POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Near Optimal Incremental Sampling-Based Path Planning for Free-Flying Space Manipulator

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Andrea Allevi**

Student ID: 920304
Advisor: Prof. Mauro MASSARI
Academic Year: 2021-22

Contents

Contents **i**

List of Figures **iii**

List of Tables **v**

Abstract **vii**

Sommario **ix**

1 Introduction **1**

1.1 Literature Review . . . 3

1.1.1 Introduction to Orbital Robotics . . . 3

1.1.2 Introduction to Path Planning Problem . . . 8

1.2 Thesis Recap and Goals . . . 14

1.3 Thesis Organization . . . 15

2 System Modelling **17**

2.1 Reference Frames . . . 17

2.2 Translational Motion Model . . . 19

2.2.1 Clohessy-Wiltshire Equations . . . 19

2.3 Rotational Motion Model . . . 20

2.3.1 Target attitude Kinematics . . . 20

2.3.2 Target Attitude Dynamics . . . 21

2.3.3 Grasping Point Kinematics . . . 21

2.4 Free-Flying Space Manipulator Kinematics Model . . . 23

2.4.1 Direct Kinematics . . . 24

2.4.2 Differential Kinematics . . . 28

2.4.3 Inverse Kinematics . . . 30

3 Spacecraft-Manipulator System Path Planning **35**
3.1 Path Planning Problem Formalization 35
3.2 Proposed Path Planning Approach 37
3.2.1 Trajectory Planning of the Chaser Base CoM 39
3.2.2 Trajectory Planning of the Chaser Base Attitude 45
3.2.3 Trajectory Planning of the Robotic Manipulator 46

4 Results **63**
4.1 Simulation Setup 63
4.1.1 Target and Chaser Spacecraft Models 63
4.1.2 Path Planning Algorithm Parameters 67
4.2 Simulation Results 70
4.3 Analysis on Variation of Parameters 75
4.3.1 LQR/APF Proposed Algorithm 75
4.3.2 Near-Optimal Incremental Sampling Based Algorithm 77

5 Conclusions **83**
5.1 Future Developments 83

Bibliography **85**

List of Figures

2.1 Local Vertical Local Horizon frame 18
2.2 Safe and unsafe region in target proximity 23
2.3 Space manipulator system 25
2.4 Deanvit-Hartenberg kinematic parameters [93] 26
2.5 Representation of vectors needed for computing Jacobian matrices [80] . . 29
2.6 Scheme of Inverse Kinematics algorithm with Jacobian pseudo-inverse [80] 31

3.1 Proposed path planning approach scheme 38
3.2 Weighting matrices components Q_{jj} and R_{jj} as function of distance d . . . 41
3.3 2D representation of potential function $U(\boldsymbol{r}) = M(\boldsymbol{r})\, Z(\boldsymbol{r})$ 43
3.4 APF control acceleration $\mathbf{u}_{\text{APF}}$ vector field 44
3.5 Scheme of the NOISBPP algorithm 48
3.6 Sampling limits definition 51

4.1 Target body and grasping point geometry 64
4.2 Chaser spacecraft and robotic manipulator 65
4.3 Planned trajectory of the end-effector position in LVLH frame 71
4.4 Chaser base control acceleration components in LVLH frame 72
4.5 Planned trajectory of the end-effector position in Chaser base Body frame 73
4.6 Manipulability index M time history 73
4.7 Shoulder joints $i = 1, 2, 3$ time histories 74
4.8 Elbow joint $i = 4$ time history 74
4.9 Wrist joints $i = 5, 6, 7$ time histories 75
4.10 Maneuver duration t_f varying LQR gain update interval Δt_{LQR} 76
4.11 Maneuver fuel consumption $C_{f_{tot}}$ varying LQR gain update interval Δt_{LQR} . 77
4.12 End-effector trajectory varying interval Δt_{int} 81

List of Tables

2.1 Denavit-Hartenberg parameters and their geometric meaning 26

4.1 Target Body orbit, dimensions and inertia properties 64
4.2 Grasping Point position and orientation with respect Target Body 64
4.3 Chaser base main properties 65
4.4 DH parameters for customized SSMRS-type manipulator 66
4.5 Robotic manipulator additional parameters 67
4.6 Joint angles excursion limits 67
4.7 Joints maximum velocity and acceleration 67
4.8 Adaptive LQR/APF algorithm input parameters 68
4.9 Near-Optimal Incremental Sampling Based algorithm input parameters . . 69
4.10 Inverse Kinematics algorithm input parameters 69
4.11 Numerical simulation inititial conditions 70
4.12 Planned maneuver duration and total fuel consumption 72
4.13 Simulations results with $N_S = 50$ 78
4.14 Joint angles time histories classification with $N_S = 50$ 79
4.15 Joint velocities time histories classification with $N_S = 50$ 79
4.16 Simulations results with $N_S = 100$ 80
4.17 Joint angles time histories classification with $N_S = 100$ 80
4.18 Joint velocities time histories classification with $N_S = 100$ 80

Abstract

This thesis proposes a solution to the problem of planning the trajectory of a satellite equipped with a robotic manipulator in charge of hooking up a second client satellite. This problem, in the context of the study of On-Orbit Servicing missions and active removal of space debris, is increasing interest in sustainable and efficient future use of space. The thesis aims to define a flexible approach to the problem and to propose an algorithm for planning the motion of the robotic arm based on an incremental definition of the trajectory through the sampling of the configuration space and guided towards the target by an optimization process .

Keywords: space robotics, path planning, sampling based

Sommario

Il presente lavoro di tesi propone una soluzione al problema di pianificazione della traiettoria di un satellite equipaggiato con un manipolatore robotico incaricato di agganciare un secondo satellite cliente. Questo problema, nel contesto dello studio di missioni di On-Orbit Servicing e rimozione attiva dei detriti spaziali sta aumentando di interesse per un uso futuro sostenibile ed efficiente dello spazio. La tesi si propone di definire una strategia flessibile di approccio al problema e di proporre un algoritmo per la pianificazione del moto del braccio robotico basato su una definizione incrementale della traiettoria attraverso il campionamento dello spazio delle configurazioni e guidata versol'obiettivo da un processo di ottimizzazione.

Parole chiave: robotica spaziale, pianificazione del moto, campionamento

1 | Introduction

The advances in space robotics are considered one of the milestones for future successful space exploration and exploitation [66].
Earth orbit exploitation is a primary element for the success of many terrestrial applications such as navigation, meteorology, risk monitoring and global access to wireless internet: these services are possible thank to successful launches and operations of artificial spacecraft into different Earth orbits.
However, space is not just for immediate practical purposes, but it represents also the new frontier to expand the human knowledge: the *Hubble Space Telescope (HST)* in more than thirty years of service has revolutionized the observation of the universe revealing its origin and the new knowledge learned aboard the *Interantional Space Station (ISS)* represents the foundations for the development and implementation of future space exploration initiatives such as ARTEMIS, the program that will bring humankind back to the moon [81].
Inserting telescopes and large stations into orbit requires on-orbit assembly capabilities as these structures are too big for launchers. For instance, the construction of the ISS has been performed by astronauts supported by robotic arms on-board the station itself. Thus, robotic arms and space robots are considered promising technologies for future missions for in orbit construction and assembly.
The technological growth of terrestrial applications requires a larger number of spacecraft to answer the growing demand. However, Earth orbits are limited and their overpopulation can increase the number of potential collisions in orbit. To mitigate these risks, several solutions have been proposed to capture and de-orbit non-functioning spacecraft and debris. For instance, a net and an harpoon have been proposed for capturing a target spacecraft and drag sail for de-orbiting [83]. Also, others have suggested robotic arms for capturing debris [7, 40]. The use of robotic arms also allows to consider another possible solution for cleaning up space from non-functioning satellites, instead of capturing and de-orbiting: on-orbit servicing can expand the operative live of spacecraft and it can enable the reuse of the available hardware.
Satellites equipped with robotic arms can be used to provide solutions for a multitude

of in-orbit applications, like active debris removal, maintenance of space system, repair, assembly of large structures, inspection and refuelling [66]. Several studies [15, 45, 87] have demonstrated that these robotic operations would result in savings in term of cost effectiveness and in higher safety. Moreover, the usefulness of robotic arms is underlined by the fact that they have been used for decades on-board the ISS to help astronauts during Extra Vehicular Activities (EVA).
Despite the potential of robotic manipulator, their use on-board smaller spacecraft is limited to only a fewer number of demonstration missions such as the *ETS VII* [64, 97] and *Orbital Express* [29]. This is due to the high complexity involved in on-orbit servicing missions, where the space robot has to interact with both cooperative or uncooperative target. A cooperative target has an active control system, whereas an uncooperative target can be any object or another faulty or tumbling satellite.
In order to perform a successful on-orbit capture of both cooperative and uncooperative target, several phases of operation should work efficiently. The principal ones are the following [28]:

- The *approaching phase* where the far range rendezvous maneuver is completed. At the end of this phase the space robot has reached the target on its orbit and it remains at a safe distance from the tumbling target;
- The *capturing phase* where the space robot acquires knowledge of the target motion, the target physical properties, plans the capture maneuver and then executes it; this phase comprises four sequential operations:
  - *Observing*, where the space robot uses vision, LiDAR or radar sensors to collect data about the target and to estimate its the motion.
  - *Final Approaching*, where the space robot firstly plans the strategy for the final approach and the motion of the end-effector to reach the grasping point and then it executes the desired plan thanks to the control system.
  - *Impact and mating*, where the end-effector of the manipulator is physically in contact with the target.
  - *Post-capturing stabilisation*, where the mated space robot and target are considered as one system and the space robot's controller has to perform stabilisation.
- The *de-orbit/orbit correction phase* where the control system of the spacecraft base takes over the control of the mated system to remove the target from its original orbit. This phase is the final one of a debris removal or orbit correction mission.

