. This fact is especially true in case of the analysis of cataloged fragmentations where data specialization could reduce the number of data points to few than a hundred and would consequently result in a highly dispersed or “uncertain” survival analysis of specialized data.

With this in mind, this work adopts the first approach and acknowledges its limitations in the statistical analysis of breakups of rocket bodies and payloads described in the following sections.

3.1.3 Rocket bodies related breakups

The data used for the statistical analysis of rocket bodies consists of a subset of the sample described in Section 3.1.1 and is made of 5185 large, unmanned rocket bodies, 160 cataloged fragmentation due to 7 relevant breakup causes (i. e. aerodynamics, anomalous, collision, deliberate, propulsion, unconfirmed, unknown) and a total number of 6480 cataloged fragments. As in case of the survival analysis of the data sample of the “collective”, in this particular subset of data, fragmentation events linked to the same parent object (four in total) are excluded from the survival analysis to eliminate any correlation between consecutive fragmentation events and decrease the “false” number of objects being subject to the same class of event. Finally, when analyzing data due to a particular breakup event, failures due to other events observed in the data are considered an additional source of censoring that is accounted in the analysis, similarly to what was done in [114] for satellite reliability analysis.

3.1.3.1 Probability of breakup of rocket bodies

The survival probability of rocket bodies and therefore their probability of breakup are obtained by applying the Kaplan-Meier estimator (see Eqs. (3.10) and (3.11)) to the previously mentioned data. The result of the estimator is shown in Fig. 3.5, where for readability only the

probabilities due to the events causing the most prominent reduction of the survival probability of rocket bodies are represented.

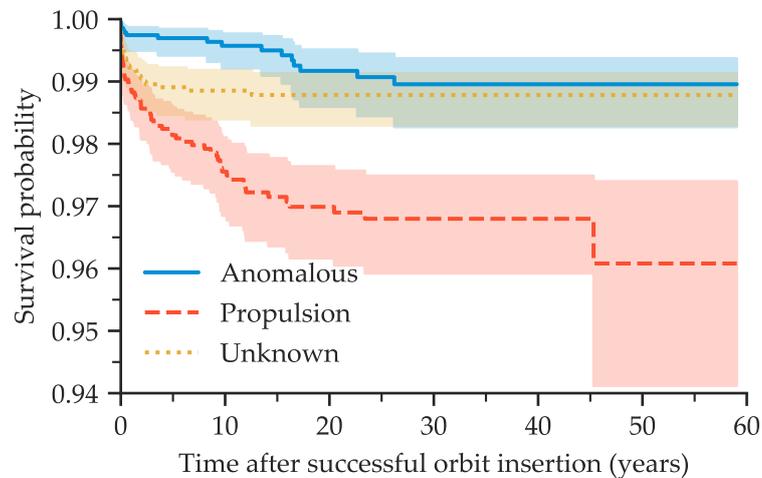


Figure 3.5: Rocket body survival probability due to anomalous, propulsion and unknown events with 95% confidence interval.

Assuming a successful launch and orbit insertion of a rocket body, from the figure can be seen that overall there is more than 96% chance that the same object will survive past the 50-year mark without experiencing any fragmentation event. This is to be attributed to the overall high reliability of space hardware and an increasing implementation of passivation and de-orbiting techniques, applied “routinely” to rocket bodies since the early ‘90s. However, the same estimate shows that rocket bodies as a collective are subject to a quite prominent decrease in reliability early-on in their lifetime, mainly due to propulsion and unknown events. This fact is known in reliability engineering as *infant mortality* and indicates a phase during which any item with an undiscovered defect fails upon activation [119]. This might be caused by the fact that most of the rocket bodies are made with a very short useful lifetime in mind, due to their singular purpose, i. e. insertion of a payload into a specific orbit.

A more gradual decrease in time of the survival probability can be observed due to anomalous events. This conclusion coupled with the definition of anomalous events described at the beginning of Section 3.1, might be attributed to *wear-out failures*, during which the breakup of components increases with time due to fatigue, corrosion etc., caused by longer than expected exposure of hardware to the harsh conditions of the space environment.

More insight can be gained from the estimate of the rocket body breakup hazard which might be derived using the Nelson estimator (see Eqs. (3.13) and (3.14)). The result is illustrated in Fig. 3.6 where once again for readability only the failure rates due to the three most frequent events are represented.

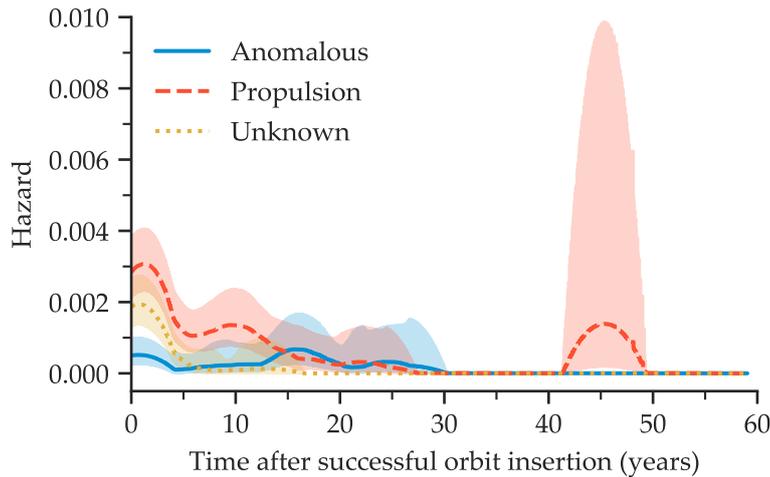


Figure 3.6: Rocket body breakup hazard estimate due to anomalous, propulsion and unknown events with 95 % confidence interval.

The figure depicts an overall low hazard estimate with a prominent failure rate within the first five years of on-orbit life mainly due to propulsion events. The shape of the corresponding failure rate function is bathtub-like, indicating a typical failure rate shape of mechanical and electronic components characterized by an initial “burn-in period” (with a high infant mortality), a “useful life period” (with a constant failure rate) and a “wear-out period” (with an increasing failure rate).

The hazard function due to unknown events exhibits an overall decreasing trend after a very high initial rate (similar to that of propulsion related events) which suggests active systems, having some sort of stored energy, be it mechanical, electrical or chemical, as a probable cause of events despite the “unknown” classification.

The hazard function due to anomalous events, after an initial high rate, appears to gradually increase in time till the 15-year mark after which it gradually decreases to zero and could be attributed to a more gradual wear-out of components after an initial “burn-in period”.

3.1.3.2 Severity of breakup of rocket bodies

Not all events have the same severity¹⁸ and frequency. In terms of the total number of events, large rocket bodies are to date mainly afflicted by two types of breakup events: propulsion and unknown. In terms of the total number of fragments, propulsion events alone are responsible for more than 80 % of the total number of generated fragments. This is depicted in Fig. 3.7, where the distribution of the cumulative number of fragments per event class is represented in the form of a *Pareto chart* [121]. It illustrates the Pareto principle, also known as the 80/20 rule or the law of the vital few, stating that roughly 80 % of the effects is caused only by 20 % of causes. Therefore, in this work the Pareto

¹⁸ Identified in this work as the median number of fragments generated by a breakup event.

principle is used to narrow down the severity analysis on propulsion related events only.

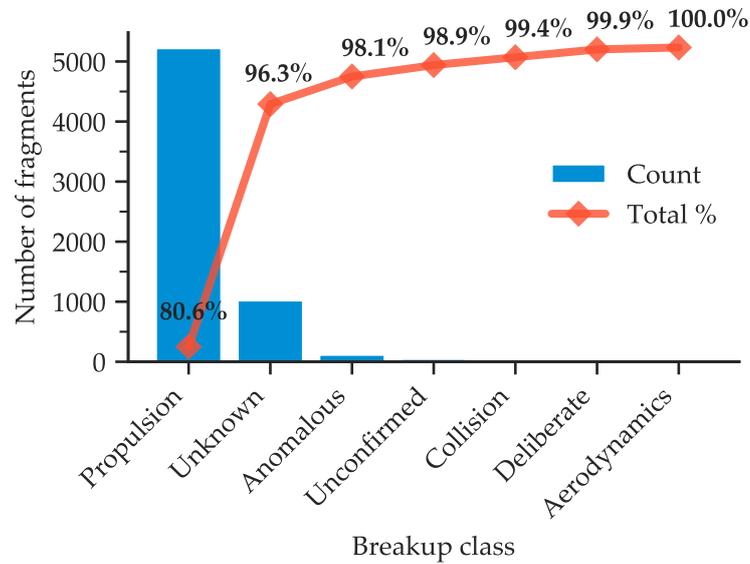


Figure 3.7: Pareto chart of breakup events of rocket bodies.

Having selected the most severe type of event for rocket bodies, the next step consists in determining the severity of such events while distinguishing between different types of rocket bodies. Considering the main function of rocket bodies, i. e. to insert payloads into orbit, one of their main subsystems to achieve this goal is the main propulsion subsystem. Therefore, a suitable method of categorizing rocket bodies can be on the propellant type used by their main propulsion system, which for the purposes of this work is identified as: cryogenic, hypergolic, petroleum or solid [122]. Using this grouping, it is possible to derive statistical values of propulsion events of each group and explore if this leads to different distribution of data among them, as it might be expected given the fundamental differences between the groups in terms of propellant type and hardware needed to use it. Fig. 3.8 is the result of such an analysis and displays the distribution of the number of generated fragments of 76 propulsion related breakups w. r. t. the propellant type of the main propulsion system of rocket bodies. The result shows that the median number of fragments generated by rocket bodies equipped with a hypergolic-based propulsion system is one order of magnitude greater than those without. This suggests that objects having hypergolic propellant on-board should be characterized by a greater severity number when compared to the others. In fact, 50% of hypergolic-based events is characterized by more than 79.5 fragments per event, with a maximum number of fragments being equal to 371. Finally, it can be noticed that the distribution of generated fragments of hypergolic-based events is much more dispersed when compared to the others and is *skewed right*, indicating

that fragmentation events generating large amounts of fragments are more spread out than those generating only few.

Analyzing the time distribution of all events it is possible to observe that the median time to event is 0.72 years, which might be attributed to the large amount of propellant still on-board at that stage of a mission. The time distribution of the most prolific events in the data sample characterized by ≥ 200 fragments per event, irrespective of the propellant type, indicates instead a median time to breakup equal to 1.05 years. It can be concluded that while determining the severity of rocket bodies not only the type of propellant of the main propulsion subsystem needs to be considered but also their on-orbit lifetime i. e. the shorter the lifetime the greater will be the expected severity of an eventual propulsion event.

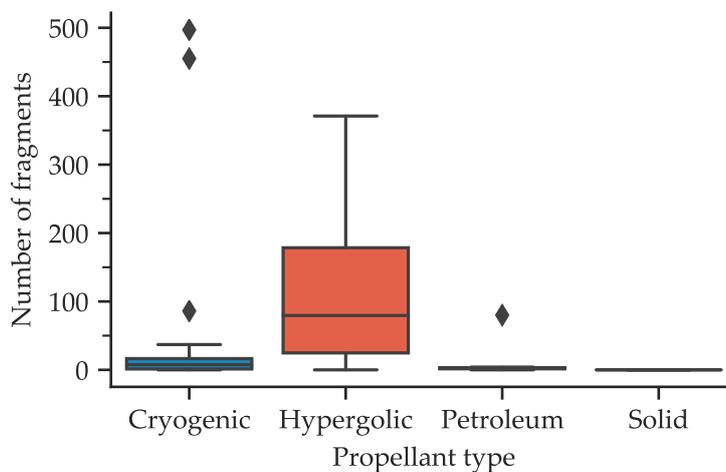


Figure 3.8: Box plot of propulsion events of rocket bodies in terms of the number of fragments per propellant type.

3.1.4 Payloads related breakups

The data used for the statistical analysis of payloads consists, as in case of rocket bodies, of a subset of the sample described in Section 3.1.1 related to payloads and is made of 5002 large, unmanned payloads, 191 cataloged fragmentations due to 9 relevant breakup causes (i. e. accidental, aerodynamics, anomalous, collision, deliberate, electrical, propulsion, unconfirmed, unknown) and a total number of 9535 cataloged fragments. As in case of the survival analysis of rocket bodies, fragmentation events linked to the same parent object (32 in total) are excluded from the survival analysis to eliminate any correlation between consecutive fragmentation events and decrease the “false” number of objects being subject to the same class of event. Finally, when analyzing data due to a particular breakup event, failures due to other events observed in the data form an additional source of

censoring that is accounted in the analysis, similarly to what was done in [114] for satellite reliability analysis.

3.1.4.1 Probability of breakup of payloads

The result of the survival analysis of the previously mentioned data is visible in Figs. 3.9 and 3.10, where only the survival probability functions of the most relevant events for this class of objects is represented for readability. Similarly, to the results of the survival analysis of rocket bodies (see Fig. 3.5), it is found that upon a successful launch and in-orbit insertion of an object there is more than 96% chance that the same object will survive past the 50-year mark without experiencing any fragmentation event. However, unlike rocket bodies that exhibit quite a prominent infant mortality, the reduction of the reliability of payloads is more gradual in time, as it might be expected from hardware whose purpose is to remain functional in-orbit for a longer period. In fact, excluding electrical and unknown events, responsible for an initial decrease of the reliability within the first two years, the overall hazard function of payloads is dominated by anomalous events that increases with time. This fact coupled with the definition of anomalous events (see beginning of Section 3.1), could be attributed to a more gradual wear-out of components of satellites exposed to the harsh environment of space.

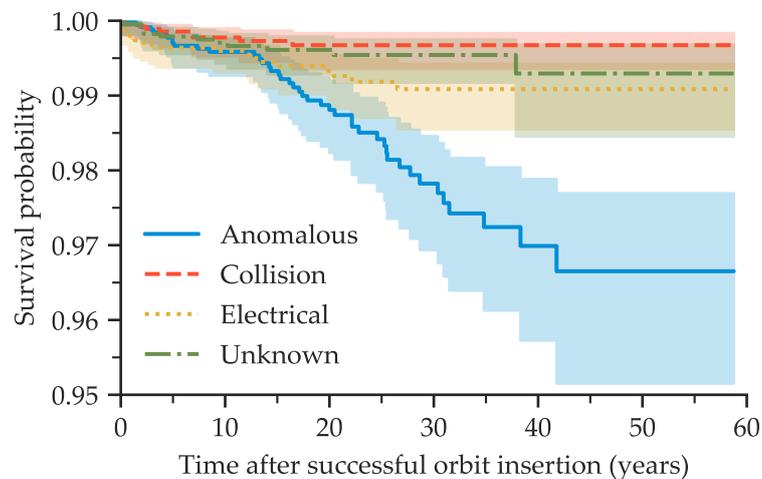


Figure 3.9: Payload survival probability due to anomalous, collision, electrical and unknown events with 95% confidence interval.

The failure rate of payloads caused by a collision event is estimated to be contained within the first 20 years of in-orbit lifetime of an object and mostly lower than that caused by other events.

The unknown events cause a bathtub-like hazard curve, hinting at active systems as probable offenders, despite the “unknown” classification of breakup events. Furthermore, the curve resembles a scaled-

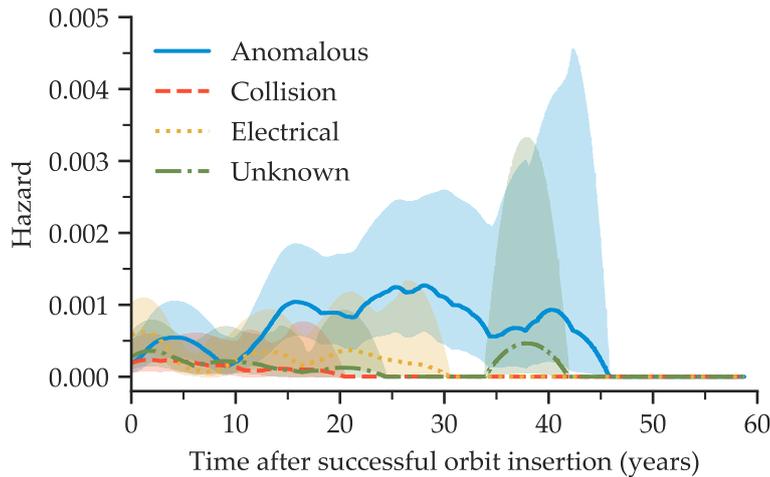


Figure 3.10: Payload breakup hazard estimate due to anomalous, collision, electrical and unknown events with 95 % confidence interval.

down version of that of rocket bodies caused by propulsion events (see Fig. 3.6), hinting at propulsion subsystems as the most likely culprit, although a more in depth analysis of this hypothesis (out of the scope of this work) is needed before any definitive conclusion can be drawn.

3.1.4.2 Severity of breakups of payloads

Payload related events are characterized by a distribution, in terms of total number of events, dominated by anomalous and deliberate events. However, as in case of rocket body events, that distribution is different considering the total amount of fragments generated by each breakup class. In fact, around 88 % of the total number of cataloged fragments can be attributed to only three types of events: deliberate, electrical and anomalous events (see Fig. 3.11). Therefore, excluding deliberate events due to their “intentional” nature, in this work the Pareto principle is used to further focus the severity analysis on the electrical and anomalous events, although the collision and unknown events are also included in what follows for completeness.

Categorizing the payload related events according to one of the four mentioned classes, it is possible to derive statistical values and explore the distributions of data among the groups, as it was done for rocket bodies with a goal to determine the severity of each type of event. Fig. 3.12 is the result of such an analysis applied on the number of generated fragments per breakup event. Differently from propulsion events of rocket bodies, the outcome of the analysis points out an overall low median number of fragments per event across all groups while highlighting the potential severity of electrical and collision events to generate large amounts of cataloged fragments with just one

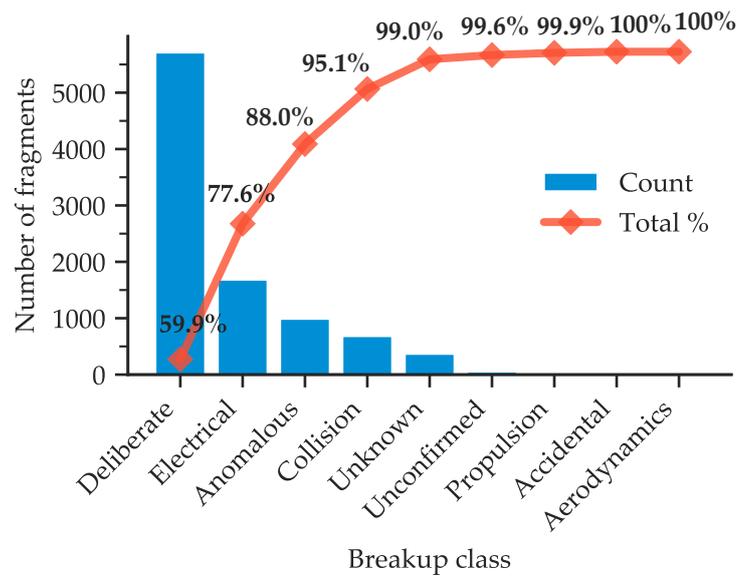


Figure 3.11: Pareto chart of breakup events of payloads.

event (see the outliers of the two types of events). Furthermore, the electrical and unknown events exhibit much more dispersed distributions of generated fragments when compared to the other two and are both skewed right, implying a larger distribution of events generating significant amounts of fragments when compared to those causing just few. Nevertheless, their distributions are much more contained w. r. t. those of propulsion-related events of rocket bodies and exhibit no discernible time-pattern. This fact might be attributed to a generally fail-safe design of payloads, meant to be reliable for years, if not decades, unlike rocket bodies which useful lifetime is generally in the order of days, if not weeks.

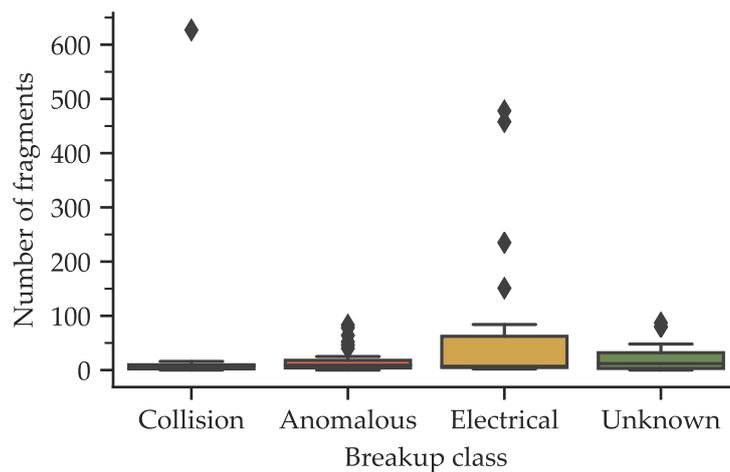


Figure 3.12: Box plot of breakup events of payloads in terms of fragments grouped by event class.

The median times to an event characterized by a number of fragments larger than 12^{19} are found to be equal to: 9.09 (for collision events), 24.53 (for anomalous events), 2.36 (for electrical events) and 8.20 (for unknown events) years.

¹⁹ *The largest median number of fragments among the classes.*

3.2 CORE TERMINOLOGY AND METHODOLOGY

The overall methodology of this work consists of the development of an ontology for ADR capture method inference, based on physical and dynamical properties of potential targets. To do so, at first the precise domain of interest and scope of the ontology are defined. Then, main classes and their hierarchy are determined. Finally, axioms allowing classification of ADR capture methods, associated with potential targets, are specified, based on the statistical analysis of on-orbit fragmentations (described in Section 3.1) and state-of-the-art capture methods characteristics.

3.2.1 Introduction into knowledge representation

Knowledge is everything, but unstructured knowledge is useless and thus can be considered as nothing. Therefore, it is important to capture and store knowledge in an unambiguous form effectively and be able to reuse it in the future for activities such as design, construction and/or verification of systems for example.

The knowledge could be information describing entire systems or specific subsystems. It could define the interaction of a system with an environment or be used as a formal vocabulary within a domain of interest.

The collection, storage, sharing and reuse of knowledge in an explicit and formal scheme is a subject as old as history. One of the classical methods of knowledge representation (the taxonomy) structures the knowledge in a classification able to store information through the membership property of things or concepts being classified (see Section 2.3.2). Hence, the relationship between things is of a single type, i. e. by association. Another classical method to represent knowledge is the ontology. The word ontology consists of two Greek words, *ὄντος* (ontos, “being”) and *-λογία* (-logia, “logical discourse”), describing a method for the “logical discourse of being”, where the knowledge is structured in classes, properties and restrictions of the latter, allowing a richer description of the domain of discourse when compared to a taxonomy [123]. This permits not only to determine a formal common vocabulary of a domain, but also to infer new knowledge from existing one through the usage of a semantic reasoner. It also enables an easier reuse of the domain knowledge, given the formal, explicit definition of the underlying information and assumptions. Finally, it

allows separating the domain and operational knowledge from each other, thus supporting reuse of an already developed framework in another related field of discourse [9].

Modeling a domain depends greatly on the expected application and extensions in mind. Therefore, it can be stated that there is *no one “correct” method* for modeling a domain with an ontology. In fact, deciding how detailed or general an ontology should be based on its application. The development process of an ontology is necessarily iterative and will most likely continue through the entire life cycle of the ontology, as new features are added or bugs are found and corrected after an evaluation of the same. Finally, it needs to be remembered that an ontology should be a model the reality, i. e. its concepts should be as close as possible to real-world objects (physical or logical) and their relationship [9].

Among the existing knowledge representation languages that can be used to define a domain-ontology, the most common one is the 2009 specification of the Web Ontology Language (OWL) (also known as OWL2), developed by the World Wide Web Consortium (W3C) for the Semantic Web. Ontologies developed with OWL2 store the information about the domain of interest into Semantic Web documents, capable of representing *individuals, properties, classes* and *relationship* among them [124]. Those terms are defined in this work as [9, 125]:

INDIVIDUALS also known as *instances*, represent specific objects of the domain of interest and are often defined as “instances of classes”.

PROPERTIES also known as *slots*, represent relations between two individuals, in case of *object properties*; an individual and a data value, in case of *data properties*; an individual and an annotation, in case of *annotation properties*.

CLASSES also known as *sets* or *concepts*, describe the physical concepts of the domain of discourse and are characterized by a formal membership requirements defined within their axioms. A class can have subclasses that define concepts into more detail than the “parent” superclass and can be seen as a collection of individuals.

These concepts and their relations are illustrated in Fig. 3.13 depicting an exemplary diagram of the developed ontology. For example, the object property `has_attitude`²⁰ ties the individual `2002-009A` to the individual `2002-009A_Attitude`, representing the attitude state of the former. Similarly, the data properties `has_angular_period` and `has_angular_rate` relate the individual `2002-009A_Attitude` to data values `164.5` and `2.19` (expressed in `s` and `deg s-1`), respectively. Finally, the individual `2002-009A` is classified both as a kind of `Payload` and `Space_Object_Type`, since the former is a subclass of the latter.

²⁰ The typewrite font is used hereafter to highlight the terms as defined within the domain-ontology.

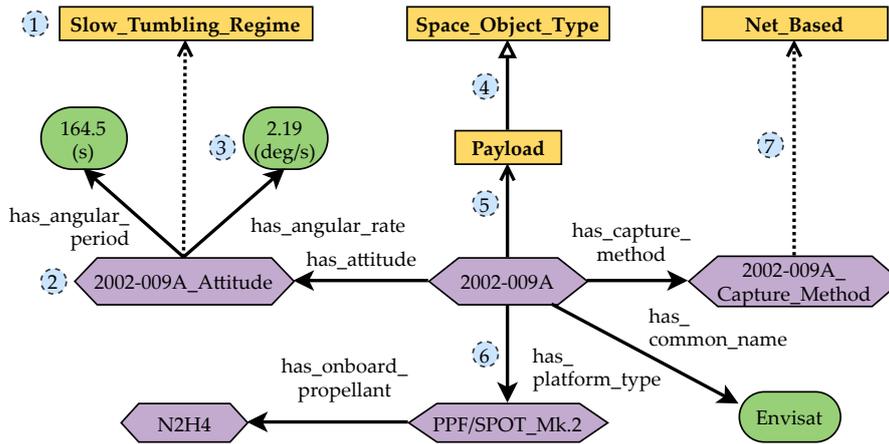


Figure 3.13: Example diagram of an ontology. The conventions used in the diagram are the following: (1) yellow squares represent classes, (2) purple hexagons represent individuals, (3) green rounded squares depict numerical or string values, (4) closed hollow arrows portray subclass or subproperty relations, (5) opened arrows illustrate “type of” relations, (6) semi-closed solid arrows depict properties of individuals, (7) dashed arrows portray inferred axioms (adapted from [12]).

3.2.2 Domain of interest and scope

The methodology used to develop ontology for active debris removal (TRACER) is primarily the “Simple Knowledge-Engineering Methodology” [9], with some hints taken from the “Unified Process for Ontology” method [126]. Overall it consists of an iterative approach where at first a rough draft of the ontology is developed, starting from the initially defined *requirements set* and a *lexicon*, defined as a collection of most important terms of the domain. Then, the lexicon is transformed into a *glossary*, defined as a collection of most important terms of the domain with accompanying definitions. Subsequently, the ontology is revised and enriched based on the analysis of the existing draft. After several iterations of the previous step, the ontology is finally formalized and tested using a representative set of data against the initial set of requirements.

The domain of TRACER is that of IDOs (i. e. payloads and rocket bodies defined in Section 2.2.2) and ADR capture methods. Therefore, at least in its current version, it does not cover the domain of de-orbit technologies, although it is formulated with that domain in mind.

The scope of the ontology is to create a standardized framework for collection, storage and sharing of characteristics of IDOs for ADR, by leveraging attributes of a modern knowledge representation technique. This way not only a method for efficient storage of complex information is provided but also a platform for further analysis of the domain knowledge from the existing data. One such analysis consists of infer-

ring, for a specific object, the most suited ADR capture method(s) and provide a human-readable explanation of the inference.

The competency questions (CQs) of the ontology which identify its requirements are as follows:

- cq1** How could a domain knowledge about IDOs be captured in a standardized, formal, machine-interpretable way useful to ADR?
- cq2** What are the minimum parameters needed to characterize an IDO for an ADR capture phase?
- cq3** How can the degree of hazard of an IDO to an ADR capture phase be represented?
- cq4** How could the most suited ADR capture method be inferred?
- cq5** How could the input of data into the ontology be simplified and made compatible with an existing space debris catalog, such as DISCOS [117]?

The intended users of the ontology are space debris domain experts, ADR mission planners and decision makers, that should have at their disposal a standardized way for data collection, storage and access of complex domain knowledge, such as that of IDOs and ADR. By using such a framework the parameter space of each object is kept hidden from the user as much as possible while at the same time allowing queries and knowledge inference from the existing data via a semantic reasoner.

3.2.3 Main classes and class hierarchy

Classes are the main building blocks of an OWL2 ontology and their identification generally involves analyzing and extracting terms from the existing documentation, technical manuals, standards and/or similar ontologies [9, 126]. In this work this was done by extracting the most important terms from the preceding taxonomy of LEO space debris (see [15, 19]), whose purpose was similar, albeit more limited, due to the inherent limitations of the used knowledge representation method.

Classes identified as sufficient to characterize an IDO, unambiguously for a capture maneuver, and included within the developed ontology are: *Attitude_Regime*, *Onboard_Propellant*, *Space_Object_Type*, *Breakup_Criticality* and *ADR_Capture_Method*.

Additional classes, such as *Launch_Vehicle*, *Orbital_Regime*, *Spacecraft_Platform* and *Stage_Type*, are introduced for completeness, but are deemed nonessential for the purpose of this work. Inclusion of further classes although possible, was excluded in order to keep the ontology simple and clutter free.

The class hierarchy can then be established with a top-down, bottom-up or combined approach (middle-out) and in any case assuming a “kind of” relationship between a subclass and its superclass, i. e. “If a class A is a superclass of class B, then every instance of B is also an instance of A.” [9].

The hierarchy of classes in TRACER consists, as illustrated in Fig. 3.14, of four levels²¹, each specifying a domain concept into more detail. Starting from the top of Fig. 3.14, the `Attitude_Regime` class characterizes instances describing the attitude state of an object. It contains two additional layers specifying the exact type of attitude state of an object, which in this study is confined to either being stable or tumbling. The former is defined as an angular state of an object having the norm (i. e. magnitude) of the angular velocity vector equal to zero deg s^{-1} i. e. $\|\omega_t\| = \omega_t = 0 \text{ deg s}^{-1}$. The latter is instead further detailed into fast, medium and slow tumbling regimes each defined as the angular rate of an object being: $0 < \omega_t < 5 \text{ deg s}^{-1}$, $5 \leq \omega_t < 18 \text{ deg s}^{-1}$, $18 \leq \omega_t < \infty \text{ deg s}^{-1}$, respectively.

An example definition of the `Medium_Tumbling_Regime` class in the Manchester OWL Syntax²² used in TRACER is illustrated in Listing 3.1 and is to be interpreted as follows. An instance will be classified as a type of `Medium_Tumbling_Regime`, if it is a type of `Attitude_Regime` and has an existential restriction some along the functional data property `has_angular_rate` having a value within the $[5.0, 18.0) \text{ deg s}^{-1}$ interval.

The `Onboard_Propellant` class defines the propellant type of the main propulsion system of an IDO, or better of its platform (i. e. bus or propulsion unit, in case of a payload or rocket body, respectively). Based on the state of matter of the oxidizer and fuel used by the propulsion system, a distinction is made between `Liquid_`, `Solid_` and `Hybrid_Propellant` [122]. Two additional classes are added for completeness, i. e. `No_Propellant` and `Other_Propellant`, in case of spacecraft having no propulsion system or an onboard propellant that cannot be classified within one of the previous classes (e. g. butane, xenon, etc.). Lastly, the `Liquid_Propellant` class, is further subdivided into `Cryogenic_`, `Hypergolic_` and `Petroleum_Propellant` subclasses, based on the specific oxidizer-fuel mixture used by the propulsion system, as defined in [122].

The type of IDO is specified with the `Space_Object_Type` class, which is further subdivided in the two most relevant object types

²¹ Excluding the all encompassing class `owl:Thing` and the “housekeeping” classes: `Domain_Thing` and `Value_Partition`.

²² See [127] for more details.

Listing 3.1: Definition of the `Medium_Tumbling_Regime` class using Manchester OWL Syntax.

```
Attitude_Regime
and (has_angular_rate some xsd:float[>= 5.0f, < 18.0f])
```

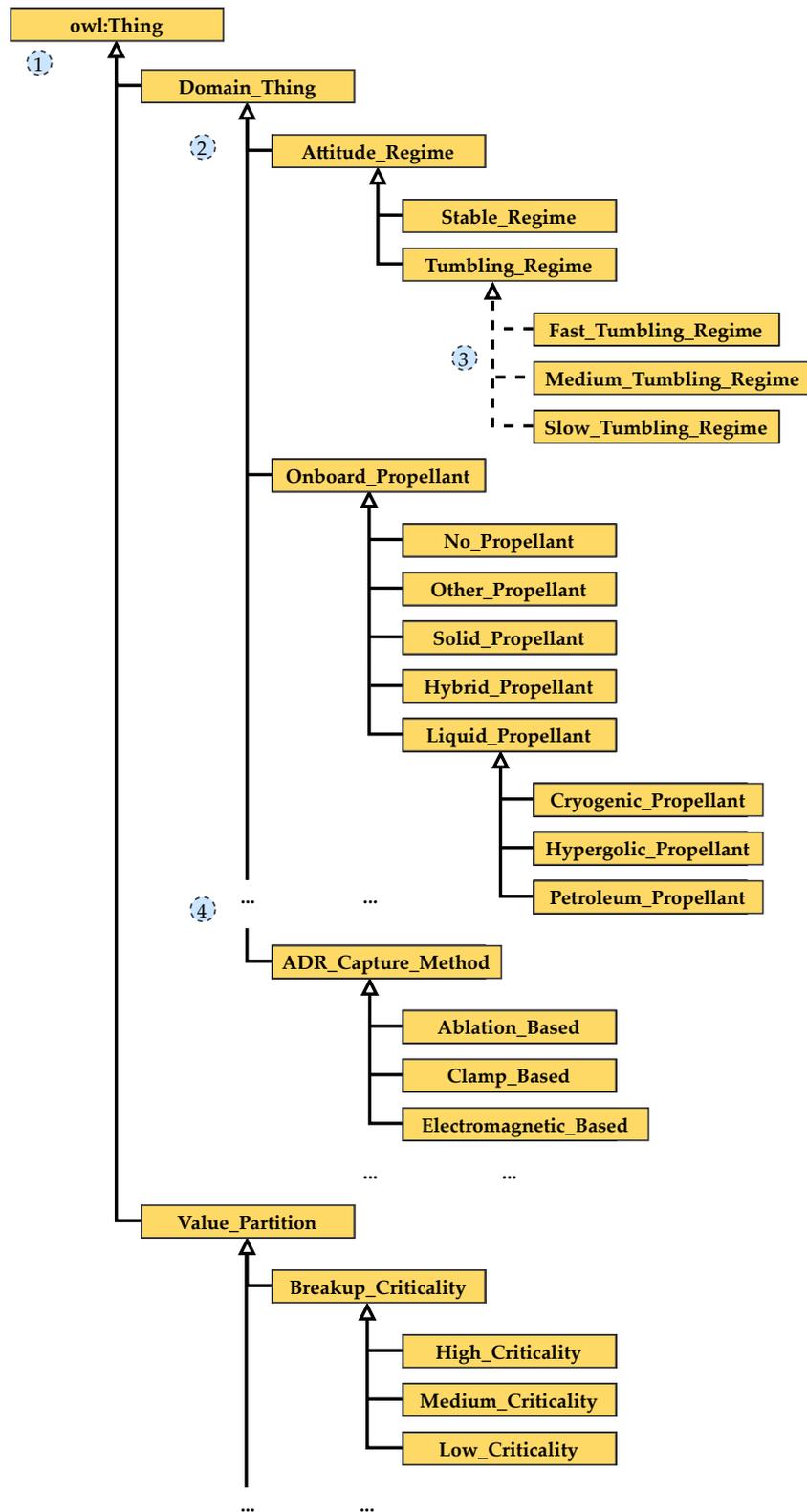


Figure 3.14: Extract from class hierarchy of TRACER. The conventions used in the diagram are the following: (1) yellow squares represent classes, (2) closed hollow arrows portray subclass relations, (3) dashed arrows portray inferred axioms, (4) three dots indicate an omission of the existing structure (adapted from [12]).

Table 3.1: Severity levels definition (adapted from [128, Table 5-1]).

Severity		
Category	Consequences description	Number
Catastrophic	Loss of system Severe environmental effects	4
Critical	Major damage to flight systems Major environmental effects	3
Major	Major mission degradation	2
Negligible	Minor mission degradation	1

identified in this work, i.e. `Payload` and `Rocket_Body`, defined as in Section 2.1.

The remaining main classes, i.e. `Breakup_Criticality` and `ADR_Capture_Method`, represent the core of the developed ontology, as they are instrumental in the ADR capture method inference.

The `Breakup_Criticality` class identifies the breakup risk of an object, due to its inherent probability of breakup, as a consequence of variety of causes, defined in Section 3.1.

The `ADR_Capture_Method` class instead defines the suitability of an ADR capture technology, described in Section 2.2.2, to capture an object based on its breakup criticality and level of uncooperativeness. The latter term is defined in this research as the degree of difficulty to capture a particular object due to its dynamical and physical properties.

3.2.4 Breakup criticality

In this work, the breakup criticality of an object is formalized in alignment with the ESA's standard on Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA) (ECSS-Q-ST-30-02C standard) [128]. Therefore, it is defined as a product of a severity and probability of occurrence of a fragmentation event of an object, i.e. of the associated severity number (SN) and probability number (PN) defined in Tables 3.1 and 3.2, having the worst possible consequences for an ADR capture maneuver. Consequently, a distinction is made not only between different object classes, i.e. rocket bodies and payloads, but also between non-passivated and passivated objects.

The identification of the worst possible fragmentation events and evaluation of SNs and PNs is performed based on the results of the statistical analysis of fragmentation events illustrated in Section 3.1.

Table 3.2: Probability levels definition (adapted from [128, Table 5-2]).

Probability		
Level	Limit	Number
Probable	$P > 10^{-1}$	4
Occasional	$10^{-2} < P \leq 10^{-1}$	3
Remote	$10^{-4} < P \leq 10^{-2}$	2
Extremely remote	$P \leq 10^{-4}$	1

The worst possible breakup event of rocket bodies is identified as either an anomalous or propulsion-related event, depending on whether the object is passivated or not. The exclusion of unknown events as a cause of breakups of passivated rocket bodies is due to the fact that the related breakup hazard estimate shown in Fig. 3.6 exhibits a high infant mortality rate, typical of active systems not inactive ones.

The severity of the two identified types of events, distinguished by the onboard propellant type, is listed in Table 3.3. In it the “No. of fragments” column details the observed median number of fragments per breakup event. The threshold of 1.05 years in the “Parent objects” column indicates the orbit age of an object after which a propulsion related event can be expected to be statistically less severe, possibly due to a depletion/venting of most of the stored on-board propellant (see Section 3.1.3). Therefore, the catastrophic severity (i. e. SN₄) is assigned to propulsion events of parent objects having an orbit age lower or equal to 1.05 years, irrespective of their onboard propellant type. The negligible severity instead is assigned to propulsion events of non-passivated parent objects using petroleum or solid propellant types. The same severity number is assigned to the anomalous class of events of passivated rocket bodies as they should not be able to cause propulsion events.

The probability of occurrence of those events is estimated using the Kaplan-Meier estimator expressed in Eq. (3.10) and the relation between $R(t)$ and $F(t)$ written in Eq. (3.3). This way it is possible to assert that the estimated probability of failure $\hat{F}(t)$ of a rocket body due to a breakup event at $t = 5$ yr is equal to 1.86×10^{-2} , which corresponds according to Table 3.3 to an occasional probability or PN₃. In case of a passivated rocket body that instead corresponds to $\hat{F}(t) = 3.04 \times 10^{-3}$ or a remote probability, i. e. PN₂.

In case of payloads, the worst possible breakup event of non-passivated spacecraft is found to be a combination of anomalous, electrical and unknown events (see Section 3.1.4). The combination of anomalous and collision events is instead taken into account in case of passivated spacecraft. The severities of those events are both

Table 3.3: Severity numbers of worst breakup events of rocket bodies in terms of the median number of fragments (RBs = rocket bodies, N/A = not applicable) (reprinted from [12]).

SN	Breakup class	Propellant type	No. of fragments	Parent objects
4	Propulsion	All	≥ 200	Non-passivated RBs with orbit age ≤ 1.05 years
3	Propulsion	Hypergolic	79.5	Non-passivated RBs with orbit age > 1.05 years
2	Propulsion	Cryogenic	7.5	
1	Propulsion	Petroleum & Solid	2.5	
1	Anomalous	N/A	1	Passivated RBs

Table 3.4: Severity numbers of worst breakup events of payloads in terms of the median number of fragments (PLs = payloads).

SN	Breakup class	No. of fragments	Parent objects
2	Anomalous +	8	Non-passivated PLs
	Electrical +		
	Unknown		
2	Anomalous + Collision	8.5	Passivated PLs

assigned the major severity level, i. e. SN = 2, due to their similar median number of generated fragments (see Table 3.4).

The probability of occurrence of mentioned events is estimated, as in case of rocket bodies, using the Kaplan-Meier estimator. However, since in both passivated and non-passivated cases the dominant probability function from the fifth year is that of anomalous events (see Fig. 3.9), no significant difference was found between the derived probabilities at that point in time. In fact, the estimated probability of a non-passivated and passivated payload experiencing a breakup even at $t = 5$ yr is equal to 8.66×10^{-3} and 4.42×10^{-3} , respectively, corresponding in both cases to a remote probability or PN2.

Using the derived severity and probability numbers of the worst possible fragmentation events of identified object classes, it is possible,

Table 3.5: Criticality matrix (adapted from [128, Table 5-3] and reprinted from [12]).

Severity Level	Severity Number (SN)	Probability Limits			
		$\leq 10^{-4}$	$\leq 10^{-2}$	$\leq 10^{-1}$	$> 10^{-1}$
		Probability Number (PN)			
		1	2	3	4
Catastrophic	4	4	8	12	16
Critical	3	3	6	9	12
Major	2	2	4	6	8
Negligible	1	1	2	3	4

as mentioned at the beginning of the subsection, to calculate their related criticality numbers and identify those that are critical. In accordance with [128], in this work an item is considered as critical for capture when one of the following conditions is fulfilled:

- the severity of its worst fragmentation event is defined as catastrophic, i. e. its SN = 4,
- the criticality number of its worst fragmentation event is greater or equal to eight, i. e. its CN ≥ 8 , in conformance with the criticality matrix defined in Table 3.5.

In these cases, any close contact with a target is to be avoided and only methods capable of achieving a capture from a significant stand-off distance (e. g. ≥ 50 m) should be considered. Moreover, in these cases special care should be exerted during the capture and stabilization maneuvers to avoid shocks and sources of sparks that might trigger a catastrophic breakup.

The remaining criticality levels are defined in this work as:

MEDIUM if the criticality number of its worst fragmentation event is equal to six, i. e. its CN = 6.

LOW if the severity level of its worst fragmentation event is considered negligible, or its criticality number is lower or equal to four, i. e. its CN ≤ 4 .

In these cases, more capture methods could be employed, depending not only on the CN but also on physical and dynamical properties of an object, such as its type, attitude regime, etc.

The Breakup_Criticality class in TRACER is therefore identified as a superclass of the High_Criticality, Medium_Criticality and

Low_Criticality classes (see Fig. 3.14), each defined using the previously delineated criticality levels. An example definition of the High_Criticality class in Manchester OWL Syntax is illustrated in Listing 3.2 which formally declares the previously identified conditions for an object to be critical. In other words, if an individual is a member of the class Breakup_Criticality and has a breakup severity that is a member of a class SN4_Catastrophic_Severity or has a breakup severity and probability individuals that are members of respective severity and probability classes such that the product of their numbers is greater or equal to eight, then it must be a member of the High_Criticality class.

3.2.5 ADR capture method

In this work, the selection of the most suited ADR capture method(s) for a specific target is determined via the evaluation of its previously defined breakup risk and degree of uncooperativeness. This way, not only the safety of the capture maneuver is considered during the evaluation, but also its degree of difficulty, which is manifested by the physical and dynamical properties of an object.

The characteristics of an object used to define its degree of uncooperativeness are: a) angular rate, b) existence of a dedicated grapple feature, c) type of material of the capture interface, d) mechanical clearance of the capture interface. The definition of those characteristics is summarized in Table 3.6 and detailed hereafter based on the perceived capabilities of current state-of-the-art capture methods.

The *angular rate* is expressed in the ontology as a data property and describes the attitude state of an object. The latter is identified as being either stable or tumbling, in accordance with what is outlined in Section 3.2.3. The thresholds defining the tumbling states, visible in Table 3.6, are chosen based on: a) maximum value of the relative angular rate that a state-of-the-art robotic manipulator should be able cope with (i. e. 5 deg s^{-1}) [42, 129], b) value of the relative angular rate above which any synchronization effort would be considered difficult

Listing 3.2: Definition of the High_Criticality class using Manchester OWL Syntax.

```

Breakup_Criticality
and (is_breakup_criticality_of some ((has_breakup_probability
    some PN4_Probable_Probability) and (has_breakup_severity some
    SN2_Major_Severity))
or ((has_breakup_probability some (PN3_Occasional_Probability or
    PN4_Probable_Probability)) and (has_breakup_severity some
    SN3_Critical_Severity))
or (has_breakup_severity some SN4_Catastrophic_Severity))

```

Table 3.6: Target characteristics defining levels of uncooperativeness ($\omega_t =$ angular rate of a target, $A =$ area centered on the capture interface of a target) (reprinted from [12]).

Characteristic	Definition
Angular rate	Stable: $\omega_t = 0 \text{ deg s}^{-1}$ Slow tumbling: $0 < \omega_t < 5 \text{ deg s}^{-1}$ Medium tumbling: $5 \leq \omega_t < 18 \text{ deg s}^{-1}$ Fast tumbling: $18 \leq \omega_t < \infty \text{ deg s}^{-1}$
Grapple feature existence	True: " <i>dedicated</i> " grapple feature exists False: " <i>dedicated</i> " grapple feature does not exist
Capture interface material	Isotropic: <i>e. g. metal, ceramics, polymer</i> Anisotropic: <i>e. g. composite materials</i>
Capture interface clearance	Narrow: $A < 0.28 \text{ m}^2$ Broad: $A \geq 0.28 \text{ m}^2$

(i. e. 18 deg s^{-1}) [7]. Consequently, objects having angular rates greater or equal to 18 deg s^{-1} are assigned a high degree of uncooperativeness and should be captured only with contactless "capture" methods.

The *grapple feature existence* is defined as a Boolean data type property. A "*dedicated*" grapple feature is identified in this work as a surface feature, with a regular enough geometry (e. g. an LAR, common on many spacecraft), that can be easily grappled. Otherwise, a capture has necessarily to be performed on some other feature, not envisioned to be grappled, or even a surface. Therefore, the existence of a grapple feature is considered advantageous for the capture maneuver since it can be approximated with a more common berthing operation, commonly used in the context of loading/unloading of cargo from the ISS.

The *capture interface material* is defined as a string data type property and reflects the versatility (and reliability) of a capture method. Considering the directional dependent mechanical properties of an anisotropic type of material, it is associated with capture methods capable of distributing applied contact forces (such as clamp or net-based methods). An isotropic material is, on the other hand, expected to be able to withstand concentrated loads, independent of their direction, which is why it is instead correlated with capture methods that are anticipated to exert such forces (e. g. manipulators or harpoon-based methods).

The *capture interface clearance* is expressed as a string data type. It reflects the overall complexity of the approach and capture operations and is defined as an area, A , enclosed by a circle centered on the capture interface. The threshold value of 0.28 m^2 is defined using the combination of ESA's recommendations on the mechanical clearance

of mechanisms [130] and the value of a maximum achievable precision of a typical GNC system, in all three axis, during a berthing maneuver (i. e. 0.1 m), using the formula $A = \pi \times (3 \times 0.1)^2$ [34]. The smaller the interface clearance, the more precise the capture maneuver needs to be. Therefore, the smaller the interface clearance of a target, the higher its degree of uncooperativeness.

Using the defined traits, the most cooperative targets are identified as those having: stable to low tumbling attitude regimes and a “dedicated” grapple feature. The most uncooperative targets are defined instead as those having fast tumbling attitude regimes, irrespective of other features.

The ADR capture method(s) selection for a particular target, i. e. the classification of the associated ADR capture method(s) instance(s), is performed by applying the defined characteristics to the axioms of the ADR_Capture_Method subclasses as detailed in Table 3.7.

An example definition of the Manipulator_Based class in Manchester OWL Syntax using the axioms defined in Table 3.7 is illustrated in Listing 3.3. From the outlined axioms it can be deduced that *manipulator-based methods* are designated as a preferred solution for capturing objects (irrespective of their type) having: a low breakup criticality, stable to medium tumbling attitude and “dedicated” grappling feature. A non-existing grappling feature would make any usage of a manipulator-based capture method more complex and less safe. Therefore, in these cases the most suited capture methods are either *clamp- or tether-based*, selected on the basis of object type, capture interface material and required clearance.

Clamp-based solutions are deemed suitable only for rocket bodies, due to their expected lack of appendages that would otherwise complicate capture maneuvers.

Harpoon-based solutions, in contrast, are associated with payloads, due to the expected higher efficacy of these methods on flat surfaces rather than on the curved ones, commonly found on rocket bodies.

Finally, *contactless solutions* are coupled exclusively with objects having high levels of uncooperativeness, since any capture effort using the previously mentioned methods would be difficult and expensive

Listing 3.3: Definition of the Manipulator_Based class using Manchester OWL Syntax.

```
ADR_Capture_Method
and (is_capture_method_of some (has_attitude some (not (
  Fast_Tumbling_Regime))))
and (is_capture_method_of some (has_breakup_criticality some
  Low_Criticality))
and (is_capture_method_of some (has_platform_type some (
  has_grapple_feature value true)))
```

Table 3.7: Class axioms used for the classification of ADR capture methods (N/A = not applicable, RB = rocket body, PL = payload) (reprinted from [12]).

Capture method class	Object type	Breakup criticality	Passiv. state	Attitude regime	Grapple feature	Capture interface	
						Material	Clearance
Manipulator_Based	N/A	Low	N/A	Stable-Medium tumbling	True	N/A	N/A
Clamp_Based	RB	Low	N/A	Stable-Medium tumbling	False	N/A	Broad
Net_Based	N/A	Low- Medium	N/A	Stable-Medium tumbling	N/A	N/A	N/A
Harpoon_Based	PL	N/A	True	Stable-Slow tumbling	False	Isotropic	N/A
Plume_Impingement	PL	Low- Medium	N/A	Fast tumbling	N/A	N/A	N/A
Electromagnetic_Based	RB	Low- Medium	N/A	Fast tumbling	N/A	N/A	N/A
Ablation_Based	PL	N/A	True	Fast tumbling	N/A	N/A	N/A
No_Solution	N/A	High	False	Fast tumbling	N/A	N/A	N/A

(in terms of propellant). The selection between different contactless methods is made on the basis of the estimated breakup criticality of objects, i. e. on the “required” stand-off distance that each method requires to achieve a successful “capture” maneuver. Furthermore, *plume impingement* and *ablation-based methods* are deemed more suitable for payloads, considering that the efficacy of both is maximized on flat surfaces rather than on the curved ones [40]. However, payloads usually present a small percentage of conductive material, with respect to their overall mass, making them challenging targets for electromagnetic-based methods [40]. As a consequence, *electromagnetic-based methods* are bound, in this work, to rocket bodies, usually containing a large percentage of conducting material with respect to their overall mass.

No solution is found suitable for targets having: a high breakup criticality, fast tumbling attitude regime and non-passivated state. The reason behind this result arises from the current unavailability of ADR capture methods that could safely tackle targets having those characteristics.

3.3 ONTOLOGY DEVELOPMENT

The domain-ontology developed within this research is implemented within the TRACER software library. Its development workflow is depicted in Fig. 3.15 and is divided into two main processes: database generation and software implementation. The former includes collection and pre-processing of raw structured and unstructured data of individuals of the ontology, using the Python programming language [131]. The latter consists instead of activities leveraging the output of previous tasks to implement the methodology, described in Section 3.2, employing the ontology editor Protégé Desktop [132]. The workflow also includes tasks to import the desired data into the ontology, using the *Owlready2* Python module [133] and perform unitary tests to assure ontology consistency.

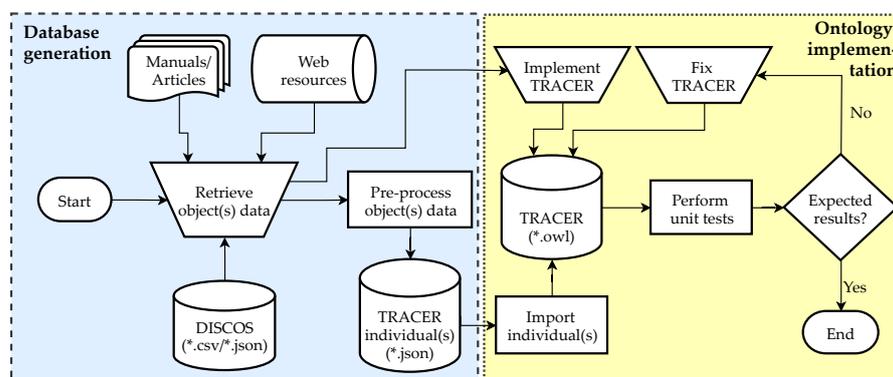


Figure 3.15: Development workflow of TRACER software library (refer to the ISO 5807:1985 standard [134] for the definition of symbols).

3.3.1 Database generation and import

The first step in the database generation process consists of collection and extraction of needed input data. The data sources considered in TRACER are both, structured and unstructured, due to the current nonexistence of a single data source containing all the information required for ADR capture methods selection, defined in Section 3.2.5. The ESA's DISCOS database comes very close with respect to the required amount of data and provides a machine-to-machine interface the DISCOSweb API, through which retrieval of structured object data is possible. For this reason, TRACER is developed to "interface" with the output of the DISCOSweb API, although a manual query to the maintainers of the database is currently required due to the existing access restrictions and limited default information available through the API itself.

The structured data expected by TRACER consists of the following target characteristics: a) identification properties, b) physical dimension, c) orbital properties, d) launch and reentry dates (if any), e) activity status, f) onboard propellant type, g) launcher name and country of origin.

The unstructured data about the target expected by TRACER consists instead of: a) attitude state, b) passivation state, c) onboard fuel and oxidizer types, d) launch and reentry dates (if any), e) spacecraft bus or propulsion unit type, f) grappling feature existence, g) potential capture interface properties. Sources of such data might consist of web resources, such as: Encyclopedia Astronautica [135], Gunter's Space Page [136], Earth Observation Portal [137], RussianSpaceWeb.com [138]; user's manuals/guides of launchers, such as Ariane 5 [139], Atlas V [140]; academic publications in the fields of space debris monitoring and modeling [141–144].

Once the required data is extracted from the chosen sources, the pre-processing step of the workflow involves integrating that data and creating a database of individuals compatible with the developed ontology. The integration of data is done via a custom script written in Python 3 (version 3.6.9) programming language. The script leverages the Python's pandas [145] and numpy [146] libraries (versions 0.25.2 and 1.16.5, respectively) to import, merge, format and manipulate the extracted data accordingly. For the computation of breakup probabilities and their storage, the Python libraries `lifelines` [147] and `dill` [148] (versions 0.24.5 and 0.3.1.1, respectively) are used within the mentioned script. The `lifelines` module, built on top of the pandas library, implements in Python the survival analysis, described in Section 3.1, and provides an API to effectively estimate the probability of an event based on historical data. `dill` on the other hand allows serialization of a model developed with `lifelines` and its storage in a file so that it can be used without any previous computation of a serialized model.

The import of the compiled data into the domain-ontology is performed using a method for the manipulation of semantic models, adapted for TRACER and developed within the D-Rock [149] and Q-Rock [150] projects of the Robotics Innovation Center – DFKI GmbH. The method allows transparent manipulation of ontologies using the Python 3 programming language and an ontology programming interfaces, *Owlready2* (version 0.21) [133], while minimizing the required user input and therefore the possibility of human error. The *Owlready2* module allows loading of OWL2 ontologies as Python objects, manipulating them, performing reasoning on them via *Hermit* [151] or *Pellet* [152] semantic reasoners, and finally saving them. It uses similarities between object models and ontologies to enable high level access to OWL2 ontologies via Python notation, therefore allowing an easy-to-use and highly concise syntax.

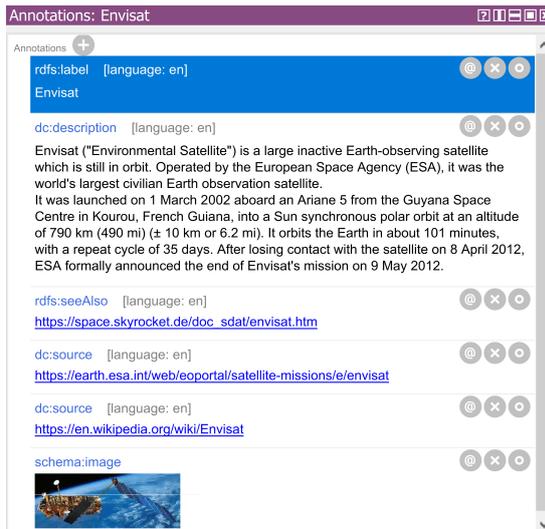
3.3.2 Ontology implementation

The software implementation of TRACER consists in using the open-source framework *Protégé* which provides users with a suite of tools to develop, edit and manage domain models. It fully supports the Semantic Web standards of the W3C, such as the Resource Description Framework (RDF) and OWL2 [124, 132]. It is supported by a strong community of users and large documentation with an aim to help novices as well as experienced users. The *Protégé Desktop* in particular, is a desktop version of the *Protégé* framework, allowing creation, editing and refactoring of ontologies via a highly customizable GUI. It allows a connection to description logic reasoners (i. e. semantic reasoners) that might be used to probe an ontology for inconsistencies as well as to infer new knowledge. Finally, the program provides an option to display human-readable inference explanations, useful to verify justifications of inferred classifications [153]. In view of these considerations, the *Protégé Desktop* (version 5.5.0) is selected in this research as the tool of choice for the implementation, testing and visualization of results of TRACER.

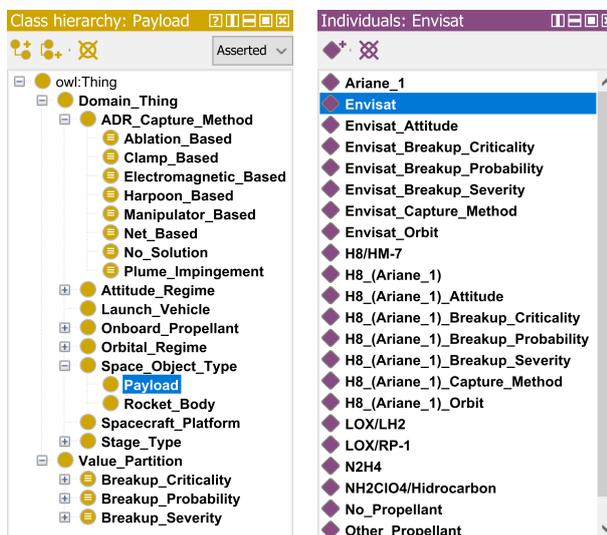
The particularities of the TRACER implementation in *Protégé Desktop* (named from now on simply *Protégé*) consist of: a) minimal hierarchy of classes, b) usage of ontology design pattern (ODP), c) differentiation between space debris objects and key characteristics, in terms of individuals.

The statistical metrics of the developed ontology are displayed in Table 3.8 while an illustration of its implementation within *Protégé* is illustrated in Fig. 3.16, representing several views available within the editor, i. e. the Annotations, Class hierarchy, Instances and Property assertions views.

The Annotations view, visible in Fig. 3.16a, displays the annotations and descriptions of the example instance of TRACER (e. g. 2002-009A/-

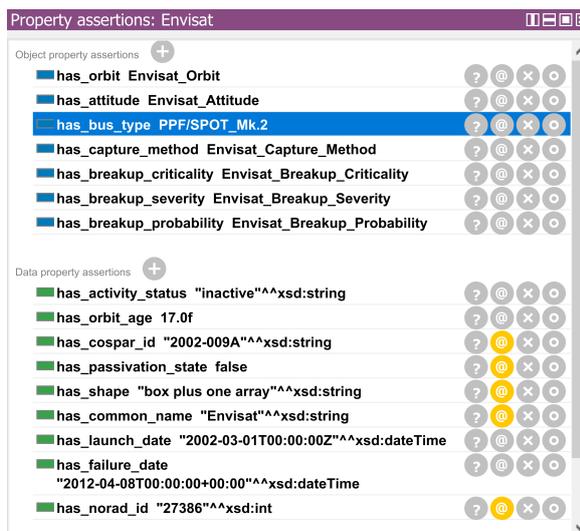


(a) Annotations view.



(b) Class hierarchy view.

(c) Instances view.



(d) Property assertions view.

Figure 3.16: TRACER within Protégé Desktop.

Table 3.8: TRACER statistical metrics (refer to the Protégé documentation [154] for the description of individual parameters).

Parameter	Count
Axiom	521
Logical axiom	247
Declaration axiom	150
Class	56
Object property	34
Data property	42
Annotation property	23

Envisat object). The view highlights the flexibility of an ontology to include additional information about an object, such as a label, description, source, etc., that could enrich the knowledge of the object enclosed within the semantic relations, a feature useful to document modeling decisions and/or refer to sources of information.

The Class hierarchy view, illustrated in Fig. 3.16b, displays the asserted hierarchy of the ontology (similarly to Fig. 3.14), structured as described in Section 3.2.3. The visible minimal classes hierarchy is intentional in order to reduce the user input and leverage instead the capability of a semantic reasoner to classify the ontology autonomously (i. e. compute the class hierarchy from the imposed axioms). Additionally, the structure reflects a design choice to group the ontology classes into either the *Domain_Thing* or *Value_Partition*, “housekeeping” superclasses, implemented to separate the ordinary classes from the “ODP classes”.

The ODP used in TRACER is the *Value Partition*. This ODP is considered as a “good practice” in ontology development since its usage allows more robust, cleaner and easier to maintain ontologies [155]. The pattern addresses the problem of representing a descriptive feature of an object with a constrained set of possible values (also known as feature space) [155, 156]. An example in case of TRACER can be the “breakup criticality” property of a space debris object, which can be regarded as a feature to be represented and its values “high”, “medium” and “low” as its feature space. This way attributes are separated from elements being described, enforcing an easier to maintain modeling [155].

The Instances view, depicted in Fig. 3.16c, illustrates the design choice to differentiate between a space debris object itself and its attributes, similarly to what was done in the existing state-of-the-art research [61, 62, 64]. The relationship with a parent object is guar-

anteed via object properties, visible in the Property assertions view (see Fig. 3.16d). In this manner, six additional individuals/instances are created for each space debris object describing its: a) attitude, b) orbit, c) breakup criticality, d) probability, e) severity, f) capture method. Two further individuals are used to represent the platform type and onboard propellant of an object, allowing their reuse in case of objects having the same platform and/or onboard propellant. Consequently, a total of eight individuals are used in TRACER to complement the data properties expressed within the space debris object individual.

The advantages of the described software implementation, especially when used in combination with Protégé, comprise of an easy to use and maintain ontology in which the characteristics of an object are represented as data properties while the contextual information and relationship with other individuals as object properties. Furthermore, the deductive reasoning capabilities of an ontology can be extended, if required, via the Semantic Web Rule Language (SWRL) [157], for which Protégé provides a development environment. This way, the `orbit_age` of an object, for example, can be deduced by a reasoner based on the defined SWRL rules, instead of being derived in the pre-processing step. However, the provided SWRL development environment was found to be very limited in terms of debugging capabilities and was therefore dismissed in favor of a more familiar Python environment.

The disadvantages of the described software implementation are mainly tied to “limitations” of the Protégé itself which requires a quite pedantic and repetitive insertion of class restrictions, individuals and properties, without the possibility to automatize the process within the editor itself. For this reason, the `owlready2` is used to populate TRACER with the required data and might be used in the future to manage the developed ontology in more depth, using a more familiar Python environment.

3.4 ONTOLOGY APPLICATION

The effectiveness of the developed ontology was tested by applying it onto a database of representative objects. The employed application workflow is illustrated in Fig. 3.17 and consists of two main processes: the data input and ontology query. The former encompasses the collection, pre-processing and input of desired data into TRACER, while the latter involves the knowledge inference from TRACER, via a software reasoner, and query of results via a GUI.

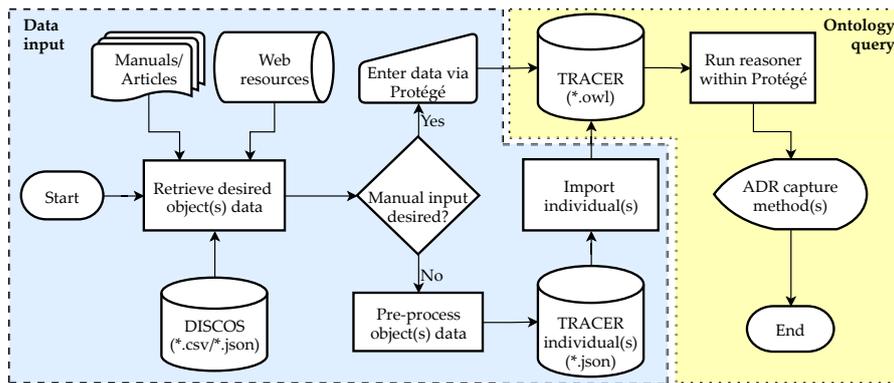


Figure 3.17: Usage workflow of TRACER software library (refer to the ISO 5807:1985 standard [134] for the definition of symbols) (reprinted from [12]).

3.4.1 Example database

The example database created for the evaluation of TRACER, consists of 240 ontology individuals, related to 30 large intact cataloged objects (19 payloads and 11 rocket bodies), for which information about their attitude states was available from publicly available resources. The database was assembled using the available data from DISCOS and the unstructured data sources mentioned in Section 3.3.1. Objects having identical properties (bus/propulsion platform and angular rate) were excluded from the database. Only one of the considered objects had at the time of generation of the database²³ an active status and its COSPAR ID is 2014-037A. The rest, either had an unknown status or could be considered inactive, being deactivated on purpose or having experienced a catastrophic failure which forced their early end-of-life disposal. The characteristics of the first 10 objects of the database are illustrated in Table 3.9.

²³ 7 July 2019.

The most represented platforms within the database are the: Uragan Block IIv bus (with five related objects), H10 propulsion unit (with two related objects), Étage à Propergols Stockables (i. e. Storable Propellant Stage) (EPS) L9 propulsion unit (with two related objects) and “ADEOS” bus (with two related objects).

The distribution of objects within the orbital classes, as defined in DISCOS, is the following: 15 in LEO, eight in medium Earth orbit (MEO), four in GEO transfer orbit (GTO), two in highly eccentric Earth orbit (HEO) and one in GEO.

The median in-orbit age²⁴ of objects was found to be 22.89 and 19.3 years (with interpercentile ranges being 13.7 to 25.51 and 17.55 to 25.89 years) for payloads and rocket bodies, respectively. The distribution of their breakup probabilities, categorized by their passivation state, is illustrated in Fig. 3.18.

²⁴ Since launch and till 7 July 2019.

The assumptions made during the consolidation of the database were the following:

Table 3.9: Characteristics of first 10 objects of the example database (RB = rocket body, PL = payload, LEO = low Earth orbit, MEO = medium Earth orbit, GTO = geostationary Earth transfer orbit) (reprinted from [12]).

COSPAR ID	Type	Orbit	CN	Passiv. state	Angular rate (deg s ⁻¹)	Orbit age (years)
1978-018B	RB	LEO	3	False	67.8	41.39
1978-121A	PL	LEO	6	False	2	40.54
1989-001B	PL	MEO	6	False	38.88	30.49
1990-005H	RB	LEO	3	True	0	29.46
1990-045A	PL	MEO	6	False	38.96	29.14
1991-084C	RB	GTO	6	False	1.74	27.56
1992-052A	PL	LEO	6	True	32.1	26.9
1993-061A	PL	LEO	6	False	2	25.78
1994-021A	PL	MEO	6	False	3.03	25.24
1994-021B	PL	MEO	6	False	8.41	25.24

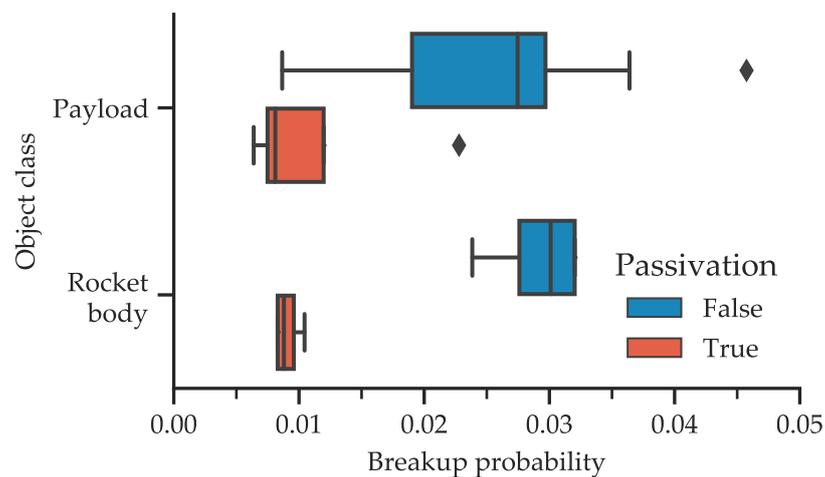


Figure 3.18: Box plot of breakup probabilities of example objects per object class and passivation state.

Table 3.10: ADR capture methods classification results per capture method class (all numeric values in the table are median values except those within the “No. of individuals” column; CN = criticality number, PN = probability number, SN = severity number) (reprinted from [12]).

ADR capture method class	No. of individuals	CN	PN	SN	Angular rate (deg s ⁻¹)	Orbit age (years)
Ablation_Based	1	6	3	2	32.1	26.9
Electromagnetic_Based	1	3	3	1	67.8	41.39
Manipulator_Based	11	3	2	1	2	17.35
Net_Based	26	6	3	2	2	18.57
Plume_Impingement	3	6	3	2	38.88	29.14

- Passivated state of an object was assumed as True only where the documentation supporting such a state was found, e. g. existence of a deactivation date, flight numbers of Ariane launches, etc.
- Grapple feature existence was assumed as True and its material isotropic for all objects within the database, considering that they should all be fitted with either an LAR or a launch vehicle adapter (LVA) (or in some cases even with an Ariane structure for auxiliary payloads (ASAP)). Those interfaces, while might be made of a composite material (especially true for LVAs), should all present metallic junction surfaces, considered to be suitable as potential grapple features.
- Failure date and type recorded within the database refer to a catastrophic type of failure of an object which would force its premature shutdown, if at all possible.

3.4.2 Classification results

The classification of the example database was performed using the Pellet semantic reasoner within the Protégé Desktop, on a 64-bit PC platform equipped with an Intel® Core™ i7-3630QM CPU, clocked at 2.40 GHz, and 16 GiB of RAM. The required average time for a classification was found to be 13.5 s. With lower numbers of individuals, e. g. 142 and 72 (corresponding to 20 and 10 cataloged objects, respectively), the average inference computation time was found to be 3.51 s and 1.25 s, respectively.

The results of the inferred classification of ADR capture methods are summarized in Table 3.10 as well as Figs. 3.19 to 3.21.

The distribution of the classifications results over the two considered classes of target objects, i. e. payloads and rocket bodies is represented in Fig. 3.19. 26 capture methods have been classified within the Net_ - Based class, 11 within the Manipulator_ Based class, three within the Plume_ Impingement class and one within the Electromagnetic_ Based and Ablation_ Based classes, respectively.

The distribution of the classification over the breakup criticality numbers and attitude regimes of related targets are illustrated in Fig. 3.20 and Fig. 3.21, respectively.

Overall 12 capture methods were simultaneously classified in two classes, suggesting that, under the considered assumptions, more than one capture method was found to be suitable for the associated targets. The two capture method pairs involved in the simultaneous classification were the following: (Manipulator_ Based, Net_ Based) and (Ablation_ Based, Plume_ Impingement). The number of occurrence of these results was 11 times and one time, respectively.

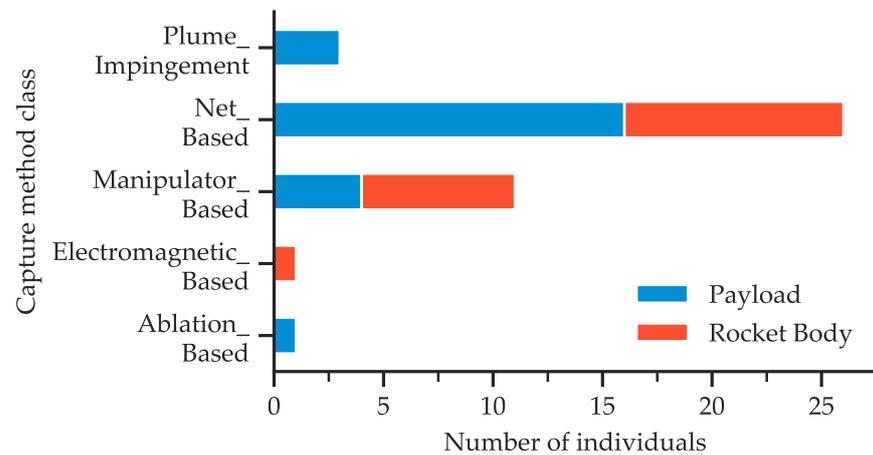


Figure 3.19: ADR capture methods classification results per target object class (adapted from [12]).

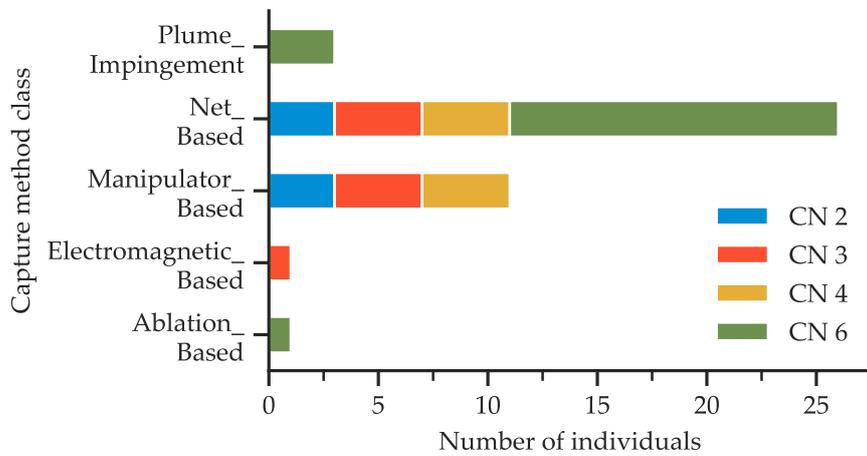


Figure 3.20: ADR capture methods classification results per target object criticality number (CN = criticality number) (adapted from [12]).

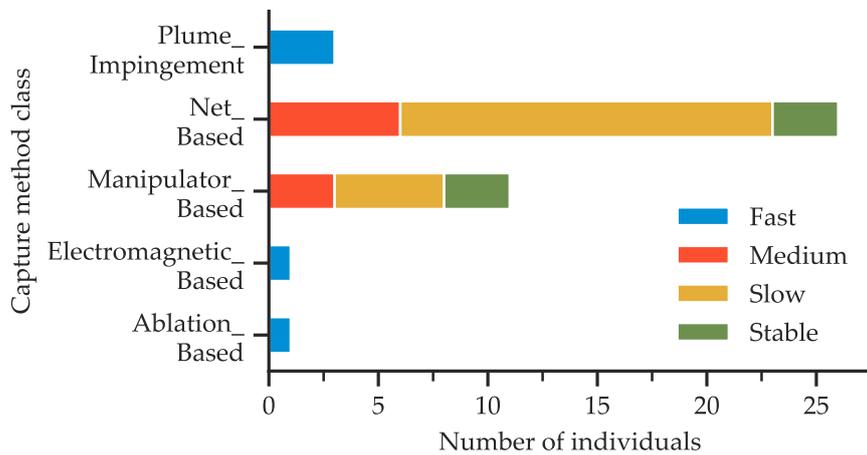


Figure 3.21: ADR capture methods classification results per target object attitude regime (adapted from [12]).

3.5 ANALYSIS AND DISCUSSION

This section analyses the results presented in Section 3.4 and discusses strengths and weaknesses of the developed ontology in light of the competency questions defined in Section 3.2.2.

3.5.1 Results analysis

The overall classification results (illustrated in Table 3.10) point toward the `Net_Based` class as the most numerous one, holding $\approx 87\%$ of all individuals of capture methods. The second most numerous class is the `Manipulator_Based`, holding a total of $\approx 37\%$ of individuals, while the remaining three classes are found to collect, in total, only $\approx 17\%$. These figures were expected and are to be attributed to the assumptions made during the creation of the example database (defined in Section 3.4.1) and restrictions imposed to the `ADR_Capture_Method` subclasses, (defined in Table 3.7). Indeed, the database is dominated by slow/medium tumbling objects, having low/medium breakup criticalities, which makes them ideal targets for contact-based capture methods, such as nets or manipulators. Furthermore, all targets were assumed to have a grapple feature that made them suitable for manipulator-based methods, especially in case of objects having low to medium tumbling rates and low breakup criticalities. Additionally, certain targets were associated with both net- and manipulator-based methods, as their capture methods satisfied more than one class membership constraint. This result is illustrated in the mirrored classification distribution of capture methods associated with targets having low CNs (i. e. $2 \leq \text{CN} \leq 4$) depicted in Fig. 3.20.

The classification distribution over target object classes, as depicted in Fig. 3.19, was also an anticipated result that can be justified by the overall higher number of payloads within the example database. In fact, the number of payloads within the database is almost double with respect to that of rocket bodies and this is reflected in the overall higher number of payloads being associated with the most generic capture method defined, the net-based. The manipulator-based method is instead associated with a higher number of target rocket bodies, due to the higher number of rocket bodies (seven) with an overall lower median CN with respect to that of payloads.