In this thesis is addressed the problem of planning the final approaching phase and the trajectory of the end-effector for capturing an uncooperative tumbling target. In particular, this thesis work proposes a path planning algorithm whose purpose is to find a feasible solution for the success of the described mission.
Handling cooperative and uncooperative targets is a challenging task if the target is spinning or tumbling: in this case approaching and capturing tasks become highly difficult because the chaser has to account for the rotating grasping point and for the motion of the target appendages that can hit and damage it during the approaching phase. Due to these complications, a proper strategy, compatible with all mission frameworks and constraints, has to be formulated for the success of the final approaching and for the capturing phases.

1.1. Literature Review

This section first provides an overview of previous and future planned orbital missions using a space robot to highlight the current interest of the thesis work. Then a thorough literature review on path planning for orbital space robots is presented to introduce useful concepts to understand the proposed work.

1.1.1. Introduction to Orbital Robotics

The manipulation capability offered by autonomous and tele-operated *space robots* is crucial for undertaking those orbital missions such as in-orbit assembly, maintenance, repair, refuelling and de-orbiting of space debris that can be inaccessible or too dangerous for humans [38, 92]. A space robot, called also *space manipulator system*, consists of a spacecraft base equipped with one or more *robotic manipulators* with grappling devices on them, called *end-effectors*, which allow them to capture a *target* body, that can be a spacecraft or another orbiting object [66]. Two major types of space robots can be distinguished: large manipulators mounted on the ISS, whose behaviour can be considered similar to a terrestrial fixed-base manipulator, and manipulators mounted on smaller spacecraft base, that are characterized by a more complex dynamics due to the coupling between the robotic arm and the spacecraft base.

ISS Manipulators

The first robotic manipulator successfully used in-orbit was the Canadian manipulator, launched on the Space Shuttle Columbia during mission STS-2 in 1981 and called *Shuttle Remote Manipulator System (SRMS)*; this system is also known as *Canadarm-1* [98]. It

performed several in-orbit operations before it was replaced by the *Space Station Remote Manipulator System (SSRMS)*, or *Canadarm-2*. The SSRMS is a bigger and better version of Canadarm-1 , with 17.6 m long, and 7 DoF, which offers more dexterity to support the astronauts on ISS during their Extra Vehicular Activities (EVA) [75].
Canadarm-1 and Canadarm-2 have been also used in EVA to support astronauts during their missions such as the series of servicing missions of the Hubble space telescope from 1993 to 2009 [91].
Others space manipulators and test missions were developed by different space agencies:

- the Japan Aerospace Exploration Agency (JAXA) built the *Japanese Experiment Module Remote Manipulator System (JEMRMS)* dedicated to support experiments on the Japanese Experiment Module (JEM) on the ISS. [77].
- the European Space Agency (ESA) developed a 7 DoF arm for the Russian segment of the ISS, called the *European Robotic Arm (ERA)* [11, 12].
- the German Aerospace Centre also known as the Deutsches zentrum für Luft- und Raumfahrt (DLR) successfully achieved a remotely controlled mission with their *Robot Technology Experiment (ROTEX)* gripper. It was a semi-autonomous 6 DoF gripper controlled both by astronauts on the Space Shuttle and tele-operators from the ground station. The main focus of this mission was to demonstrate the use of the sensor technologies and control algorithms under zero gravity, developed at that time [32]

Space Robots for On Orbit Servicing (OOS)

On-orbit servicing for spacecrafts comes from the necessity of expanding their operative life: despite spacecrafts go through a number of tests before launch, their reliability is not completely guaranteed due to different kinds of failures that can occur in orbit and that can end their operations and their use earlier than expected [25].
The concept of the OOS of spacecraft was first proposed in the 1960s and was applied in many missions within the last century: in 1993, during the Skylab mission, astronauts first showed that OOS was practical in a series of on-orbit operations, including the solar array and the microwave antenna reparation and the arrangement of some critical units to upgrade the performance of Skylab [46]. The development of the Space Shuttle program, which was executed first with Columbia in 1981, brought huge progresses to the technology of the OOS of spacecraft. Before the Hubble Space Telescope (HST) Servicing Mission, between years 1993 and 2009, astronauts had successfully executed several important operations using the Space Shuttle Challenger to the Solar Maximum

Satellite (SMM) launched in 1980. In 1997, ETS-VII was the first satellite equipped with a space robot, making it the first autonomous on-orbit servicing spacecraft and promoting the fast development of technologies for on-orbit assembly and maintenance.

- **Maintenance and repair missions**

  In 1997, JAXA carried the first successful autonomous OOS mission using a space robot. They developed the *Experimental Test Satellite VII (ETS-VII)* to test technologies for autonomous rendezvous and docking. The system consisted of a 6 DoF manipulator mounted on a chaser spacecraft base to dock with a target spacecraft and conduct several tests using a robotic manipulator. The coupled chaser/target spacecraft first latched together and then various tele-operated and robotic tasks were conducted. As the two spacecraft were docked without using the arm, the post-docking manipulator motion was similar to the motion of a terrestrial fixed base manipulator [97].
  In 2007, an advanced OOS mission called *Orbital Express* was led by Defence Advances Research Projects Agency (DARPA): they demonstrated how one spacecraft would service another spacecraft, involving autonomous rendezvous and docking to perform refuelling and replacement of components. This mission was designed to demonstrate that orbital servicing of spacecraft is feasible using Robotics and Autonomous Systems (RAS) technology [29]. DARPA also developed, along with the US Naval Research Laboratory (NRL), the *Front-end Enabling Near-term Demonstration (FREND)* which is a 7 DOF robotic arm, designed with the aim of capturing spacecraft that were not equipped with a grappling fixture during their design [16]. This arm is now used in a program called *PHOENIX* whose aim is demonstrating a new method for spacecraft assembly in Geosynchronous Earth Orbits (GEO) [54].
  The technologies developed in this program were tested only in laboratories that are used to prove the OOS concept on Earth. For instance, the *OOS-SIM* is a robotic experimental facility built at the DLR that replicates a close proximity scenario under realistic orbital dynamics conditions. Unlike others rendezvous simulation facilities, the OOS-SIM has recently been adapted to simulate a chaser spacecraft approaching and capturing a target using a robotic arm [2].

- **Refuelling missions**

  Refuelling has recently gained attention, since extending the mission life-cycle of satellites would help decrease creation of space debris and create a more sustainable future for space expolitation. In May 2013, NASA used Canadarm 2 of the ISS and the Canadian Dextre Robot to accomplish Robotic Refuelling Mission (RRM).

The RRM tools have been designed to demonstrate how tiny caps can be retrieved and stowed in space. The mission paved the way towards future uses showing how remotely-operated robot could extend the lives of the hundreds of satellites residing in GEO orbit [62].
Following the success of RRM mission, NASA planned *On-orbit Servicing And Manufacturing 1 (OSAM-1)* mission, formerly called Restore-L. OSAM-1 is an upcoming mission to demonstrate and perform autonomous rendezvous, refueling and relocation of a target satellite in a polar LEO orbit not designed to be serviced [61]. Another similar mission, known as *Robotic Servicing of Geosynchronous Satellites (RSGS)* and led by DARPA, will demonstrate the use of a space robot for refueling and servicing a client satellite in GEO [76].

- **Tele-Operations**

  Tele-Operation covers the entire range of task execution at a remote location and it can combine robotic manipulation with the human intelligence to react proptly to unforseen situation that may occur during a mission. The effects of communication time delays due to round-trip and processing times [67] and signal dropouts as well as of the misperception of the human operator that has to rely only on real time images with a limited field of view, make the execution of tasks on dynamic environments with telerobotic methods, such as the capture of a tumbling target, still challenging today.
  Some demonstrations have been performed to test technologies needed for success of tele-operated missions: in the contest of *Multi-purpose End-To-End Robotic Operation Network (METERON)*, led by ESA and with partners NASA, Roscosmos, and DLR, several experiments to validate advanced technologies for space robotics operation and tele-manipulation have been conducted. These experiments aimed to demonstrate supervised autonomous tele-operations and how robots, despite significant communication times, can be commanded to solve complex tasks [47, 48]. NASA designed *Robonaut I* and *Robonaut II* for a wide variety of intra-vehicular activities on the ISS [1, 18]. The robot can be tele-operated by coupling its stereo vision and dexterous capability with the user through augmented reality and finger tracking [68].

Space Robots for in-orbit assembly

Large structures such as the ISS or large aperture space telescopes cannot be launched all at once in one launch vehicle, but they have to be assembled in-orbit. As the ISS took over a decade to be built by astronauts, many space agencies are focused on finding better

cost-effective and time-effective solutions for the post-ISS era [31, 33].
The PHOENIX concept developed for servicing in GEO has been upgraded to perform construction of geostationary satellites in orbit. The concept is based on the aggregation in orbit of several modular satellites called *Hyper-Integrated Satlets (HISats)* [54].
Tethered Unlimited introduced the *SpiderFab* system consisting of a multi-arm space robot equipped with several containers for the raw material needed to build different structures in-orbit [33].
Recently, the US Naval Academy (USNA) started its program to advance autonomous robotic assembly, known as the *Intelligent Space Assembly Robot (ISAR)* [89]. This program aims to demonstrate a semi-autonomous assembly using small space robots, called *RSat* consisting of a 3U cubesat base equipped with two 7 DoF robotic arms and vision sensors for navigation [90].
Space assembly is also the main focus of NASA's space technology mission directorate as they develop for the Commercial Infrastructure for Robotic Assembly and Services (CIRAS) program the *Tension Actuated LIghtweight in Space MANipulator (TALISMAN)*, a compact and very dexterous arm consisting of a lightweight foldable truss and a novel concept of hinges that enables 360°rotation [20].
In addition to the assembly of orbital structures and spacecraft, a space robot can be used for self-assembly. In this framework, the Dragon Fly project led by NASA is focused on self-assembly of satellites in-orbits [104].