The association of the remaining classes (`Plume_Impingement`, `Electromagnetic_Based` and `Ablation_Based`) with one target class or the other was also an awaited outcome that is a direct consequence of class restrictions specified in Table 3.7.

The total absence of individuals within the `Clamp_Based` or `Harpoon_Based` classes reflects the very nature of the example database, which does not include rocket bodies or payloads with missing grapple features.

3.5.2 Discussion

The overall scope of this work was twofold: a) create a domain-ontology for data collection, storage and sharing of characteristics of IDOs, b) enable a discovery of new domain knowledge, such as the inference of most suited ADR capture methods. The presented results confirm the capability of TRACER to fulfill the mentioned objectives, within the constraints defined in Section 3.2.2. In fact, using TRACER a user can capture, in a standardized, formal and machine-interpretable way, the domain knowledge of IDOs (in particular of payloads and rocket bodies) useful to ADR. Additionally, the application of the ontology onto an example database displays how a knowledge representation method can be used to infer the most suited ADR capture methods, based on the existing data. The ontology also provides a way to represent the degree of hazard of an IDO to an ADR capture phase, due to its fragmentation potential and inability to support a capture maneuver. Finally, TRACER is provided with tools necessary to transparently handle the input data from an existing database of space debris into an ontology, a detail often overlooked, or at least not sufficiently documented, by the existing state-of-the-art methods.

The required inference time points towards an ontology with an exponential complexity, as demonstrated by the exponentially increasing computation time with the number of individuals. This suggests the limitation of the current implementation to deal with a classification of large number of space objects in one run. Indeed, in case of 100 space objects (which would roughly translate into 700 ontology individuals), the expected computation time would require around 90 h on the PC platform mentioned in Section 3.4.2. Nevertheless, the computation time can be drastically reduced by simply halving the number of objects to be classified in one run. In fact, considering 50 objects, or circa 360 individuals, to be classified within TRACER in one run, the expected computation time should be around 4 min. Hence, this limitation is not currently seen as a serious impediment of TRACER, especially knowing that the developed repository includes automation tools adequate to classify large number of IDOs in batches of 50 objects at a time, for example. Moreover, in case of large groups of objects to be analyzed, consisting mainly of spacecraft of the same platform/bus (such as in case of planned mega-constellations), objects might be further grouped by attitude regimes (as defined in Section 3.2.5) and the classification might be performed only on representative objects of each group, while extrapolating results for the rest. In this manner, the entire group can be analyzed without the need to process each individual.

Main limitations of the current version of TRACER consist of:

- its inability to classify all possible types of orbital regimes of parent objects,

- omission of shape, size and mass properties from characteristics used to determine the degree of uncooperativeness of a target (see Table 3.6),
- omission of a cost parameter within the class axioms used for the ADR capture methods classification (detailed in Table 3.7).

Further constraint of TRACER is its dependence from the unstructured data that are not always easily retrievable, e. g. attitude states of cataloged objects, their onboard propellant types, passivation states, etc. Therefore, further research in these areas is considered of paramount importance, towards a goal of either building a comprehensive database of objects and their properties, or developing a machine learning algorithm that could infer the required data from the existing ones.

Finally, TRACER in its current form does provide an answer only to one phase of an ADR mission, i. e. the capture phase. Other phases, such as the close-range rendezvous or disposal (e. g. de-orbit), have not been considered. Therefore, for TRACER to represent a comprehensive ADR planning solution, something that was out of the scope of the current work, all possible mission phases of a generic ADR mission should be included, e. g. as separate ontologies.

4

ROBOTIC CAPTURE OF AN UNCOOPERATIVE TARGET

Disclaimer on content reuse: this chapter is based on the following publications: [11, 16, 17]ⁱ.

Among the proposed ADR capture technologies, those involving space robotics are at present the most mature ones. Moreover, these technologies are among the most versatile ones given the high number of DOF they can generally control. However, no robotic spacecraft has ever performed an autonomous capture of a truly uncooperative, tumbling object. In addition, most of the existing studies on the matter have been confined to target objects with relatively low tumbling rates. In this context the second objective of this thesis is to develop a motion planning method for a spacecraft equipped with a robotic manipulator, based on optimal control, to facilitate the capture of an uncooperative, rapidly tumbling target.

The chapter begins with a brief overview of the mathematical modeling of a space robot. This is followed by an introduction to optimal control and more specifically to the formulation of an optimal control problem and ways to solve it numerically. The specific optimal control problem considered by the developed motion planner is formulated next as well as its software implementation. This leads to the description of the numerical simulations used to evaluate the developed planner. The chapter concludes with the analysis of results of the simulations and a discussion about the strengths as well as weaknesses of the developed motion planner in light of the illustrated results.

4.1 MATHEMATICAL MODELING OF A SPACE ROBOT

With respect to a ground-based robot, a space robot and especially a free-floating one (defined in Section 2.4.2) exhibits, during the usage of its manipulator, a dynamical coupling between the latter and its base due to the conservation of momentum. In fact, any capture maneuver performed with a space robot, even of a stationary target, is inherently more complex when compared to that of a ground-based robot, as the resulting attitude disturbance of the base needs to be accounted

ⁱ Some of the text in this chapter is reprinted/adapted by permission from Springer Nature Customer Service Centre GmbH: Springer International Publishing, *Stardust Final Conference. Astrophysics and Space Science Proceedings*. "Trajectory Generation Method for Robotic Free-Floating Capture of a Non-cooperative, Tumbling Target", DOI: 10/gf93ms, M. Jankovic and F. Kirchner. © 2018.

for in the planning and control algorithms to achieve a successful capture. This in turn implies that in case of a space robot to solve a capture maneuver appropriately the kinematic information alone is not sufficient and the evaluation of dynamic terms is needed [158], revealing an inherent coupling between the equations of motion.

This section introduces the kinematic and dynamic equation of a space robot and describes the main notation that is be used in the rest of the chapter.

4.1.1 Kinematic equations

A space manipulator, schematically depicted in Fig. 4.1, can be represented as an *open kinematic chain*, consisting of a series of n rigid bodies (i. e. *links*) connected by kinematic pairs, or *joints*, defining the DOF of the manipulator. One end of the open chain is connected to the spacecraft (*base*) while the other, named *end-effector*, is connected to the opposite end of the chain and allows a capture and manipulation of objects. The pose of the end-effector in Cartesian space is a result of a composition of poses of each link in the chain with respect to each other [159]. Differently from a ground-based robot, where the base is fixed and its pose does not influence that of the end-effector, in a space robot the base reacts to the motion of the end-effector and its pose needs to be accounted for in the equations of motion.

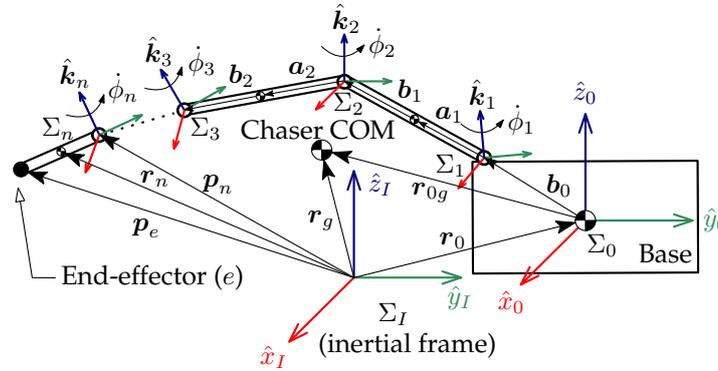


Figure 4.1: Schematic model of a single-arm space robot having $n + 1$ number of rigid links and n joints (COM = center of mass, Chaser = space robot, \hat{k}_i = rotation axis of joint i , $\dot{\phi}_i$ = angular velocity of joint i , a_i = vector pointing from joint i to the COM of link i , b_i = vector pointing from the COM of link i to joint $i + 1$, Σ_i = reference frame of joint i , p_e = end-effector position, p_i = position of joint i , r_i = position of the COM of link i , r_g = position of the COM of the chaser, r_{0g} = position of the COM of the chaser with respect to the base reference frame (Σ_0), r_0 = base position) (based on [158, Fig. 2.1]).

With a reference to Fig. 4.1, the position vector of the end-effector, $p_e \in \mathbb{R}^3$, can be expressed w. r. t. the inertial coordinate frame, Σ_I

(assumed to exist in the vicinity of the overall system (space robot plus target)) as [160]:

$$\mathbf{p}_e = \mathbf{r}_0 + \mathbf{b}_0 + \sum_{i=1}^n \mathbf{l}_i \quad (4.1)$$

where $\mathbf{r}_0 \in \mathbb{R}^3$ denotes the position of the center of mass (COM) of the base spacecraft²⁵. $\mathbf{l}_i = \mathbf{a}_i + \mathbf{b}_i \in \mathbb{R}^3 \forall i = 1, \dots, n$, with \mathbf{a}_i being the link vector pointing from joint i to the COM of link i and \mathbf{b}_i representing a link vector pointing from the COM of link i to joint $i + 1$ of the manipulator [158, 160].

²⁵ Named for simplicity from hereafter simply as base.

Note that, in what follows all vectors (i. e. their components), unless specified otherwise, are expressed relative to Σ_I and are typeset in bold, e. g. \mathbf{v} . Bold symbols are also used for matrices, e. g. \mathbf{M} , as the vector components in a reference frame can be interpreted as a 3×1 column matrix. Variables typeset with a right arrow or hat, e. g. \vec{v} or \hat{v} , are instead reserved for physical vectors or unit vectors, respectively. Therefore, in general a column vector $\mathbf{v} = [v_x, v_y, v_z]^T \in \mathbb{R}^3$ will provide the component representation of a physical vector quantity \vec{v} in the reference frame Σ_I such that it is possible to write:

$$\vec{v} = v_x \hat{\mathbf{x}}_I + v_y \hat{\mathbf{y}}_I + v_z \hat{\mathbf{z}}_I \quad (4.2)$$

where $\{\hat{\mathbf{x}}_I, \hat{\mathbf{y}}_I, \hat{\mathbf{z}}_I\}$ are a set of unit vectors in Σ_I .

Differentiating Eq. (4.1) on both sides w. r. t. time and expressing the angular velocity of a link as a function of the angular velocity of the base, $\boldsymbol{\omega}_0 \in \mathbb{R}^3$, and joint velocities, $\dot{\boldsymbol{\phi}} \in \mathbb{R}^n$, the linear velocity of the end-effector, $\mathbf{v}_e \in \mathbb{R}^3$, can be expressed as [158]:

$$\mathbf{v}_e = \mathbf{v}_0 + \boldsymbol{\omega}_0 \times (\mathbf{p}_e - \mathbf{r}_0) + \sum_{i=1}^n \left\{ \hat{\mathbf{k}}_i \times (\mathbf{p}_e - \mathbf{p}_i) \right\} \dot{\phi}_i \quad (4.3)$$

where $\mathbf{v}_0 \in \mathbb{R}^3$ is the linear velocity of the base, $\hat{\mathbf{k}}_i$ is a unit vector associated with a rotational axis of joint i and $\mathbf{p}_i \in \mathbb{R}^3$ is the position of joint i .

Expressing the angular velocity of the end-effector, $\boldsymbol{\omega}_e \in \mathbb{R}^3$, as a function of $\dot{\boldsymbol{\phi}}$

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_0 + \sum_{i=1}^n \hat{\mathbf{k}}_i \dot{\phi}_i \quad (4.4)$$

the differential kinematics of the end-effector of a space manipulator, outlined in Eqs. (4.3) and (4.4), can be expressed in a more compact form as [160]:

$$\mathbf{v}_e \equiv \begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_0 \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \mathbf{J}_m \dot{\boldsymbol{\phi}} \quad (4.5)$$

where $\mathcal{V}_e \in \mathbb{R}^6$ is the spatial velocity vector of the end-effector. $J_0 \in \mathbb{R}^{6 \times 6}$ and $J_m \in \mathbb{R}^{6 \times n}$ represent the Jacobian matrices of the base and manipulator, respectively, and are defined as [160]:

$$J_0 \equiv \begin{bmatrix} \mathbf{I}_{3 \times 3} & -[\mathbf{p}_{0e}]_{\times} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad \mathbf{p}_{0e} = \mathbf{p}_e - \mathbf{r}_0 \quad (4.6)$$

$$J_m \equiv \begin{bmatrix} \hat{\mathbf{k}}_1 \times (\mathbf{p}_e - \mathbf{p}_1) & \cdots & \hat{\mathbf{k}}_n \times (\mathbf{p}_e - \mathbf{p}_n) \\ \hat{\mathbf{k}}_1 & \cdots & \hat{\mathbf{k}}_n \end{bmatrix} \quad (4.7)$$

where $\mathbf{0}_{3 \times 3}$ is a 3×3 matrix of zeros and $\mathbf{I}_{3 \times 3}$ is a 3×3 matrix of ones. The operator $[\cdot]_{\times} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ represents a mapping from a vector to a skew-symmetric matrix and is defined, considering a generic vector $\mathbf{v} \in \mathbb{R}^3$, as:

$$[\mathbf{v}]_{\times} \equiv \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix} \quad (4.8)$$

such that a vector cross product $\mathbf{v}_1 \times \mathbf{v}_2$ can be replaced with a matrix-vector product $[\mathbf{v}_1]_{\times} \mathbf{v}_2$ [161].

Eq. (4.5) displays the dependency of the differential kinematics of a space robot not only from joint variables but also from the base variables which complicates the solution even of a relatively simple forward kinematics problem. The reason for this lies in the nonholonomic constraint a space robot according to which the pose of the end-effector depends not only on the configuration of the base and manipulator but also on the overall history of the maneuver that was executed to reach that specific pose. For this reason the kinematics of the space manipulator is often defined not at the position level but at a velocity level as expressed in Eq. (4.5). Moreover, the knowledge of the kinematics alone is not enough for solving the inverse kinematics of a space robot as it also requires the knowledge of its dynamic characteristics [158].

The differential kinematic equation of a space robot (see Eq. (4.5)) can be reformulated in a more canonical form similar to that of ground-based manipulators as [158]:

$$\mathcal{V}_e = J^* \dot{\boldsymbol{\phi}} + J_0 \mathbf{H}_0^{-1} \begin{bmatrix} \mathbf{P} \\ \mathbf{L} - \mathbf{r}_0 \times \mathbf{P} \end{bmatrix} \quad (4.9)$$

where J^* is the *generalized Jacobian matrix* [160]

$$J^* = J_m - J_0 \mathbf{H}_0^{-1} \mathbf{H}_{0m} \in \mathbb{R}^{6 \times n} \quad (4.10)$$

$\mathbf{H}_0 \in \mathbb{R}^{6 \times 6}$ is the inertia matrix of the base and $\mathbf{H}_{0m} \in \mathbb{R}^{6 \times n}$ is the coupling inertia matrix the definitions of which are omitted here for

the sake of brevity but can be found in [158, 160, 162]. $\mathbf{P} \in \mathbb{R}^3$ and $\mathbf{L} \in \mathbb{R}^3$ represent the linear and angular momentum of the whole system w. r. t. Σ_I and are not assumed to be constant.

The generalized Jacobian matrix in Eq. (4.10) can be seen as an extended and more generic form of a Jacobian matrix of ground-based manipulators as it is a function of both kinematic and dynamic parameters. In fact, the more massive the base is, the closer J^* is to J_m which corresponds essentially to a Jacobian matrix of a ground-based (i. e. fixed-based) manipulator only defined w. r. t. Σ_I [160]. Therefore, it can be expected that with increasing base to manipulator ratio the planning and control of the end-effector will be easier as the reaction forces will be smaller.

For what concerns the rotational kinematics of the base, the minimal representation of an attitude state requires at least three parameters, each for each angle about an axis of rotation, e. g. a 3-angle, roll-pitch-yaw (XYZ), sequence $\Gamma = [\phi, \theta, \psi]^T$ also known as the Tait-Bryan angles. However, this minimal representation is subject to a singularity, also known as the *gimbal lock*, that occurs when the rotational axis of the second parameter becomes parallel to the axis of either the first or third parameter. In this configuration, the system loses one degree of freedom, the transformation cannot be inverted and only the sum or difference of the first and last parameters can be obtained [161]. Using an orthogonal rotation matrix representation solves this issue at the expense of the larger number of parameters²⁶ and the number of operations necessary to compute two successive rotations. A more elegant solution involves instead quaternions which are an extension of complex numbers and can represent an orientation of an object without singularity with four parameters and one constraint.

²⁶ Nine against the previous three.

A generic quaternion $\mathcal{Q} \in \mathbb{H} \cong \mathbb{R}^4$ can be expressed as [159, 161]:

$$\begin{aligned}\mathcal{Q} &= \{\eta, \epsilon\} \\ &= \eta + \epsilon \\ &= \eta + \epsilon_x i + \epsilon_y j + \epsilon_z k\end{aligned}\tag{4.11}$$

where $\eta \in \mathbb{R}$ and $\epsilon = [\epsilon_x, \epsilon_y, \epsilon_z]^T \in \mathbb{R}^3$ are the scalar and vector parts of \mathcal{Q} , respectively, while i, j and k are the orthogonal complex numbers defined such that [161]:

$$i^2 = j^2 = k^2 = ijk = -1$$

In order to represent rotations with quaternions a constraint of unit magnitude or norm, $\|\cdot\|$, must be added to Eq. (4.11) such that

$$\|\mathcal{Q}\| = \sqrt{\eta^2 + \epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2} = 1\tag{4.12}$$

Geometrically, a quaternion with such a constraint, denoted from now on as $\hat{\mathcal{Q}} \in \mathbb{S}^3$, can be conceived as a point on a three-dimensional unit

sphere in \mathbb{R}^4 [163] and is similar to another four parameters representation of rotations, the angle-axis. In fact, $\dot{\mathbf{Q}}$ can be considered as a rotation of a generic angle θ about a generic unit vector \hat{v} such that its components can be expressed in terms of the angle-axis representation as [159]:

$$\eta = \cos \frac{\theta}{2}, \quad \epsilon = \sin \frac{\theta}{2} \hat{v} \quad (4.13)$$

For this very reason the unit quaternion representation is not unique, i. e. a rotation by $-\theta$ about $-\hat{v}$ is equivalent in terms of $\dot{\mathbf{Q}}$ to the rotation by θ about \hat{v} . If θ is confined within $[-\pi, \pi]$ than $\eta \geq 0$ and the quaternion representation is unique [164].

Consider two coordinate systems Σ_1 and Σ_2 and let $\dot{\mathbf{Q}}_1 = \{\eta_1, \epsilon_1\}$ and $\dot{\mathbf{Q}}_2 = \{\eta_2, \epsilon_2\}$ denote their orientations w. r. t. Σ_I , respectively. The composition of the two rotations using the quaternion notation can be expressed as [159]:

$$\dot{\mathbf{Q}}_1 \otimes \dot{\mathbf{Q}}_2 = \left\{ \eta_1 \eta_2 - \epsilon_1^T \epsilon_2, \eta_1 \epsilon_2 + \eta_2 \epsilon_1 + \epsilon_1 \times \epsilon_2 \right\} \quad (4.14)$$

where \otimes stands for a quaternion (i. e. Hamiltonian) product.

Introducing the quaternion conjugate

$$\dot{\mathbf{Q}}^{-1} = \{\eta, -\epsilon\} \quad (4.15)$$

as the equivalent to an inverse of a rotation matrix, it is easy to observe that if $\dot{\mathbf{Q}}_2 = \dot{\mathbf{Q}}_1^{-1}$ than their product is equal to an identity quaternion $\{1, \mathbf{0}\}$ which corresponds to a zero rotation [159].

The relative orientation between the two frames Σ_1 and Σ_2 , in terms of a unit quaternion variation, can be expressed as [159]:

$$\begin{aligned} \Delta \dot{\mathbf{Q}} &= \{\Delta \eta, \Delta \epsilon\} \\ &= \dot{\mathbf{Q}}_1 \otimes \dot{\mathbf{Q}}_2^{-1} \end{aligned} \quad (4.16)$$

where [159, 164]

$$\Delta \eta = \eta_1 \eta_2 + \epsilon_1 \epsilon_2^T \quad (4.17)$$

$$\Delta \epsilon = \eta_1 \epsilon_2 - \eta_2 \epsilon_1 - [\epsilon_1]_{\times} \epsilon_2 \quad (4.18)$$

When the two frames are aligned, it can be seen that $\Delta \dot{\mathbf{Q}} = \{1, \mathbf{0}\}$. This implies that the two frames coincide if and only if $\Delta \epsilon = \mathbf{0}$, as through the normality constraint expressed in Eq. (4.12) it can be inferred that $\Delta \eta = \pm 1$.

The time derivative of a unit quaternion is however not a unit quaternion. It is a regular quaternion and can be considered as a 4-vector [161]. It provides a relationship between a unit quaternion $\dot{\mathbf{Q}}$ and an angular velocity ω , the so-called *quaternion propagation* [159], and can be defined as [161]:

$$\dot{\mathbf{Q}} = \frac{1}{2} \dot{\mathbf{Q}} \otimes \dot{\omega} \in \mathbb{H} \quad (4.19)$$

where $\dot{\mathbf{Q}} = [0, \boldsymbol{\omega}^T]^T$ is a pure quaternion of $\boldsymbol{\omega}$ expressed in the rotating, i. e. non-inertial, frame of reference. In terms of its components \mathbf{Q} can be expressed as [159]:

$$\dot{\eta} = -\frac{1}{2} \boldsymbol{\epsilon}^T \boldsymbol{\omega} \quad (4.20)$$

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} (\eta \mathbf{I}_{3 \times 3} - [\boldsymbol{\omega}]_{\times}) \boldsymbol{\epsilon} \quad (4.21)$$

Using the fact that η can be defined according to Eq. (4.12) as:

$$\eta = \sqrt{1 - \epsilon_x^2 - \epsilon_y^2 - \epsilon_z^2} \quad (4.22)$$

a one-to-one mapping between $\boldsymbol{\omega}$ and \mathbf{Q} can be expressed as [165]:

$$\dot{\mathbf{Q}} = \frac{1}{2} \boldsymbol{\Omega} \boldsymbol{\omega} \quad (4.23)$$

where the matrix $\boldsymbol{\Omega}$ is defined as [165]:

$$\boldsymbol{\Omega} \equiv \begin{bmatrix} \sqrt{1 - \epsilon_x^2 - \epsilon_y^2 - \epsilon_z^2} & -\epsilon_z & \epsilon_y \\ \epsilon_z & \sqrt{1 - \epsilon_x^2 - \epsilon_y^2 - \epsilon_z^2} & -\epsilon_x \\ -\epsilon_y & \epsilon_x & \sqrt{1 - \epsilon_x^2 - \epsilon_y^2 - \epsilon_z^2} \end{bmatrix} \quad (4.24)$$

Eq. (4.23) can be used instead of Eq. (4.19) for any given \mathbf{Q} if $\theta \neq \pm\pi$, as in those cases $\boldsymbol{\Omega}$ does not have a full rank. This is easily verifiable by analyzing its determinant defined as [165]:

$$\det(\boldsymbol{\Omega}) = \frac{1}{\sqrt{1 - \epsilon_x^2 - \epsilon_y^2 - \epsilon_z^2}} \quad (4.25)$$

The advantages of using Eq. (4.23) over Eq. (4.19) consists mainly of the reduced dimensions of the nonlinear system, controllable nature of the linearized system and a simpler stability analysis [165].

4.1.2 Dynamic equations

The dynamics of a space robot, just as that of a fixed-base robot, can be typically expressed in either the Newton-Euler formalism, by using the Newton's and Euler's dynamic equations of a rigid body, or the Lagrangian formalism that leverages the kinetic and potential energy of the system [163]. In this work the equations of motion of a space robot are derived using the Lagrangian formalism due to its conceptual simplicity and ability to concisely express the main characteristics of dynamic equations.

The first step in the Lagrangian formulation of dynamic equations of a generic n -link open chain manipulator is to select a set of *generalized*

coordinates, $\mathbf{q} \in \mathbb{R}^n$, that describe the configuration of the system and identify the corresponding *generalized forces*, $\boldsymbol{\tau} \in \mathbb{R}^n$. Once the generalized coordinates and forces have been selected the Lagrangian of the system, $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$, can be defined as a function of the generalized coordinates [163]

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{T}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}) \quad (4.26)$$

where $\mathcal{T}(\mathbf{q}, \dot{\mathbf{q}})$ represents the kinetic energy of the system and $\mathcal{U}(\mathbf{q})$ its potential energy. The equations of motion of the system are then expressed in terms of the Lagrangian by the *Euler-Lagrange equations* with external forces²⁷ as [163]:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} \quad (4.27)$$

²⁷ As the external forces of the standard Euler-Lagrange equations are zero [163].

In case of the space robot composed of $n + 1$ rigid-links (illustrated in Fig. 4.1) and subject to the microgravity environment, the generalized coordinates, $\mathbf{q} \in \mathbb{R}^{6+n}$, and their time derivatives, $\dot{\mathbf{q}} \in \mathbb{R}^{6+n}$, can be defined as:

$$\mathbf{q} = \begin{bmatrix} \mathbf{r}_0 \\ \dot{\mathbf{Q}}_0 \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_m \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega}_0 \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}}_0 \\ \dot{\mathbf{q}}_m \end{bmatrix} \quad (4.28)$$

where $\mathbf{q}_0 \in \mathbb{R}^6$ and $\mathbf{q}_m \in \mathbb{R}^n$ are the generalized coordinates of the base and manipulator, respectively. Consequently, the vector of generalized forces, $\boldsymbol{\tau} \in \mathbb{R}^{6+n}$, is defined as:

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_m \end{bmatrix} \quad \text{where} \quad \boldsymbol{\tau}_0 = \begin{bmatrix} \mathbf{f}_0 \\ \mathbf{n}_0 \end{bmatrix} \quad (4.29)$$

where $\boldsymbol{\tau}_0 \in \mathbb{R}^6$ is the vector of generalized forces of the base composed of forces, $\mathbf{f}_0 \in \mathbb{R}^3$, and torques, $\mathbf{n}_0 \in \mathbb{R}^3$, applied to its COM. $\boldsymbol{\tau}_m \in \mathbb{R}^n$ is the vector of generalized forces of the manipulator corresponding to joint torques applied to the n joints of the robot about their axes.

The potential energy of such a system can be assumed to be equal to zero while its kinetic energy can be defined as [160]:

$$\mathcal{T} \equiv \frac{1}{2} \sum_{i=0}^n \left\{ \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i + m_i \dot{\mathbf{r}}_i^T \dot{\mathbf{r}}_i \right\} \quad (4.30)$$

where $\boldsymbol{\omega}_i$ is the angular velocity of link i , $\mathbf{I}_i \in \mathbb{R}^{3 \times 3}$ is the inertia tensor of a link i expressed in its body frame, m_i is the mass of the link i and $\dot{\mathbf{r}}_i$ is the time derivative of \mathbf{r}_i .

Reformulating Eq. (4.30) in terms of the \mathbf{v}_0 , $\boldsymbol{\omega}_0$ and $\dot{\boldsymbol{\phi}}$ and collecting the mass and inertia properties of the system into \mathbf{H}_0 and \mathbf{H}_{0m} terms,

the kinetic energy of the space robot and therefore its Lagrangian can be expressed as:

$$\mathcal{L} = \mathcal{T} = \frac{1}{2} \begin{bmatrix} \dot{q}_0^T & \dot{q}_m^T \end{bmatrix} \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0m} \\ \mathbf{H}_{0m}^T & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_m \end{bmatrix} \quad (4.31)$$

where $\mathbf{H}_m \in \mathbb{R}^{n \times n}$ is the inertia matrix of the manipulator, equivalent to that of a ground-based one.

Substituting Eq. (4.31) into Eq. (4.27), computing the derivatives and properly rearranging the resulting terms, the associated equations of motion can be obtained in a canonical form, i. e. as a set of second-order differential equations of generalized coordinates, \mathbf{q} , as [70]:

$$\begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_{0m} \\ \mathbf{H}_{0m}^T & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_m \end{bmatrix} + \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_m \end{bmatrix} \quad (4.32)$$

where the nonlinear centripetal and Coriolis forces of the base and manipulator are contained in $\mathbf{c}_0 \in \mathbb{R}^{6 \times 1}$ and $\mathbf{c}_m \in \mathbb{R}^{n \times 1}$ terms, respectively.

In a more compact form Eq. (4.32) can be rewritten as a *joint space dynamic model* [159] of a space robot as:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \boldsymbol{\tau} \quad (4.33)$$

where $\mathbf{H} \in \mathbb{R}^{6+n \times 6+n}$ is the symmetric, positive definite inertia matrix of the overall system and $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{6+n \times 6+n}$ is the *Coriolis matrix* containing the nonlinear velocity dependent centripetal and Coriolis terms. The derived equations of motion such as Eq. (4.33) apply to the case of a free-floating space robot as well, i. e. a robot without the control authority of the base. In this specific scenario, the generalized coordinates are reduced only to the active joint angles of the manipulator, i. e. $\mathbf{q} \equiv \mathbf{q}_m$ and $\boldsymbol{\tau} \equiv \boldsymbol{\tau}_m$. Moreover, assuming for simplicity that the initial momentum of the system is equal to zero, the dependency of Eq. (4.33) from the uncontrolled base variables (v_0 and ω_0) can be eliminated such that \mathbf{H} and \mathbf{C} can be redefined as [158]:

$$\mathbf{H}(\mathbf{q}) = \mathbf{H}^*(\mathbf{q}) \equiv \mathbf{H}_m - \mathbf{H}_{0m}^T \mathbf{H}_0^{-1} \mathbf{H}_{0m} \in \mathbb{R}^{n \times n} \quad (4.34)$$

and

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \dot{\mathbf{H}}^* \dot{\mathbf{q}} - \frac{\partial}{\partial \boldsymbol{\phi}} \left(\frac{1}{2} \dot{\boldsymbol{\phi}}^T \mathbf{H}^* \dot{\boldsymbol{\phi}} \right) \in \mathbb{R}^{n \times n} \quad (4.35)$$

where \mathbf{H}^* is known as the *generalized inertia matrix* of space manipulators [70]. Observing Eq. (4.34) and Eq. (4.35), important properties of the dynamic equations of a free-floating space robot can be noted. First, the relationship between \mathbf{H} and \mathbf{C} existing for ground-based robots is valid also in case of space robots. Second, \mathbf{H}^* can be seen as an extension of \mathbf{H}_m of ground-based robots and just like \mathbf{J}^* the right

hand side (r. h. s.) of Eq. (4.34) is composed of two terms. The first one is a function of \mathbf{H}_m which is completely independent of the base dynamics. The second one is a function of \mathbf{H}_0 and \mathbf{H}_{0m} and represents the coupling between the manipulator and the base. Finally, it should be noted that as the mass properties of the base approach infinity $\mathbf{H}^* \rightarrow \mathbf{H}_m$, i. e. with the increase of the mass of the base, \mathbf{H}^* would reduce to that of a ground-based manipulator just as it was the case of \mathbf{J}^* [158].

4.2 INTRODUCTION TO OPTIMAL CONTROL

Optimal control aims at finding inputs to a dynamical system, over a defined time period, such that a performance index (also known as the objective or cost function) is optimized²⁸ while satisfying certain constraints of motion. Due to the complexity of most applications, solving an OCP in a closed form is either very difficult or outright impossible. Therefore, we must resort to numerical methods to solve the problem at hand. Classical numerical methods for solving an optimal control problem can be grouped essential into two categories: *indirect methods* and *direct methods* [166].

²⁸ Minimized or maximized depending on the application.

Indirect methods use the *calculus of variations* to analytically determine the first-order optimality conditions of the original OCP which leads to a multiple-point *boundary-value problem* that is discretized and solved numerically. Therefore, the original problem is solved indirectly through the system of differential equations that satisfy endpoint and/or interior point conditions [166].

Direct methods use a discretization to transform the original infinite-dimensional OCP into a large, sparse, finite-dimensional nonlinear programming (NLP) problem which is solved and refined until a sufficiently accurate solution is found [166, 167].

The choice of one over the other method depends on the type of the problem to be solved and the amount of time that can be dedicated to its software implementation. An indirect method, due to its analytical derivation of the first-order optimality conditions will in general produce more accurate solution when it converges. However, deriving such conditions, especially in case of a complex problem, is challenging, tedious, error-prone and sometimes even impossible [166]. Moreover, the region of convergence of an indirect method is generally smaller when compared to that of a direct method, thus requiring a better initial guess [168]. Finally, implementing an indirect method in a general-purpose software program might be more difficult due to the need to analytically derive the optimality conditions [166]. Because of these difficulties, the category of methods chosen as the basis for the optimization-based motion planner is that of direct methods and more specifically that of direct collocation capable of handling complex

problems and converge with poor initial guesses. What follows is an introduction into the general formulation of an OCP and direct methods to solve it.

4.2.1 Optimal control problem

Consider a dynamical system defined by a set of ordinary differential equations (ODEs), expressed here in explicit form and referred to as the *state equations* [167]

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t) \quad (4.36)$$

where $\mathbf{y}(t) \in \mathbb{R}^m$ and $\mathbf{u}(t) \in \mathbb{R}^n$ are the *state* and *control* variables, respectively, while $\mathbf{p} \in \mathbb{R}^p$ are the *static parameters*. $t \in \mathbb{R}$ is the independent variable of the system such that $t_I \leq t \leq t_F$, with t_I and t_F being the initial and final time, respectively.

A generic OCP related to the dynamic system described by Eq. (4.36) can be formally defined as finding the *decision variables* $\mathbf{y}(t)$, $\mathbf{u}(t)$, \mathbf{p} , t_I and t_F that optimize the *performance index* [166, 167]

$$J = \phi(\mathbf{y}(t_I), t_I, \mathbf{y}(t_F), t_F, \mathbf{p}) + \int_{t_I}^{t_F} w(\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t) dt \quad (4.37)$$

expressed here in the *Bolza form* composed of the boundary and path integral terms, i. e. the *Mayer* and *Lagrange* terms, respectively [167].

The decision variables are subject to the *dynamic constraints*, defined by the state equations (see Eq. (4.36)), and the *algebraic* (i. e. control) *path constraints* of the form

$$\mathbf{g}_L \leq \mathbf{g}(\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t) \leq \mathbf{g}_U \quad (4.38)$$

where $\mathbf{g}_L \in \mathbb{R}^q$ and $\mathbf{g}_U \in \mathbb{R}^q$ are the lower and upper algebraic path constraint vectors, respectively. In addition, the solution must satisfy the *boundary conditions* [168]

$$\boldsymbol{\psi}_L \leq \boldsymbol{\psi}(\mathbf{y}(t_I), \mathbf{u}(t_I), t_I, \mathbf{y}(t_F), \mathbf{u}(t_F), t_F, \mathbf{p}) \leq \boldsymbol{\psi}_U \quad (4.39)$$

as well as simple *state* and *control bounds* [167]

$$\mathbf{y}_L \leq \mathbf{y}(t) \leq \mathbf{y}_U, \quad \mathbf{u}_L \leq \mathbf{u}(t) \leq \mathbf{u}_U \quad (4.40)$$

where $\boldsymbol{\psi}(\mathbf{y}(t_I), \mathbf{u}(t_I), t_I, \mathbf{p}) \equiv \boldsymbol{\psi}_I$ and $\boldsymbol{\psi}(\mathbf{y}(t_F), \mathbf{u}(t_F), t_F, \mathbf{p}) = \boldsymbol{\psi}_F$ are the initial and final conditions of the system, respectively, while the subscripts $[\cdot]_L$ and $[\cdot]_U$ indicate the lower and upper values of variables, respectively.

4.2.2 Nonlinear programming problem

An OCP generally involves continuous function of $\mathbf{y}(t)$ and $\mathbf{u}(t)$. In contrast, an NLP problem implicates finding a finite number of variables \mathbf{x} that optimize a performance index without violating a set of finite constraints \mathbf{c} . Therefore, it might be convenient to see an OCP as an infinite-dimensional extension of an NLP problem [167].

Formally an NLP problem can be defined as finding the vector $\mathbf{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$ of decision variables that minimizes the scalar objective function [167]

$$F(\mathbf{x}) \tag{4.41}$$

subject to the $m \leq n$ constraints

$$\mathbf{c}_L \leq \mathbf{c}(\mathbf{x}) \leq \mathbf{c}_U \tag{4.42}$$

and simple bounds

$$\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \tag{4.43}$$

The equality constraints can be imposed simply by setting $\mathbf{c}_L = \mathbf{c}_U$.

The *Karush-Kuhn-Tucker* necessary conditions for a vector \mathbf{x}^* to be a constrained optimum of an NLP problem given by Eqs. (4.41) to (4.43) require that [167]:

- \mathbf{x}^* is feasible, i. e. Eqs. (4.42) and (4.43) are satisfied;
- the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ corresponding to Eq. (4.42) and Eq. (4.43), respectively, satisfy

$$\mathbf{g} = \mathbf{G}^T \boldsymbol{\lambda} + \boldsymbol{\nu} \tag{4.44}$$

where $\mathbf{g} \equiv \nabla_{\mathbf{x}} F = [\partial F / \partial x_1, \dots, \partial F / \partial x_n]^T$ is the n -dimensional gradient vector of the objective function $F(\mathbf{x})$ and $\mathbf{G} \equiv \partial \mathbf{c} / \partial \mathbf{x}$ is the Jacobian of the constraint vector $\mathbf{c}(\mathbf{x})$.

- the Lagrange multipliers for the inequalities are:
 - nonpositive for active upper bounds,
 - zero for strictly satisfied constraints,
 - non-negative for active lower bounds;
- the Jacobian of the active constraints have a full row rank.

Numerical methods able to solve NLP problems defined by Eqs. (4.41) to (4.43) can essentially be grouped into two categories: *gradient methods* and *heuristic methods*.

Gradient methods are *local methods*, meaning that, upon convergence, the found optimal solution will generally by a local optimum.

The main idea behind gradient methods is to find a solution in a deterministic and iterative manner [166]. The most widely used gradient methods are the sequential quadratic programming (SQP) and interior-point (also known as the barrier) methods. An SQP algorithm consists of solving a series of quadratic programming subproblems each used to determine the *search direction* of the solution and the *step length* providing the direction along which to change the current guess and magnitude of that change, respectively. In contrast, an interior-point algorithm replaces a constrained optimization problem by a sequence of unconstrained problems in which the *barrier function* is used to constraint the search of the optimal solution within a feasible region, hence the name interior-point [166, 167].

Heuristic methods are *global methods*, i. e. able to provide a globally optimal solution, when available, and the search for the solution is performed in a stochastic manner. Two well-known classes of heuristic methods used in the optimal control are genetic algorithms (GAs) and particle swarm optimization (PSO) algorithms. A GA mimics evolutionary processes in genetics to find genes with the highest fitness²⁹ within a population that are then passed over to the later generations via a recombination (i. e. mutation) mechanism, thus improving the solution after each generation. A PSO algorithm is a population-based optimization method that shares many similarities with GAs and is roughly based on the idea of swarms of animals. The optimal solution is obtained iteratively by generational searches which do not involve a mutation of candidate solutions, called particles, unlike in case of GAs [166].

²⁹ Intended here as the quality of a particular gene [166].

4.2.3 Direct transcription

In a direct method, the solution to the original OCP is found by performing a *discretize then optimize* philosophy which consists essentially out of three steps [167]:

1. convert (i. e. transcribe) the continuous functions of the dynamic system into a finite set of variables using discretization;
2. solve the finite-dimensional problem, i. e. the NLP subproblem, using a parameter optimization method;
3. assess the accuracy of the conversion (i. e. transcription) and if necessary repeat the previous steps.

When the transcription involves only the control variables of the problem, the method is called the *control parameterization method*. The two main techniques belonging to this class of methods are the *direct shooting* and *direct multiple shooting*. In case where both the state and control variables are transcribed, the method is called the *state and*

control parameterization method. Its two most common techniques are the *direct collocation* and *pseudospectral method* [166].

The direct shooting is the most basic control parameterization approach where the solution is composed of the parameters defining $\mathbf{u}(t)$ such that the performance index is optimized and all constraints satisfied. The state variables $\mathbf{y}(t)$ are found by integrating the equations of motion, therefore ensuring that the dynamic constraints are satisfied. The cost function is determined using a quadrature approximation³⁰ consistent with the numerical integrator used to solve the equations of motion [166]. The method derives its name from an early practical application of the method where the goal was to find an angle of a cannon such that, when a cannonball is fired, it hits a target [167]. The method is appealing due to its simplicity and ability to describe the problem with a relatively small number of variables. However, it presents one significant disadvantage which is high sensibility of the final conditions to even smallest changes in initial conditions³¹, which can lead to highly nonlinear problem constraints that are very difficult to solve [167]. Therefore, this method lends itself well to problems requiring simple control with few path constraints, such as launch and orbit transfer problems [168].

³⁰ Numerical approximation of a definite integral.

³¹ The so called “tail wagging the dog” behavior [167].

To overcome the difficulties of the direct shooting method, one way is to break the original problem into many shorter ones, i. e. “shooting not as far” [167]. This approach is called direct multiple shooting and is an extension of the direct shooting method. In it the time interval $[t_I, t_F]$ is divided into $M + 1$ subintervals and over each the aforementioned direct shooting method is used to find the values of the state at the beginning of each subinterval, along with the parameters of the control variables. In order to enforce the continuity between the subintervals, additional constraints, called *defect* constraints or simply, *defects*, are enforced at the interface of each subinterval [166]. An obvious consequence of this method over the original shooting method is the increased size of the problem. Nevertheless, it represents a marked improvement over the shooting method in terms of robustness and therefore can be used in case of more challenging problems. Another benefit is the ability to speed up the solution of a problem by exploiting parallel computing [167, 168].

Direct collocation methods are a class of transcription approaches where the decision variables, $\mathbf{u}(t)$ and $\mathbf{y}(t)$, are approximated as piecewise polynomial functions, i. e. splines, with the unknowns of the optimization being the coefficients of the splines. The feasibility of the solution is guaranteed by ensuring that the equations of motion are satisfied at a finite number of points distributed evenly in the time domain, known as *collocation points*. The objective function is approximated using a quadrature consistent with the collocation method used to approximate the system dynamics. The constraints of the problem are handled by enforcing them at specific collocation points [166, 169].

A generic formulation of a direct collocation method is expressed in what follows. Suppose that in a generic single phase OCP, represented by Eqs. (4.36) to (4.40), the time domain can be uniformly divided into n intervals [167]

$$t_I = t_1 < t_2 < \dots < t_M = t_F \quad (4.45)$$

where $t_{k+1} = t_k + h$, $h = t_f/M$, $M \equiv n + 1$ and the fixed points in time are assigned the name: node, mesh, knot or grid points. Let us denote with $\mathbf{y}_k \equiv \mathbf{y}(t_k)$, $\mathbf{u}_k \equiv \mathbf{u}(t_k)$ and $\mathbf{f}_k \equiv \mathbf{f}(\mathbf{y}(t_k), \mathbf{u}(t_k), \mathbf{p}, t_k)$ the value of the state variable, control and state equations at a specific grid point t_k , respectively. In addition, let us denote with $\bar{\mathbf{u}}_k \equiv \mathbf{u}(\bar{t})$ the value of the control at the midpoint of an interval where $\bar{t} = 1/2(t_k + t_{k+1})$.

Assuming the state and control as a set of NLP variables such as for example [167]:

$$\mathbf{x} = [\mathbf{u}_1, \mathbf{y}_1, \mathbf{u}_2, \mathbf{y}_2, \dots, \mathbf{u}_M, \mathbf{y}_M]^T \quad (4.46)$$

the state equations, expressed in Eq. (4.36), can be replaced by a finite set of defect constraints, $\zeta_k \forall k = 1, \dots, M - 1$, derived by the numerical integration scheme. For example, if a forward Euler method is used, the defects are defined as [167]:

$$c_k(\mathbf{x}) \equiv \zeta_k = \mathbf{y}_{k+1} - \mathbf{y}_k - h \mathbf{f}(\mathbf{y}_k, \mathbf{u}_k) = 0 \quad (4.47)$$

since in this case the state equation can be approximated as [167]:

$$\dot{\mathbf{y}} \approx \frac{\mathbf{y}_{k+1} - \mathbf{y}_k}{h} \quad (4.48)$$

As a result of the transcription, the constraints of the original problem expressed in Eqs. (4.36), (4.38) and (4.39) are replaced by the NLP constraints expressed in Eq. (4.42) where [167]

$$c(\mathbf{x}) = [\zeta_1, \zeta_2, \dots, \zeta_{M-1}, \psi_I, \psi_F, \mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_M]^T \quad (4.49)$$

with

$$c_L = [\mathbf{0}, \dots, \mathbf{0}, \mathbf{g}_{1,L}, \dots, \mathbf{g}_{M,L}]^T \quad (4.50)$$

$$c_U = [\mathbf{0}, \dots, \mathbf{0}, \mathbf{g}_{1,U}, \dots, \mathbf{g}_{M,U}]^T \quad (4.51)$$

The first n defect vectors in Eq. (4.49) are set equal to zero to approximately satisfy the differential equations of Eq. (4.36). The nonlinear path constraints in Eq. (4.38) are handled by simply imposing them at grid points while the boundary conditions in Eq. (4.39) are enforced by equality constraints on ψ_I and ψ_F . The objective function given in Eq. (4.37) is approximated by writing it as a function of \mathbf{x} and using the appropriate quadrature. Finally, the state and control bounds in Eq. (4.40) are handled by enforcing them as simple bounds on NLP variables. It needs to be noted that while all discretization schemes

require the path constraints and variable bounds to be imposed at the grid points, the Hermite-Simpson and Runge-Kutta discretization methods require them imposed also at the interval midpoints [167].

The expression of NLP variables and defect constraints of four most common *K-stage Runge-Kutta* discretization schemes are summarized hereafter as a reference [167]:

Euler method

Variables:

$$\mathbf{x} = [\mathbf{y}_1, \mathbf{u}_1, \dots, \mathbf{y}_M, \mathbf{u}_M]^T \quad (4.52)$$

Defects:

$$\zeta_k = \mathbf{y}_{k+1} - \mathbf{y}_k - h_k \mathbf{f}_k \quad (4.53)$$

Classical Runge-Kutta method

Variables:

$$\mathbf{x} = [\mathbf{y}_1, \mathbf{u}_1, \bar{\mathbf{u}}_2, \dots, \bar{\mathbf{u}}_M, \mathbf{y}_M, \mathbf{u}_M]^T \quad (4.54)$$

Defects:

$$\zeta_k = \mathbf{y}_{k+1} - \mathbf{y}_k - \frac{1}{6} (\mathbf{k}_1 + 2 \mathbf{k}_2 + 2 \mathbf{k}_3 + \mathbf{k}_4) \quad (4.55)$$

where

$$\mathbf{k}_1 = h_k \mathbf{f}_k \quad (4.56)$$

$$\mathbf{k}_2 = h_k \mathbf{f} \left(\mathbf{y}_k + \frac{1}{2} \mathbf{k}_1, \bar{\mathbf{u}}_{k+1}, t_k + \frac{h_k}{2} \right) \quad (4.57)$$

$$\mathbf{k}_3 = h_k \mathbf{f} \left(\mathbf{y}_k + \frac{1}{2} \mathbf{k}_2, \bar{\mathbf{u}}_{k+1}, t_k + \frac{h_k}{2} \right) \quad (4.58)$$

$$\mathbf{k}_4 = h_k \mathbf{f} (\mathbf{y}_k + \mathbf{k}_3, \mathbf{u}_{k+1}, t_{k+1}) \quad (4.59)$$

Trapezoidal method

Variables:

$$\mathbf{x} = [\mathbf{y}_1, \mathbf{u}_1, \dots, \mathbf{y}_M, \mathbf{u}_M]^T \quad (4.60)$$

Defects:

$$\zeta_k = \mathbf{y}_{k+1} - \mathbf{y}_k - \frac{h_k}{2} (\mathbf{f}_k + \mathbf{f}_{k+1}) \quad (4.61)$$

Hermite-Simpson method

Variables:

$$\mathbf{x} = [\mathbf{y}_1, \mathbf{u}_1, \bar{\mathbf{u}}_2, \dots, \bar{\mathbf{u}}_M, \mathbf{y}_M, \mathbf{u}_M]^T \quad (4.62)$$

Defects:

$$\zeta_k = \mathbf{y}_{k+1} - \mathbf{y}_k - \frac{h_k}{6} (\mathbf{f}_k + 4\bar{\mathbf{f}}_{k+1} + \mathbf{f}_{k+1}) \quad (4.63)$$

where

$$\bar{\mathbf{y}}_{k+1} = \frac{1}{2} (\mathbf{y}_k + \mathbf{y}_{k+1}) + \frac{h_k}{8} (\mathbf{f}_k - \mathbf{f}_{k+1}) \quad (4.64)$$

$$\bar{\mathbf{f}}_{k+1} = \mathbf{f} \left(\bar{\mathbf{y}}_{k+1}, \bar{\mathbf{u}}_{k+1}, t_k + \frac{h_k}{2} \right) \quad (4.65)$$

The collocation scheme of particular interest for the rest of this work is Eq. (4.63), also known as the *Lobatto IIIA method* of order four [167]. In fact, while the trapezoidal collocation method (see Eq. (4.61)) approximates the control trajectory and system dynamics as piecewise linear functions, i. e. linear splines, the Hermite-Simpson collocation method approximates them as piecewise quadratic functions, i. e. quadratic splines, thus providing a higher-order accurate solution. In addition, while the state trajectory of the former method is approximated as a quadratic spline, the latter uses a cubic Hermite spline which has a continuous first derivative [169]. Finally, all Lobatto methods share a unique property which is that the boundaries and knot points of the trajectory are also collocation points [167].

Pseudospectral methods, also known as global orthogonal collocation methods, are closely related to the methods previously described, as they use the same collocation idea but the state and control trajectories are approximated using a linear combination of *global polynomial* basis functions, such as Chebyshev or Lagrange polynomials. This way an optimal solution is found by keeping fixed the number of segments (i. e. meshes) of a piecewise polynomial and varying its degree, as opposed to changing the number of meshes and keeping fixed the degree of the polynomial as done in case of direct collocation methods [166].

4.3 SEMISYNCHRONIZING APPROACH PROBLEM

As mentioned at the beginning of the chapter, employing a space robot to capture even a still target is a challenging task due to the dynamical coupling between the manipulator and its base. This imposes a nonholonomic constraint on the system and introduces dynamic singularities which location and existence cannot be determined using

only the kinematics, as in case of ground-based robots. The challenge is even more exacerbated when considering a capture of a target having high angular momentum (i. e. high angular rate and/or inertia). If this quantity is not considered appropriately in the approach phase of a capture maneuver, it could pose difficulties to the ACS of the spacecraft in the post-capture phase, as on-board storing of large amounts of momenta might be dangerous or even impossible. The mainstream solution to this problem currently is to achieve a complete relative motion synchronization between the chaser and target, such that the latter is fixed in the body frame of the former, while relying on the ACS and a feed-forward control chaser to counteract any reaction forces induced by the manipulator onto the base [104]. Doing so will result in an overall simpler³² and safer maneuver at the expense of high consumption of on-board propellant which might work well for a single-target mission but would pose a serious limitation in case of a multi-target mission. To overcome this limitation Ma, Wei, and Yuan [10] recently conceived a semisynchronizing (i. e. partially synchronizing) strategy of a chaser spacecraft to aid the capture of a fast tumbling target. This way propellant consumption can be significantly reduced during the synchronizing maneuver while at the same time minimizing the relative angular velocity of the grappling feature within the acceptable range for a robotic arm. This work builds upon the defined semisynchronizing method to solve the motion planning problem of both the spacecraft and manipulator to capture a rapidly tumbling target and is formulated as a single-phase, nonlinear optimization. Details of the mathematical formulation of the method and its implementation are outlined in what follows.

³² From the perspective of the manipulator control.

4.3.1 Assumptions and reference mission

A generic robotic capture maneuver of an uncooperative target, i. e. its close-range rendezvous phase, can be usually divided into four phases illustrated in Fig. 4.2:

CLOSE RANGE OBSERVATION during which the chaser is assumed to perform the inspection and pose estimation of the target before the final approach from an out of the plane safety ellipse around the target able to guarantee passive safety at all times [170]. During this phase the time is not considered a critical parameter as a proper design of the safety ellipse should passively guarantee a minimum distance between the two objects.

SPACECRAFT APPROACH consists of a series of maneuvers performed by the ACS of the spacecraft in order to: a) acquire the desired capture axis of a target (generally coinciding with the main axis of rotation of a target), b) perform the final approach till reaching the berthing box, i. e. a volume relative to the target within

which the chaser must be located to achieve a successful capture, c) achieve relative attitude synchronization within the required boundaries of the manipulator. During these maneuvers the safety, i. e. collision avoidance between the bodies, must always be guaranteed either passively by a proper design of trajectories or actively via appropriately planned CAMs. Time is considered a critical parameter, especially during the final approach, as the berthing box is subject to natural drifts that are to be expected in the order of minutes.

MANIPULATOR APPROACH covers the deployment of the manipulator from a stowed configuration (if not already deployed) into a predefined initial configuration and its approach, following a pre-planned trajectory, toward a pre-selected pre-grasping point, assumed to be “rigidly attached” to the grasping feature of a target. As in the previous phase, due to the close range nature of the maneuver and tumbling state of the target, safety needs to be maintained at all times by, for example, enforcing the keep-out zone around the target. As in the previous phase, time is considered a critical parameter but mainly due to the drifting berthing box and limitations of the manipulator to compensate for resulting errors.

CAPTURE AND STABILIZATION encompasses: a) tracking phase of the grasping feature, necessary to correct any errors occurred during the previous phase while simultaneously reducing to zero the relative distance and velocity between the end-effector of the manipulator and the grasping feature, b) capture of the grasping feature involving physical contact, c) attenuation of shocks and residual velocities in order to achieve a rigid connection between the two bodies, d) stabilization of the compound through the transfer of relative angular momentum from a target to the chaser and its dissipation using the ACS of the spacecraft. Safety and time are of paramount importance during this phase as it involves contact between two floating bodies.

In this context the developed motion planner focuses on the *manipulator approach* phase, illustrated in Fig. 4.2c, and more specifically on the manipulator maneuver from a pre-defined initial configuration³³ to a pre-defined pre-grasping point that is at a safe distance from the grasping feature. The objective of the method is to provide a locally optimal motion of the robotic spacecraft and its manipulator to facilitate the capture of an uncooperative, rapidly tumbling target while minimizing the overall control effort.

³³ The manipulator is assumed to be already deployed.

ii Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer International Publishing, *Stardust Final Conference. Astrophysics and Space Science Proceedings*. "Trajectory Generation Method for Robotic Free-Floating Capture of a Non-cooperative, Tumbling Target", DOI: 10/gf93ms, M. Jankovic and F. Kirchner, © 2018.

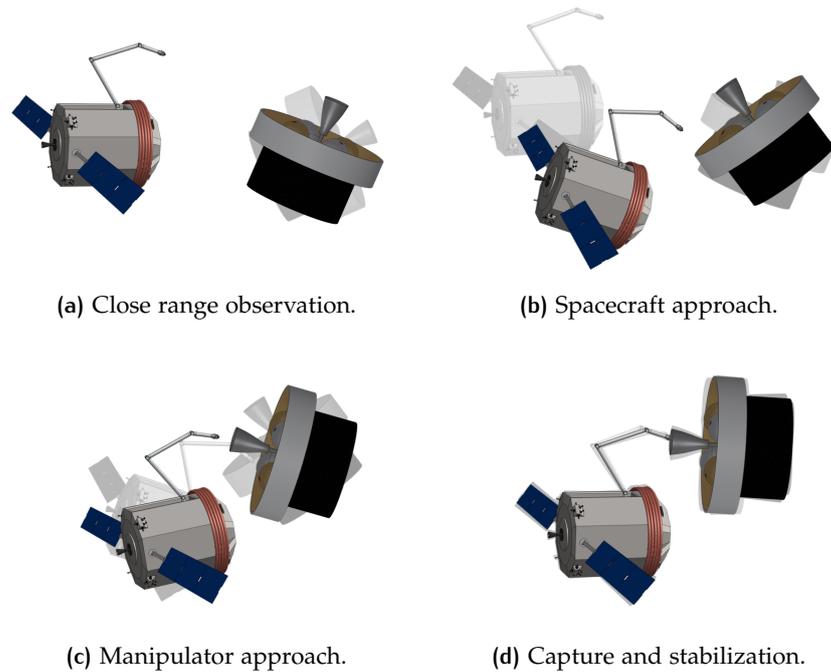


Figure 4.2: Mission phases of a robotic ADR capture maneuver (reprinted from [16]ⁱⁱ).

The assumptions made during the formulation of the problem are the following:

1. the motion planning is executed while in the *close range observation* phase and the duration of the planned approach maneuver is fixed,
2. the target is assumed to be a generic rocket body that undergoes a constant angular motion known in advance having an angular rate constrained within the $[5, 18]$ deg s^{-1} interval,
3. relative orbital dynamic effects as well as external perturbation forces, that might act on the target or chaser, are neglected,
4. the system is composed only of rigid bodies which inertial characteristics are known,
5. the mass of the chaser is constant throughout the maneuver,
6. the chaser is assumed to start the maneuver from a user-defined initial state and end it within a user-defined state space region,
7. the semisynchronizing attitude strategy developed by Ma, Wei, and Yuan [10] precedes the overall maneuver.

The assumed absence of external torques applied to the system is justifiable by a generally short duration of the approach and capture

maneuvers (≈ 30 s to 120 s) when compared to a typical orbital period of an object in LEO (≈ 5400 s).

The assumption of the fixed mass of the chaser throughout the maneuver can be rationalized by the negligibly small mass of the propellant that will be used during the maneuver compared to the overall mass of the chaser.

The aim of the assumption of the final state of the chaser to be optimized within a certain state space region stems from the desire to subject this state to the optimization while being able to constraint possible solutions.

The fixed duration of the maneuver can be rationalized by the need to achieve the capture maneuver (and by extension the approach of the manipulator) while the target is at a specific relative pose, to maximize the possibility of capture even in case of uncertainties and perturbations not considered in the problem.

The specific semisynchronizing method considered in the developed planner is the minimum relative angular velocity strategy (MRAVS), capable of achieving a minimum relative angular velocity between the chaser and target. The method defines the optimal rotation axis and angular rate via a geometrical interpretation of a torque-free motion of a rigid-body. Details of the method are reported hereafter.

Assuming a rigid-body rotating around its center of mass with $\vec{\omega}$, its total angular momentum, \vec{L} , can be expressed as [171]:

$$\vec{L} = \mathbf{I} \cdot \vec{\omega} \quad (4.66)$$

where $\mathbf{I} \in \mathbb{R}^{3 \times 3}$ is the moment of inertia tensor of the body and the dot symbol, \cdot , indicates a scalar, dot, product between two vectors.

Similarly, the rotational kinetic energy of the body can be written as [171]:

$$\mathcal{T} = \frac{1}{2} \vec{\omega} \cdot \mathbf{I} \cdot \vec{\omega} \quad (4.67)$$

or according to Eq. (4.66), equivalently as:

$$\mathcal{T} = \frac{1}{2} \vec{\omega} \cdot \vec{L} \quad (4.68)$$

The rotational dynamics of the body is described by the *Euler's equations* of motion, which relate the variation of the angular momentum to the total amount of external torque applied to the body and can be expressed relative to Σ_B as [171]:

$$\dot{\vec{L}} + \vec{\omega} \times \vec{L} = \vec{n} \quad (4.69)$$

Expressing the vector quantities in body axis components, if \mathbf{I} is constant, Eq. (4.69) can be reformulated as [172]:

$$\mathbf{I}^B \dot{\boldsymbol{\omega}} + {}^B \boldsymbol{\omega} \times (\mathbf{I}^B \boldsymbol{\omega}) = {}^B \mathbf{n} \quad (4.70)$$

In case a set of principal axis relative to the reference point are chosen as the body axis, \mathbf{I} becomes diagonal and Eq. (4.70) can be expressed as [171, 172]:

$$\begin{aligned} I_x \dot{\omega}_x + (I_z - I_y) \omega_y \omega_z &= n_x \\ I_y \dot{\omega}_y + (I_x - I_z) \omega_z \omega_x &= n_y \\ I_z \dot{\omega}_z + (I_y - I_x) \omega_x \omega_y &= n_z \end{aligned} \quad (4.71)$$

where the reference to Σ_B has been omitted for readability.

³⁴ Also known as a torque-free motion of a rigid-body.

In the absence of external torques applied to the body, its motion³⁴ is generally characterized by a time-varying angular velocity and an angular momentum vector that is constant in Σ_I . Moreover, its magnitude is independent of the considered reference frame and is also constant, i. e. $L = \|\mathbf{L}\| = \text{const}$. Geometrically this fact implies that $\vec{\omega}$ must lie within a *momentum ellipsoid* characterized by the following equation [172]:

$$\frac{\omega_x^2}{(L/I_x)^2} + \frac{\omega_y^2}{(L/I_y)^2} + \frac{\omega_z^2}{(L/I_z)^2} = 1 \quad (4.72)$$

The kinetic energy of the rigid-body subject to a torque-free motion is also constant, which geometrically indicates that $\vec{\omega}$ must also satisfy the equation [172]

$$\frac{\omega_x^2}{(2\mathcal{T}/I_x)} + \frac{\omega_y^2}{(2\mathcal{T}/I_y)} + \frac{\omega_z^2}{(2\mathcal{T}/I_z)} = 1 \quad (4.73)$$

³⁵ Also known in the literature as the inertia ellipsoid [171, 173]

which identifies the surface of an *energy ellipsoid*³⁵ fixed in Σ_B .

The intersection of the two ellipsoids described by Eqs. (4.72) and (4.73), specifies a closed path in Σ_B , called *polhode*, along which the tip of $\vec{\omega}$ moves in time (see Fig. 4.3). As $\vec{\omega}$ travels along the polhode, it is the energy ellipsoid (fixed in Σ_B) that moves in such a way that the normal to its surface at the instantaneous point of contact of $\vec{\omega}$ on the polhode is parallel to the \vec{L} which is fixed in space [171].

Recalling Eq. (4.68) and the fact that $\mathcal{T} = \text{const}$, it can be seen that in case of a torque-free motion of a rigid body [172]

$$\omega_L = \vec{\omega} \cdot \frac{\vec{L}}{L} = \frac{2\mathcal{T}}{L} = \text{const} \quad (4.74)$$

which reveals that during the motion of the body the tip of $\vec{\omega}$ always rests on a plane fixed in Σ_I at a distance $2\mathcal{T}/L$ from the center of the energy ellipsoid. Therefore, a torque-free motion of a rigid body can be pictured in Σ_I (see Fig. 4.3) as the rolling, without slipping, of its energy ellipsoid on the *invariable plane*³⁶ with the center of the ellipsoid (i. e. the COM of the body) fixed at a distance defined by Eq. (4.74) and the point of contact being the tip of $\vec{\omega}$ [171, 173]. The curve traced by $\vec{\omega}$ on the invariable plane is known as *herpolhode* and is in general

³⁶ Also known as the "fixed plane".

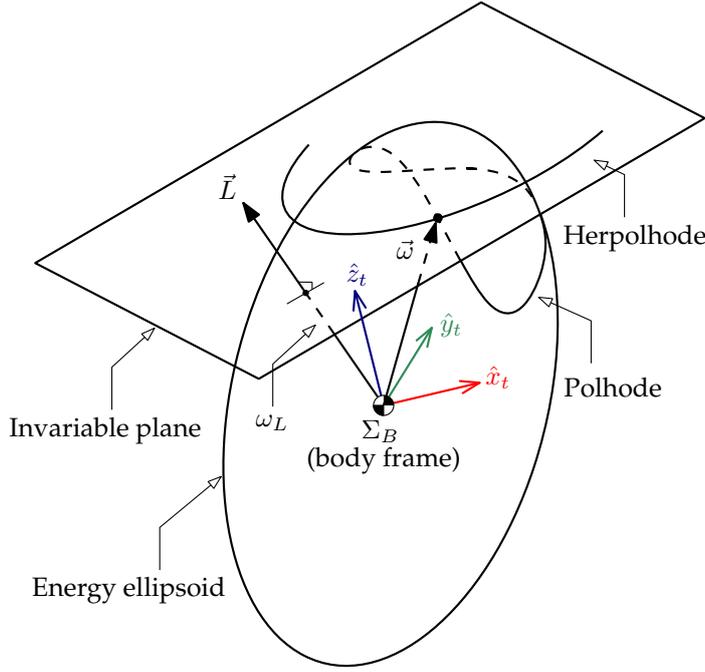


Figure 4.3: Main geometrical characteristics of a torque-free motion of a rigid body (\vec{L} = angular momentum of a body, $\vec{\omega}$ = angular velocity of a body, ω_L = fixed distance between the invariable plane and center of mass, i. e. projection of $\vec{\omega}$ onto \vec{L}) (based on [174, Fig. 4.2]).

not a closed curve [172] but is always concave to the origin [171] on the invariable plane.

MRAVS takes the advantage of the just described geometric characteristics of a torque-free motion to define the synchronization maneuver of the base. It consists of a pure rotation of the base spacecraft around the direction of the angular momentum vector of the target \vec{L}_t at an angular rate corresponding to the projection of angular velocity of the target, $\vec{\omega}_t$, onto \vec{L}_t defined by Eq. (4.74). This way, the body axis of the chaser pointing toward the COM of the target, will remain fixed in Σ_I , just like \vec{L}_t , and only a spin-up and spin-down commands will need to be issued, greatly reducing maneuver complexity and amount of propellant needed for it. The maximum reduction of the relative angular velocity of the target, λ , in percentages, using MRAVS can be calculated using the following expression [10]:

$$\lambda = \frac{\omega_L}{\|\vec{\omega}_t\|} \cdot 100 = \cos\langle\vec{\omega}_t, \vec{L}_t\rangle \quad (4.75)$$

where $\langle\vec{\omega}_t, \vec{L}_t\rangle$ denotes the angle between the two vectors. Consequently, if the angle between the two vectors in Σ_I is large, such as in case of a “wildly tumbling” target, the chaser might fail to reduce the relative angular rate to an acceptable level for the capture. This represents in fact the main limitation of MRAVS [10].

The illustration of the overall semisynchronizing approach problem of a free-flying chaser is depicted in Fig. 4.4 where Σ_I represents the inertial frame of the system of rigid bodies and is assumed to translate with the system at its orbital speed during the short duration of the capture maneuver. Σ_t and Σ_0 are the body-fixed frames of the target and chaser, respectively. The origins of those frames are located at the respective COMs and axis are assumed to be parallel to the principal axis of inertia of the target and chaser, respectively. The target is characterized by an angular velocity $\vec{\omega}_t$ and an angular momentum \vec{L}_t . Due to the tumbling motion the target a *keep-out zone*³⁷ is enforced around it and its radius is equal to the half of the maximum length of a target plus a margin. The chaser is distinguished by an initial $\vec{\omega}_0$, which magnitude is derived using Eq. (4.74) while its direction is coincident with that of \vec{L}_t . The initial configuration of the manipulator and the initial pose of the end-effector are assumed to be known. The aim of the approach maneuver is for the manipulator of the chaser to perform a “point-to-point” motion to reach a pre-grasping feature of the target while not penetrating the keep-out zone nor violating the kinematic and dynamic constraints of the chaser. In order to constraint the solution of the motion planning problem to a certain state space region, the final configuration of the chaser is constrained.

³⁷ A zone where the presence of a chaser should be avoided to reduce potential collisions.

The envisioned implementation example of the motion planner for semisynchronizing capture (POSE) within a robotic control architecture of a chaser is depicted in Fig. 4.5. The proposed control architecture is divided into two modules: an on-board (i. e. on-line) and on-ground (i. e. off-line). The latter consists of a target motion simulation and prediction module along with the developed motion planner and are situated on-ground, i. e. external to the chaser. The former instead resides within the robotics module, on-board the chaser, outside the GNC architecture, in order to enhance the computational efficiency of the on-board computer [175]. It uses the calculated off-line solution as an initial guess for the trajectory generation and control of the robotic arm in real time. The reason behind this division lies in the computational requirements of the developed motion planner, that can not be performed in a reasonable time with the computational power of nowadays on-board computers. Moreover, it is worth noting that this computationally intensive task has to be performed just once given the dynamic properties of the robotic chaser spacecraft and the geometry of a target. Thus, it makes more sense to do it on-ground and upload it to the spacecraft before the capture maneuver.

4.3.2 Optimal control problem formulation

³⁸ C-space for short [163].

Consider the definition of the configuration space of a robot³⁸, \mathcal{C} , as the space where each configuration q of the robot represents a point. Additionally, let us define the *free C-space*, $\mathcal{C}_{\text{free}}$, a subset of $\mathcal{C} \subset \mathbb{R}^n$

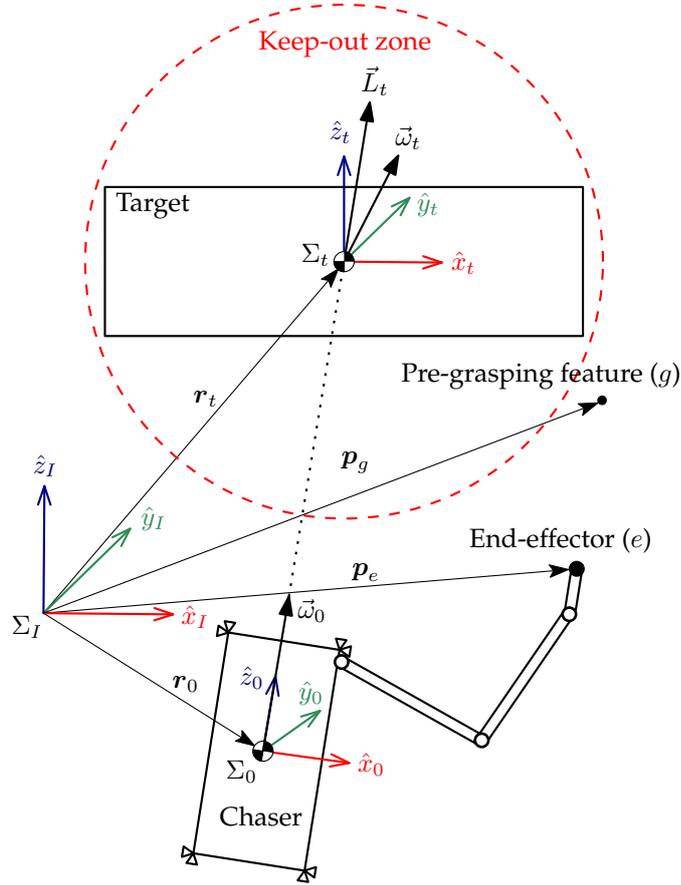


Figure 4.4: Illustration of the semisynchronizing approach problem ($\vec{L}_t =$ target angular momentum, $\vec{\omega}_t =$ target angular velocity, $\Sigma_t =$ target body frame, $r_t =$ target position, $p_g =$ pre-grasping feature position, $p_e =$ end-effector position, $\vec{\omega}_0 =$ angular velocity of the base, $\Sigma_I =$ inertial frame, $r_0 =$ base position, $\Sigma_0 =$ base body frame).

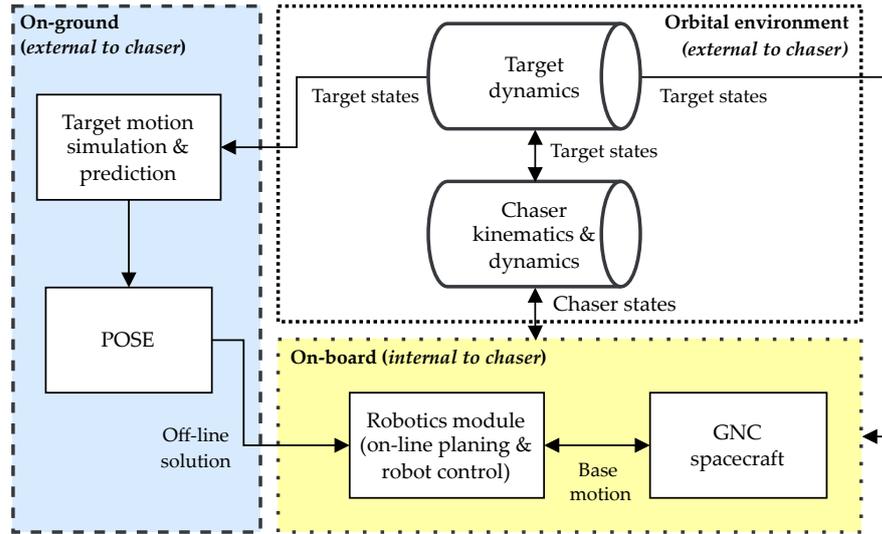


Figure 4.5: Implementation example of the motion planner (POSE) within a robotic control architecture of a chaser (GNC = guidance, navigation and control) (based on [11]).

consisting of robot configurations where the robot does not penetrate environmental obstacles and respects the joint limits. The motion planning can be expressed as finding a time t and a set of controls $u(t)$ such that the equations of motion of the robot satisfy $y(t_F) = y_F$ and $q(y(t)) \in \mathcal{C}_{\text{free}}$ for all $t \in [t_0, t_F]$, given an initial state $y(t_0) = y_0$ and a desired final state y_F of the robot [163].

Solving the motion planning problem does not have a unique answer and there are many methods to choose from, such as the *grid methods*, *sampling methods*, *virtual potential fields*, *nonlinear optimization*, etc [163]. The method used in this thesis to solve the semisynchronizing motion planning problem is the nonlinear optimization which converts the problem written as an OCP into an NLP and solves it using one of the numerical methods mentioned in Section 4.2.2. To this aim what follows is a definition of the semisynchronizing approach problem as an OCP using the formalism expressed in Section 4.2.1.

Assuming the state and control vectors (i.e. the differential and algebraic variables) of the problem defined as:

$$y(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} q_0 \\ q_m \\ \dot{q}_0 \\ \dot{q}_m \end{bmatrix}, \quad u(t) = \tau(t) = \begin{bmatrix} \tau_0 \\ \tau_m \end{bmatrix} \quad (4.76)$$

the state equations of the free-flying robot, expressed in Eq. (4.33), can be reformulated as:

$$\frac{d}{dt} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} \dot{q} \\ H^{-1}(q) [\tau - C(q, \dot{q}) \dot{q}] \end{bmatrix} \quad (4.77)$$

where $\dot{\boldsymbol{q}}$ is the vector of generalized velocities (see Eq. (4.28)).

These equations are complemented with the differential kinematics of the base

$$\dot{\boldsymbol{Q}}_0(t) = \frac{1}{2} \dot{\boldsymbol{Q}}_0 \otimes \dot{\boldsymbol{\omega}}_0 \quad (4.78)$$

which provide a relationship between $\dot{\boldsymbol{Q}}_0$ and $\boldsymbol{\omega}_0$.

The performance index is formulated as:

$$J = \int_{t_i}^{t_F} \boldsymbol{\tau}^T(t) \boldsymbol{W} \boldsymbol{\tau}(t) dt \quad (4.79)$$

where $\boldsymbol{W} \in \mathbb{R}^{6+n \times 6+n}$ is a positive-definite weight matrix. The chosen performance index can be interpreted as a “minimum energy” index which tends to uniquely define a smooth and “well-behaved” trajectory of a problem [167]. Additionally, it tends to filter large forces/torques which is generally a desired behavior in real robotic systems [169].

The algebraic path constraints of the problem are specified as a series of *no-contact constraints* on the distances between the keep-out zone around the target (see Fig. 4.4) and the spherical elements approximating the chaser:

$$d_i = (r_{c_i} + r_t) - \|\boldsymbol{p}_{c_i}(t) - \boldsymbol{p}_t\| \leq 0, \quad \forall i = 0, \dots, n \in \mathbb{R} \quad (4.80)$$

where $r_t \in \mathbb{R}$ is the radius of the keep-out zone of the target, $r_{c_i} \in \mathbb{R}$ is the radius of the i -th sphere approximating the chaser, $\boldsymbol{p}_{c_i}(t) \in \mathbb{R}^3$ and $\boldsymbol{p}_t \in \mathbb{R}^3$ are the position vectors of the spheres approximating the chaser and the keep-out zone of the target, respectively.

For an impact free capture, the final pose and spatial velocity of the end-effector, $\boldsymbol{T}_e(t_F)$ ³⁹ and $\boldsymbol{\mathcal{V}}_e(t_F)$, need to match that of the grasping feature of a target, ensuring that the end-effector can achieve the “capture” before contact and avoid velocity discontinuities that would generate high contact forces [97, 176]. During the manipulator approach phase there is no need for such conditions, as it does not involve physical contact (see Section 4.3.1) and therefore the final orientation and velocity of the end-effector are irrelevant. However, imposing such or similar conditions might ease the subsequent capture and stabilization phases. For this reason the OCP contains the boundary conditions on the position and velocity of the end-effector defined as:

$$\begin{aligned} \boldsymbol{T}_e(t_F) - \boldsymbol{T}_g(t_F) &= \mathbf{0} \\ \boldsymbol{\mathcal{V}}_e(t_F) - \boldsymbol{\mathcal{V}}_g(t_F) &= \mathbf{0} \end{aligned} \quad (4.81)$$

where $\boldsymbol{T}_g(t_F) \in \mathbf{SE}(3) \subset \mathbb{R}^{4 \times 4}$ and $\boldsymbol{\mathcal{V}}_g(t_F) \in \mathbb{R}^6$ are the final pose and spatial velocity of the pre-grasping feature, respectively.

³⁹ Where \boldsymbol{T} is a 4×4 homogeneous transformation matrix belonging to a special Euclidean group of dimensions three.

To comply with the initial and final states assumptions of the problem, defined in Section 4.3.1, additional boundary conditions are required and specified as bounds on the initial states:

$$\begin{aligned}
 \mathbf{q}_0(t_I) &= \mathbf{q}_{0,I} \\
 \mathbf{q}_m(t_I) &= \mathbf{q}_{m,I} \\
 \dot{\mathbf{q}}_0(t_I) &= \dot{\mathbf{q}}_{0,I} \\
 \dot{\mathbf{q}}_m(t_I) &= \dot{\mathbf{q}}_{m,I}
 \end{aligned} \tag{4.82}$$

as well as on the final states:

$$\begin{aligned}
 |\mathbf{q}_0(t_F)| &\leq \mathbf{q}_{0,F} + \boldsymbol{\varepsilon}_{q_0} \\
 |\mathbf{q}_m(t_F)| &\leq \mathbf{q}_{m,F} + \boldsymbol{\varepsilon}_{q_m} \\
 |\dot{\mathbf{q}}_0(t_F)| &\leq \dot{\mathbf{q}}_{0,F} + \boldsymbol{\varepsilon}_{\dot{q}_0} \\
 |\dot{\mathbf{q}}_m(t_F)| &\leq \dot{\mathbf{q}}_{m,F} + \boldsymbol{\varepsilon}_{\dot{q}_m}
 \end{aligned} \tag{4.83}$$

where $[\cdot]_{[\cdot,I]}$ and $[\cdot]_{[\cdot,F]}$ indicate constant, initial and final variables, respectively. $\boldsymbol{\varepsilon}_{[\cdot]}$ $\in \mathbb{R}^n$ is the vector of performance values of variables (e.g. \mathbf{q}_0 or $\dot{\mathbf{q}}_0$) and is used to define the final state space region of the solution.