Space Robots for Debris Removal

The Inter-Agency space Debris Coordination Committee (IADC) defines space debris all non functional objects, that can be both fragments or entire devices, made by humans that are orbiting around Earth or are reentering the atmosphere [6]. It is estimated, as of May 2022, that there are about 31500 of tracked debris objects orbiting the Earth [26]. These debris include a large variety of orbiting objects of different sizes that vary from a few millimeters such as small fragments originated from accidental explosions or collisions, to several meters such as rockets upper stages or large non-operational spacecraft like ENVISAT [38] and they represent a threat and a problem for all the functional satellites: collisions with even small fragments at orbital velocities can be highly damaging and can also produce even more space debris. If the probability of collision exceeds a prescribed threshold a Collision-Avoidance Maneuver (CAM) is performed reducing the propellant on board and decreasing the satellite operational lifetime. To mitigate the generation of additional debris and the associated risks various measures have been proposed. Some of them consist in capturing and de-orbiting large-sized objects in LEO [8, 83].

A few mission concepts and planning have been performed to validate the use of a space robot for debris removal [7, 19, 40]:

- The ESA mission called *e.Deorbit* is designed with the aim of capturing the largest debris in orbit around Earth. To de-orbit ENVISAT, ESA starts developing many technologies and considering different solutions [38] as a 7 DoF robotic manipulator mounted on a spacecraft [7]. The goal of the manipulator is to capture ENVISAT with a grapple fixture and a secure clamping mechanism [38].
- the *DEutsche Orbitale Servicing Mission (DEOS)*, started after the failure of *TECSAS* in 2006 , was a mission led by DLR with the purpose of demonstrating technologies to capture a tumbling uncooperative spacecraft using a robotic manipulator. DEOS only reached the design phase in which was planned to insert two spacecraft, a chaser and a target, into a LEO [19, 71] before being cancelled.
- the *Agora* mission wants to test the use of a robotic manipulator for capturing the upper stages of the Ariane rocket bodies [40]. which are uncontrolled tumbling objects. This mission proposes a spacecraft designed for three main features; (i) a contactless de-tumbling system based on eddy current, (ii) a de-orbiting kit and (iii) a robotic manipulator to place the de-orbiting kit on the target.

1.1.2. Introduction to Path Planning Problem

One of the fundamental needs in robotics is to have algorithms that convert high-level specifications into low-level descriptions of how to move; these low-level instructions can then be effectively executed by the robot to accomplish the task [43]. The terms *path planning*, *motion planning* and *trajectory planning* are often referred to the problem of defining these instructions for moving a robot.
The path planning problem for a space manipulator system consists in defining the trajectory of the end-effector to capture successfully the target body. The trajectory can be both designed in two different spaces [10]:

- the *Cartesian space* by directly defining instant by instant the position and the orientation of the reference frame attached to the end-effector;
- the *configuration space* by designing the trajectories of the robotic manipulator joints angles

However, for space robots, unlike terrestrial fixed-base manipulators, the position and orientation of the end effector do not depend only on the joint angles but also on the position and orientation of the spacecraft base on which the manipulator is mounted.

The dimension of the configuration space of a space robot is therefore equal to $6 + N$ where N is the number of joints that make up the robotic arm.
Moreover, the dynamic coupling between the base and the robotic manipulator impacts the position and the orientation of the end-effector and, in the planning of the trajectory, it can be neglected or not depending on the operating mode in which the space robot is assumed to work in. Several techniques, for reducing the dynamic coupling effect and avoiding singularities and collisions, have been developed in the last couple of decades. These techniques either consider paths free of obstacles or paths that can avoid collisions.

Modes of Operations

There are two main operative modes for a space manipulator system to approach and capture a target: *free- fying* and *free-floating* [22, 28]. Both modes presents an undesired dynamic coupling effect on the chaser base originating from the motion of the manipulator, but they are characterized by different behaviours:

- in the *free-flying mode* the spacecraft base is actively controlled, to maintain a fixed pose or reach one desired pose whilst the arm is in motion. In free-flying mode the effects of the dynamic coupling can be counteracted by the control system of the chaser spacecraft and for this reason they can be neglected in the planning of the trajectory of the end-effector.
- in the *free-floating mode* the spacecraft base is uncontrolled and is free to change its pose in reaction to the motion of the arm due to momentum conservation. Due to the non-integrability of the angular conservation equation, free-floating space manipulator system exhibit nonholonomic behaviour that has to be taken into account during the planning of the motion.

Both modes of operation have benefits depending on the nature of the mission. For instance, when communication between the space robot and the ground station is paramount, the free-flying mode is preferred as its controlled base can keep the antennas pointing towards Earth. Whereas when reducing fuel consumption is prioritised, the free-floating mode is more suitable due to its uncontrolled base.

Obstacle-free path planning techniques

There exists various methods for planning a path for space robots without addressing the problem of obstacles that can be present in the workspace of the space robot. Some of the most relevant methods are listed here in this section.

- **Polynomial and Sinusoidal Parametrization**

  Polynomial and sinusoidal functions can be used to design smooth trajectories between initial and final conditions that can be defined both in Cartesian space or in configuration space. Polynomial and sinusoidal trajectories are extensively used for terrestrial fixed-base manipulators and this solid background prompted the research to study their potential use in space robots. Sinusoidal functions were utilised in [94, 102, 103] to parametrize the arm joints trajectories and authors in [94, 102] used polynomial functions as arguments to the sinusoidal functions. Also, polynomial functions were used in [30, 65, 70] to design smooth joint trajectories for a free-foating space robot taking advantage of its nonholonomic nature.

- **Enhanced Disturbance Map (EDM)**

  During operations, the dynamic coupling effect can be reduced using a graphical tool called the Enhanced Disturbance Map (EDM) [24]. In [24] a map that outlines hot and cool areas in the configuration space has been introduced. Hot areas are defined as regions where small joint displacements result in a large dynamic coupling effect and hence, large attitude changes. Cool areas, on the other hand, are regions where joint displacements do not produce a large dynamic coupling. The same authors shows in [23] how the EDM can be used with the objective of minimizing fuel consumption for attitude control. The EDM can then be used to find paths that result in a minimum dynamic coupling, but it adds restriction to the workspace of the space robot that has to avoid the hot spot regions in the configuration space.

- **Reaction Null Space (RNS)**

  The concept of Reaction Null Space (RNS) was introduced in [63], which is based on the null space of the inertia coupling matrix. The idea is to find a vector of arm joint velocities or accelerations that nullifies the dynamic coupling effect, keeping constant coupling momentum. The RNS method restricts the motion of the space robot to regions where the joint angles result in a null dynamic coupling. This can be allowed in a workspace free of singularities and obstacles, but avoiding singular configurations and collisions may require the space robot to operate outside the RNS regions. A reaction-less approach based on RNS methods was presented in [69] to grasp a target rotating at constant angular speed with 2 DoF planar manipulator without affecting the supporting base pose. The RNS-based reactionless manipulation was successfully tested in experiments performed during the ETS-VII mission, where the chaser was equipped with a non redundant 6 DoF manipulator [99].

- **Optimal Path Planning**

  An optimal motion of a space robot is based on a minimisation of objective functions that describe the desired problem to be minimised. Even if optimization-based approaches have traditionally appeared to be too computationally intensive, they have received considerable attention. Most of the the optimal path planning techniques presented in the literature, need an initial guess path prior to the execution of the optimisation [35, 88]. This helps reduce the computational burden of the optimisation algorithm but does not necessarily guarantee that the pre-selected path is optimal. Most of the optimal path planning algorithm that don't consider obstacles tend to minimize dynamic coupling and disturbances on the chaser base, so as it was mentioned in [24] that the EDM can be used to find an initial path that results in a minimum dynamic coupling to aid the optimisation process.

Collision-free path planning techniques

Collision avoidance is a task that can be attained adding one or more degree of the redundancy to the space robot, as proposed in [60] and in [57]. Neverthless, in recent years, algorithm like Artificial Potential Field (APF) [50, 59] and sampling based algorithm such as Rapidly-exploring Random Trees (RRT) shows good performance in terms of collision avoidance and dealing with other hard constraints, typical of an on-orbit servicing mission.

- **Artificial Potential Field**

  One of the most widespread techniques for avoiding obstacles for robots is the APF approach. The main idea of artificial potential field theory is to construct a potential field and let the robot move under the action of the opposite of its gradient. In this way, high potential regions act as repulsive regions and local minima acts as attractive regions: high potential regions are defined in correspondence of obstacles and local minimum is defined in correspondence of the goal region. [80]. APF has been used extensively in terrestrial robots [39] and has since improved to accommodate the challenges faced by space robots [72]. In [50], APF was utilised to plan a collision-free path for free-foating space robots. Authors in [50] firstly used APF to drive the end-effector towards the grasping point, then they improved the APF method to take into account all the links of the space robot for collision avoidance; this depends on the threshold set for the safety distance between the links and obstacles. Also, the nonholonomic nature of free-foating space robots was exploited using a simple Lyapunov function as a potential function in [58] and using a Laplace potential function in [96]. The former approach also considers avoiding the limits

of the manipulator's joints and the latter uses extra spline functions to guarantee smoothness of motion.