Finally, the state and control variables of the solution need to be within certain limits during the maneuver, due to the physical limitations of joint actuators and thrusters as well as other desired constraints, such as the interval of variation of the angular velocity of the base during the maneuver. These limits are formulated as the bounds on the state variables:

$$\begin{aligned}
 |\mathbf{q}_0(t)| &\leq \mathbf{q}_{0,U} \\
 |\mathbf{q}_m(t)| &\leq \mathbf{q}_{m,U} \\
 |\dot{\mathbf{q}}_0(t)| &\leq \dot{\mathbf{q}}_{0,U} \\
 |\dot{\mathbf{q}}_m(t)| &\leq \dot{\mathbf{q}}_{m,U}
 \end{aligned} \tag{4.84}$$

as well as on the control variables:

$$\begin{aligned}
 |\boldsymbol{\tau}_0(t)| &\leq \boldsymbol{\tau}_{0,U} \\
 |\boldsymbol{\tau}_m(t)| &\leq \boldsymbol{\tau}_{m,U}
 \end{aligned} \tag{4.85}$$

where $[\cdot]_{[\cdot,U]}$ indicates constant, upper bounds of variables.

To conclude, the OCP formulation of the entire problem of the semisynchronizing approach maneuver can be summarized as:

$$\begin{aligned}
& \text{find} && \mathbf{u}(t) = \boldsymbol{\tau}^T(t), \mathbf{y}(t) = \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}, && \forall t \in [t_I, t_F] \\
& \text{minimizing} && J = \int_{t_I}^{t_F} \boldsymbol{\tau}^T(t) \mathbf{W} \boldsymbol{\tau}(t) dt \\
& \text{subject to} && \dot{\mathbf{y}}(t) = \frac{d}{dt} \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{H}^{-1}(\mathbf{q}) [\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}] \end{bmatrix} \\
& && \dot{\mathbf{Q}}_0(t) = \frac{1}{2} \dot{\mathbf{Q}}_0 \otimes \dot{\boldsymbol{\omega}}_0 \\
& && (r_{c_i} + r_t) - \|\mathbf{p}_{c_i}(t) - \mathbf{p}_t\| \leq \mathbf{0} \\
& && \mathbf{T}_e(t_F) - \mathbf{T}_g(t_F) = \mathbf{0} \\
& && \mathbf{V}_e(t_F) - \mathbf{V}_g(t_F) = \mathbf{0} \\
& && \mathbf{y}(t_I) = \mathbf{y}_I \\
& && |\mathbf{y}(t_F)| \leq \mathbf{y}_F + \varepsilon_y \\
& && |\mathbf{y}(t)| \leq \mathbf{y}_U \\
& && |\mathbf{u}(t)| \leq \mathbf{u}_U
\end{aligned} \tag{4.86}$$

4.3.3 Optimal control problem implementation

The OCP expressed in Eq. (4.86) is in general difficult to solve due to, among other things, the nonlinear dynamics of the chaser and rotational kinematics of the base. Additionally, the problem as formulated presents one major issue due to a fundamental numerical problem associated with a rotational kinematics model based on unit quaternions. A numerical integration of Eq. (4.78) without any additional constraints will most likely lead to significant violation of the normalization condition of the resulting quaternion (see Eq. (4.12)) unless it is integrated *very* accurately. In other words, even if $\|\mathbf{Q}_0(t_0)\| = 1$, due to the integration errors, one must expect that $\|\mathbf{Q}_0(t_F)\| \neq 1$, which will lead to ill-conditioning and/or slow convergence of the solution of the OCP [167]. To obviate this issue the implemented solution consists in adding to the overall problem expressed in Eq. (4.86) an algebraic constraint that explicitly assures the unit-length condition of \mathbf{Q}_0

$$\|\dot{\mathbf{Q}}_0\| - 1 = 0 \tag{4.87}$$

and substituting Eq. (4.78) with a simplified one-to-one mapping between $\boldsymbol{\omega}_0$ and $\dot{\mathbf{Q}}_0$

$$\dot{\mathbf{Q}}_0 = \frac{1}{2} \boldsymbol{\Omega}_0 \boldsymbol{\omega}_0 \tag{4.88}$$

where the matrix Ω_0 is defined in Eq. (4.24) [167]. Doing so the differential and algebraic variables of the OCP (see Eq. (4.76)) are redefined as:

$$\mathbf{y}(t) = \begin{bmatrix} \tilde{\mathbf{q}}_0 \\ \mathbf{q}_m \\ \dot{\mathbf{q}}_0 \\ \dot{\mathbf{q}}_m \end{bmatrix}, \quad \tilde{\mathbf{u}}(t) = \begin{bmatrix} \eta_0 \\ \boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_m \end{bmatrix} \quad (4.89)$$

where $\tilde{\mathbf{q}}_0 = [\mathbf{r}_0^\top, \epsilon_0^\top]^\top$. The problem re-defined in this manner, explicitly ensures that the normalization condition of the unit quaternion is respected at every grid point of the transcribed NLP, even in case of a coarse mesh. This leads to a more robust implementation of the method w. r. t. what was defined in Eq. (4.86) [167] with one caveat. A discontinuous behavior of Eq. (4.88) is to be expected for large attitude maneuvers of the base. In those cases the angle characterizing the attitude θ_0 will most certainly cross the values of $\pm\pi$, at which Ω_0 does not have a full rank (see Eq. (4.25)) [165]. This presents difficulties during the solution of related NLPs as all the numerical methods described in Section 4.2.2 assume that the user-defined functions are continuous and differentiable.

The approximation of an object using spherical elements, for collision detection purposes, must in general cover all points of the object as a union of overlapping spheres. The greater the number of spheres, the better the approximation of the actual geometry of the object [163]. However, considering that in general optimization problems having path constraints tend to be much more difficult to solve than those without [167], it can be reasonably expected that with the increase of the number of spheres the convergence of the solver will be negatively impacted. Therefore, in these cases a careful trade-off needs to be made between the number of spherical elements approximating the object and desired convergence speed. Taking into account the geometry of the semisynchronizing approach problem, illustrated in Fig. 4.4, and the related OCP, it can be expected within reason that during the desired “point-to-point” motion not all bodies constituting the chaser will be able to penetrate the keep-out zone of the target. With this in mind, for the implementation of the no-contact constraints expressed in Eq. (4.80), only the “most critical bodies”⁴⁰ of the chaser (i. e. the base, “shoulder” and “elbow” joints as well as the end-effector of the manipulator) are considered, thus keeping the number of the overall path constraints to a minimum while still ensuring a reasonable convergence speed and a collision-free path.

⁴⁰ From the perspective of “collision” with the keep-out zone.

Another issue that in general negatively impacts the solution of an OCP is its *initial guess* required by most of the state-of-the-art NLP solvers. For easy to solve problems, i. e. those having a unique global solution, a wide range of initial guesses will ensure that the solver arrives at the solution with ease. In case of difficult problems,

characterized by multiple local solutions, a poor initial guess might cause the solver to get “stuck” on a bad solution or even fail to solve an otherwise correctly formulated optimization problem [169]. For this reason, the simplest initialization techniques, such as the straight line guesses in the state and control⁴¹, were found not to be sufficient for the OCP at hand and a more methodical approach was adopted relying on the specific knowledge of the system in question.

⁴¹ *Between their initial and final values.*

Assuming the ACS of the chaser is able to completely cancel-out the dynamical coupling between the base and manipulator during the motion of the latter, the chaser can be treated as a fixed-base robot. This allows to calculate the initial guess trajectories of the base and manipulator state variables separately and as follows. Given the desired initial states of the chaser and the overall maneuver time, the initial guess trajectory of the base is obtained by propagating in time Eq. (4.71) in the absence of external torque applied to the base. The trajectory of the manipulator is instead derived by assuming a straight-line motion from a desired initial state to a desired final state in either configuration (i. e. joint) space or task (i. e. Cartesian) space, depending on the user selection. In case of a joint-space motion, which might be desired when a singularity-free trajectory is required, the position and velocity joint trajectories are obtained by a polynomial interpolation between the initial and final states. In contrast, in case of a Cartesian motion, which might be desired when a straight-line motion in task space is required, the joint trajectories of the manipulator are obtained by first interpolating the pose of the end-effector between its known initial and final states and then using an inverse kinematics algorithm to determine the corresponding joint trajectories. The initial guess of the control variables of the base and manipulator is assumed in both cases to be constant at zero.

The initial guess formulated in this manner does not satisfy the system dynamics or most of the other constraints of the OCP. However, it provides the solver with a guess that is “close enough” to the desired behavior of the chaser in order to direct it towards the desired, “correct” solution [169].

The software implementation of the described motion planner and related OCP is implemented within the POSE software library (see Fig. 4.6).

The library is developed using the MATLAB® programming language and the homonym computing environment (version 9.3.0) [177] extended using the add-ons described hereafter.

The Robotics System Toolbox™ (version 1.5) [178] is used to calculate feasible inverse kinematics solutions of the robot modeled with a unified robot description format (URDF). The same Toolbox is also used to display the calculated initial guess trajectories in a graphical representation, useful to identify anomalous guesses.

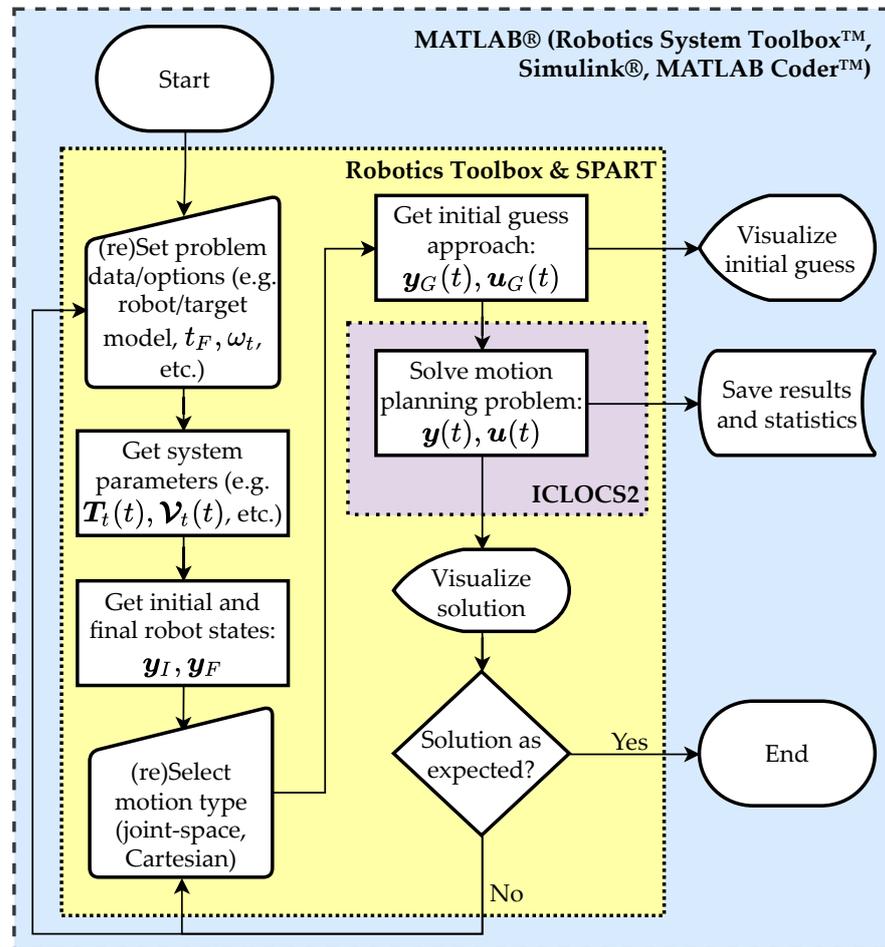


Figure 4.6: Illustration of the POSE algorithm (refer to the ISO 5807:1985 standard [134] for the definition of flowchart symbols; t_F = final time, ω_t = target angular velocity, T_t = target pose, V_t = target spatial velocity, y_I = initial state vector, y_F = final state vector, y_G = state vector guess, u_G = control vector guess, y = state vector, u = control vector).

The MATLAB Coder™ (version 3.4) [179] is employed to improve the computational efficiency of the developed code by converting it from the MATLAB code to a standalone C/C++ code and packaging it as a MEX-function for an easier integration within the existing non C/C++ code.

Simulink® (version 9.0) [180] and Simscape™ Multibody™ (version 5.1) [181] are utilized to provide a multibody simulation environment and generate 3D animations of the approach maneuver based on the solution of the OCP.

The Spacecraft Robotics Toolkit (SPART) (version 0.2) is used to derive the kinematic and dynamic equations of the free-flying chaser based on its URDF model. The Toolbox developed by Virgili-Llop [182], is an open-source modeling and control software for mobile-base robotic multibody systems with kinematic tree typologies. It was successfully evaluated on a planner hardware-in-the-loop (HIL) testbed to guide, in real-time, a robotic capture maneuver of a rotating target [105].

The open-source Robotics Toolbox (version 10.4) developed by Corke [161] and the “quaternion” class (version 1.8.0.0) developed by Tincknell [183] are used for manipulating and converting between different orientation representations, such as unit quaternions, quaternions, homogeneous transformations, etc. Additionally, both add-ons are employed for the generation of the initial guess trajectories. Namely, the Robotics Toolbox is used for the interpolation of the joint trajectories (both in the joint and task space) while the “quaternion” class is employed for the numerical propagation of the rotational motion of the base using Euler’s equations of motion formulated in Eq. (4.71).

The open-source Imperial College London Optimal Control Software (ICLOCS) (version 2.5.1 beta) developed by Nie, Faqir, and Kerrigan [184] is the software used for the implementation and solution of the OCP. It provides easy access to a variety of transcription and discretization methods and contains automated tools to improve the efficiency of the solver (within the specified accuracy) and display the obtained solution. The Toolbox also provides an API to several state-of-the-art NLP solvers, such as the open-source Interior Point Optimizer (IPOPT) (version 3.12.7) [185], which is the one employed in this work to solve the transcribed OCP via an interior-point algorithm with a filter line-search method.

4.4 NUMERICAL SIMULATIONS

POSE was evaluated in 3D simulation environment using the “crawl, walk, run” methodology consisting of different test cases, each progressively more difficult in terms of the number of active joints of the manipulator and higher angular rate of the target. Both the chaser and

target inertial characteristics are shared among all test cases in order to allow comparison among the obtained results. The following is a description of the parameters used in the test cases and illustration of the most important results.

4.4.1 Simulation test cases

⁴² Defined as a mass of both the spacecraft (known as the dry mass) and the propellant.

The chaser is modeled after the e.Deorbit spacecraft [104]. It has a wet mass⁴² of 1660 kg and is shaped as a parallelepiped, without protruding solar panels, having a length of 3 m, a width of 1.5 m and a height of 1.5 m. The safety sphere of the base spacecraft has a radius $r_{c_0} = 1.84$ m which corresponds to a half of the diagonal of the parallelepiped. The on-board manipulator is a 7 DOF robotic arm kinematically resembling that of the e.Deorbit spacecraft [104] with slightly increased length to conform with the semisynchronizing approach maneuver. It has a stretched length of 5.07 m and a mass of 80 kg. An illustration of the model of the chaser is depicted in Fig. 4.7. The principle moments of inertia of both the base spacecraft and manipulator are summarized in Table 4.1, where for simplicity the manipulator links are approximated with circular cylindrical shells.

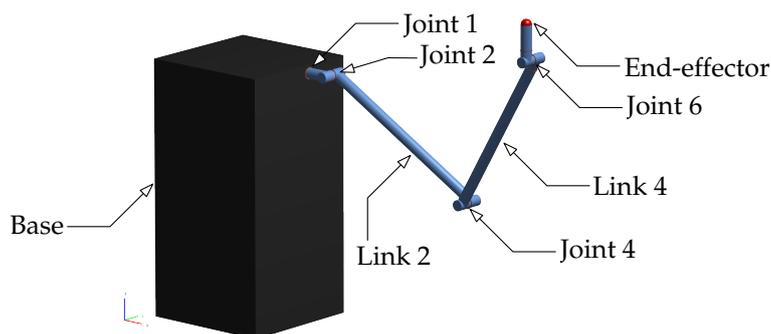


Figure 4.7: Model of a chaser spacecraft used in numerical simulations.

⁴³ 1435 kg + 20 % of mass uncertainty margin.

The target is a Kosmos-3M second stage (see Fig. 4.8) [186] approximated for simplicity as a solid circular cylinder having a mass of 1722 kg⁴³, a length of 6.5 m, a diameter of 2.4 m and principle moments of inertia summarized in Table 4.1. The radius of its keep-out zone is set to $r_t = 3.518$ m and it includes a margin of 0.03 m.

The reason for choosing this particular target stems first from it being on the very top of the list of potential ADR targets in LEO that should be considered for removal [26, 187]. Second, it presents physical properties that are compatible with the selected chaser and MRAVS semisynchronizing capture strategy.

The target is assumed to have no linear motion and three different attitude motions, each characterized by an angular velocity having dominating components around the axes of maximum inertia of the target body, i. e. its y and z axes. This is consistent with a torque-free

Table 4.1: Inertial parameters of the chaser and target modeled according to [104, 186], respectively.

Body	Mass (kg)	I_{xx} (kg m ²)	I_{yy} (kg m ²)	I_{zz} (kg m ²)
Base	1580	1481.2	1481.2	592.48
Link 1	10	0.08	0.08	0.04
Link 2	17	6.62	0.07	6.62
Link 3	10	0.03	0.03	0.04
Link4	16	5.29	0.06	5.29
Link 5	10	0.03	0.03	0.04
Link 6	10	0.03	0.04	0.03
Link 7	7	0.06	0.06	0.03
Target	1722	1239.8	6776.2	6776.2



Figure 4.8: Mock-up of a Kosmos-3M second stage carrying a model of a German SAR-Lupe reconnaissance satellite (credit: Wikimedia and reprinted from [11]).

motion of a rigid body under the action of orbital perturbation forces such as Eddy currents [188].

The values of the angular velocities of the target assumed in tests are

$${}^t\boldsymbol{\omega}_t = [0, 1.58, 4.74]^T \text{ deg s}^{-1} \quad (4.90)$$

$$= [0, 3.64, 10.91]^T \text{ deg s}^{-1} \quad (4.91)$$

$$= [0, 5.69, 17.08]^T \text{ deg s}^{-1} \quad (4.92)$$

which correspond to angular rates of

$$\|{}^t\boldsymbol{\omega}_t\| = 5 \text{ deg s}^{-1} \quad (4.93)$$

$$= 11.5 \text{ deg s}^{-1} \quad (4.94)$$

$$= 18 \text{ deg s}^{-1} \quad (4.95)$$

that cover the end points as well as the middle point of the interval $\omega_t = [5, 18] \text{ deg s}^{-1}$ defined in Section 4.3.1.

⁴⁴ *w. r. t. the base spacecraft.*

The pose parameters of the target (both relative⁴⁴ and inertial) used in tests are

$${}^0\mathbf{r}_t(t_I) = [0, 0, 5.56]^T \text{ m}, \quad {}^0\mathring{\mathbf{Q}}_t(t_I) = [1, 0, 0, 0]^T \quad (4.96)$$

$$\mathbf{r}_t(t_F) = [0, 0, 5.56]^T \text{ m}, \quad \mathring{\mathbf{Q}}_t(t_F) = [0.71, 0, 0, 0.71]^T \quad (4.97)$$

$${}^t\mathbf{p}_g = [3.38, 0, -1.4]^T \text{ m}, \quad {}^t\mathring{\mathbf{Q}}_g = [1, 0, 0, 0]^T \quad (4.98)$$

where ${}^t\mathbf{p}_g$ and ${}^t\mathring{\mathbf{Q}}_g$ are respectively the position and orientation of the pre-grasping feature of the target relative to its body frame (see Fig. 4.4). The reason for specifying only the relative initial pose of the target instead of the inertial one (see Eq. (4.96)) is due to a fact that a combination of a limited maneuver time, attitude motion of a target and the specific synchronizing strategy of the chaser results in a limited final reachable space of the manipulator. Therefore, instead of starting from defined inertial pose of the target and searching for its “optimal” capture pose (which might as well not exist), the approach implemented in this work consists in specifying the desired initial relative pose of the target along with its final inertial pose and propagating the latter backward in time, under the assumption of a torque-free, to obtain its “optimal” starting inertial pose.

The time parameters assumed for all tests are

$$t_I = 0 \text{ s}, \quad t_F = 10 \text{ s} \quad (4.99)$$

in line with those assigned in similar studies (e. g. [97, 176]) to the approach phase of the overall capture maneuver of a tumbling target. Increasing t_F and thus the maneuver time might be tempting, however, as noted by Virgili-Llop et al. [105], shorter maneuver times such as the one considered, might be preferred due to the lower computational load allowing for higher refresh rates and increased robustness

of the maneuver to uncertainties and perturbations not considered in numerical models. In addition, considering the partial nature of MRAVS and high angular rates of the target (see Eqs. (4.90) to (4.92)), a shorter maneuver is to be preferred in order to decrease the required movement ranges of the base spacecraft and manipulator.

4.4.1.1 3 DOF manipulator test cases

The first set of test cases is characterized by a chaser spacecraft having a robotic arm with only three active joints, i. e. joints one, two and four (see Fig. 4.7). This allows it to satisfy at maximum three boundary conditions of the end-effector out of the overall six defined in Eq. (4.81). Therefore, while these tests are not able to entirely satisfy the constraints of the OCP (see Eq. (4.86)), they present the advantage of a smaller number of decision variables which benefits greatly the computation time needed to find a solution. For this reason these tests were used as the first benchmark and proof of concept of the developed motion planner.

What follows is a description of the main parameters and options used to define the 3 DOF test cases in ICLOCS, based on the formalism specified in Sections 4.3.2 and 4.3.3.

Starting with the objective function, the positive-definite weight matrix of Eq. (4.79) is defined as:

$$W = I_{9 \times 9} \quad (4.100)$$

where $I_{9 \times 9} \in \mathbb{R}^{9 \times 9}$ is an identity matrix.

The number of algebraic no-contact constraints defined in Eq. (4.80) is constrained to five, corresponding to five collision-free distances between the keep-out zone of the target and the spherical elements approximating the base, the second joint, the fourth joint, the sixth joint and the end-effector of the manipulator. The radii of the spherical elements of joints and the end-effector are set to 0.18 m and 0.08 m, respectively, based on the maximum dimensions of considered joints and the link preceding the end-effector.

The final boundary conditions on the pose of the end-effector defined in Eq. (4.81) are modified to accommodate the reduced capability of the manipulator as follows:

$$\begin{aligned} \mathbf{p}_e(t_F) - \mathbf{p}_g(t_F) &= \mathbf{0} \\ \mathbf{v}_e(t_F) - \mathbf{v}_g(t_F) &= \mathbf{0} \end{aligned} \quad (4.101)$$

The boundary conditions on the initial and final states of the chaser are mainly functions of the attitude and angular velocity of the target. Therefore, each test case is characterized by a unique set of boundary conditions calculated at runtime and summarized in Table 4.2.

Table 4.2: Bounds on initial and final states of 3 DOF tests.

Parameter	$\omega_t = 5$ (deg s ⁻¹)	$\omega_t = 11.5$ (deg s ⁻¹)	$\omega_t = 18$ (deg s ⁻¹)
$r_{0,I}$ (m)	[1.76, 0, 0.29] ^T		
$\mathcal{Q}_{0,I}$	[0.93, -0.06, -0.15, 0.34] ^T	[0.96, 0.04, -0.16, -0.21] ^T	[0.7, 0.11, -0.11, -0.7] ^T
$\mathbf{q}_{m,I}$ (deg s ⁻¹)	[0, 30, 60] ^T		
$v_{0,I}$ (m s ⁻¹)	[0, 0, 0] ^T		
$\omega_{0,I}$ (deg s ⁻¹)	[0, 0, 5] ^T	[0, 0, 11.5] ^T	[0, 0, 18] ^T
$\dot{\mathbf{q}}_{m,I}$ (deg s ⁻¹)	[0, 0, 0] ^T		
$\mathbf{q}_{0,F}$ (m, N/A)	[1.76, 0, 0.29, 0.7, -0.11, -0.11, 0.7] ^T		
$\mathbf{q}_{m,F}$ (deg)	[-8.94, -7.68, 79.9] ^T		
$v_{0,F}$ (m s ⁻¹)	[0, 0, 0] ^T		
$\omega_{0,F}$ (deg s ⁻¹)	[0, 0, 5] ^T	[0, 0, 11.5] ^T	[0, 0, 18] ^T
$\dot{\mathbf{q}}_{m,F}$ (deg s ⁻¹)	[0, 0, 0] ^T		

The definition of the final state bounds (see Eq. (4.83)) is completed by specifying the vectors of performance values of final states, $\varepsilon_{[\cdot]}$, as follows:

$$\begin{aligned}\varepsilon_{q_0} &= [0.1, 0.1, 0.1, 0, 0, 0, 0]^T \text{ m, N/A} \\ \varepsilon_{q_m} &= [10, 10, 10]^T \text{ deg} \\ \varepsilon_{\dot{q}_0} &= [0.01, 0.01, 0.01, 0.1, 0.1, 0.1]^T \text{ m s}^{-1}, \text{ deg s}^{-1} \\ \varepsilon_{\dot{q}_m} &= [5, 5, 5]^T \text{ deg s}^{-1}\end{aligned}\tag{4.102}$$

where the vector components related to the base position, linear and angular velocities are defined according to the typical performance values of a GNC system of a spacecraft in a berthing box [34].

The state and control bounds/limits are defined as:

$$\begin{aligned}q_{0,U} &= [\infty, \infty, \infty, 1, 1, 1, 1]^T \text{ m, N/A} \\ q_{m,U} &= [160, 160, 160]^T \text{ deg} \\ \dot{q}_{0,U} &= [\infty, \infty, \infty, 1, 1, 1]^T \text{ m s}^{-1}, \text{ deg s}^{-1} \\ \dot{q}_{m,U} &= [30, 30, 30]^T \text{ deg s}^{-1} \\ \tau_{0,U} &= [88, 88, 88, 132, 132, 66]^T \text{ N, N m} \\ \tau_{m,U} &= [80, 80, 80]^T \text{ N m}\end{aligned}\tag{4.103}$$

where the symbol ∞ indicates no bounds on that specific variable and is used in case of position and linear velocity of the base in order not to over constraint the OCP. The limits on q_m and τ_m follow closely those of the manipulator of the e.Deorbit's chaser while the limits on \dot{q}_m have been increased from 10 deg s^{-1} to 30 deg s^{-1} ⁴⁵ to better accommodate the considered maneuver. The limits on τ_0 are the result of the assumption that the spacecraft is equipped with $24 \times 22 \text{ N}$ attitude thrusters grouped in eight clusters of three, positioned on the corners of the base.

⁴⁵ w. r. t. that of the e.Deorbit's chaser [104].

Finally, the main configuration options imposed within ICLOCS for solving the OCP with the previously defined parameters are:

INITIAL GUESS MOTION TYPE: joint-space.

DISCRETIZATION SCHEME: Hermite-Simpson method.

INITIAL GRID POINTS: 20.

DERIVATIVE GENERATION: numeric.

NLP SOLVER: IPOPT.

IPOPT CONVERGENCE TOLERANCE: 1×10^{-7} .

IPOPT μ STRATEGY: adaptive.

MESH METHOD: *automatic, aggressive* mesh refinement.

INITIAL GUESS TYPE: warm.

4.4.1.2 7 DOF manipulator test cases

The second set of test cases consists of a chaser having all seven joints of the manipulator active allowing it to satisfy all six boundary conditions of the end-effector specified in Eq. (4.81) with one degree of redundancy.

What follows is a description of the main parameters and options used to define the 7 DOF test cases in ICLOCS, based on the formalism specified in Sections 4.3.2 and 4.3.3.

Starting with the objective function, the positive-definite weight matrix of Eq. (4.79) is defined as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{I}_{10 \times 10} & \mathbf{0}_{10 \times 3} \\ \mathbf{0}_{3 \times 10} & 100 \times \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (4.104)$$

where the weight of the last three joints of the manipulator is increased from 1 to 100 (w. r. t. those used in 3 DOF tests) to limit the maximum torque and by consequence joint accelerations which were found to be quite high during early testing phases due to the low inertial properties of the corresponding links.

The number of the imposed algebraic no-contact constraints defined in Eq. (4.80) is the same as in case of 3 DOF tests, i. e. equal to five, corresponding to five collision-free distances between the keep-out zone of the target and the spherical elements approximating the base, the second joint, the fourth joint, the sixth joint and the end-effector of the manipulator, respectively. The radii of the spherical elements of joints and the end-effector are set to 0.18 m and 0.08 m, respectively, based on the maximum dimensions of considered joints and the link preceding the end-effector.

The boundary conditions on the initial and final states of the chaser are mainly functions of the attitude and angular velocity of the target. Therefore, as in case of the first set of test cases in Section 4.4.1.1, each test case is characterized by a unique set of boundary conditions calculated at runtime and summarized in Table 4.3.

The definition of the final state bounds (see Eq. (4.83)) is completed by specifying the vectors of performance values of final states, $\boldsymbol{\varepsilon}_{[\cdot]}$, as following:

$$\begin{aligned} \boldsymbol{\varepsilon}_{q_0} &= [0.1, 0.1, 0.1, 0, 0, 0, 0]^T \text{ m, N/A} \\ \boldsymbol{\varepsilon}_{q_m} &= [10, 10, 10, 10, 10, 10, 10]^T \text{ deg} \\ \boldsymbol{\varepsilon}_{\dot{q}_0} &= [0.01, 0.01, 0.01, 0.1, 0.1, 0.1]^T \text{ m s}^{-1}, \text{ deg s}^{-1} \\ \boldsymbol{\varepsilon}_{\dot{q}_m} &= [5, 5, 5, 5, 5, 5, 5]^T \text{ deg s}^{-1} \end{aligned} \quad (4.105)$$

where, as in the 3 DOF tests, the vector components related to the base position, linear and angular velocities are specified according to the typical performance values of a GNC system of a spacecraft in a berthing box [34].

Table 4.3: Bounds on initial and final states of 7 DOF tests.

Parameter	$\omega_t = 5$ (deg s ⁻¹)	$\omega_t = 11.5$ (deg s ⁻¹)	$\omega_t = 18$ (deg s ⁻¹)
$r_{0,I}$ (m)	[1.76, 0, 0.29] ^T		
$\mathcal{Q}_{0,I}$	[0.93, -0.06, -0.15, 0.34] ^T	[0.96, 0.04, -0.16, -0.21] ^T	[0.7, 0.11, -0.11, -0.7] ^T
$q_{m,I}$ (deg s ⁻¹)	[0, 30, 0, 60, 0, 60, 0] ^T		
$v_{0,I}$ (m s ⁻¹)	[0, 0, 0] ^T		
$\omega_{0,I}$ (deg s ⁻¹)	[0, 0, 5] ^T	[0, 0, 11.5] ^T	[0, 0, 18] ^T
$\dot{q}_{m,I}$ (deg s ⁻¹)	[0, 0, 0, 0, 0, 0] ^T		
$q_{0,F}$ (m, N/A)	[1.76, 0, 0.29, 0.7, -0.11, -0.11, 0.7] ^T		
$q_{m,F}$ (deg)	[-58.12, -26.01, 70.48, 78.2, 51.44, 9.42, -75.69] ^T		
$v_{0,F}$ (m s ⁻¹)	[0, 0, 0] ^T		
$\omega_{0,F}$ (deg s ⁻¹)	[0, 0, 5] ^T	[0, 0, 11.5] ^T	[0, 0, 18] ^T
$\dot{q}_{m,F}$ (deg s ⁻¹)	[0, 0, 0, 0, 0, 0] ^T		

The state and control bounds/limits are defined as:

$$\begin{aligned}
\mathbf{q}_{0,U} &= [\infty, \infty, \infty, 1, 1, 1, 1]^T \text{ m, N/A} \\
\mathbf{q}_{m,U} &= [160, 160, 160, 160, 160, 160, 160]^T \text{ deg} \\
\dot{\mathbf{q}}_{0,U} &= [\infty, \infty, \infty, 1, 1, 1]^T \text{ m s}^{-1}, \text{ deg s}^{-1} \\
\dot{\mathbf{q}}_{m,U} &= [30, 30, 30, 30, 30, 30, 30]^T \text{ deg s}^{-1} \\
\boldsymbol{\tau}_{0,U} &= [88, 88, 88, 132, 132, 66]^T \text{ N, N m} \\
\boldsymbol{\tau}_{m,U} &= [80, 80, 80, 80, 80, 80, 80]^T \text{ N m}
\end{aligned} \tag{4.106}$$

where the same considerations are valid as in case of 3 DOF test cases.

Finally, the main configuration options imposed within ICLOCS for solving the OCP with the previously defined parameters are the same as in 3 DOF test cases (see Section 4.4.1.1).

4.4.2 Results

In this section the results and performance values of the numeric simulations defined in Section 4.4.1 are illustrated focusing especially on the test case characterized by $\omega_t = 18 \text{ deg s}^{-1}$, as deemed the most challenging scenario of the semisynchronizing approach problem, due to the high angular rate of the target. The results were obtained running all test cases simultaneously, using separate instances of the MATLAB environment, on a 64-bit PC platform equipped with an Intel® Xeon® X5650 CPU, clocked at 2.67 GHz, and 24 GiB of RAM. This resulted in an average CPU and RAM utilization of 34 % and 46 %, respectively.

4.4.2.1 3 DOF manipulator test results

The initial configuration of the chaser with a 3 DOF manipulator and target having $\omega_t = 18 \text{ deg s}^{-1}$ is illustrated in Fig. 4.9.

The optimal trajectories of \mathbf{q} , $\dot{\mathbf{q}}$ and $\boldsymbol{\tau}$, obtained as a solution to the OCP, are shown in Figs. 4.10 to 4.12. For an easier interpretation of the results, the history of the base rotation, depicted in Fig. 4.10b, is represented as a continuous function of time of a 3-angle, roll-pitch-yaw (XYZ), sequence, $\boldsymbol{\Gamma}_0 = [\phi_0, \theta_0, \psi_0]^T$.

The time history of the boundary conditions of the OCP expressed in Eq. (4.101) is shown in Fig. 4.13 in terms of $\|\mathbf{p}_g(t_f) - \mathbf{p}_e(t)\|$ and $\|\mathbf{v}_g(t_f) - \mathbf{v}_e(t)\|$.

The time profiles of the algebraic no-contact constraints of the OCP outlined in Eq. (4.80) are illustrated in Fig. 4.14 as the time history of the collision distances between the five spherical elements of the chaser and the keep-out zone of the target.

Fig. 4.15 shows the comparison of $\dot{\mathbf{q}}_m$ trajectories of all 3 DOF test cases.

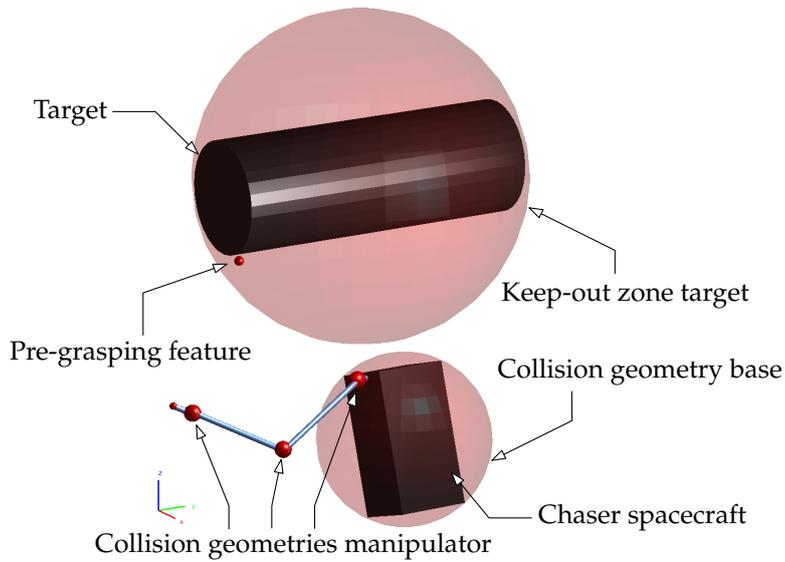


Figure 4.9: Initial configuration of the chaser and target in case of 3 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$.

Table 4.4: Performance summary of 3 DOF tests (min. = minimized, ref. = refinement, iter. = iteration(s), FE = function evaluations).

Parameter	5 deg s^{-1}	11.5 deg s^{-1}	18 deg s^{-1}
Min. objective	3.98×10^3	6.70×10^3	2.03×10^4
Mesh-ref. iter.	4	2	2
1st mesh iter.	17	17	21
Total iter.	64	32	44
CPU time (s)	2.34×10^3	1.32×10^3	3×10^3
Time for FE (%)	99.67	99.65	99.73

The performance of the planner in case of all 3 DOF test cases is summarized in Table 4.4 featuring the final values of the minimized objective function, number of mesh refinement iterations, number of iterations within the first mesh, the total number of iterations, the total computational time employed and the percentage of that time used to perform function evaluations.