Free-flying space robots can also benefit from the APF method by applying a three dimensional space APF developed in [36, 72] for a terrestrial robot. Also, the effectiveness of using an APF method on a free-flying space robot was verified in a laboratory experiment [101] . This experiment involved the trajectory planning of a planar space robot to capture a mock satellite whilst avoiding several obstacles. Other applications of APF were demonstrated in [59], but only flying platforms without a robotic arm were considered.

APF method has proved to be highly effective for mobile robots and terrestrial manipulators. Also, its utilisation in free-oating and free-flying space robots showed that this method can be used for orbital operations. The strengths of APF are the limited computational effort required and the intuitive working principle, the drawback on the other hand is related to the non-awareness of fuel consumption. To overcome this issue, adaptive APF that continually update the shape of the potential field in optimized way [9] and solutions based on the combination of a Linear Quadratic Regulator has been proposed [5].

- **Sampling Based Path Planning**

  Sampling based path planning is based on the idea that it is not needed a complete description of the configuration space, but it can be explored by a sampling scheme [43]. One of the algorithms based on this principle are the Rapid Exploring Random Trees (RRT). . This approach grows a tree from a root, located at the initial configuration, towards unexplored areas in the workspace map that are defined by a random selection of state space nodes [44]. It is known to have algebraic and differerential constraints to respectively represent obstacles and nonholonomy. Since RRT takes into account the nonholonomy of systems, authors in [4] used it to find a path for a free-floating space robot subject to joint limits and obstacles in the workspace. In the same context, RRT was used in [74] for planning the motion of a free-floating space robot, where the tree was constructed in the configuration space and obstacles, in Cartesian space, were checked at each new node.

  One of the main issues related to RRT is the difficulties to reach a precise goal configuration. To overcome this issue two solutions were proposed: a bi-drectional tree, one growing from the initial condition and the other growing from the final configuration [73], and an hybrid guiding growth: the tree is constructed in configuration space and the growth is steered towards a desired final position expoliting direct kinematics.

- **Other techniques for collision-free path planning**

  Optimisation techniques can be used to avoid obstacles as presented in [41], where three cost functions related to obstacles, singularities and spacecraft translation were optimised to result in a collision-free path.In an effort to reduce the computation time, an off-line generated look-up table is used in [42] to seed the optimizer. For a space robot, obstacles are not only objects in the workspace obstructing its motion, but they can be represented by components of the target itself, such as solar panels or antennas. Therefore, the space robot should avoid potential collisions with the target during its approach. In this context, an optimisation-based algorithm was developed in [14] to avoid singularities and collisions with the target satellite. Also, authors in [85] used a convex optimisation to design a trajectory for a space robot whilst keeping a desired distance between the chaser space robot and the target satellite. On the other hand, the method presented in [37] is based on a general numerical optimisation to avoid collision with a tumbling target satellite.

1.2. Thesis Recap and Goals

The aim of this thesis work is to propose a flexible path planning approach for a generic proximity operations conducted by a free-flying space manipulator system. In particular the thesis concentrates on planning the trajectory of the end-effector to perform a successful capture of an orbiting, uncooperative, tumbling body.
This thesis work presents an original path planning algorithm based on an optimal-guided, incremental exploration of configuration space through sampling. The strength of the algorithm is that it is able to compute the trajectory of the end-effector passing from joint angles regardless of how the trajectory of the base is obtained. The issues on the other hand are related to the need of an inverse kinematics algorithm to fully defined the joint trajectory and attain a successful capture.
The mission framework and the proposed path planning approach are based on the following assumptions:

- the orbit of the target body is circular or near circular;
- the chaser is initially in proximity of the target body, at a relative distance d $\leq 100m$;
- both the chaser and target object are composed of rigid bodies;
- disturbances such as gravity gradient, solar radiation pressure and atmospheric drag are neglected;
- the state, the geometry and the inertia properties of the chaser and of the tumbling target are known;
- the mass of the chaser remains constant since the amount of propellant used is small when compared with the chaser mass,
- the chaser operates in *free-flying* mode, which means that the position and the orientation of the chaser base are actively controlled.
- the rotational motion of the target is bounded, so that the trajectories of the grasping point and of the appendages can be addressed to fixed locations in local vertical local horizon frame

1.3. Thesis Organization

The thesis work is organized in the following Chapters:

- **Chapter 2**
  Introduces the mathematical models used in the thesis for describing the mission framework; in particular is presented the linear dynamics model governing the relative translational motion in orbit between a target and a chaser spacecraft, then are presented the equations to model the behaviour of an orbiting tumbling rigid body and the kinematics relation to track the trajectories of its edges and finally a description of the kinematics relations that rule the motion of a space free-flying manipulator system are given.

- **Chapter 3**
  The path planning problem is formalized with all the constraints considered and the proposed solution approach is explained; in particular a path planning approach based on the combination of Linear Quadratic Regulator and Artificial Potential Function is described and the original Near-Optimal Incremental Sampling Based path planning algorithm is presented in detail.

- **Chapter 4**
  Gathers all the most relevant results of the numerical simulations performed to validate the path planning approach and the algorithms described in *Chapter 3* Firstly all the input parameters of the simulations are presented, then simulation results are discussed and a critical analysis of issues and performances with respect a variation of parameters is conducted.

- **Chapter 5**
  Contains a brief summary of the thesis work and of the analysis of the results, with an emphasis on the possible future developments.

2 | System Modelling

The purpose of this Chapter is to present mathematical models that are used in this thesis work with a basic theoretical background.
The first section introduces reference frames used for the description of translational and rotational motions of target and chaser spacecrafts. The second and third ones present models used to deal with relative orbital motion and spacecraft attitude motion. The last section focuses on kinematics of the spacecraft-manipulator system.

2.1. Reference Frames

A reference frame $\mathcal{F}$ is defined by the location of its origin $\boldsymbol{o}$ and the orientation of a set of three mutually orthogonal unit-length direction vectors $\{\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}\}$.
Three types of reference frames are introduced [27]:

- *Inertial reference frames*: to describe the orientation of the orbit and the position of the spacecraft relative to the Earth;
- *Spacecraft local orbital reference frames*: to describe the motion relative to a specific point in orbit or to another spacecraft;
- *Spacecraft body and geometric frames*: to describe the rotational motion of the spacecraft with respect to its center of mass and to describe features relative to the geometry and to a particular point of the spacecraft.

All reference frames introduced are right-handed frames.

Inertial Reference Frame $\mathcal{I}_{ECI}$

Earth-Centered Inertial (ECI) frame $\mathcal{I}_{ECI} = \left\{\boldsymbol{O}_{ECI}; \hat{\boldsymbol{X}}_{ECI}, \hat{\boldsymbol{Y}}_{ECI}, \hat{\boldsymbol{Z}}_{ECI}\right\}$ is used to define the position and the orientation of all others reference frames in absolute way, relative to the Earth. Its origin $\boldsymbol{O}_{ECI}$ is located in the center of the Earth, $\hat{\boldsymbol{X}}_{ECI}$ lies on equatorial plane and points toward the direction of vernal equinox, $\hat{\boldsymbol{Z}}_{ECI}$ points toward the north pole and $\hat{\boldsymbol{Y}}_{ECI} = \hat{\boldsymbol{Z}}_{ECI} \times \hat{\boldsymbol{X}}_{ECI}$ completes the right-handed frame.

Local Orbital Reference Frame $\mathcal{L}_{LVLH}$

Local vertical Local Horizon (LVLH) frame $\mathcal{L}_{LVLH} = \{\boldsymbol{o}_{LVLH}; \hat{\boldsymbol{x}}_{LVLH}, \hat{\boldsymbol{y}}_{LVLH}, \hat{\boldsymbol{z}}_{LVLH}\}$ is commonly used to describe the relative motion between a chaser and a target spacecraft during orbital close proximity operations and rendezvous [13].
It is a moving frame since its origin $\boldsymbol{o}_{LVLH}$ is attached to the CoM of the target, as illustrated in Figure 2.1; $\hat{\boldsymbol{x}}_{LVLH}$ is directed as the target's position vector $\boldsymbol{r}_T$ from center of $\mathcal{I}_{ECI}$ and pointing outwards, $\hat{\boldsymbol{z}}_{LVLH}$ is normal to the orbital plane of the target spacecraft and it is directed as the target's orbital angular momentum vector, $\hat{\boldsymbol{y}}_{LVLH} = \hat{\boldsymbol{z}}_{LVLH} \times \hat{\boldsymbol{x}}_{LVLH}$ completes the right-handed frame, lying in in target's orbital plane and pointing in the direction of target's local horizon.