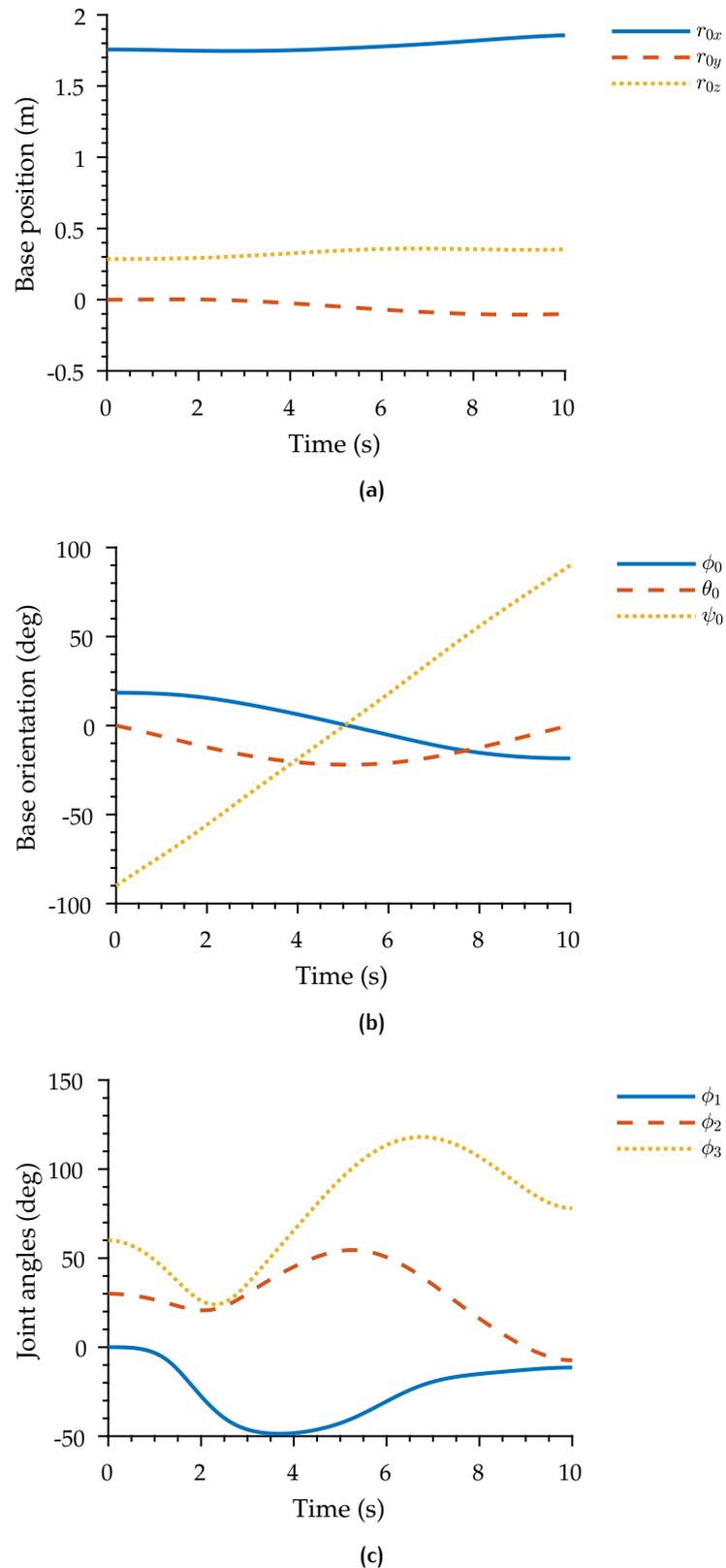
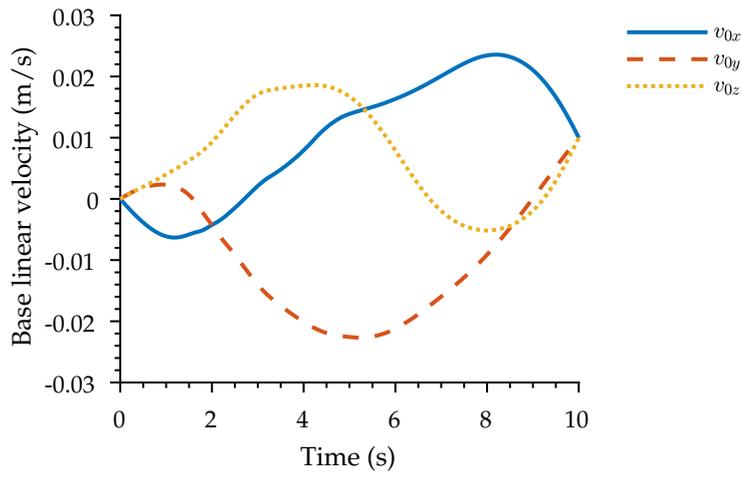
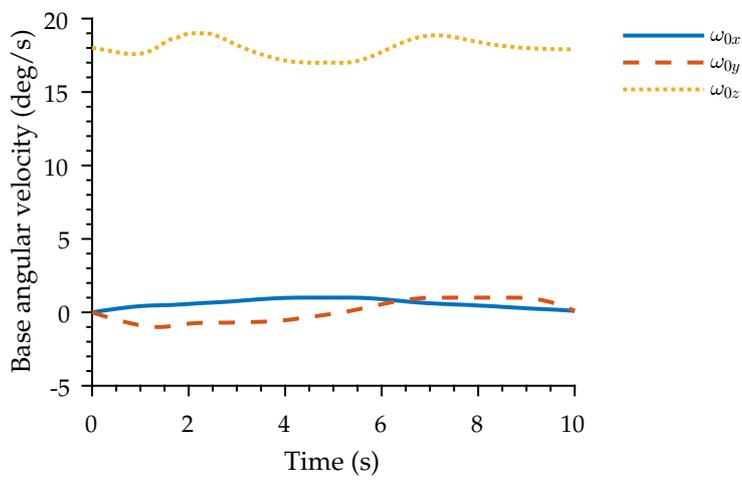


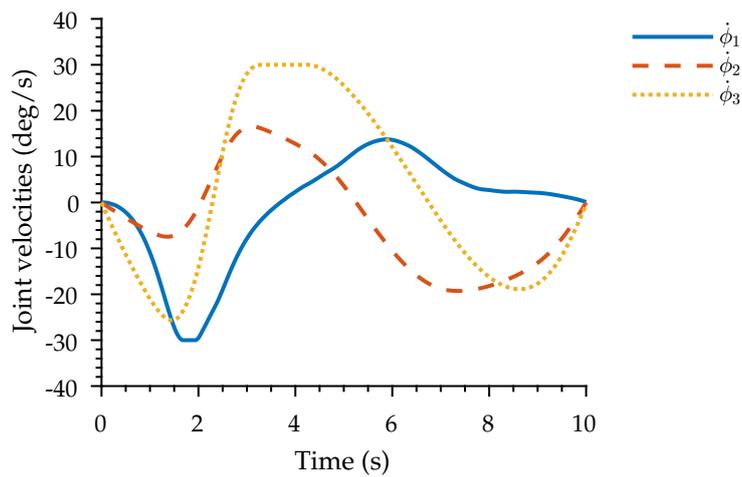
Figure 4.10: Optimal trajectories of the generalized coordinates of the base and manipulator (q_0 and q_m) in case of 3 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (the base orientation is expressed in a roll-pitch-yaw (XYZ) angle sequence).



(a)



(b)



(c)

Figure 4.11: Optimal trajectories of the time derivatives of generalized coordinates of the base and manipulator (\dot{q}_0 and \dot{q}_m) in case of 3 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (base angular velocity is expressed in Σ_0).

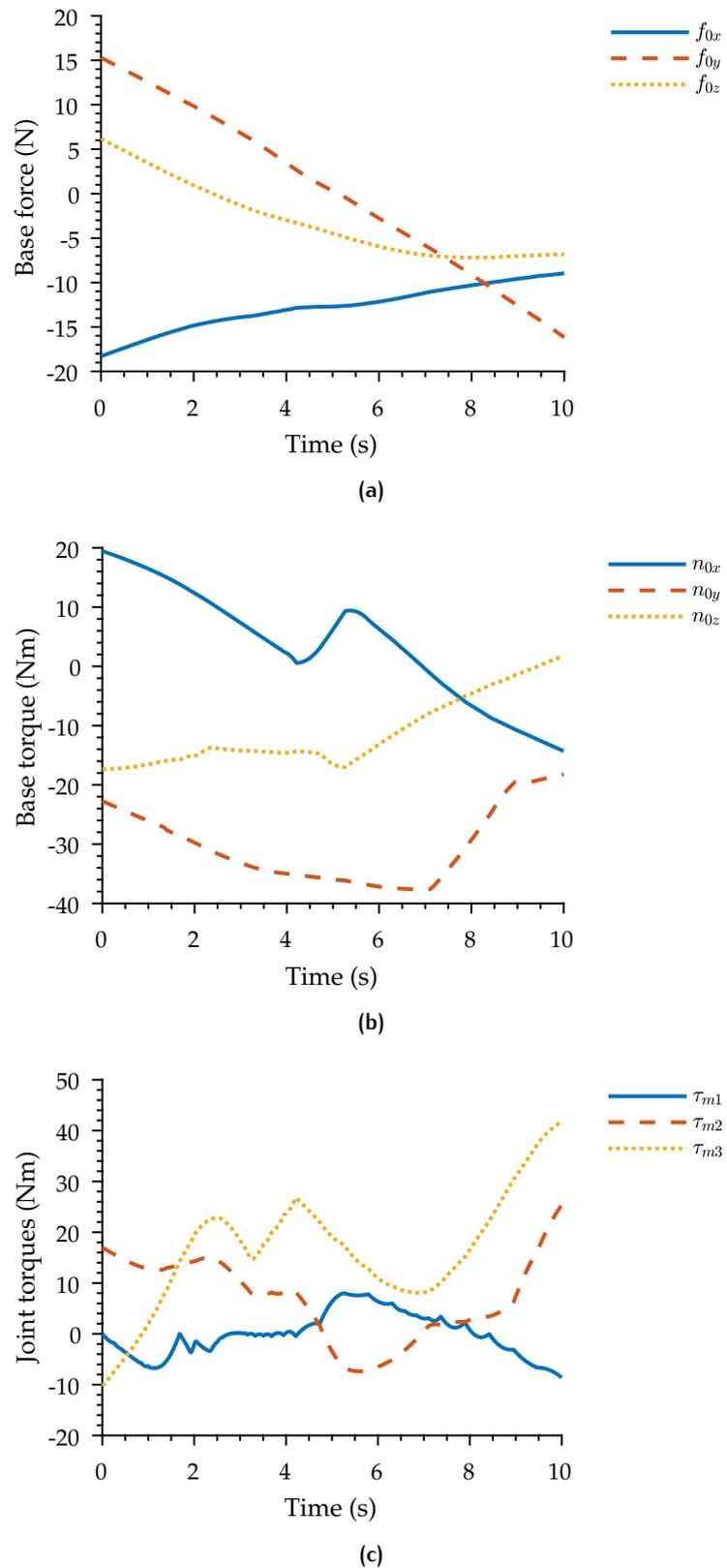


Figure 4.12: Optimal trajectories of the generalized forces of the base and manipulator (τ_0 and τ_m) in case of 3 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (base torque is expressed in Σ_0).

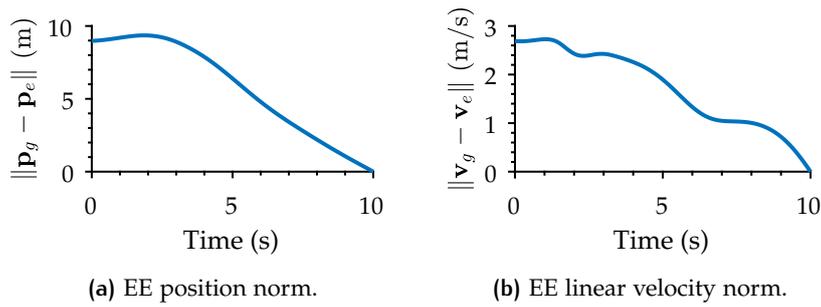


Figure 4.13: Euclidean norms of the relative position and linear velocity of the end-effector (EE) (p_e and v_e) with respect to the state of the pre-grasping feature at the end of the maneuver ($p_g(t_f)$ and $v_g(t_f)$) in case of 3 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$.

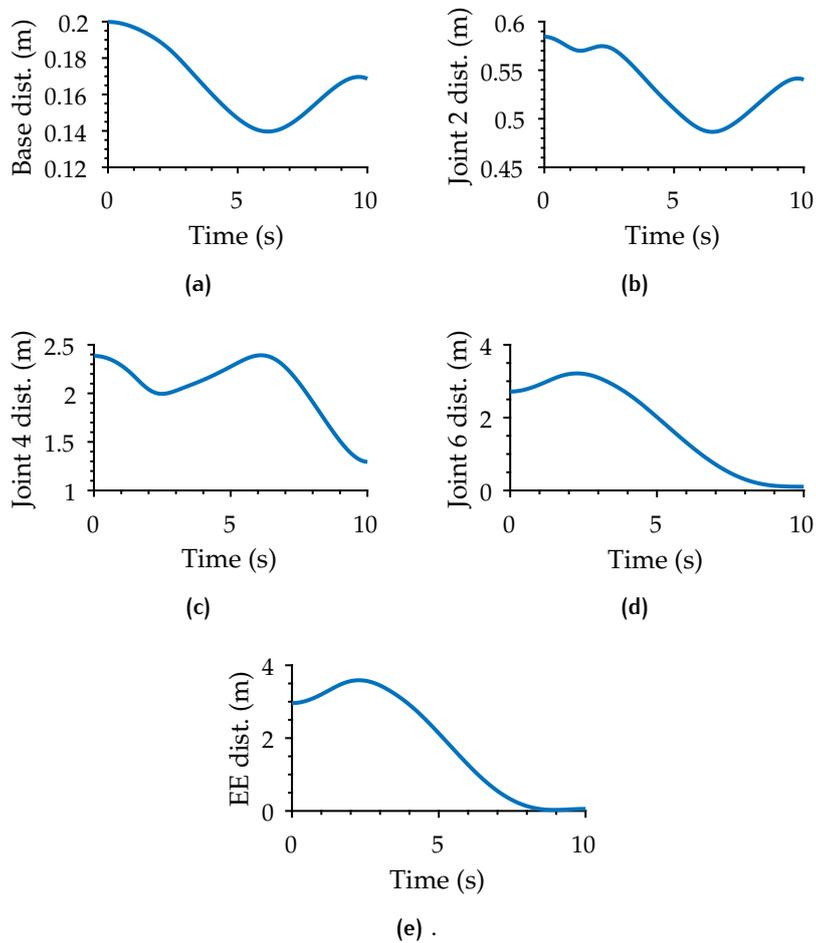
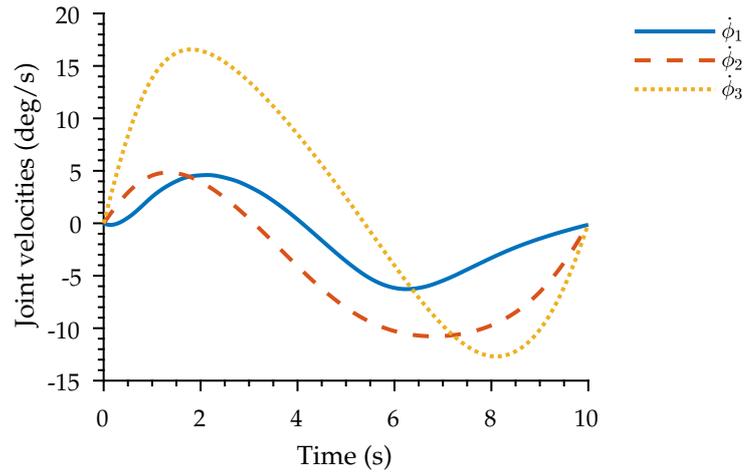
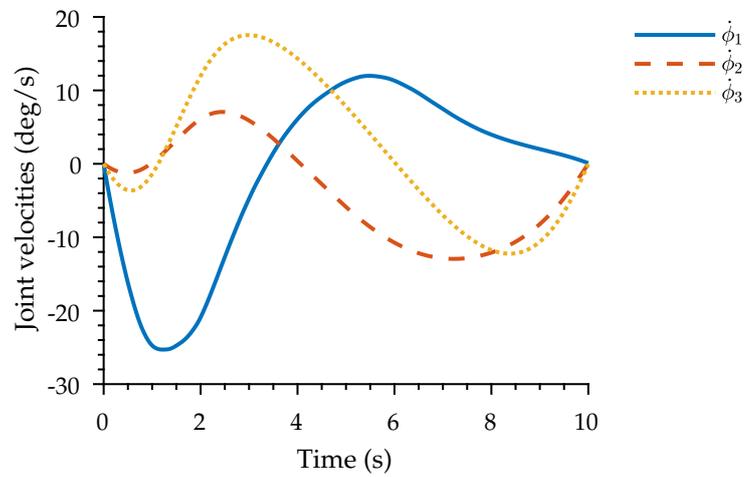
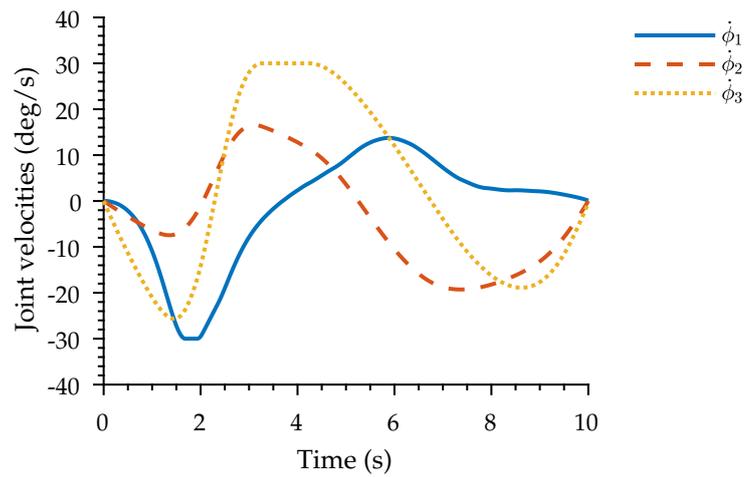


Figure 4.14: Distances between collision geometries of the robot and the keep-out zone of the target in case of 3 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (dist. = distance, EE = end-effector).

(a) Joint velocities for $\omega_t = 5 \text{ deg s}^{-1}$.(b) Joint velocities for $\omega_t = 11.5 \text{ deg s}^{-1}$.(c) Joint velocities for $\omega_t = 18 \text{ deg s}^{-1}$.Figure 4.15: Optimal trajectories of joint velocities (\dot{q}_m) of all 3 DOF tests.

4.4.2.2 7 DOF manipulator test results

The initial configuration of the chaser with a 7 DOF manipulator and target having $\omega_t = 18 \text{ deg s}^{-1}$ is shown in Fig. 4.16.

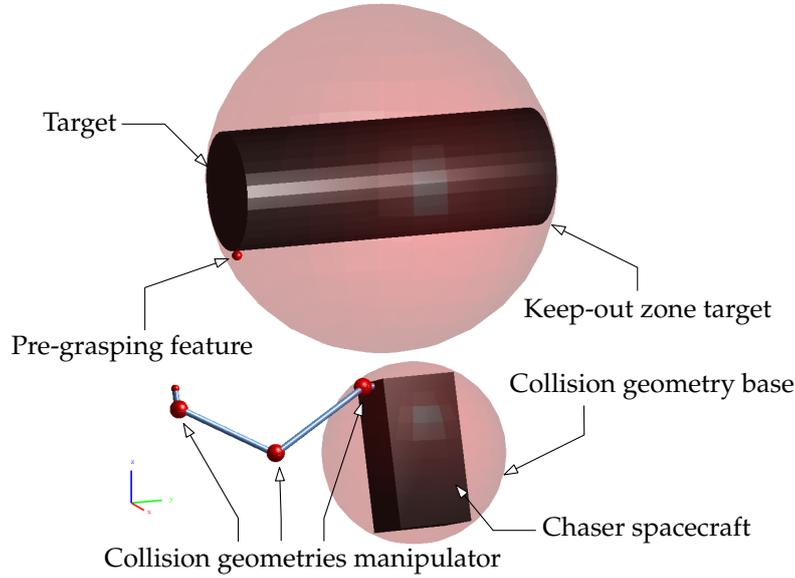


Figure 4.16: Initial configuration of the chaser and target in case of 7 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$.

The optimal trajectories of \mathbf{q} , $\dot{\mathbf{q}}$ and $\boldsymbol{\tau}$, obtained as a solution of the OCP, are shown in Figs. 4.17 to 4.19. Once again, for an easier interpretation of the results, the history of the base rotation, depicted in Fig. 4.17b, is represented as a continuous function of time of a 3-angle, roll-pitch-yaw (XYZ), sequence, Γ_0 .

The time history of the boundary conditions of the OCP are illustrated in Fig. 4.20 in terms of $\|\mathbf{p}_g(t_f) - \mathbf{p}_e(t)\|$, α^{46} , $\|\mathbf{v}_g(t_f) - \mathbf{v}_e(t)\|$ and $\|\boldsymbol{\omega}_g(t_f) - \boldsymbol{\omega}_e(t)\|$,

⁴⁶ The angle between $\mathcal{Q}_g(t_f)$ and $\mathcal{Q}_e(t)$.

The time profiles of the algebraic no-contact constraints of the OCP are depicted in Fig. 4.21 as the time history of the collision distances between the five spherical elements of the chaser and the keep-out zone of the target.

The comparison of $\dot{\mathbf{q}}_m$ trajectories of all 7 DOF test cases is shown in Fig. 4.22.

The performance of the planner in case of all 7 DOF test cases is summarized in Table 4.5 featuring, as in case of 3 DOF test cases, the final values of the minimized objective function, number of mesh refinement iterations, number of iterations within the first mesh, the total number of iterations, the total computational time employed and the percentage of that time used to perform function evaluations.

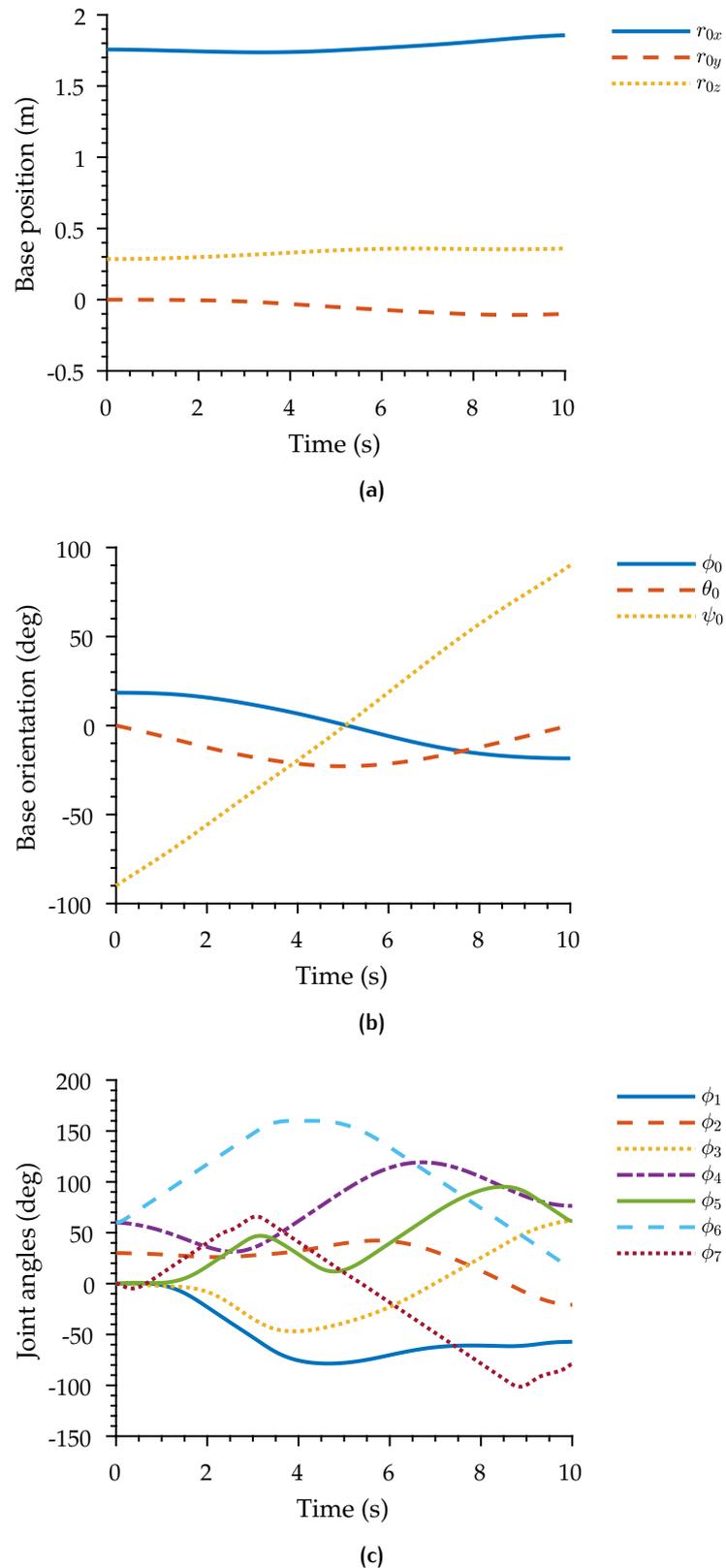
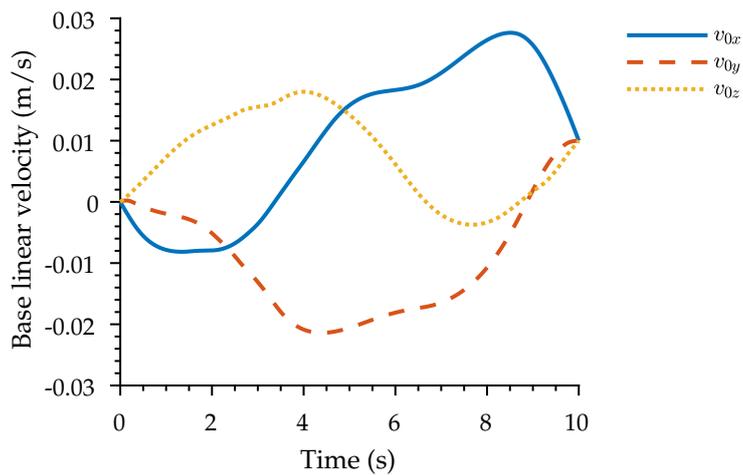
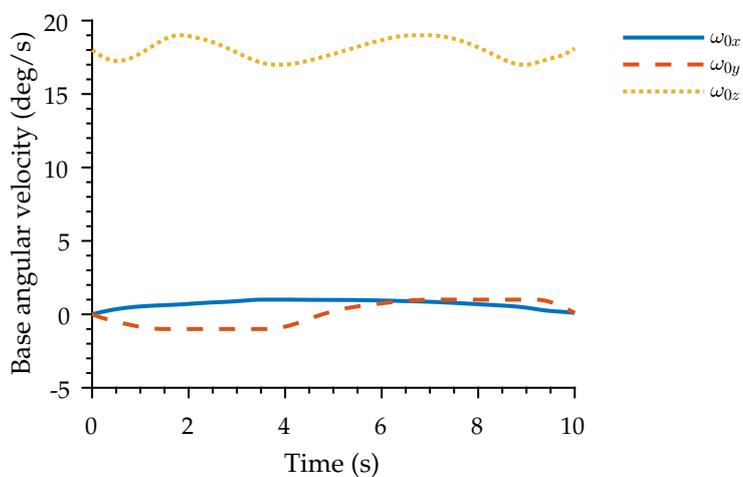


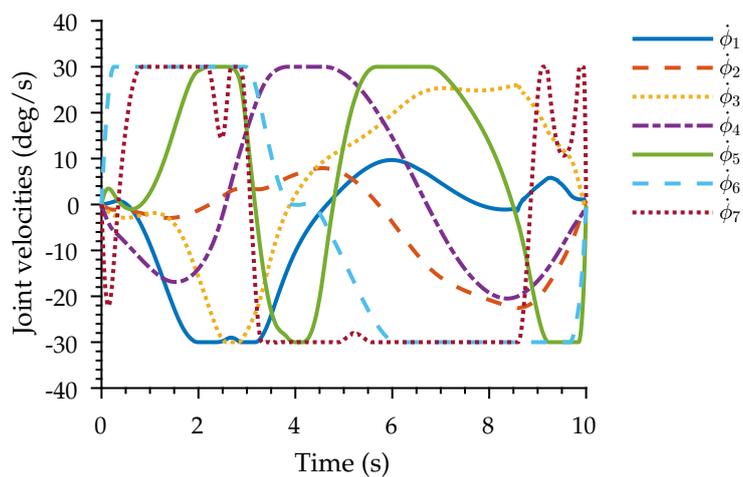
Figure 4.17: Optimal trajectories of the generalized coordinates of the base and manipulator (q_0 and q_m) in case of 7 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (the base orientation is expressed in a roll-pitch-yaw (XYZ) angle sequence).



(a)



(b)



(c)

Figure 4.18: Optimal trajectories of the time derivatives of generalized coordinates of the base and manipulator (\dot{q}_0 and \dot{q}_m) in case of 7 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (base angular velocity is expressed in Σ_0).

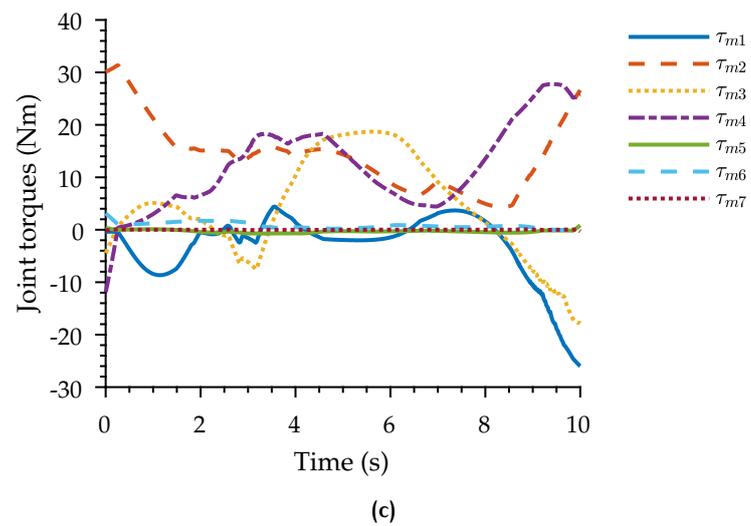
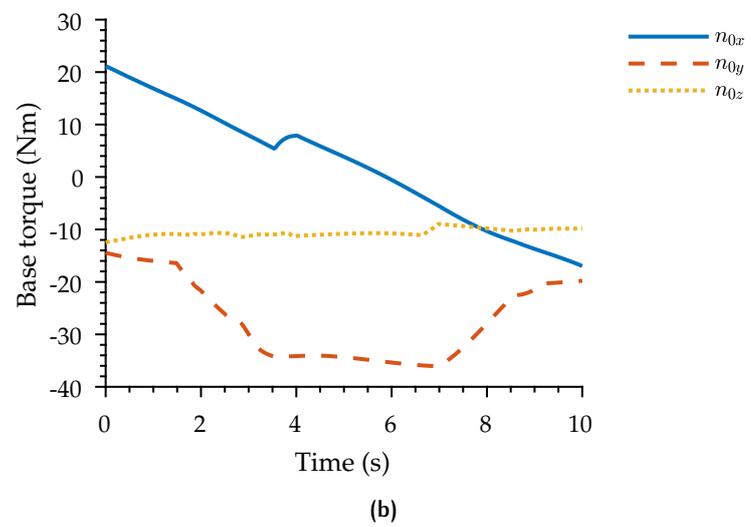
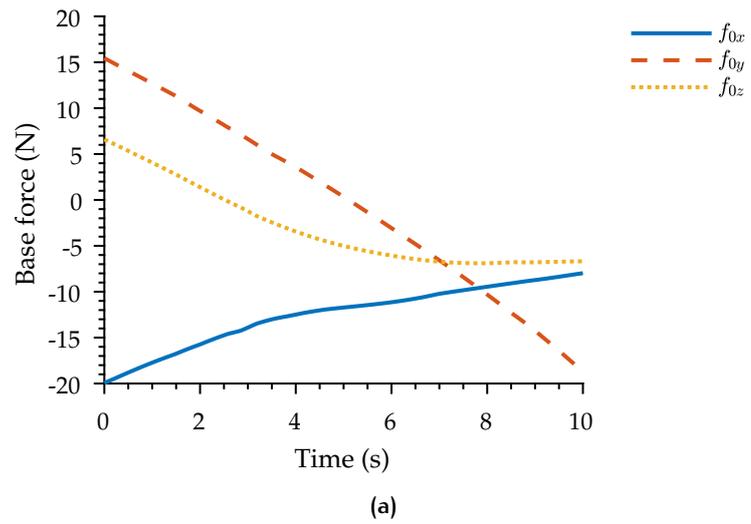


Figure 4.19: Optimal trajectories of the generalized forces of the base and manipulator (τ_0 and τ_m) in case of 7 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (base torque is expressed in Σ_0).

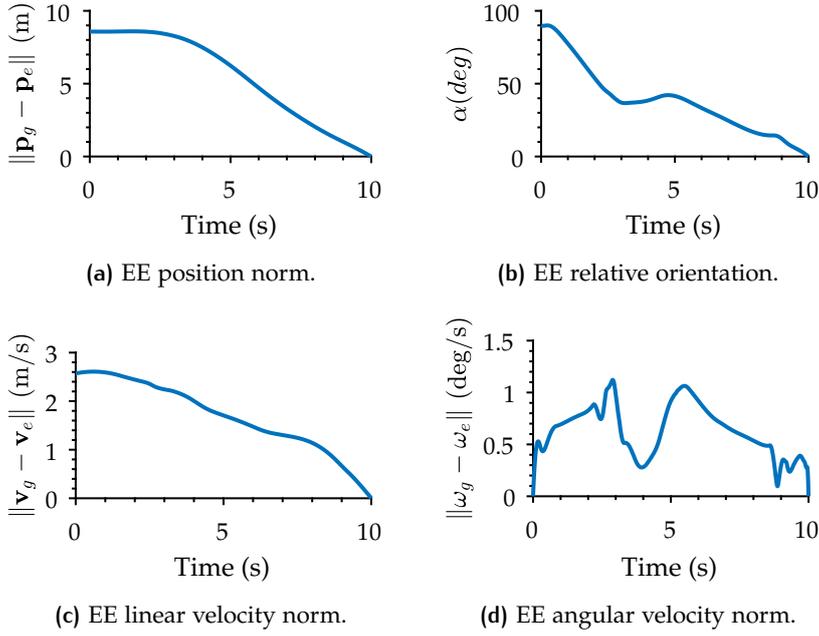


Figure 4.20: Euclidean norms of the relative position, linear and angular velocity as well as relative orientation of the end-effector (EE) (\mathbf{p}_e , \mathbf{v}_e , $\boldsymbol{\omega}_e$ and $\dot{\mathbf{Q}}_e$) with respect to the state of the pre-grasping feature at the end of the maneuver ($\mathbf{p}_g(t_f)$, $\mathbf{v}_g(t_f)$, $\boldsymbol{\omega}_g(t_f)$ and $\dot{\mathbf{Q}}_g(t_f)$) in case of 7 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ ($\alpha =$ angle between $\dot{\mathbf{Q}}_g(t_f)$ and $\dot{\mathbf{Q}}_e(t)$).

Table 4.5: Performance summary of 7 DOF tests (min. = minimized, ref. = refinement, iter. = iteration(s), FE = function evaluations).

Parameter	5 deg s^{-1}	11.5 deg s^{-1}	18 deg s^{-1}
Min. objective	3.88×10^3	7.02×10^3	2.05×10^4
Mesh-ref. iter.	3	4	3
1st mesh iter.	147	226	533
Total iter.	264	417	929
CPU time (s)	8.3×10^4	12.59×10^4	25.99×10^4
Time for FE (%)	99.84	99.83	99.72

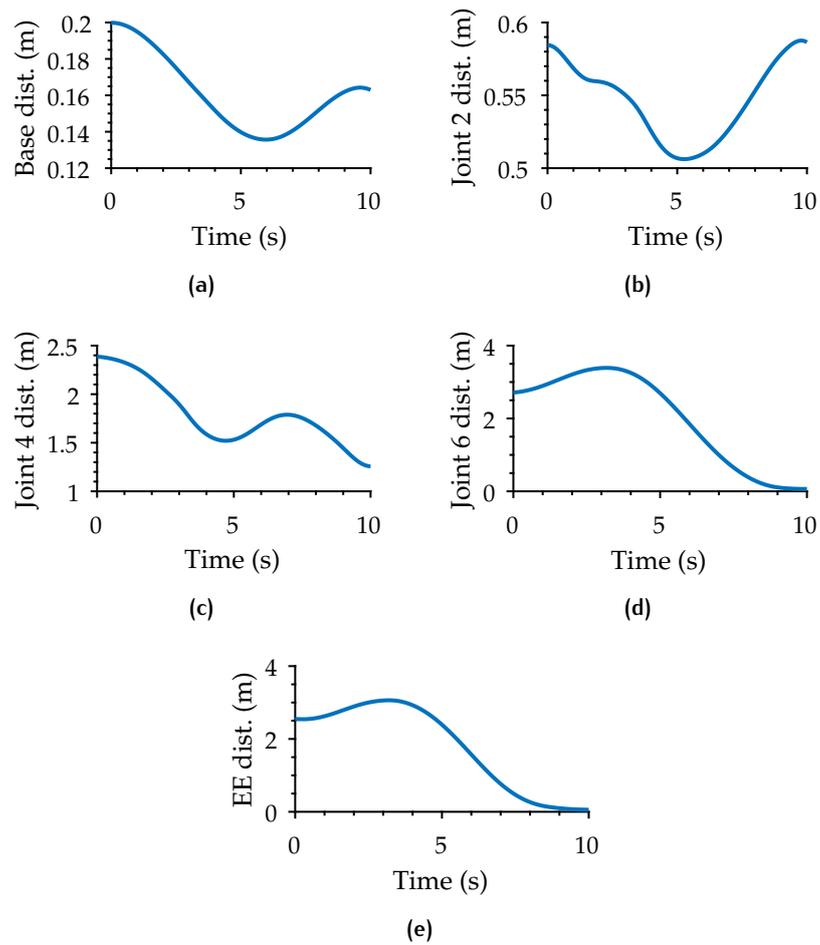
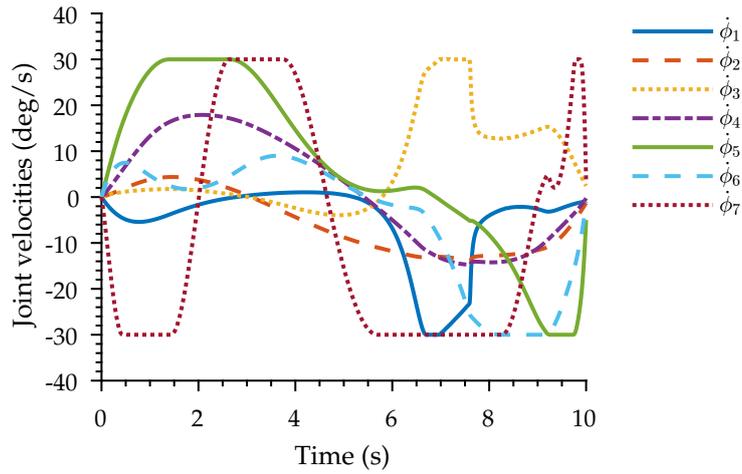
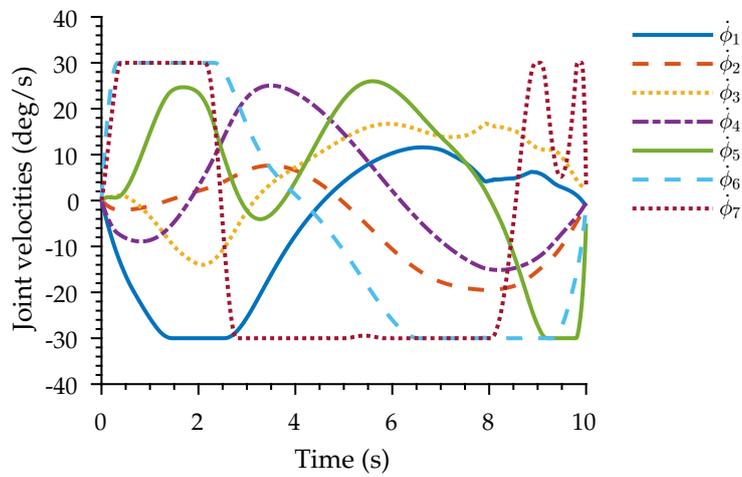
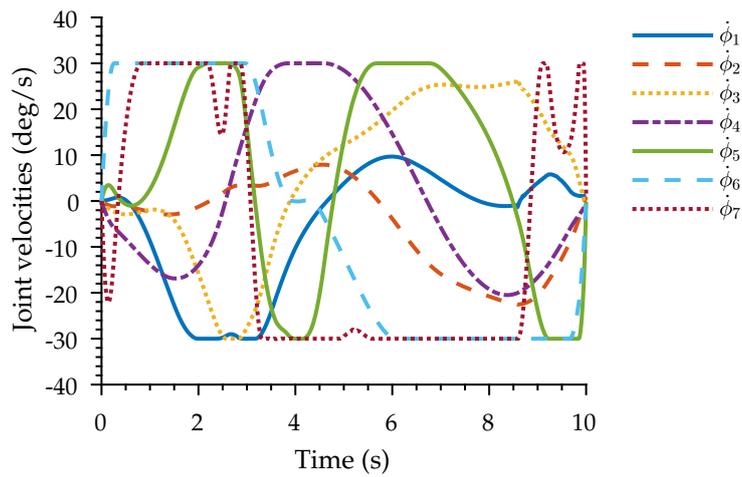


Figure 4.21: Distances between the collision geometries of the robot and the keep-out zone of the target in case of 7 DOF manipulator and $\omega_t = 18 \text{ deg s}^{-1}$ (dist. = distance, EE = end-effector).