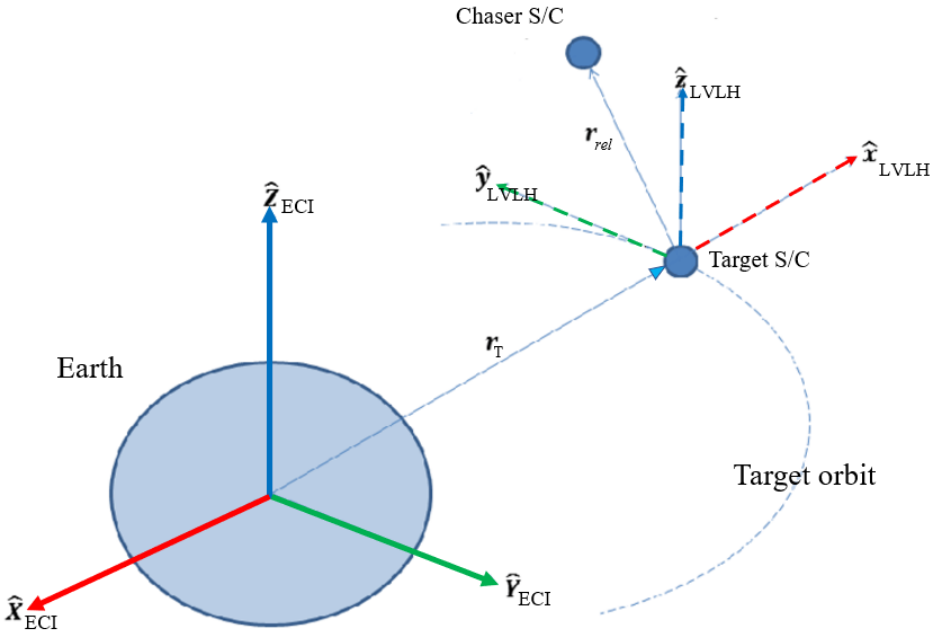


Figure 2.1: Local Vertical Local Horizon frame

Spacecraft Body Frames $\mathcal{B}_{Tb}$ and $\mathcal{B}_{Cb}$

Target's body $\mathcal{B}_{Tb} = \{\boldsymbol{o}_{Tb}; \hat{\boldsymbol{x}}_{Tb}, \hat{\boldsymbol{y}}_{Tb}, \hat{\boldsymbol{z}}_{Tb}\}$ and Chaser's body $\mathcal{B}_{Cb} = \{\boldsymbol{o}_{Cb}; \hat{\boldsymbol{x}}_{Cb}, \hat{\boldsymbol{y}}_{Cb}, \hat{\boldsymbol{z}}_{Cb}\}$ frames have their origins $\boldsymbol{o}_{Tb}$ and $\boldsymbol{o}_{Cb}$ attached respectively to the CoMs of the target and of the chaser's base. Their coordinate axes $\{\hat{\boldsymbol{x}}_{Tb}, \hat{\boldsymbol{y}}_{Tb}, \hat{\boldsymbol{z}}_{Tb}\}$ and $\{\hat{\boldsymbol{x}}_{Cb}, \hat{\boldsymbol{y}}_{Cb}, \hat{\boldsymbol{z}}_{Cb}\}$ are assumed to coincide with *principal axes of inertia* and to point towards fixed directions of the target's and chaser's bodies respectively.

2.2. Translational Motion Model

This section briefly introduces the dynamics model used in this thesis work to describe the relative translational motion of the chaser with respect to the target body
As anticipated in section 2.1, for in-orbit close proximity operations it is convenient to refer the motion of the chaser with respect to LVLH frame centered in the CoM of the target body. In the most general case, the relative motion in LVLH frame is described by a set of nonlinear differential equations [82]. However, the close-proximity operation framework justifies the two following assumptions:

1. the relative position of the chaser with respect to target $\boldsymbol{r}_{rel}$ is far smaller than the target's position $\boldsymbol{r}_T$ in $\mathcal{I}_{ECI}$, $r_{rel} \ll r_T$, where $r_{rel} = \|\boldsymbol{r}_{rel}\|$ and $r_T = \|\boldsymbol{r}_T\|$
2. the orbit of the target body is circular or near-circular.

If these conditions hold true, the relative motion of the chaser with respect to the target body can be accurately described by a simplified set of linear differential equations, known as *Hill's* or *Clohessy-Wiltshire equations*.
An extensive derivation of Clohessy-Wiltshire equations from general equations of relative motion is given in [84].

2.2.1. Clohessy-Wiltshire Equations

The relative motion in LVLH frame of a chaser spacecraft with respect to a target body that is in a circular orbit around a central body can be described by Clohessy-Wiltshire (CW) equations (Eq. (2.1)):

$$\begin{cases} \ddot{x} - 3n_T^2 x - 2n_T \dot{y} = u_x \\ \ddot{y} + 2n_T \dot{x} = u_y \\ \ddot{z} + n_T^2 z = u_z \end{cases} \tag{2.1}$$

where x, y and z are the coordinates of the CoM of the chaser base in $\mathcal{L}_{LVLH}$, n_T is the mean orbital motion of the target body and $\mathbf{u} = [u_x, u_y, u_z]^T$ denotes control accelerations in $\mathcal{L}_{LVLH}$ exerted by thrusters on the chaser to maneuver it.
Thrust's and control acceleration's components in $\mathcal{L}_{LVLH}$ are related each other by:

$$u_x = \frac{F_x}{m_c} \qquad u_y = \frac{F_y}{m_c} \qquad u_z = \frac{F_z}{m_c}$$

where m_c is the mass of the chaser spacecraft. It is assumed that three forces in the three directions of space can be generated independently to maneuver the chaser.
The set (2.1) of second-order differential equations can be converted in a set of first-order differential equations, typically referred as *state space form* (Eq. (2.2)):

$$\dot{\mathbf{x}}_{\text{cw}} = f(\mathbf{x}_{\text{cw}}, \mathbf{u}, t) = \mathbf{A}_{\text{cw}}\mathbf{x}_{\text{cw}} + \mathbf{B}_{\text{cw}}\mathbf{u} \tag{2.2}$$

where $\mathbf{x}_{\text{cw}}(t) = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ represents the state vector and state matrix $\mathbf{A}_{\text{cw}}$ and state input matrix $\mathbf{B}_{\text{cw}}$ are given by Eqs. (2.3):

$$\mathbf{A}_{\text{cw}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n_T^2 & 0 & 0 & 0 & 2n_T & 0 \\ 0 & 0 & 0 & -2n_T & 0 & 0 \\ 0 & 0 & -n_T^2 & 0 & 0 & 0 \end{bmatrix} \qquad \mathbf{B}_{\text{cw}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

2.3. Rotational Motion Model

In the framework of rendezvous and proximity operations, estimating the rotational motion of the target body is crucial for a safe and successful capture. Indeed an accurate description of the rotational motion of the target is needed for two main purposes:

1. track the trajectory and the motion of the grasping point to define the objective of the end-effector;
2. track the trajectories of the target body edges and appendages to define safe and unsafe regions around the target.

This section is divided in three parts: the first two parts give a description of the equations of the kinematics and of the dynamics that governs the rotational motion of the target body. In the third and last part is reported the procedure to characterize the motion of the grasping point once given the motion of the whole target body.

2.3.1. Target attitude Kinematics

The attitude of the target's body frame $\mathcal{B}_{Tb}$ with respect to LVLH frame is represented by using *unit quaternions*.
A unit quaternion $\boldsymbol{q}$ is a four-component vector with unit norm made up of a three-vector

part $\boldsymbol{q}_{1:3}$ and a scalar part q_4 [52]:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_{1:3} \\ q_4 \end{bmatrix} \quad \text{where} \quad \boldsymbol{q}_{1:3} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}; \qquad \|\boldsymbol{q}\| = 1$$

Attitude kinematics equation for quaternions is given by [52]:

$$\dot{\boldsymbol{q}}_{\mathrm{Tb}} = \frac{1}{2}\,\boldsymbol{\Xi}(\boldsymbol{q}_{\mathrm{Tb}})\,\boldsymbol{\omega}_{\mathrm{Tb}}^{\mathrm{Tb}} \tag{2.4}$$

where $\boldsymbol{q}_{\mathrm{Tb}}$ represents the attitude of $\mathcal{B}_{Tb}$ with respect to $\mathcal{L}_{LVLH}$, $\boldsymbol{\omega}_{\mathrm{Tb}}^{\mathrm{Tb}}$ represents the angular speed of the target's body frame with respect LVLH frame, expressed in $\mathcal{B}_{Tb}$ and matrix $\boldsymbol{\Xi}(\boldsymbol{q})$ is defined as follow [52]:

$$\boldsymbol{\Xi}(\boldsymbol{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{2.5}$$

2.3.2. Target Attitude Dynamics

The rotational dynamics of the target spacecraft is governed by Euler's equation that can be represented as [52]:

$$\mathbb{J}\dot{\boldsymbol{\omega}}_{\mathrm{Tb}}^{\mathrm{Tb}} = \mathbb{J}\boldsymbol{\omega}_{\mathrm{Tb}}^{\mathrm{Tb}} \times \boldsymbol{\omega}_{\mathrm{Tb}}^{\mathrm{Tb}} \tag{2.6}$$

where $\mathbb{J}$ denotes the inertia matrix of the target spacecraft expressed in $\mathcal{B}_{Tb}$ and $\boldsymbol{\omega}_{\mathrm{Tb}}^{\mathrm{Tb}}$ denotes the angular speed of the target's body frame with respect to LVLH frame, expressed in $\mathcal{B}_{Tb}$. The vector representing control torques is not present since it is assumed that target is non-cooperative and uncontrolled.
Euler's equation (2.6) and attitude kinematics equation (2.4) provide a complete description of the rotational motion of the target.
For more details related to attitude kinematics and dynamics, the reader is advised to refer to [52], from which most of the material present in subsections 2.3.1 and 2.3.2 has been taken.

2.3.3. Grasping Point Kinematics

To perform a successful capture, the motion of a defined grasping point on the the target body must be tracked.

If the vector $\boldsymbol{r}_{gr}^{\text{Tb}}$ denotes the position of the grasping point with respect to the target CoM expressed in the Target Body frame $\mathcal{B}_{Tb}$, its position $\boldsymbol{r}_{gr}^{\text{LVLH}}$ in LVLH frame can be computed as follow:

$$\boldsymbol{r}_{gr}^{\text{LVLH}} = \boldsymbol{R}_{\text{Tb}}^{\text{LVLH}}\, \boldsymbol{r}_{gr}^{\text{Tb}} \tag{2.7}$$

The rotation matrix $\boldsymbol{R}_{\text{Tb}}^{\text{LVLH}}$ can be obtained from quaternion $\boldsymbol{q}_{\text{Tb}}$ trough the following relation [52]:

$$\boldsymbol{R}_{\text{Tb}}^{\text{LVLH}}(\boldsymbol{q}_{\text{Tb}}) = \boldsymbol{\Xi}^{T}(\boldsymbol{q}_{\text{Tb}})\, \boldsymbol{\Psi}(\boldsymbol{q}_{\text{Tb}}) \tag{2.8}$$

where the matrix $\boldsymbol{\Xi}^{T}(\boldsymbol{q})$ has been defined in Equation (2.5), and matrix $\boldsymbol{\Psi}(\boldsymbol{q})$ is defined as [52]:

$$\boldsymbol{\Psi}(\boldsymbol{q}) = \begin{bmatrix} q_4 & q_3 & -q_2 \\ -q_3 & q_4 & q_1 \\ q_2 & -q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{2.9}$$

The Equation (2.7) is valid for every point of the target body and it can be used also to compute the position of the edges of the appendages in LVLH frame and track their envelope, as shown in Figure 2.2

The linear velocity of the grasping point $\dot{\boldsymbol{r}}_{\text{gr}}^{\text{LVLH}}$ with respect to LVLH frame is:

$$\dot{\boldsymbol{r}}_{gr}^{\text{LVLH}} = \boldsymbol{\omega}_{\text{Tb}}^{\text{LVLH}} \times \boldsymbol{r}_{gr}^{\text{LVLH}} \tag{2.10}$$

where the vector $\boldsymbol{\omega}_{\text{Tb}}^{\text{LVLH}} = \boldsymbol{R}_{\text{Tb}}^{\text{LVLH}}\, \boldsymbol{\omega}_{\text{Tb}}^{\text{Tb}}$ represents the angular velocity of the Target Body frame $\mathcal{B}_{Tb}$ with respect to LVLH frame, expressed in LVLH frame.

The capture is possible only if the end-effector approach the grasping point with a precise orientation. For this reason is useful to define a reference frame with its origin attached to the grasping point, denoted by $\mathcal{B}_{gr} = \{\boldsymbol{o}_{gr}; \hat{\boldsymbol{x}}_{gr}, \hat{\boldsymbol{y}}_{gr}, \hat{\boldsymbol{z}}_{gr}\}$.

The grasping point reference frame $\mathcal{B}_{gr}$ is here defined to coincide with the end-effector reference frame in the instant of the capture.

The orientation of the grasping point reference frame with respect to LVLH frame can be represented by rotation matrix $\boldsymbol{R}_{gr}^{\text{LVLH}}$, that can be obtained as follow:

$$\boldsymbol{R}_{gr}^{\text{LVLH}} = \boldsymbol{R}_{\text{Tb}}^{\text{LVLH}}\, \boldsymbol{R}_{gr}^{\text{Tb}} \tag{2.11}$$

where the rotation matrix $\boldsymbol{R}_{\text{Tb}}^{\text{LVLH}}$ is obtained from Equation (2.8) and the rotation matrix $\boldsymbol{R}_{gr}^{\text{Tb}}$ represents the orientation of the the grasping point reference frame $\mathcal{B}_{gr}$ with respect

to Target Body frame $\mathcal{B}_{Tb}$ and it is defined as follow:

$$\boldsymbol{R}_{gr}^{\mathrm{Tb}} = [\hat{\boldsymbol{x}}_{gr}^{\mathrm{Tb}}, \hat{\boldsymbol{y}}_{gr}^{\mathrm{Tb}}, \hat{\boldsymbol{z}}_{gr}^{\mathrm{Tb}}] \tag{2.12}$$

with $\hat{\boldsymbol{x}}_{gr}^{\mathrm{Tb}}, \hat{\boldsymbol{y}}_{gr}^{\mathrm{Tb}}, \hat{\boldsymbol{z}}_{gr}^{\mathrm{Tb}}$ denoting the axes of grasping point frame expressed in $\mathcal{B}_{Tb}$.
The orientation of grasping point reference frame with respect to LVLH frame can also be represented by the unit quaternion $\boldsymbol{q}_{gr}$, that can be extracted from rotation matrix $\boldsymbol{R}_{gr}^{\mathrm{LVLH}}$ with the method proposed in [3].

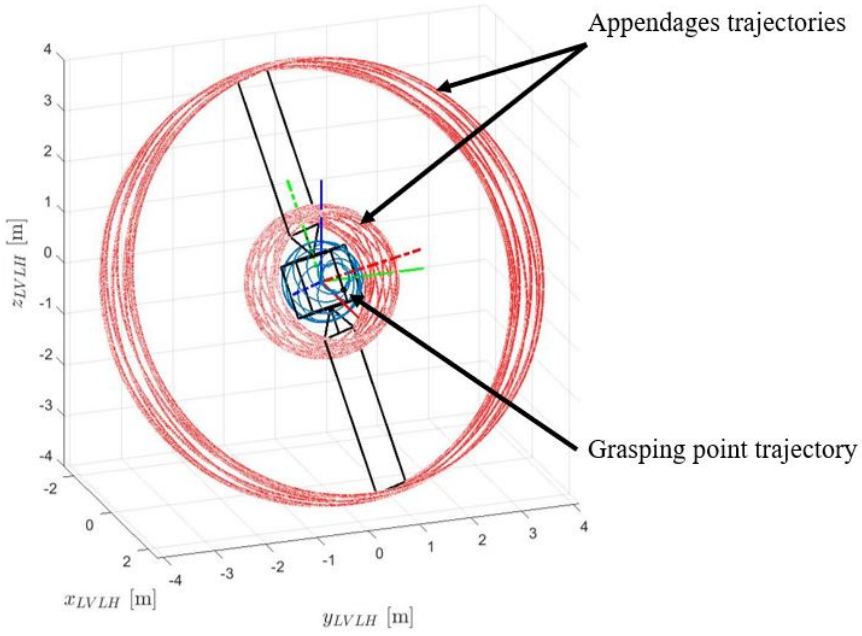


Figure 2.2: Safe and unsafe region in target proximity

2.4. Free-Flying Space Manipulator Kinematics Model

A kinematics model of a free-flying space manipulator is needed to describe the motion of the system, in particular of the *end-effector*, in LVLH frame, without taking into account forces and torques applied to generate the motion.
A free-flying space manipulator system, depicted in Figure 2.3 consists of a N degrees of freedom (DoF) robotic manipulator mounted on a six DoF spacecraft base [56] and its motion with respect to LVLH frame is a combination of the kinematics of the base and the kinematics of the robotic arm.
This section completes the analysis of the free-flying manipulator system kinematics presenting the equations to describe the motion of the end-effector in LVLH frame.

In particular, three problems are addressed in this section:

1. *Direct kinematics:* it describes the end-effector position and orientation as a function of joints variables;
2. *Differential kinematics:* it describes the relationship between the joints motion and the end-effector motion in terms of velocities, through the Jacobian matrix;
3. *Inverse kinematics:* it consists of the determination of the joint variables corresponding to a given end-effector position and orientation.

Without losing generality, it is assumed that all joints of the robotic arm are rotary joints.

2.4.1. Direct Kinematics

The aim of direct kinematics is to compute the position and the orientation of the end-effector with respect to LVLH frame, represented respectively by end-effector position vector $\boldsymbol{r}_{\mathrm{EE}}^{\mathrm{LVLH}}$ and end-effector rotation matrix $\boldsymbol{R}_{\mathrm{EE}}^{\mathrm{LVLH}}$.
As previously mentioned, for a free-flying manipulator the position and orientation of the end-effector depend on both joint variables $\theta_i\,, i = 1, 2, ..., N$ and on chaser base position and orientation with respect to LVLH frame. The direct kinematics function f_{DK} can be written as:

$$\boldsymbol{T}_{\mathrm{EE}}^{\mathrm{LVLH}} = f_{DK}(\boldsymbol{\theta}, \boldsymbol{r}_{\mathrm{Cb}}^{\mathrm{LVLH}}, \boldsymbol{R}_{\mathrm{Cb}}^{\mathrm{LVLH}}, \text{ geometric parameters}) \tag{2.13}$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_N]^T$ represents the vector of joint variables, $\boldsymbol{r}_{\mathrm{Cb}}^{\mathrm{LVLH}}$ and $\boldsymbol{R}_{\mathrm{Cb}}^{\mathrm{LVLH}}$ denote here, respectively, the position vector and the rotation matrix of the chaser base with respect to LVLH frame and $\boldsymbol{T}_{\mathrm{EE}}^{\mathrm{LVLH}} \in \mathbb{R}^{4\times4}$ represents *homogeneous transformation matrix* from end-effector frame to LVLH frame defined as:

$$\boldsymbol{T}_{\mathrm{EE}}^{\mathrm{LVLH}} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{EE}}^{\mathrm{LVLH}} & \boldsymbol{r}_{\mathrm{EE}}^{\mathrm{LVLH}} \\ \boldsymbol{0}_{1,3} & 1 \end{bmatrix} \tag{2.14}$$