(a) Joint velocities for $\omega_t = 5 \text{ deg s}^{-1}$.(b) Joint velocities for $\omega_t = 11.5 \text{ deg s}^{-1}$.(c) Joint velocities for $\omega_t = 18 \text{ deg s}^{-1}$.Figure 4.22: Optimal trajectories of joint velocities (\dot{q}_m) of all 7 DOF tests.

4.5 ANALYSIS AND DISCUSSION

The results presented in Section 4.4 are here analyzed and strengths as well as weaknesses of the developed motion planner (POSE) are discussed.

4.5.1 Results analysis

Overall the results reveal that under the assumptions and parameters defined in Sections 4.3.1 and 4.4.1 the developed optimization-based motion planner (POSE) is able to find (locally) optimal trajectories for a space robot to approach the pre-grasping feature of an uncooperative, rapidly tumbling target.

More specifically, the performance values of the test cases presented in Tables 4.4 and 4.5 highlight that the 3 DOF tests require one to two orders of magnitude less CPU time to converge to a solution. This is due to the lower number of state variables and an initialization that is better suited for the 3 DOF tests than for the 7 DOF ones, as evidenced by the maximum numbers of iterations of the solver within the first mesh. The same tables show that the computation time needed to find a solution of an 18 deg s^{-1} test case is in general longer which was expected due to the overall more difficult conditions of this edge case.

The comparison between the optimal trajectories of q and \dot{q} of the two test cases shown in Figs. 4.10, 4.11, 4.17 and 4.18, reveals no significant difference in the profiles of the base variables. In fact, as expected, the figures illustrate the base spacecraft predominantly rotating around its body z axis with ω_{0z} oscillating (within the bounds) around the nominal value of 18 deg s^{-1} , during the motion of the manipulator. Differently the optimal trajectories of joint angles and velocities, shown in Figs. 4.10c, 4.11c, 4.17c and 4.18c, highlight no direct correlation between the two test cases, as it was expected, due to different robot kinematics. Nevertheless, the results present some similarities. In fact, while in both test cases the optimal profiles of the joint angles are predominantly within the specified interval of $\pm 160 \text{ deg}$ (see Figs. 4.10c and 4.17c), those of the joint velocities (see Figs. 4.11c and 4.18c) show the tendency of the last active joints of the arm to reach the limits ($\pm 30 \text{ deg s}^{-1}$) shortly after the start of the maneuver, despite the fact that corresponding joint actuators are far from their saturation limits (see Figs. 4.12c and 4.19c). This result is most certainly due to the imposed low bounds on joint velocities (i. e. $\pm 30 \text{ deg s}^{-1}$), high angular rate of the target as well as the imposed performance index (see Eq. (4.79)) which does not account for joint velocities. Indeed, analyzing the profiles of the joint velocities of all the test cases, graphically summarized in Figs. 4.15 and 4.22, it can be inferred that with lower angular rates of the target the degree of saturation of joint

velocities of the manipulator is non-existent in case of 3 DOF tests and more contained in case of 7 DOF test cases.

The optimal trajectories of τ_0 shown in Figs. 4.12a, 4.12b, 4.19a and 4.19b reiterate the similarities between the base profiles of all test cases. As expected, they are dominated by τ_{0x} and τ_{0y} needed to correct the base velocities induced by the movement of the manipulator.

The optimal profiles of τ_m of the two test cases presented in Figs. 4.12c and 4.19c reveal joint torques trajectories well within the limits of the actuators (set to ± 80 N m) with no direct correlation between the two test cases, as expected. The very low median values of joint torques of the last three joints in Fig. 4.19c are to be attributed to the high weights assigned to those joints in the objective function (see Eq. (4.104)). The non-smooth nature of all τ profiles is to be attributed to the control variable of the OCP appearing linearly in the differential equations reported in Eq. (4.86). In fact, every time the control appears linearly in the problem, the optimal control history can be expected to have a discontinuous appearance that will introduce inaccuracies in the solution.

Finally, the analysis of Figs. 4.13, 4.14, 4.20 and 4.21 reveals that the found optimal solutions satisfy both the boundary end-effector conditions and algebraic path constraints. This indicates that using these results the end-effector is able to match the pose and velocities of the pre-grasping feature at t_f ⁴⁷ while assuring a collision-free trajectory.

⁴⁷ More specifically, only position and linear velocity in case of 3 DOF tests.

4.5.2 Discussion

The objective of this research was to develop an optimization-based motion planner for a free-flying robot to facilitate the capture of a rapidly tumbling, uncooperative target with a multi-target mission in mind. The analysis of the results, presented in the previous section, confirms the ability of POSE⁴⁸ to find (locally) optimal motions of a free-flying robot from fixed start states to a region of fixed goal states while avoiding the keep-out zone of the target and satisfying other constraints, such as the end-effector pose and velocity constraints, joint and control limits. The planner extends the MRAVS semisynchronizing strategy, detailed in [10], by explicitly including the nonholonomic behavior of the space robot and formulating the underlying motion planning problem as a nonlinear optimization. Compared to the state-of-the-art in the field detailed in Section 2.4.2, the developed method is able to provide (locally) optimal approach motions of both the spacecraft and manipulator to a rapidly tumbling target (i. e. with $\omega_t \in [5, 18]$ deg s⁻¹) while reducing the relative angular velocity between target and chaser to a minimum, minimizing control requirements, leveraging the redundant nature of the on-board manipulator and ensuring collision avoidance. This way the planner can be used to enable not only multi-target capture missions by means of a free-flying

⁴⁸ Under the assumptions defined in Section 4.3.1.

robot but also single-target missions that would benefit from lower propellant requirements and ultimately the overall lower cost of a mission. However, POSE is not free of limitations and the main ones are:

- relatively long computation times,
- saturation of joint velocities,
- limited number of no-contact constraints,
- discontinuous control,
- inability to provide a solution for “widely tumbling” targets,
- limited ability to plan trajectories with large attitude maneuvers of the base spacecraft.

The long computation times (pronounced in case of a 7 DOF manipulator) are to be attributed to two factors. The first one being the selected computing environment (i. e. MATLAB) and library to compute the kinematic and dynamic equations of a free-flying robot (i. e. SPART). In fact, from the performance values of the two tests cases summarized in Tables 4.4 and 4.5, it can be seen that the planner spends most of the time in function evaluations. The second factor is the less than ideal initial guess provided to the solver, especially in 7 DOF edge cases, as evidenced by the large number of iterations within the first mesh needed to converge to a solution, visible in Table 4.5. A better initial guess would ensure that the solver arrives at the solution quickly in just few steps, as it is the case in 3 DOF tests. To improve the performance of the developed planner future research will have to address these issues, i. e. find a better way to initialize the edge cases and employ a more optimized multibody dynamics library. However, note that long computation times are not seen as a major limitation of POSE. In fact, as per the assumptions detailed in Section 4.3.1, the motion planning is to be executed during the close range observation phase of the mission during which the time is not considered a critical parameter. Therefore, a computation can be executed from a safe distance. Nevertheless, this limitation hinders the amount of tests that can be performed in certain time period which makes the whole experience less pleasant, especially if a user wants to explore the combination of methods/parameters that could lead to better results. For this reason, the presented numeric simulations were developed using the “crawl, walk, run” methodology, where easier to solve tests were followed by progressively difficult and longer ones.

The tendency of the planner to saturate the joint velocities, shortly after the start of the maneuver, can be attributed to the imposed low bounds on joint velocities, high angular rate of the target and the selected performance index. Excluding the modification of the

state bounds and angular rate of the target, since they might not be alterable, the only way to directly influence the profiles of joint velocities would be to include them inside the performance index, as done by Lampariello and Hirzinger [97] for example. This was outside of the scope of the present work and future research will have to explore this possibility, if the saturation of joint velocities needs to be limited.

The limited number of no-contact constraints included in the planner is the result of a trade-off made to ensure a reasonable convergence speed of the solver while still providing a $\mathcal{C}_{\text{free}}$ of the chaser (see Section 4.3.3). While this approach is suitable for the specific problem at hand, as evidenced by Figs. 4.14 and 4.21, it is not the most conservative one as it does not cover all the points of the robot and it does not consider possible self-collisions. Future research will need to address this issue by resorting to a more generic collision-detection routine that considers every point of the chaser and may have at its core an efficient distance-measurement algorithm, such as the Gilbert-Johnson-Keerthi algorithm [189].

A discontinuous control solution can be expected anytime the control appears linearly in the differential equations of an OCP. In these cases, formulating a problem with only one phase will inevitably lead to a continuous control function that is a reasonably good approximation of a solution but is nevertheless only an approximation and therefore will be characterized by discretization inaccuracies in the neighborhood of the discontinuities. Additionally, the convergence time of the solver might also suffer. The correct way to deal with this issue is to explicitly introduce the discontinuous behavior of the control function within the formulation of the OCP by defining it as a collection of N phases. Future research striving for a more efficient and precise solution will have to investigate in detail this issue, determine the locations of discontinuities and introduce them explicitly as phase boundaries in the problem formulation [167].

The inability of the planner to provide a solution for “widely tumbling” targets stems from an inherent limitation of MRAVS to provide an acceptable solution in case of a large angle between the angular momentum and angular velocity of a target, as mentioned in Section 4.3.1. This also implies that in case of an ω_t having dominant ω_{tx} and/or ω_{ty} components, the $T_g(t_F)$ will need to be revisited in order to ensure that the pre-grasping feature is within the reachable space of the manipulator. The initial state of the chaser will not need to be updated as it will be automatically calculated by the planner based on the expected target motion and MRAVS, as detailed in Section 4.3.1.

The limited ability of the planner to find optimal trajectories requiring large attitude maneuvers of the base spacecraft is directly tied to the three-parameter implementation of the rotational kinematics of the base (see Eq. (4.88)) and the non-native support of quaternions

⁴⁹ *Typical of large maneuvers.*

within ICLOCS. For this reason, whenever the attitude of the base crosses the $\pm\pi$ values⁴⁹, a discontinuous behavior of Eq. (4.88) at those points is to be expected as $\mathbf{\Omega}_0$ does not have a full rank at those points (see Eq. (4.25)). The most common consequences of such functions are slow convergence, small steps in the line search and possible ill-conditioning of the Hessian matrix [167]. Considering the relatively short maneuver time assumed in Section 4.4.1 this limitation was not relevant for the considered test cases. However, future research aiming at longer maneuvers might need to consider a more generic method for optimization-based planning that correctly handles the group structure of 3D rotations represented by unit quaternions. An example of such a method and related solver (ALTRO) might be found in [190].

5

CONCLUSIONS AND FUTURE WORK

The research presented in this thesis addresses the capture phase of an ADR mission from the point of view of the selection of the most suited ADR capture method(s) and capture of an uncooperative, rapidly tumbling target with a space robot. This chapter provides a brief summary of the main contributions of the thesis and emphasizes their implications in the context of the existing state-of-the-art. Finally, it provides a future perspective of the presented work.

5.1 CONCLUSIONS

The ever-increasing population of space debris has long been recognized by the scientific community as a critical issue that needs to be addressed with urgency. In fact, there is a consensus among researchers that the implemented mitigation measures should be coupled with ADR in the near future if we are to stabilize the current space debris environment. However, choosing the right ADR method is currently a difficult task mainly due to the information paradox characterizing the space debris domain. Past studies have addressed this problem with modern knowledge representation methods but only in the context of the space situational awareness. None has explicitly addressed the domain of ADR and most of them appear to overlook the handling of the input data from existing databases of objects. To bridge this gap, the first objective of this work was to develop a domain-ontology, for data processing, collection, storage and sharing of characteristics of large, intact objects, able to perform automatic ADR capture method(s) selection. The developed ontology, TRACER, defines a minimal set of physical and dynamical parameters of an object deemed sufficient to infer, via a semantic reasoner, its most suited ADR capture method(s), safety wise. This way, not only the management, but also the discovery of the new knowledge is facilitated. At the same time, TRACER is equipped with tools to transparently handle the input of data from an existing space debris catalog, e. g. DISCOS, thus reducing user input and possibility of a human error. The practicality and validity of TRACER were demonstrated by applying it onto a database of representative objects, for which attitude states were able to be obtained from publicly available resources. The classification results point toward net-based capture methods as the most frequently associated with targets, followed by manipulator-based methods, as it

was expected, considering the nature of objects within the example database.

Once most suited capture method(s) is(are) selected, the next step involves defining the capture maneuver itself which presents its own challenges depending on the selected capture method. For those not requiring a physical contact (e.g. propulsion-based devices), such maneuver might be relatively simple, even in case of a rapidly tumbling target, as the “capture” will occur from a stand-off distance. For others, this might not be the case, as the capture will involve close-range proximity operations and eventually a contact with the target which is not a trivial task in case it is rapidly tumbling. Indeed, in case of space robotics, it can be argued that capturing a target with high angular momentum (i.e. high angular rate and/or high inertia) remains an open challenge despite the 30 years of research on the topic and demonstration missions. The solution to this issue is made even more difficult due to the nonholonomic behavior of a space robot which induces dynamical coupling between the base and manipulator during its motion. It is not surprising that the mainstream solution to the robotic capture of a rapidly tumbling target assumes a complete relative motion synchronization and a fixed base spacecraft during the motion of the manipulator. This approach while suitable for single-target missions would pose serious limitations to multi-target missions due to the significant amount of propellant required to complete each one. To facilitate future, autonomous, robotic missions capable of capturing multiple, rapidly tumbling targets, the second objective of this work was to develop an optimization-based motion planner able to find (locally) optimal coordinated approach trajectories of both the base spacecraft and manipulator. The developed motion planner, POSE, exploits the semisynchronizing strategy MRAVS [10] to find (locally) optimal trajectories of the state and control variables of the chaser that minimize the overall control effort subject to dynamic and collision constraints, boundary conditions on the pose and velocity of the end-effector as well as state and control bounds. To ensure locally optimal solutions within a reasonable time, POSE provides an initialization routine that generates an initial guess of the base and manipulator trajectories separately, assuming the base subject only to rotational motion around its body z -axis and the manipulator subject to a “point-to-point” motion from a desired initial state to a desired final state. The results of numerical simulations provide the insight into the performance of POSE and suggest its ability to provide optimal, well-behaved state trajectories of both the base spacecraft and manipulator under a range of angular rates of the target and different kinematic configurations of the robot.

5.2 FUTURE WORK

The work presented in this thesis, like any other research work, is far from being complete. Main areas of interest that might further it are outlined hereafter and are grouped in the two topics covered by this dissertation.

Space debris ontology for ADR

With reference to the developed domain-ontology TRACER recommendations for future research might cover:

- **Extending the class hierarchy.** The class hierarchy of TRACER was developed with simplicity and readability in mind, as detailed in Section 3.2.3. Hence, it is characterized by a minimum necessary number of classes and subclasses to sufficiently characterize an IDO for a capture maneuver. On the other hand, future research might strive to achieve a more generic and comprehensive ontology. In this case, the class hierarchy should be extended with a greater variety of classes and subclasses such as orbital regimes, launch vehicle types, spacecraft platforms, etc.
- **Estimating survival probability using parametric models.** To estimate the survival probability of spacecraft, TRACER employs a nonparametric model, i. e. the Kaplan-Meier estimator, (see Section 3.1.2). Even though this approach provides the most accurate representation of the survival probability, it is not the most convenient one to use [114]. Therefore, future work might consider using a parametric model. One of the most commonly used in reliability analysis is the Weibull distribution [114].
- **Expanding the class axioms of `ADR_Capture_Method` subclasses.** TRACER in its current form lacks axioms related to physical characteristics of a target (e. g. shape, size or mass) that might reflect in more detail the usability of a capture method. For instance, tether-based devices could be more appropriate for targets with compact geometries as appendages could potentially be structurally damaged during the capture creating even more debris. In addition, the ontology does not include any cost parameter which might be of particularly interest for future commercial ADR services. Future work should most certainly address these limitations in the form of additional axioms to be applied to the subclasses of the `ADR_Capture_Method` class detailed in Section 3.2.5.
- **Increasing data input automation and resilience to missing data.** The current data input of TRACER relies heavily on a manual scraping of information from the Internet, due to the nonex-

istence of a single data source containing all the necessary information (see Section 3.3.1). This procedure is time-consuming and error prone. Future research might consider automatizing this process via an automated data scraping. Considering the unstructured nature of the required data, it is to be expected that in some cases this automated data extraction might prove to be challenging especially when using simple scraping techniques such as the *text pattern matching*. Therefore, more advanced scraping techniques based on machine learning and computer vision are recommended. In some cases even this approach will fail due to the nonexistence of the data in the public repositories. To overcome this issue, the resilience of TRACER to missing data should be increased, by employing for example machine learning algorithms that could infer the required data from the existing one (e. g. see [64]).

Robotic capture of an uncooperative target

In regard to the developed motion planner, POSE, future research might consider:

- **Improving the performance of the motion planner.** Analyzing the numeric simulations outlined in Section 4.4.2, long computation times can be observed, especially in case of a 7 DOF manipulator with high ω_t . This phenomenon can be attributed mainly to the following three factors: a) formulation of the OCP as a single-phase problem, b) selected computation environment and libraries, c) provided initialization method. Future research striving to improve the performance of POSE will have to address this issue by investigating the role of each of the three factors in the overall performance and providing an alternative solution to those that mainly affect the numeric performance of the planner. For example, the underlying OCP might be re-formulated as a multi-phase problem or an alternative initialization routine might be developed, e. g. based on a solution of a simplified planning problem.
- **Generalizing the collision-free constraints.** The planner in its current implementation considers only a limited number of fixed no-contact constraints between the chaser and target. This choice stems from the necessity to keep the number of collision geometries to a minimum to ensure a collision-free path and a reasonable convergence speed. As proved by the results illustrated in Section 4.4.2, this approach is well suited for the specific problem at hand. A more conservative approach will need to cover all the points of the robot and consider possible self-collisions. To achieve this, future research might resort to more

generic collision-detection routines such as those implemented by optimization-based planners used in ground robotics, e. g. Trajectory Optimization for Motion Planning (Trajopt) [191] and Stochastic Trajectory Optimization for Motion Planning (STOMP) [192].

- **Analyzing the robustness of the planner.** Based on the analysis of the results detailed in Section 4.5 and under the assumptions defined in Section 4.3.1, it can be asserted that POSE is able to fulfill its objective. Nevertheless, it must be noted that while the presumed assumptions are reasonable, they might not always hold true, especially in case of a real mission where uncertainties and estimation errors are to be expected. Future research is strongly recommended to analyze the robustness of the planner to variations of the assumptions in order to assess its applicability to generic mission conditions and improve upon those results.
- **Performing tests on a HIL platform.** As evident from Section 4.4, the performance of POSE was tested only in a simulation environment. Considering the specific mission phase under study (i. e. the manipulator approach), this is appropriate as no contact is involved and both the dynamics of the chaser and target can be simulated numerically with high-accuracy and without any loss of generality. Still, validating a tool only in simulation might conceal issues or alternative directions that would arise only during an experimental work. For this reason, future work should consider complementing numerical tests with experimental ones using HIL platforms able to simulate microgravity conditions on ground, such as: air-bearings testbeds [193], neutral buoyancy facilities [194], suspension [195] or robotic systems [196].

BIBLIOGRAPHY

- [1] H. Klinkrad. *Space Debris: Models and Risk Analysis*. Astronautical Engineering. Berlin Heidelberg: Springer-Verlag, 2006. ISBN: 978-3-540-25448-5. DOI: 10/c8zt5p (cit. on pp. 1, 8, 10, 12).
- [2] V. Agrapov et al. *IAA Situation Report on Space Debris - 2016*. Research report 978-2-917761-56-4. Paris, France: International Academy of Astronautics (IAA), 2017, pp. 1–170. URL: <https://bit.ly/2MP0Azf> (visited on 10/16/2019) (cit. on pp. 1, 2, 8, 10, 12–14, 16, 17, 20, 42).
- [3] ESA. *About Space Debris*. Feb. 21, 2018. URL: <https://bit.ly/2Be1wYf> (visited on 10/16/2019) (cit. on pp. 1, 12).
- [4] D. J. Kessler and B. G. Cour-Palais. “Collision Frequency of Artificial Satellites: The Creation of a Debris Belt.” In: *Journal of Geophysical Research* 83.A6 (June 1978), pp. 2637–2646. DOI: 10/ct44fz (cit. on pp. 1, 10).
- [5] M. Kaplan et al. “Engineering Issues for All Major Modes of In Situ Space Debris Capture.” In: *AIAA SPACE 2010 Conference & Exposition*. AIAA SPACE. Anaheim, CA, USA: American Institute of Aeronautics and Astronautics (AIAA), 2010, pp. 1–20. ISBN: 978-1-60086-966-2. DOI: 10/gf93hz (cit. on pp. 2, 16).
- [6] R. Biesbroek et al. “The e.Deorbit CDF Study: A Design Study for the Safe Removal of a Large Space Debris.” In: *6th European Conference on Space Debris*. Vol. 6. Darmstadt, Germany: ESA Space Debris Office, Apr. 2013, pp. 1–8. URL: <https://bit.ly/2P2b7Kb> (visited on 10/16/2019) (cit. on pp. 2, 18, 21, 25).
- [7] S. Matsumoto et al. “Satellite Capturing Strategy Using Agile Orbital Servicing Vehicle, Hyper-OSV.” In: *2002 IEEE International Conference on Robotics and Automation*. Vol. 3. IEEE, 2002, pp. 2309–2314. ISBN: 0-7803-7272-7. DOI: 10/bg8j2k (cit. on pp. 2, 70).
- [8] B. Naasz et al. “Flight Dynamics and GN&C for Spacecraft Servicing Missions.” In: *AIAA Guidance, Navigation, and Control Conference*. Toronto, ON, Canada: American Institute of Aeronautics and Astronautics (AIAA), Aug. 2, 2010, pp. 1–20. DOI: 10/ggc2v5 (cit. on pp. 3, 18, 33).
- [9] N. F. Noy and D. L. Mcguinness. “Ontology Development 101: A Guide to Creating Your First Ontology.” In: (2001). URL: <https://stanford.io/3b04tQu> (cit. on pp. 3, 26, 60–63).

- [10] C. Ma, C. Wei, and J. Yuan. "Semisynchronizing Strategy for Capturing a High-Speed Tumbling Target." In: *Journal of Guidance, Control, and Dynamics* 41.12 (2018), pp. 2615–2632. DOI: 10/gfqx63 (cit. on pp. 3, 39, 104, 106, 109, 143, 148).
- [11] M. Jankovic, J. Paul, and F. Kirchner. "GNC Architecture for Autonomous Robotic Capture of a Non-Cooperative Target: Preliminary Concept Design." In: *Advances in Space Research* 57.8 (Apr. 15, 2016), pp. 1715–1736. ISSN: 02731177. DOI: 10/f8j2gw (cit. on pp. 4, 5, 7, 87, 112, 121).
- [12] M. Jankovic et al. "Space Debris Ontology for ADR Capture Methods Selection." In: *Acta Astronautica* 173 (Aug. 1, 2020), pp. 56–68. ISSN: 0094-5765. DOI: 10/ggsd2b (cit. on pp. 4, 5, 7, 41, 61, 64, 67, 68, 70, 72, 79–83).
- [13] M. Jankovic and J. Paul. "Robotic Active Debris Removal and On-Orbit Servicing." In: *Asteroid and Space Debris Manipulation: Advances from the Stardust Research Network*. Progress in Astronautics and Aeronautics. Reston, VA, USA: AIAA, Jan. 2016, pp. 329–372. ISBN: 978-1-62410-323-0. URL: <https://bit.ly/2qeUGzw> (visited on 10/17/2019) (cit. on pp. 4, 7).
- [14] M. Jankovic, N. Thiry, and P. Bourdon. "Lasers for Asteroid and Debris Deflection." In: *Asteroid and Space Debris Manipulation: Advances from the Stardust Research Network*. Progress in Astronautics and Aeronautics. Reston, VA, USA: AIAA, Jan. 2016, pp. 291–328. ISBN: 978-1-62410-323-0. URL: <https://bit.ly/2IXYHit> (visited on 10/17/2019) (cit. on p. 4).
- [15] M. Jankovic and F. Kirchner. "Taxonomy of LEO Space Debris Population for ADR Capture Methods Selection." In: *Stardust Final Conference*. Ed. by M. Vasile et al. Astrophysics and Space Science Proceedings. Springer International Publishing, 2018, pp. 129–144. ISBN: 978-3-319-69956-1. DOI: 10/gf93mr (cit. on pp. 4, 5, 7, 41, 62).
- [16] M. Jankovic and F. Kirchner. "Trajectory Generation Method for Robotic Free-Floating Capture of a Non-cooperative, Tumbling Target." In: *Stardust Final Conference*. Ed. by M. Vasile et al. Astrophysics and Space Science Proceedings. Springer International Publishing, 2018, pp. 111–127. ISBN: 978-3-319-69956-1. DOI: 10/gf93ms (cit. on pp. 4, 5, 7, 87, 106).
- [17] M. Jankovic et al. "Robotic System for Active Debris Removal: Requirements, State-of-the-Art and Concept Architecture of the Rendezvous and Capture (RVC) Control System." In: *5th CEAS Air & Space Conference Proceedings*. Delft, The Netherlands: Council of European Aerospace Societies (CEAS), 2015, pp. 1–15. URL: <https://tinyurl.com/yjz3a86c> (visited on 10/11/2019) (cit. on pp. 4, 5, 87).

- [18] K. Kumar et al. "Agora: Mission to Demonstrate Technologies to Actively Remove Ariane Rocket Bodies." In: *66th International Astronautical Congress (IAC)*. Jerusalem, Israel: International Astronautical Federation (IAF), Oct. 2015, pp. 1–16. URL: <https://iafastro.directory/iac/archive/browse/IAC-15/A6/6/28851/> (cit. on p. 4).
- [19] M. Jankovic and F. Kirchner. "Taxonomy of LEO Space Debris Population for ADR Selection." In: *67th International Astronautical Congress (IAC)*. Guadalajara, Mexico: International Astronautical Federation (IAF), Oct. 2016, pp. 1–15. URL: <https://bit.ly/2V0jnPb> (visited on 10/17/2019) (cit. on pp. 5, 41, 62).
- [20] S. Vyas, M. Jankovic, and F. Kirchner. "Momentum Based Classification for Robotic Active Debris Removal." In: *71st International Astronautical Congress (IAC)*. Online: International Astronautical Federation (IAF), Oct. 2020, pp. 1–11. DOI: 10/gkptq2 (cit. on p. 5).
- [21] ESA Space Debris Office. *Space Environment Statistics*. Space Debris User Portal. Jan. 9, 2020. URL: <https://bit.ly/36I2sm3> (visited on 01/10/2020) (cit. on pp. 7–11, 42, 43).
- [22] ESA Space Debris Office. *Space Debris by the Numbers*. Jan. 9, 2020. URL: <https://bit.ly/360UmYZ> (visited on 10/05/2021) (cit. on p. 8).
- [23] ESA Space Debris Office. *Fragmentation Event Database Statistics*. ESA's fragmentation database. July 13, 2019. URL: <https://bit.ly/2teuw1R> (visited on 01/12/2020) (cit. on pp. 10, 26, 41).
- [24] D. McKnight, R. Maher, and L. Nagl. "Refined Algorithms for Structural Breakup Due to Hypervelocity Impact." In: *International Journal of Impact Engineering*. Hypervelocity Impact Proceedings of the 1994 Symposium 17.4 (Jan. 1, 1995), pp. 547–558. ISSN: 0734-743X. DOI: 10/cwvp45 (cit. on p. 10).
- [25] N. L. Johnson et al. "NASA's New Breakup Model of Evolve 4.0." In: *Advances in Space Research* 28.9 (Jan. 1, 2001), pp. 1377–1384. ISSN: 0273-1177. DOI: 10/bjhx4v (cit. on p. 10).
- [26] J.-C. Liou. "An Active Debris Removal Parametric Study for LEO Environment Remediation." In: *Advances in Space Research* 47.11 (June 1, 2011), pp. 1865–1876. ISSN: 0273-1177. DOI: 10/btsrbz (cit. on pp. 13, 15–17, 120).
- [27] IADC Steering and Working Group 4. *IADC Space Debris Mitigation Guidelines*. Technical report IADC-02-01, Rev 1. Vienna, Austria: Inter-Agency Space Debris Coordination Committee (IADC), 2007, pp. 1–10. URL: <https://bit.ly/3adlQcC> (visited on 10/16/2019) (cit. on pp. 13, 14).

- [28] ISO TC20/SC14. *ISO 24113:2019 - Space Systems – Space Debris Mitigation Requirements*. Standard ISO 24113:2019. Geneva, Switzerland: International Organization for Standardization (ISO), July 2019, pp. 1–13. URL: <https://bit.ly/373w0L7> (visited on 01/17/2020) (cit. on p. 13).
- [29] United Nations Office for Outer Space Affairs. *Space Debris Mitigation Guidelines of the Committee on the Peaceful Uses of Outer Space*. Technical report. Vienna, Austria: United Nations (UN), Jan. 2010, pp. 1–12. URL: <https://bit.ly/2G1PPqd> (visited on 01/17/2020) (cit. on p. 13).
- [30] D. McKnight et al. *A Handbook for Post-Mission Disposal of Satellites Less Than 100 Kg*. Technical report. Paris, France: International Academy of Astronautics (IAA), May 2019, pp. 1–109. URL: <https://bit.ly/3aob7w8> (visited on 01/18/2020) (cit. on p. 14).
- [31] ESA Space Debris Office. *ESA's Annual Space Environment Report*. Technical report GEN-DB-LOG-00288-OPS-SD. Darmstadt, Germany: ESA, Sept. 29, 2020, pp. 1–88. URL: <http://tinyurl.com/y3aovlb5> (visited on 09/29/2020) (cit. on pp. 14, 42, 43).
- [32] J.-C. Liou et al. *Stability of the Future LEO Environment*. Technical report IADC-12-08, Rev. 1. Vienna, Austria: Inter-Agency Space Debris Coordination Committee (IADC), Jan. 2013, pp. 1–26. URL: <https://bit.ly/2G5rW0u> (visited on 01/19/2020) (cit. on pp. 15, 16).
- [33] C. Bonnal et al. “Just in Time Collision Avoidance-A Review.” In: *2nd IAA Conference on Space Situational Awareness (ICSSA)*. Washington, DC, USA, Jan. 2020, pp. 1–28 (cit. on p. 17).
- [34] W. Fehse. *Automated Rendezvous and Docking of Spacecraft*. Ed. by M. J. Rycroft and W. Shyy. 2008th ed. Vol. 13. New York, USA: Cambridge University Press, 2003. ISBN: 1-139-44068-3. URL: <https://bit.ly/3aweYqT> (visited on 01/20/2020) (cit. on pp. 17, 18, 71, 125, 126).
- [35] G. S. Aglietti et al. “RemoveDEBRIS: An in-Orbit Demonstration of Technologies for the Removal of Space Debris.” In: *The Aeronautical Journal* 124.1271 (Jan. 2020), pp. 1–23. ISSN: 0001-9240, 2059-6464. DOI: 10/ggh8s5 (cit. on pp. 18, 24).
- [36] C. Bombardelli and J. Pelaez. “Ion Beam Shepherd for Contactless Space Debris Removal.” In: *Journal of Guidance, Control, and Dynamics* 34.3 (2011), pp. 916–920. DOI: 10/ft6xh4 (cit. on p. 19).
- [37] T. V. Peters and D. Escorial Olmos. “COBRA Contactless De-tumbling.” In: *CEAS Space Journal* 8.3 (Sept. 1, 2016), pp. 143–165. ISSN: 1868-2510. DOI: 10/ggh8s6 (cit. on p. 19).

- [38] C. R. Phipps et al. "Removing Orbital Debris with Lasers." In: *Advances in Space Research* 49.9 (May 1, 2012), pp. 1283–1300. ISSN: 0273-1177. DOI: 10/f3w7qx (cit. on p. 19).
- [39] R. Soulard et al. "ICAN: A Novel Laser Architecture for Space Debris Removal." In: *Acta Astronautica* 105.1 (Dec. 1, 2014), pp. 192–200. ISSN: 0094-5765. DOI: 10/f6ttwm (cit. on p. 19).
- [40] N. Ortiz Gomez. "Eddy Currents Applied to Space Debris Objects." PhD thesis. University of Southampton, Sept. 2017. 267 pp. URL: <https://bit.ly/2vb4opq> (visited on 02/24/2020) (cit. on pp. 19, 73).
- [41] H. Schaub and Z. Sternovsky. "Active Space Debris Charging for Contactless Electrostatic Disposal Maneuvers." In: *Advances in Space Research* 53.1 (Jan. 1, 2014), pp. 110–118. ISSN: 0273-1177. DOI: 10/f5qwqk (cit. on p. 19).
- [42] C. Bonnal, J.-M. Ruault, and M.-C. Desjean. "Active Debris Removal: Recent Progress and Current Trends." In: *Acta Astronautica* 85 (Apr. 1, 2013), pp. 51–60. ISSN: 0094-5765. DOI: 10/gf9rr6 (cit. on pp. 19, 69).
- [43] M. Shan, J. Guo, and E. Gill. "Review and Comparison of Active Space Debris Capturing and Removal Methods." In: *Progress in Aerospace Sciences* 80 (Jan. 1, 2016), pp. 18–32. ISSN: 0376-0421. DOI: 10/f78qqt (cit. on pp. 21, 22).
- [44] H. Hakima and M. R. Emami. "Assessment of Active Methods for Removal of LEO Debris." In: *Acta Astronautica* 144 (Mar. 2018), pp. 225–243. ISSN: 0094-5765. DOI: 10/gc2frb (cit. on pp. 21, 22, 24).
- [45] P. Couzin, X. Roser, and L. Strippoli. "Comparison of Active Debris Removal Architectures." In: *63rd International Astronautical Congress (IAC)*. Naples, Italy: International Astronautical Federation (IAF), Oct. 2012, pp. 1–9. URL: <https://bit.ly/2RVLydt> (visited on 01/27/2020) (cit. on pp. 21, 24).
- [46] B. Chamot. "Mission and System Architecture Design for Active Removal of Rocket Bodies in Low Earth Orbit." MA thesis. Ecole Polytechnique Federale de Lausanne (EPFL), 2012. 1-76. URL: <https://bit.ly/2REf5YY> (visited on 01/23/2020) (cit. on p. 21).
- [47] N. van der Pas et al. "Target Selection and Comparison of Mission Design for Space Debris Removal by DLR's Advanced Study Group." In: *Acta Astronautica* 102 (Sept. 1, 2014), pp. 241–248. ISSN: 0094-5765. DOI: 10/f6d9jq (cit. on pp. 21, 24).

- [48] V. Braun, E. Schulz, and C. Wiedemann. “Cost Estimation for the Active Debris Removal of Multiple Priority Targets.” In: *40th COSPAR Scientific Assembly*. Moscow, Russia, Aug. 2014, pp. 1–14. URL: <https://bit.ly/37wmFM0> (visited on 01/27/2020) (cit. on pp. 21, 24).
- [49] L. Innocenti. *CDF Study Report: E.Deorbit, e.Deorbit Assessment*. Technical report CDF-135(C). Noordwijk, Netherlands: ESTEC - ESA, Sept. 2012, pp. 1–265 (cit. on pp. 21, 25).
- [50] H. G. Lewis and A. Lidtke. “Active Debris Removal: Consequences of Mission Failure.” In: *65th International Astronautical Congress (IAC)*. Toronto, ON, Canada: International Astronautical Federation (IAF), Sept. 2014, p. 1. URL: <https://bit.ly/2t4GAmg> (visited on 01/27/2020) (cit. on p. 24).
- [51] R. Biesbroek et al. “E. Deorbit - ESA’s Active Debris Removal Mission.” In: *7th European Conference on Space Debris*. Darmstadt, Germany: ESA Space Debris Office, Apr. 2017, pp. 1–10. URL: <https://bit.ly/37yoMij> (visited on 01/27/2020) (cit. on p. 25).
- [52] L. Orman. *Information Paradox: Drowning in Information, Starving for Knowledge*. IEEE Technology and Society. June 29, 2017. URL: <https://bit.ly/2oKFSIe> (visited on 10/11/2019) (cit. on p. 25).
- [53] J. Naisbitt. *Megatrends: Ten New Directions Transforming Our Lives*. 6th ed. New York, NY, USA: Warner Books, 1982. 1–290. ISBN: 0-446-35681-6 (cit. on p. 25).
- [54] B. Kasenchak. *What Are Knowledge Graphs (vs. Other Kinds of KOS)?* Synaptica LLC. Dec. 9, 2019. URL: <https://bit.ly/3aMVLkZ> (visited on 01/27/2020) (cit. on p. 26).
- [55] J. Stichbury. *WTF Is a Knowledge Graph?* Hacker Noon. Apr. 26, 2017. URL: <https://bit.ly/2RRp2m4> (visited on 01/27/2020) (cit. on p. 26).
- [56] H. Paulheim. “Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods.” In: *Semantic Web 8* (Dec. 2016), pp. 489–508. DOI: 10/gc9zxx (cit. on p. 26).
- [57] L. Ehrlinger and W. Wöß. “Towards a Definition of Knowledge Graphs.” In: *Posters and Demos Track of 12th International Conference on Semantic Systems (SEMANTiCS2016) and 1st International Workshop on Semantic Change & Evolving Semantics (SuCESS16)*. Leipzig, Germany, Sept. 2016, pp. 1–5. URL: <https://bit.ly/2RtVtb2> (visited on 01/27/2020) (cit. on p. 26).
- [58] SAIC. *Space-Track.Org*. 2019. URL: <https://bit.ly/2V0nXNs> (visited on 10/12/2019) (cit. on pp. 26, 28).