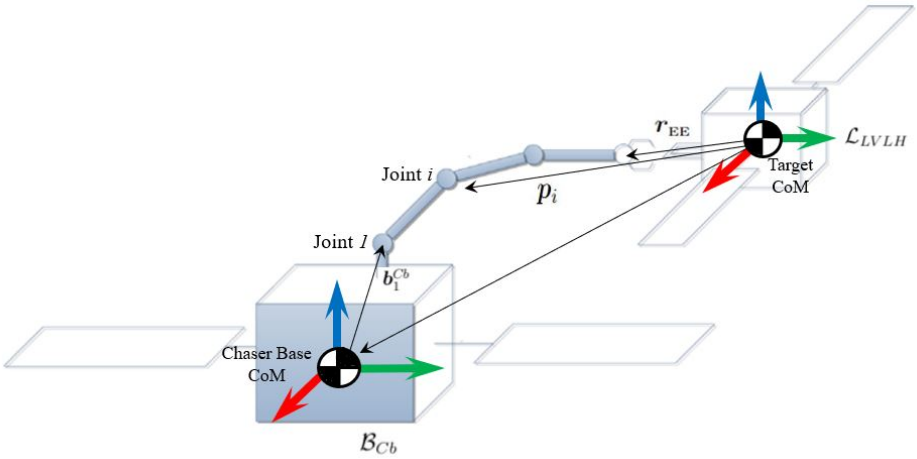


Figure 2.3: Space manipulator system

The procedure to compute $\boldsymbol{T}_{\text{EE}}^{\text{LVLH}}$ and hence $\boldsymbol{r}_{\text{EE}}^{\text{LVLH}}$ and $\boldsymbol{R}_{\text{EE}}^{\text{LVLH}}$ is explained in [93] and it is here briefly reported:

- the robotic arm is composed by N revolute joints and $N-1$ links;
- joints and links are numbered sequentially: joint 1 is attached to the chaser base, at the other end of the arm the end-effector is treated as a fictious joint $N+1$;
- for each joint i is built a joint coordinate system $\mathcal{J}_i = \{\boldsymbol{\varrho}_i; \hat{\boldsymbol{x}}_{\mathcal{J}_i}, \hat{\boldsymbol{y}}_{\mathcal{J}_i}, \hat{\boldsymbol{z}}_{\mathcal{J}_i}\}$ using *Denavit-Hartenberg (DH) convention* [80, 93] where:
  - the axis $\hat{\boldsymbol{z}}_{\mathcal{J}_i}$ is parallel to joint i rotation axis $\hat{\boldsymbol{k}}_i$;
  - the origin $\boldsymbol{\varrho}_i$ is at intersection of $\hat{\boldsymbol{z}}_{\mathcal{J}_i}$ and the common normal between $\hat{\boldsymbol{z}}_{\mathcal{J}_{i-1}}$ and $\hat{\boldsymbol{z}}_{\mathcal{J}_i}$, in correspondence of the joint i;
  - the axis $\hat{\boldsymbol{x}}_{\mathcal{J}_i}$ points from $\boldsymbol{\varrho}_i$ along the direction of the common normal between $\hat{\boldsymbol{z}}_{\mathcal{J}_{i-1}}$ and $\hat{\boldsymbol{z}}_{\mathcal{J}_i}$;
  - the axis $\hat{\boldsymbol{y}}_{\mathcal{J}_i} = \hat{\boldsymbol{z}}_{\mathcal{J}_i} \times \hat{\boldsymbol{x}}_{\mathcal{J}_i}$ completes the right-handed frame.

  As only exceptions, the origin $\boldsymbol{\varrho}_1$ of $\mathcal{J}_1$ as well as the directions of $\hat{\boldsymbol{x}}_{\mathcal{J}_1}$ and $\hat{\boldsymbol{y}}_{\mathcal{J}_1}$ are arbitrarily selected, based on the geometry of the interface between the chaser base and the manipulator arm.

Adopting DH convention, the geometrical relationship between subsequent joint coordinate systems $\mathcal{J}_i$ and $\mathcal{J}_{i+1}$ is a function of the four *DH parameters* $d_i, \theta_i, \alpha_i, c_i$, whose geometric meaning is explained in Table 2.1 and shown in Figure 2.4.

DH parameter	Geometric meaning
d_i	distance with sign between $\boldsymbol{\varrho}_i$ and $\boldsymbol{\varrho}_{i+1}$ along $\hat{\boldsymbol{z}}_{\mathcal{J}_i}$
θ_i	rotation from $\hat{\boldsymbol{x}}_{\mathcal{J}_i}$ to $\hat{\boldsymbol{x}}_{\mathcal{J}_{i+1}}$ about $\hat{\boldsymbol{z}}_{\mathcal{J}_i}$
α_i	rotation from $\hat{\boldsymbol{z}}_{\mathcal{J}_i}$ to $\hat{\boldsymbol{z}}_{\mathcal{J}_{i+1}}$ about $\hat{\boldsymbol{x}}_{\mathcal{J}_{i+1}}$
c_i	distance with sign between $\boldsymbol{\varrho}_i$ and $\boldsymbol{\varrho}_{i+1}$ along $\hat{\boldsymbol{x}}_{\mathcal{J}_{i+1}}$

Table 2.1: Denavit-Hartenberg parameters and their geometric meaning

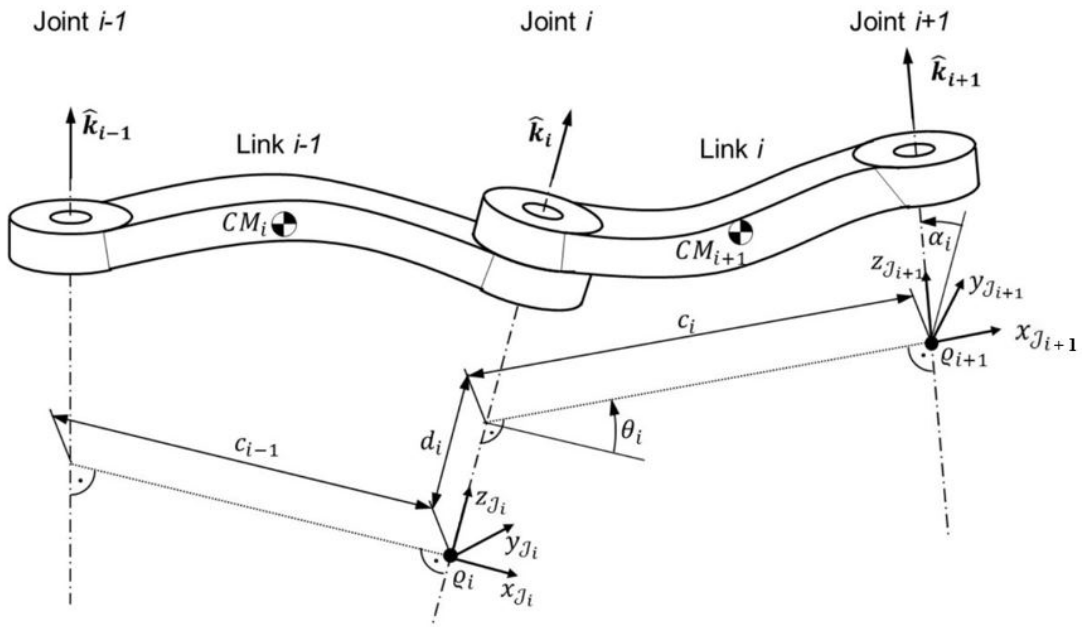


Figure 2.4: Deanvit-Hartenberg kinematic parameters [93]

For a revolute joint, θ_i is the only variable and it denotes the *joint angle*, while d_i, α_i and c_i represents *fixed geometric parameters* of the manipulator link.
The homogeneous transformation matrix $\boldsymbol{T}^{\mathcal{J}_i}_{\mathcal{J}_{i+1}}$ from joint frame $\mathcal{J}_{i+1}$ to joint frame $\mathcal{J}_i$ can be expressed as a function of DH parameters as follow:

$$\boldsymbol{T}^{\mathcal{J}_i}_{\mathcal{J}_{i+1}} = \boldsymbol{A}(\theta_i, d_i, \alpha_i, c_i) \tag{2.15}$$

where the DH transformation matrix function $\boldsymbol{A}(\theta_i, d_i, \alpha_i, c_i)$ is defined as:

$$\boldsymbol{A}(\theta_i, d_i, \alpha_i, c_i) := \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & c_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & c_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.16}$$

Therefore, the position and the orientation with respect to LVLH frame of each joint coordinate system $\mathcal{J}_i$ for $i = 2, ..., N+1$ can be computed recursively by a product of DH transformation matrices, as follow:

$$\boldsymbol{T}_{\mathcal{J}_i}^{\text{LVLH}} = \boldsymbol{T}_{\mathcal{J}_{i-1}}^{\text{LVLH}} \boldsymbol{T}_{\mathcal{J}_i}^{\mathcal{J}_{i-1}} \quad \text{where} \quad \boldsymbol{T}_{\mathcal{J}_i}^{\mathcal{J}_{i-1}} = \boldsymbol{A}(\theta_{i-1}, d_{i-1}, \alpha_{i-1}, c_{i-1}) \tag{2.17}$$

The resulting homogeneous transformation matrix $\boldsymbol{T}_{\mathcal{J}_i}^{\text{LVLH}}$ contains the rotation matrix $\boldsymbol{R}_{\mathcal{J}_i}^{\text{LVLH}}$ and the position vector $\boldsymbol{p}_i^{\text{LVLH}}$ of joint coordinate frame $\mathcal{J}_i$ with respect to LVLH frame:

$$\boldsymbol{T}_{\mathcal{J}_i}^{\text{LVLH}} = \begin{bmatrix} \boldsymbol{R}_{\mathcal{J}_i}^{\text{LVLH}} & \boldsymbol{p}_i^{\text{LVLH}} \\ \boldsymbol{0}_{1,3} & 1 \end{bmatrix} \tag{2.18}$$