- [59] M. P. Wilkins et al. "Towards an Artificial Space Object Taxonomy." In: *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*. Vol. 1. Maui, Hawaii, USA: Maui Economic Development Board, Inc., Sept. 2013, pp. 1–18. URL: <https://bit.ly/2EWSWPH> (visited on 01/27/2020) (cit. on p. 26).
- [60] C. Frueh et al. "Taxonomy and Classification Scheme for Artificial Space Objects." In: *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*. Vol. 1. Maui, Hawaii, USA: Maui Economic Development Board, Sept. 2013, pp. 1–13. URL: <https://bit.ly/2ryiH50> (cit. on p. 26).
- [61] A. P. Cox et al. "The Space Object Ontology." In: *2016 19th International Conference on Information Fusion (FUSION)*. Heidelberg, Germany: IEEE, July 2016, pp. 146–153. ISBN: 978-0-9964527-4-8. URL: <https://bit.ly/360WeNw> (cit. on pp. 27, 77).
- [62] R. J. Rovetto. "An Ontological Architecture for Orbital Debris Data." In: *Earth Science Informatics* 9.1 (Mar. 6, 2016), pp. 67–82. ISSN: 1865-0473. DOI: 10/gfxj3m (cit. on pp. 27, 77).
- [63] R. Furfaro et al. "Resident Space Object Characterization and Behavior Understanding via Machine Learning and Ontology-based Bayesian Networks." In: *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*. Wailea, Maui, Hawaii, USA: Maui Economic Development Board, Inc., Sept. 1, 2016, pp. 1–14. URL: <https://bit.ly/31kcbeR> (visited on 10/17/2019) (cit. on p. 27).
- [64] B. Liu, L. Yao, and D. Han. "Harnessing Ontology and Machine Learning for RSO Classification." In: *SpringerPlus* 5.1 (Dec. 26, 2016). ISSN: 2193-1801. DOI: 10/gf9zg3 (cit. on pp. 27, 77, 150).
- [65] S. Le May et al. "Leveraging Web Data and Graph Structures to Support Rapid Space Object Identification." In: *69th International Astronautical Congress (IAC)*. Bremen, Germany: International Astronautical Federation (IAF), 2018, pp. 1–10 (cit. on p. 27).
- [66] Union of Concerned Scientists (UCS). *UCS Satellite Database*. 2019. URL: <https://bit.ly/33BozbR> (visited on 10/17/2019) (cit. on p. 28).
- [67] D. Reintsema, K. Landzettel, and G. Hirzinger. "DLR's Advanced Telerobotic Concepts and Experiments for On-Orbit Servicing." In: *Advances in Telerobotics*. Ed. by M. Ferre et al. Springer Tracts in Advanced Robotics. Berlin, Heidelberg: Springer, Berlin, Heidelberg, 2007, pp. 323–345. ISBN: 978-3-540-71364-7. DOI: 10/cn56x8 (cit. on p. 28).

- [68] H. Ueno et al. "Space Robotic Mission Concepts for Capturing Stray Objects." In: *The Journal of Space Technology and Science* 18.2 (2002), pp. 1–8. DOI: 10/gbgj8 (cit. on p. 28).
- [69] R. H. M. Miller. *Space Applications of Automation, Robotics and Machine Intelligence Systems (ARAMIS). Volume 3: ARAMIS Overview*. Aug. 1, 1982. URL: <https://go.nasa.gov/320oqyW> (visited on 11/29/2019) (cit. on p. 28).
- [70] K. Yoshida et al. "Space Robotics." In: *Springer Handbook of Robotics*. 2nd ed. Springer Handbooks. Berlin, Heidelberg: Springer International Publishing, 2016, pp. 1423–1463. ISBN: 978-3-319-32550-7. DOI: 10/fph4 (cit. on pp. 28–31, 95).
- [71] Canadian Space Agency (CSA). *Canadarm2, the Canadian Robotic Arm on the Space Station*. 2019. URL: <https://bit.ly/2pCxS6R> (visited on 10/20/2019) (cit. on p. 30).
- [72] A. Kauderer. *Dextre Successfully Completes Its First Official Job*. NASA. Apr. 2, 2011. URL: <https://tinyurl.com/y6br4on5> (visited on 01/06/2021) (cit. on p. 30).
- [73] M. Garcia. *50 Years Ago: The First Automatic Docking in Space*. NASA. URL: <https://go.nasa.gov/2QRxktt> (visited on 01/06/2020) (cit. on p. 30).
- [74] K. Yoshida. "Achievements in Space Robotics." In: *IEEE Robotics Automation Magazine* 16.4 (Dec. 2009), pp. 20–28. ISSN: 1070-9932, 1558-223X. DOI: 10/b32pkh (cit. on p. 31).
- [75] S. Nolet. "Development of a Guidance, Navigation and Control Architecture and Validation Process Enabling Autonomous Docking to a Tumbling Satellite." PhD thesis. Boston, MA, USA: Massachusetts Institute of Technology (MIT), 2007. 1–324. URL: <https://tinyurl.com/yyq5uqft> (visited on 10/20/2019) (cit. on pp. 31, 34).
- [76] J. Delaval. *From Active Debris Removal to In-Orbit Servicing: The Legacy of e.Deorbit*. The Clean Space blog. Nov. 26, 2018. URL: <https://bit.ly/38D5l8x> (visited on 02/17/2020) (cit. on p. 32).
- [77] S. Estable et al. "Capturing and Deorbiting Envisat with an Airbus Spacetug. Results from the ESA e.Deorbit Consolidation Phase Study." In: 69th International Astronautical Congress (IAC). Bremen, Germany: International Astronautical Federation (IAF), Oct. 2018, pp. 1–15. URL: <https://bit.ly/2SZWrLI> (visited on 02/19/2020) (cit. on p. 32).
- [78] ESA. *ESA Commissions World's First Space Debris Removal*. Sept. 12, 2019. URL: <https://bit.ly/390r39Y> (visited on 02/19/2020) (cit. on p. 32).

- [79] E. Ackerman. *How NASA Will Grapple and Refuel a Satellite in Low Earth Orbit*. IEEE Spectrum: Technology, Engineering, and Science News. Feb. 10, 2019. URL: <https://bit.ly/2SyQ436> (visited on 02/17/2020) (cit. on p. 33).
- [80] B. Reed. *Restore-L: Robotic Servicing Mission*. Satellite Servicing Projects Division. Jan. 31, 2020. URL: <https://sspd.gsfc.nasa.gov/> (visited on 02/17/2020) (cit. on p. 33).
- [81] A. Flores-Abad et al. "A Review of Space Robotics Technologies for On-Orbit Servicing." In: *Progress in Aerospace Sciences* 68.10-11 (July 2014), pp. 1–26. ISSN: 03760421. DOI: 10/gfwsx2 (cit. on pp. 33, 35).
- [82] E. Papadopoulos. "On the Dynamics and Control of Space Manipulators." PhD thesis. Massachusetts Institute of Technology, 1990. URL: <https://bit.ly/2rcvjPM> (visited on 11/11/2019) (cit. on p. 34).
- [83] S. Dubowsky and E. Papadopoulos. "The Kinematics, Dynamics, and Control of Free-Flying and Free-Floating Space Robotic Systems." In: *IEEE Transactions on Robotics and Automation* 9.5 (Oct. 1993), pp. 531–543. ISSN: 1042-296X, 2374-958X. DOI: 10/d9t836 (cit. on pp. 34, 36).
- [84] K. Yoshida, D. Dimitrov, and H. Nakanishi. "On the Capture of Tumbling Satellite by a Space Robot." In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: IEEE, Oct. 2006, pp. 4127–4132. ISBN: 1-4244-0258-1. DOI: 10/dttv4v (cit. on pp. 34, 36, 40).
- [85] E. Papadopoulos and S. Dubowsky. "Dynamic Singularities in Free-floating Space Manipulators." In: *Space Robotics: Dynamics and Control*. Ed. by Y. Xu and T. Kanade. The Kluwer International Series in Engineering and Computer Science. Boston, MA, USA: Springer US, 1993, pp. 77–100. ISBN: 978-1-4615-3588-1. DOI: 10/fwrcw7 (cit. on pp. 34, 35).
- [86] E. Papadopoulos and S. Dubowsky. "On the Nature of Control Algorithms for Space Manipulators." In: *IEEE International Conference on Robotics and Automation Proceedings*. May 1990, 1102–1108 vol.2. DOI: 10/c84h5v (cit. on p. 34).
- [87] J. Virgili-Llop and M. Romano. "Simultaneous Capture and Detumble of a Resident Space Object by a Free-Flying Spacecraft-Manipulator System." In: *Frontiers in Robotics and AI* 6 (2019). ISSN: 2296-9144. DOI: 10/ggbgkw (cit. on pp. 34, 38).
- [88] Z. Vafa and S. Dubowsky. "The Kinematics and Dynamics of Space Manipulators: The Virtual Manipulator Approach." In: *The International Journal of Robotics Research* 9.4 (Aug. 1, 1990), pp. 3–21. ISSN: 0278-3649. DOI: 10/fgvpwv (cit. on p. 34).

- [89] W. Xu et al. "Non-Holonomic Path Planning of a Free-Floating Space Robotic System Using Genetic Algorithms." In: *Advanced Robotics* 22.4 (Jan. 1, 2008), pp. 451–476. ISSN: 0169-1864. DOI: 10/fv7nts (cit. on pp. 35, 36).
- [90] Y. Nakamura and R. Mukherjee. "Nonholonomic Path Planning of Space Robots via a Bidirectional Approach." In: *IEEE Transactions on Robotics and Automation* 7.4 (Aug. 1991), pp. 500–514. ISSN: 2374-958X. DOI: 10/bj3b28 (cit. on p. 35).
- [91] E. Papadopoulos. "Path Planning For Space Manipulators Exhibiting Nonholonomic Behavior." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 1. Raleigh, NC, USA: IEEE, 1992, pp. 669–675. ISBN: 978-0-7803-0738-4. DOI: 10/bt8htg (cit. on p. 35).
- [92] D. Nenchev, K. Yoshida, and Y. Umetani. "Analysis, Design and Control of Free-flying Space Robots Using Fixed-attitude-restricted Jacobian Matrix." In: *5th International Symposium on Robotics Research*. Cambridge, MA, USA: MIT Press, 1990, pp. 251–258. ISBN: 978-0-262-13253-4. URL: <https://bit.ly/2QNBP8p> (visited on 11/24/2019) (cit. on p. 35).
- [93] D. Nenchev, Y. Umetani, and K. Yoshida. "Analysis of a Redundant Free-Flying Spacecraft/Manipulator System." In: *IEEE Transactions on Robotics and Automation* 8.1 (Feb. 1992), pp. 1–6. ISSN: 2374-958X. DOI: 10/d4xhnd (cit. on p. 35).
- [94] K. Yoshida, K. Hashizume, and S. Abiko. "Zero Reaction Maneuver: Flight Validation with ETS-VII Space Robot and Extension to Kinematically Redundant Arm." In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 1. May 2001, 441–446 vol.1. DOI: 10/b5k4zg (cit. on p. 35).
- [95] J. Virgili Llop et al. "Autonomous Capture of a Resident Space Object by a Spacecraft with a Robotic Manipulator: Analysis, Simulation and Experiments." In: *AIAA/AAS Astrodynamics Specialist Conference*. AIAA SPACE Forum. American Institute of Aeronautics and Astronautics (AIAA), Sept. 9, 2016. DOI: 10/dhvm (cit. on p. 35).
- [96] W. Xu et al. "Target Berthing and Base Reorientation of Free-Floating Space Robotic System after Capturing." In: *Acta Astronautica* 64.2 (Jan. 1, 2009), pp. 109–126. ISSN: 0094-5765. DOI: 10/bqgz8h (cit. on p. 35).
- [97] R. Lampariello and G. Hirzinger. "Generating Feasible Trajectories for Autonomous On-Orbit Grasping of Spinning Debris in a Useful Time." In: *IEEE International Conference on Intelligent Robots and Systems*. Tokyo, Japan: DLR, 2013, pp. 5652–5659. ISBN: 978-1-4673-6358-7. DOI: 10/ggfvqx (cit. on pp. 35, 113, 122, 145).

- [98] A. Flores-Abad et al. "Optimal Capture of a Tumbling Object in Orbit Using a Space Manipulator." In: *Journal of Intelligent & Robotic Systems* 86.2 (May 1, 2017), pp. 199–211. ISSN: 1573-0409. DOI: 10/gbghrz (cit. on p. 35).
- [99] S. Dubowsky, E. E. Vance, and M. A. Torres. "The Control of Space Manipulators Subject to Spacecraft Attitude Control Saturation Limits." In: *NASA Conference on Space Telerobotics*. Vol. 4. Pasadena, CA, USA: JPL, California Inst. of Tech, Jan. 31, 1989, pp. 409–418. URL: <https://go.nasa.gov/2t0N84T> (visited on 11/07/2019) (cit. on p. 36).
- [100] M. A. Torres and S. Dubowsky. "Minimizing Spacecraft Attitude Disturbances in Space Manipulator Systems." In: *Journal of guidance, control, and dynamics* 15.4 (1992), pp. 1010–1017. DOI: 10/dmx8tp (cit. on p. 36).
- [101] Z. Vafa and S. Dubowsky. "On the Dynamics of Space Manipulators Using the Virtual Manipulator with Applications to Path Planning." In: *Space Robotics: Dynamics and Control*. Ed. by Y. Xu and T. Kanade. The Kluwer International Series in Engineering and Computer Science. Boston, MA, USA: Springer US, 1993, pp. 45–76. ISBN: 978-1-4615-3588-1. DOI: 10/cvmv7x (cit. on p. 36).
- [102] D. Dimitrov. "Dynamics and Control of Space Manipulators During a Satellite Capturing Operation." PhD thesis. Sendai, Miyagi, Japan: Tohoku University, 2006. 166 pp. URL: <https://bit.ly/2P3uBhi> (visited on 10/20/2019) (cit. on pp. 36, 37).
- [103] F. Aghili. "Pre- and Post-Grasping Robot Motion Planning to Capture and Stabilize a Tumbling/Drifting Free-Floater with Uncertain Dynamics." In: *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, May 2013, pp. 5461–5468. DOI: 10/ggc3cg (cit. on p. 37).
- [104] S. Jaekel et al. "Design and Operational Elements of the Robotic Subsystem for the e.Deorbit Debris Removal Mission." In: *Frontiers in Robotics and AI* 5 (2018). ISSN: 2296-9144. DOI: 10/ggc3m7 (cit. on pp. 37, 104, 120, 121, 125).
- [105] J. Virgili-Llop et al. "A Convex-Programming-Based Guidance Algorithm to Capture a Tumbling Object on Orbit Using a Spacecraft Equipped with a Robotic Manipulator." In: *The International Journal of Robotics Research* 38.1 (Jan. 1, 2019), pp. 40–72. ISSN: 0278-3649. DOI: 10/ggbqjc (cit. on pp. 37, 119, 122).
- [106] H. Nagamatsu, T. Kubota, and I. Nakatani. "Capture Strategy for Retrieval of a Tumbling Satellite by a Space Robotic Manipulator." In: *Proceedings of IEEE International Conference on Robotics and Automation*. Vol. 1. Minneapolis, MN, USA: IEEE, Apr. 1996, 70–75 vol.1. ISBN: 0-7803-2988-0. DOI: 10/d3ndxg (cit. on p. 38).

- [107] Y. Tsuda and S. Nakasuka. "New Attitude Motion Following Control Algorithm for Capturing Tumbling Object in Space." In: *Acta Astronautica* 53.11 (Dec. 1, 2003), pp. 847–861. ISSN: 0094-5765. DOI: 10/bjcm99 (cit. on p. 39).
- [108] Z. Ma, O. Ma, and B. N. Shashikanth. "Optimal Control for Spacecraft to Rendezvous with a Tumbling Satellite in a Close Range." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: IEEE, Oct. 2006, pp. 4109–4114. DOI: 10/crzsk5 (cit. on p. 39).
- [109] M. Oda and Y. Ohkami. "Coordinated Control of Spacecraft Attitude and Space Manipulators." In: *Control Engineering Practice* 5.1 (Jan. 1, 1997), pp. 11–21. ISSN: 0967-0661. DOI: 10/c3p5q3 (cit. on p. 39).
- [110] S. Matunaga, T. Kanzawa, and Y. Ohkami. "Rotational Motion-Damper for the Capture of an Uncontrolled Floating Satellite." In: *Control Engineering Practice* 9.2 (Feb. 1, 2001), pp. 199–205. ISSN: 0967-0661. DOI: 10/bpvdown (cit. on p. 39).
- [111] S. Yoshikawa and K. Yamada. "Impulsive Control for Angular Momentum Management of Tumbling Spacecraft." In: *Acta Astronautica* 60.10 (May 1, 2007), pp. 810–819. ISSN: 0094-5765. DOI: 10/dq25f6 (cit. on p. 40).
- [112] S.-I. Nishida and S. Kawamoto. "Strategy for Capturing of a Tumbling Space Debris." In: *Acta Astronautica* 68.1 (Jan. 1, 2011), pp. 113–120. ISSN: 0094-5765. DOI: 10/cwks2z (cit. on p. 40).
- [113] F. Sugai et al. "Detumbling an Uncontrolled Satellite with Contactless Force by Using an Eddy Current Brake." In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Tokyo, Japan: IEEE, Nov. 2013, pp. 783–788. DOI: 10/gghbtd (cit. on p. 40).
- [114] J. H. Saleh and J.-F. Castet. *Spacecraft Reliability and Multi-State Failures: A Statistical Approach*. John Wiley, 2011. 206 pp. ISBN: 978-0-470-68791-8. URL: <https://bit.ly/2SZrKre> (cit. on pp. 42, 44–51, 56, 149).
- [115] F. Letizia, S. Lemmens, and H. Krag. "Environment Capacity as an Early Mission Design Driver." In: *70th International Astronautical Congress (IAC)*. Washington, DC, USA: International Astronautical Federation (IAF), 2019, pp. 1–15 (cit. on p. 42).
- [116] F. Letizia, S. Lemmens, and H. Krag. "Environment Capacity as an Early Mission Design Driver." In: *Acta Astronautica* (May 1, 2020). ISSN: 0094-5765. DOI: 10/ggtvns (cit. on p. 42).
- [117] ESA Space Debris Office. *Database and Information System Characterising Objects in Space (DISCOS)*. 2019. URL: <https://bit.ly/2MnBxnT> (visited on 10/11/2019) (cit. on pp. 43, 62).

- [118] D. G. Kleinbaum and M. Klein. *Survival Analysis: A Self-Learning Text*. 2nd ed. Statistics for Biology and Health. New York: Springer-Verlag, 2005. ISBN: 978-0-387-29150-5. DOI: 10/d74brq (cit. on pp. 44, 45, 47).
- [119] M. Rausand and A. Høyland. *System Reliability Theory: Models, Statistical Methods, and Applications*. 2nd ed. Vol. 396. Wiley Series in Probability and Statistics. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2004. 644 pp. ISBN: 978-0-470-31690-0. DOI: 10/bcwb2t (cit. on pp. 44–49, 52).
- [120] J. F. Lawless. *Statistical Models and Methods for Lifetime Data*. 2nd ed. Vol. 362. Wiley Series in Probability and Statistics. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2003. 663 pp. ISBN: 978-1-118-03300-5. DOI: 10/cqstk3 (cit. on pp. 45, 49, 50).
- [121] Wikipedia contributors. *Pareto Chart*. In: *Wikipedia*. Feb. 20, 2020. URL: <https://bit.ly/32tRAXr> (visited on 02/26/2020) (cit. on p. 53).
- [122] R. A. Braeunig. *Rocket Propellants*. Rocket & Space Technology. 2008. URL: <https://bit.ly/2Pvvd52> (visited on 02/27/2020) (cit. on pp. 54, 63).
- [123] IEEE Robotics and Automation Society. Standing Committee for Standards Activities. “IEEE Standard Ontologies for Robotics and Automation.” In: *IEEE Std 1872-2015* (Apr. 2015), pp. 1–60. ISSN: null. DOI: 10/gf9rr3 (cit. on p. 59).
- [124] W3C OWL Working Group. *OWL 2 Web Ontology Language Document Overview (Second Edition)*. Dec. 2012. URL: <https://bit.ly/395C4nh> (visited on 02/27/2020) (cit. on pp. 60, 75).
- [125] M. Horridge et al. *A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. Manual. Manchester, UK: The University Of Manchester, Mar. 2011, pp. 1–108. URL: <https://bit.ly/1YTKHW8> (visited on 09/03/2019) (cit. on p. 60).
- [126] A. De Nicola, M. Missikoff, and R. Navigli. “A Software Engineering Approach to Ontology Building.” In: *Information Systems* 34.2 (Apr. 1, 2009), pp. 258–275. ISSN: 0306-4379. DOI: 10/c9hxdx (cit. on pp. 61, 62).
- [127] M. Horridge and P. F. Patel-Schneider. *OWL 2 Web Ontology Language Manchester Syntax (Second Edition)*. The World Wide Web Consortium (W3C). Dec. 11, 2012. URL: <https://bit.ly/2pVW6oX> (visited on 05/13/2020) (cit. on p. 63).
- [128] ECSS Secretariat. *Space Product Assurance: Failure Modes, Effects (and Criticality) Analysis (FMEA/FMECA)*. Standard ECSS-Q-ST-30-02C. Noordwijk, The Netherlands: ESA Requirements and Standards Division / ESA-ESTEC, Mar. 2009, pp. 23–24. URL:

- <http://www.ecss.nl/> (visited on 09/03/2019) (cit. on pp. 65, 66, 68).
- [129] M. M. Castronuovo. "Active Space Debris Removal—A Preliminary Mission Analysis and Design." In: *Acta Astronautica* 69.9 (Nov. 1, 2011), pp. 848–859. ISSN: 0094-5765. DOI: 10/c84vm2 (cit. on p. 69).
- [130] ECSS Secretariat. *Space Engineering: Mechanisms*. Standard ECSS-E-ST-33-01C Rev.2. Noordwijk, The Netherlands: ESA Requirements and Standards Division / ESA-ESTEC, Mar. 2019, p. 45. URL: <http://www.ecss.nl/> (visited on 09/03/2019) (cit. on p. 71).
- [131] Python Software Foundation. *3.6.10 Documentation*. Python. May 16, 2020. URL: <https://bit.ly/2zN8bLu> (visited on 05/16/2020) (cit. on p. 73).
- [132] M. A. Musen. "The Protégé Project: A Look Back and a Look Forward." In: *AI Matters* 1.4 (June 16, 2015), pp. 4–12. DOI: 10/gf2vm9 (cit. on pp. 73, 75).
- [133] J.-B. Lamy. "Owlready: Ontology-oriented Programming in Python with Automatic Classification and High Level Constructs for Biomedical Ontologies." In: *Artificial Intelligence in Medicine* 80 (July 1, 2017), pp. 11–28. ISSN: 0933-3657. DOI: 10/gb4pjz (cit. on pp. 73, 75).
- [134] ISO IEC JTC 1/SC 7. *ISO 5807:1985 - Information Processing— Documentation Symbols and Conventions for Data, Program and System Flowcharts, Program Network Charts and System Resources Charts*. Standard ISO 5807:1985. Geneva, Switzerland: International Organization for Standardization (ISO), Feb. 1985, pp. 1–25. URL: <https://tinyurl.com/yfev8zp3> (visited on 10/11/2021) (cit. on pp. 73, 79, 118).
- [135] M. Wade. *Encyclopedia Astronautica*. 2019. URL: <https://bit.ly/2pBCa45> (visited on 03/01/2020) (cit. on p. 74).
- [136] G. D. Krebs. *Gunter's Space Page - Information on Spaceflight, Launch Vehicles and Satellites*. Feb. 20, 2020. URL: <https://bit.ly/2oKqprF> (visited on 03/01/2020) (cit. on p. 74).
- [137] ESA. *eoPortal - Earth Observation Directory & News*. 2020. URL: <https://bit.ly/2pDGTlX> (visited on 03/01/2020) (cit. on p. 74).
- [138] A. Zak. *RussianSpaceWeb.Com*. Feb. 27, 2020. URL: <https://bit.ly/2o56DqM> (visited on 03/01/2020) (cit. on p. 74).
- [139] *Ariane 5 User's Manual*. Issue 5, Revision 2. Manual. Boulevard de l'Europe, BP 177 91006, Evry-Courcouronnes Cedex -France: Arianespace S.A., Oct. 2016, pp. 1–271. URL: <https://bit.ly/2pz0eVe> (visited on 09/03/2019) (cit. on p. 74).

- [140] *Atlas 5 User's Guide*. Revision 11. Manual. 9100 East Mineral Circle, Centennial, CO 80112: United Launch Alliance, LLC., Mar. 2010, pp. 1–420. URL: <https://bit.ly/2oKM6I6> (visited on 09/03/2019) (cit. on p. 74).
- [141] E. Linder et al. "Extraction of Spin Periods of Space Debris from Optical Light Curves." In: *66th International Astronautical Congress (IAC)*. Jerusalem, Israel: International Astronautical Federation (IAF), Oct. 2015, pp. 1–9. DOI: 10/ggm343 (cit. on p. 74).
- [142] T. Schildknecht, H. Krag, and T. Flohrer. "Determining, Monitoring and Modelling the Attitude Motion of Potential ADR Targets." 2016 Clean Space Industrial Days (Noordwijk, Netherlands). May 2016. URL: <https://bit.ly/2oQpnKD> (visited on 09/03/2019) (cit. on p. 74).
- [143] T. Schildknecht et al. "Attitude States of Space Debris Determined from Optical Light Curve Observations." In: *1st IAA Conference on Space Situational Awareness (ICSSA)*. Orlando, FL, USA: Bern Open Repository and Information System (BORIS), Nov. 2017, pp. 1–4. DOI: 10/ggm349 (cit. on p. 74).
- [144] J. Silha et al. "Debris Attitude Motion Measurements and Modelling by Combining Different Observation Techniques." In: *7th European Conference on Space Debris*. Ed. by T. Flohrer and F. Schmitz. Vol. 7. ESA Space Debris Office, Darmstadt, Germany: ESA Space Debris Office, June 2017, pp. 1–12. URL: <https://bit.ly/2VCS7F8> (visited on 09/03/2019) (cit. on p. 74).
- [145] J. Reback et al. *Pandas: Vo.25.2*. Version 0.25.2. Oct. 2019. DOI: 10/ggwd6w (cit. on p. 74).
- [146] S. van der Walt, S. C. Colbert, and G. Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation." In: *Computing in Science Engineering* 13.2 (Mar. 2011), pp. 22–30. ISSN: 1558-366X. DOI: 10/d8k4p9 (cit. on p. 74).
- [147] C. Davidson-Pilon et al. *Lifelines: Vo.24.5*. Zenodo, May 1, 2020. DOI: 10/dvv7 (cit. on p. 74).
- [148] M. McKerns et al. "Building a Framework for Predictive Science." In: *10th Python in Science Conference (SCIPY)*. 2011, pp. 1–12. DOI: 10/ggwd62 (cit. on p. 74).
- [149] DFKI GmbH. *D-Rock - Models, Methods and Tools for the Model Based Software Development of Robots*. May 15, 2020. URL: <https://tinyurl.com/y8h37v2k> (visited on 05/16/2020) (cit. on p. 75).
- [150] DFKI GmbH. *Q-Rock - AI-based Qualification of Deliberative Behaviour for a Robotic Construction Kit*. May 15, 2020. URL: <https://tinyurl.com/yyn59uqa> (visited on 05/16/2020) (cit. on p. 75).

- [151] B. Glimm et al. "Hermit: An OWL 2 Reasoner." In: *Journal of Automated Reasoning* 53.3 (Oct. 1, 2014), pp. 245–269. ISSN: 1573-0670. DOI: 10/ggwfdj (cit. on p. 75).
- [152] E. Sirin et al. "Pellet: A Practical OWL-DL Reasoner." In: *Journal of Web Semantics* 5.2 (June 2007), pp. 51–53. DOI: 10/bhcnh9 (cit. on p. 75).
- [153] The Board of Trustees of the Leland Stanford Junior University. *Protégé Products*. Protégé. 2019. URL: <https://stanford.io/2oIrRuG> (visited on 03/02/2020) (cit. on p. 75).
- [154] The Board of Trustees of the Leland Stanford Junior University. *Protégé 5 Documentation*. Sept. 1, 2020. URL: <https://tinyurl.com/ygbjdcfs> (visited on 10/11/2021) (cit. on p. 77).
- [155] University of Manchester. *Ontology Design Patterns (ODPs) Public Catalog*. July 9, 2009. URL: <https://bit.ly/2nWZaKE> (visited on 08/11/2019) (cit. on p. 77).
- [156] B. Rodriguez-Castro, M. Ge, and M. Hepp. "Alignment of Ontology Design Patterns: Class As Property Value, Value Partition and Normalisation." In: *On the Move to Meaningful Internet Systems: OTM 2012*. Ed. by R. Meersman et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 682–699. ISBN: 978-3-642-33615-7. DOI: 10/ggm584 (cit. on p. 77).
- [157] I. Horrocks et al. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. May 2004. URL: <https://bit.ly/2o5Q3Hq> (visited on 08/02/2019) (cit. on p. 78).
- [158] T. C. Nguyen Huynh. "Adaptive Reactionless Control of a Space Manipulator for Post-Capture of an Uncooperative Tumbling Target." Montreal, Canada: McGill University, 2013. 1-155. URL: <https://bit.ly/35TffrK> (visited on 12/02/2019) (cit. on pp. 88–91, 95, 96).
- [159] B. Siciliano et al. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2009. ISBN: 978-1-84628-641-4. DOI: 10/c9k3kd (cit. on pp. 88, 91–93, 95).
- [160] K. Yoshida and Y. Umetani. "Control of Space Manipulators with Generalized Jacobian Matrix." In: *Space Robotics: Dynamics and Control*. Ed. by Y. Xu and T. Kanade. The Kluwer International Series in Engineering and Computer Science. Boston, MA: Springer US, 1993, pp. 165–204. ISBN: 978-1-4615-3588-1. DOI: 10/dftsn3 (cit. on pp. 89–91, 94).

- [161] P. Corke. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised, Extended And Updated Edition*. 2nd ed. Springer Tracts in Advanced Robotics. Springer International Publishing, 2017. ISBN: 978-3-319-54412-0. DOI: 10/fph5 (cit. on pp. 90–92, 119).
- [162] M. Wilde et al. “Equations of Motion of Free-Floating Spacecraft-Manipulator Systems: An Engineer’s Tutorial.” In: *Frontiers in Robotics and AI* 5 (2018). ISSN: 2296-9144. DOI: 10/ghk2gt (cit. on p. 91).
- [163] K. M. Lynch and F. C. Park. *Modern Robotics: Mechanics, Planning, and Control*. 1st edition. Cambridge: Cambridge University Press, May 25, 2017. 544 pp. ISBN: 978-1-107-15630-2. URL: <https://tinyurl.com/y7oluzh3> (visited on 11/30/2019) (cit. on pp. 92–94, 110, 112, 116).
- [164] J. Yuan. “Closed-Loop Manipulator Control Using Quaternion Feedback.” In: *IEEE Journal on Robotics and Automation* 4.4 (Aug. 1988), pp. 434–440. ISSN: 2374-8710. DOI: 10/fc7pjw (cit. on p. 92).
- [165] Y. Yang. “Quaternion Based Model for Momentum Biased Nadir Pointing Spacecraft.” In: *Aerospace Science and Technology* 14.3 (Apr. 1, 2010), pp. 199–202. ISSN: 1270-9638. DOI: 10/c4m4km (cit. on pp. 93, 116).
- [166] A. Rao. “A Survey of Numerical Methods for Optimal Control.” In: *2009 AAS/AIAA Astrodynamics Specialist Conference*. Vol. 135. San Diego, CA, USA: Published for the American Astronautical Society by Univelt, Aug. 2009, pp. 497–528. URL: <https://tinyurl.com/ya5fu7dk> (visited on 11/10/2020) (cit. on pp. 96, 97, 99, 100, 103).
- [167] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. 2nd ed. Advances in Design and Control. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, Jan. 2010. 1–441. ISBN: 978-0-89871-688-7. DOI: 10/b4cf64 (cit. on pp. 96–103, 113, 115, 116, 145, 146).
- [168] J. T. Betts. “Survey of Numerical Methods for Trajectory Optimization.” In: *Journal of Guidance, Control, and Dynamics* 21.2 (Mar. 1, 1998), pp. 193–207. DOI: 10/c27qkb (cit. on pp. 96, 97, 100).
- [169] M. Kelly. “An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation.” In: *SIAM Review* 59.4 (Jan. 1, 2017), pp. 849–904. ISSN: 0036-1445. DOI: 10/gcj6tr (cit. on pp. 100, 103, 113, 117).

- [170] B. W. Barbee et al. "Guidance and Navigation for Rendezvous and Proximity Operations with a Non-Cooperative Spacecraft at Geosynchronous Orbit." In: *George H. Born Symposium*. Boulder, Colorado, USA: NASA Goddard Space Flight Center, 2010, pp. 1–21. URL: <https://tinyurl.com/y3zjjwef> (visited on 12/08/2020) (cit. on p. 104).
- [171] H. Goldstein, C. P. Jr, and J. Safko. *Classical Mechanics*. 3rd edition. San Francisco, NJ: Pearson, June 15, 2001. 664 pp. ISBN: 978-0-201-65702-9. URL: <https://tinyurl.com/ycjhzcy9> (visited on 11/09/2020) (cit. on pp. 107–109).
- [172] G. Avanzini. *Spacecraft Attitude Dynamics and Control*. Mar. 2009. URL: <https://tinyurl.com/y2jeufgo> (visited on 03/09/2020) (cit. on pp. 107–109).
- [173] F. L. Markley and J. L. Crassidis. "Attitude Kinematics and Dynamics." In: *Fundamentals of Spacecraft Attitude Determination and Control*. Ed. by F. L. Markley and J. L. Crassidis. Space Technology Library. New York, NY: Springer, 2014, pp. 67–122. ISBN: 978-1-4939-0802-8. DOI: 10/fph3 (cit. on p. 108).
- [174] P. C. Hughes. *Spacecraft Attitude Dynamics*. Mineola, N.Y: Dover Publications, Dec. 17, 2004. 592 pp. ISBN: 978-0-486-43925-9. URL: <https://tinyurl.com/ygnjsprw> (visited on 11/24/2020) (cit. on p. 109).
- [175] A. Ellery. "An Engineering Approach to the Dynamic Control of Space Robotic On-Orbit Servicers." In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 218.2 (Feb. 1, 2004), pp. 79–98. ISSN: 0954-4100. DOI: 10/fjpwjv (cit. on p. 110).
- [176] R. Lampariello. "Motion Planning for the On-orbit Grasping of a Non-cooperative Target Satellite with Collision Avoidance." In: *10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2010)*. Sapporo, Japan: JAXA, 2010, pp. 1–8 (cit. on pp. 113, 122).
- [177] The MathWorks, Inc. *MATLAB Documentation*. 2021. URL: <https://tinyurl.com/y2gep5w6> (visited on 02/08/2021) (cit. on p. 117).
- [178] The MathWorks, Inc. *Robotics System Toolbox Documentation*. 2021. URL: <https://tinyurl.com/y3ym7ktz> (visited on 02/08/2021) (cit. on p. 117).
- [179] The MathWorks, Inc. *MATLAB Coder Documentation*. 2021. URL: <https://tinyurl.com/y5at4p7k> (visited on 02/08/2021) (cit. on p. 119).
- [180] The MathWorks, Inc. *Simulink Documentation*. 2021. URL: <https://tinyurl.com/y6mcdgs3> (visited on 02/13/2021) (cit. on p. 119).

- [181] The MathWorks, Inc. *Simscape Multibody Documentation*. 2021. URL: <https://tinyurl.com/y2jdpypf> (visited on 02/13/2021) (cit. on p. 119).
- [182] J. Virgili-Llop. *SPART: SPACecraft Robotics Toolkit*. Version 0.1. Monterey, CA, USA: Naval Postgraduate School, 2017. URL: <https://tinyurl.com/yxd6awz1> (visited on 11/17/2019) (cit. on p. 119).
- [183] M. Tincknell. *Quaternion*. Version 1.8.0.0. Aug. 24, 2017. URL: <https://tinyurl.com/y2wl2u76> (visited on 02/08/2021) (cit. on p. 119).
- [184] Y. Nie, O. Faqir, and E. C. Kerrigan. "ICLOCS2: Try This Optimal Control Problem Solver Before You Try the Rest." In: *2018 UKACC 12th International Conference on Control (CONTROL)*. 2018 UKACC 12th International Conference on Control (CONTROL). Sept. 2018, pp. 336–336. DOI: 10/ghd8bt (cit. on p. 119).
- [185] A. Wächter and L. T. Biegler. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming." In: *Mathematical Programming* 106.1 (Mar. 1, 2006), pp. 25–57. ISSN: 1436-4646. DOI: 10/c6j59p (cit. on p. 119).
- [186] E. Kyle. *Kosmos 3M Data Sheet*. Space Launch Report. Oct. 30, 2005. URL: <https://tinyurl.com/2jj3tjd6> (visited on 07/04/2021) (cit. on pp. 120, 121).
- [187] D. McKnight et al. "Identifying the 50 Statistically-Most-Concerning Derelict Objects in LEO." In: *Acta Astronautica* 181 (Apr. 1, 2021), pp. 282–291. ISSN: 0094-5765. DOI: 10/ghwh42 (cit. on p. 120).
- [188] N. Praly et al. "Study on the Eddy Current Damping of the Spin Dynamics of Space Debris from the Ariane Launcher Upper Stages." In: *Acta Astronautica* 76 (2012), pp. 145–153. ISSN: 00945765. DOI: 10/gkptqn (cit. on p. 122).
- [189] E. Gilbert, D. Johnson, and S. Keerthi. "A Fast Procedure for Computing the Distance between Complex Objects in Three-Dimensional Space." In: *IEEE Journal on Robotics and Automation* 4.2 (Apr. 1988), pp. 193–203. ISSN: 2374-8710. DOI: 10/d7q7k3 (cit. on p. 145).
- [190] B. E. Jackson, K. Tracy, and Z. Manchester. "Planning with Attitude." In: *IEEE Robotics and Automation Letters* (Jan. 2021). URL: <https://tinyurl.com/yjsecnhb> (cit. on p. 146).
- [191] J. Schulman et al. "Motion Planning with Sequential Convex Optimization and Convex Collision Checking." In: *The International Journal of Robotics Research* 33.9 (Aug. 1, 2014), pp. 1251–1270. ISSN: 0278-3649. DOI: 10/f6j8k3 (cit. on p. 151).

- [192] M. Kalakrishnan et al. "STOMP: Stochastic Trajectory Optimization for Motion Planning." In: *2011 IEEE International Conference on Robotics and Automation*. May 2011, pp. 4569–4574. DOI: 10/fxbq67 (cit. on p. 151).
- [193] ESA. *ACROBAT – Air Cushion ROBotic plATform*. ESA - Automation and Robotics. 2021. URL: <https://tinyurl.com/yhxeapph> (visited on 05/03/2021) (cit. on p. 151).
- [194] ESA. *ESA's Neutral Buoyancy Facility at EAC*. 3/02/20215. URL: <https://tinyurl.com/yfy3bewx> (visited on 05/03/2021) (cit. on p. 151).
- [195] NASA. *NASA - MicroGravity Test Facility*. Ames Technology Capabilities and Facilities. Mar. 29, 2008. URL: <https://tinyurl.com/yeqe4z9n> (visited on 05/03/2021) (cit. on p. 151).
- [196] DLR. *On-Orbit Servicing Simulator (OOS-SIM)*. DLR - Institute of Robotics and Mechatronics. 2020. URL: <https://tinyurl.com/yj7znjuo> (visited on 05/03/2021) (cit. on p. 151).

DECLARATION

I hereby declare that I have:

1. produced the doctoral thesis without unauthorized outside help,
2. not used any sources or aids other than those indicated by me,
3. marked the places taken from the works used, either literally or in terms of content, as such.

Bremen, March 28, 2022

Marko Jankovic

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both \LaTeX and $\text{L}\text{\AA}\text{X}$:

<https://bitbucket.org/amiede/classicthesis/>

Final Version as of March 28, 2022 (v2.0).