The rotation matrix $\boldsymbol{R}_{\mathcal{J}_i}^{\text{LVLH}}$ is useful to derive the directions $\hat{\boldsymbol{k}}_i$ of the joint rotation axes in LVLH frame, since in DH convention the rotation axis of a revolute joint is defined as the z-axis of the corresponding joint frame:

$$\boldsymbol{R}_{\mathcal{J}_i}^{\text{LVLH}} = [\hat{\boldsymbol{x}}_{\mathcal{J}_i}^{\text{LVLH}}, \hat{\boldsymbol{y}}_{\mathcal{J}_i}^{\text{LVLH}}, \hat{\boldsymbol{z}}_{\mathcal{J}_i}^{\text{LVLH}}] \quad \longrightarrow \quad \hat{\boldsymbol{k}}_i^{\text{LVLH}} = \boldsymbol{R}_{\mathcal{J}_i}^{\text{LVLH}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.19}$$

The homogeneous transformation matrix $\boldsymbol{T}_{\mathcal{J}_1}^{\text{LVLH}}$ from joint coordinate frame $\mathcal{J}_1$ to LVLH frame is needed for the recursive application of Equation (2.17). The joint 1 is attached to the base of the chaser spacecraft and its coordinate frame $\mathcal{J}_1$ has a constant position $\boldsymbol{b}_1^{Cb}$ and orientation $\boldsymbol{R}_{\mathcal{J}_i}^{\text{Cb}}$ with respect to the chaser's base body frame $\mathcal{B}_{Cb}$. Its homogeneous transformation matrix $\boldsymbol{T}_{\mathcal{J}_1}^{\text{LVLH}}$ can be computed as follow:

$$\boldsymbol{T}_{\mathcal{J}_1}^{\text{LVLH}} = \boldsymbol{T}_{\text{Cb}}^{\text{LVLH}} \boldsymbol{T}_{\mathcal{J}_1}^{\text{Cb}} \tag{2.20}$$

where $\boldsymbol{T}_{\text{Cb}}^{\text{LVLH}}$ is the homogeneous transformation matrix from chaser base body frame $\mathcal{B}_{Cb}$ to LVLH frame and $\boldsymbol{T}_{\mathcal{J}_1}^{\text{Cb}}$ is the homogeneous transformation matrix from joint coordinate

frame $\mathcal{J}_1$ to $\mathcal{B}_{Cb}$, respectively defined as:

$$\boldsymbol{T}_{\text{Cb}}^{\text{LVLH}} = \begin{bmatrix} \boldsymbol{R}_{\text{Cb}}^{\text{LVLH}} & \boldsymbol{r}_{\text{Cb}}^{\text{LVLH}} \\ \boldsymbol{0}_{1,3} & 1 \end{bmatrix} \qquad \boldsymbol{T}_{\mathcal{J}_1}^{\text{Cb}} = \begin{bmatrix} \boldsymbol{R}_{\mathcal{J}_1}^{\text{Cb}} & \boldsymbol{b}_1^{Cb} \\ \boldsymbol{0}_{1,3} & 1 \end{bmatrix} \tag{2.21}$$

The direct kinematics function f_{DK} in Equation (2.13) can be finally re-written as:

$$\boldsymbol{T}_{\text{EE}}^{\text{LVLH}} = \boldsymbol{T}_{\text{Cb}}^{\text{LVLH}} \boldsymbol{T}_{\mathcal{J}_1}^{\text{Cb}} \left(\prod_{i=2}^{N+1} \boldsymbol{T}_{\mathcal{J}_i}^{\mathcal{J}_{i-1}} \right) \boldsymbol{T}_{\text{EE}}^{\mathcal{J}_{N+1}} \tag{2.22}$$

where homogeneous transform matrix $\boldsymbol{T}_{\text{EE}}^{\mathcal{J}_{N+1}}$ is arbitrary and it is used to rotate according to the actual geometry of the end-effector the fictitious joint frame $\mathcal{J}_{N+1}$ defined by DH convention. The homogeneous transform matrix $\boldsymbol{T}_{\text{EE}}^{\mathcal{J}_{N+1}}$ is defined as:

$$\boldsymbol{T}_{\text{EE}}^{\mathcal{J}_{N+1}} = \begin{bmatrix} \boldsymbol{R}_{\text{EE}}^{\mathcal{J}_{N+1}} & \boldsymbol{b}_{\text{EE}}^{\mathcal{J}_{N+1}} \\ \boldsymbol{0}_{1,3} & 1 \end{bmatrix} \tag{2.23}$$

where $\boldsymbol{b}_{\text{EE}}^{\mathcal{J}_{N+1}}$ and $\boldsymbol{R}_{\text{EE}}^{\mathcal{J}_{N+1}}$ represent respectively the position vector and the rotation matrix of the end-effector frame with respect the fictitious joint frame $\mathcal{J}_{N+1}$.

2.4.2. Differential Kinematics

The goal of differential kinematics is to describe the kinematics of the system at the velocity level, expressing the end-effector linear velocity $\dot{\boldsymbol{r}}_{\text{EE}}^{\text{LVLH}}$ and angular velocity $\boldsymbol{\omega}_{\text{EE}}^{\text{LVLH}}$ as a function of the joint velocities $\dot{\boldsymbol{\theta}} = [\dot{\theta}_1, \dot{\theta}_2, ..., \dot{\theta}_N]^T$ and of the linear and angular velocities of the chaser base, represented respectively by $\dot{\boldsymbol{r}}_{\text{Cb}}^{\text{LVLH}}$ and $\boldsymbol{\omega}_{\text{Cb}}^{\text{LVLH}}$.
The relationship between the velocities of the end-effector and the velocities of the joint and of the chaser base is linear and it can be expressed as [55, 79, 93]:

$$\begin{bmatrix} \dot{\boldsymbol{r}}_{\text{EE}} \\ \boldsymbol{\omega}_{\text{EE}} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \dot{\boldsymbol{X}}_{\text{Cb}} \\ \dot{\boldsymbol{\theta}} \end{bmatrix} \qquad \text{with} \quad \boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{\text{Cb}} & \boldsymbol{J}_{\text{m}} \end{bmatrix} \tag{2.24}$$

where $\dot{\boldsymbol{r}}_{\text{EE}}$ $\boldsymbol{\omega}_{\text{EE}}$ are respectively the linear and the angular velocities expressed in LVLH frame, $\boldsymbol{J} \in \mathbb{R}^{6\times(6+N)}$ represents the Jacobian matrix of the overall free-flying space manipulator system, $\boldsymbol{J}_{\text{Cb}} \in \mathbb{R}^{6\times6}$ is the Jacobian matrix of the chaser spacecraft base, $\boldsymbol{J}_{\text{m}} \in \mathbb{R}^{6\times N}$ is the robotic manipulator Jacobian, $\dot{\boldsymbol{X}}_{\text{Cb}} = [\dot{\boldsymbol{r}}_{\text{Cb}}^{\text{LVLH}}; \boldsymbol{\omega}_{\text{Cb}}^{\text{LVLH}}]$ and $\dot{\boldsymbol{\theta}}$ represent vectors containing the velocities of the chaser spacecraft base and the velocities of the joints respectively.
In order to compute the *Jacobian of the overall free-flying manipulator system* $\boldsymbol{J}$, it can

be convenient to proceed separately for the contribution of the chaser base motion and the contribution of joints motion:

- the *Jacobian of the chaser spacecraft base* $\boldsymbol{J}_{\text{Cb}}$ relates the contribution of the chaser base linear and angular velocities to the end-effector velocities and it can be expressed as follow [93]:

$$\boldsymbol{J}_{\text{Cb}} = \begin{bmatrix} \boldsymbol{I}_{3,3} & -\left[\boldsymbol{r}_{\text{EE, Cb}}^{\text{LVLH}}\right]^{\times} \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_{3,3} \end{bmatrix} \tag{2.25}$$

  where

$$\boldsymbol{r}_{\text{EE, Cb}}^{\text{LVLH}} = \boldsymbol{r}_{\text{EE}}^{\text{LVLH}} - \boldsymbol{r}_{\text{Cb}}^{\text{LVLH}} \tag{2.26}$$

  represents the relative position of the end-effector with respect to the CoM of the chaser spacecraft base and $[\,\cdot\,]^{\times}$ denotes the skew symmetric matrix operator.

- the *Jacobian of the robotic manipulator* $\boldsymbol{J}_{\text{m}}$ relates the contribution of the joint velocities to the end-effector velocities and it can be assembled as follow [93]:

$$\boldsymbol{J}_{\text{m}} = \begin{bmatrix} \hat{\boldsymbol{k}}_1 \times (\boldsymbol{r}_{\text{EE}} - \boldsymbol{p}_1) & \dots & \hat{\boldsymbol{k}}_{\text{i}} \times (\boldsymbol{r}_{\text{EE}} - \boldsymbol{p}_{\text{i}}) & \dots & \hat{\boldsymbol{k}}_{\text{N}} \times (\boldsymbol{r}_{\text{EE}} - \boldsymbol{p}_{\text{N}}) \\ \hat{\boldsymbol{k}}_1 & \dots & \hat{\boldsymbol{k}}_{\text{i}} & \dots & \hat{\boldsymbol{k}}_{\text{N}} \end{bmatrix} \tag{2.27}$$

  where $\hat{\boldsymbol{k}}_{\text{i}}$ is the direction of the i^{th} joint rotation axes in LVLH frame and $(\boldsymbol{r}_{\text{EE}} - \boldsymbol{p}_{\text{i}})$ represents the relative position of the end-effector with respect to the i^{th} joint.
  For each joint, according to Equations (2.18) and (2.19), the vectors $\hat{\boldsymbol{k}}_{\text{i}}$ and $\boldsymbol{p}_{\text{i}}$ are available once the homogeneous transform matrix $\boldsymbol{T}_{\mathcal{J}_i}^{\text{LVLH}}$ has been computed.

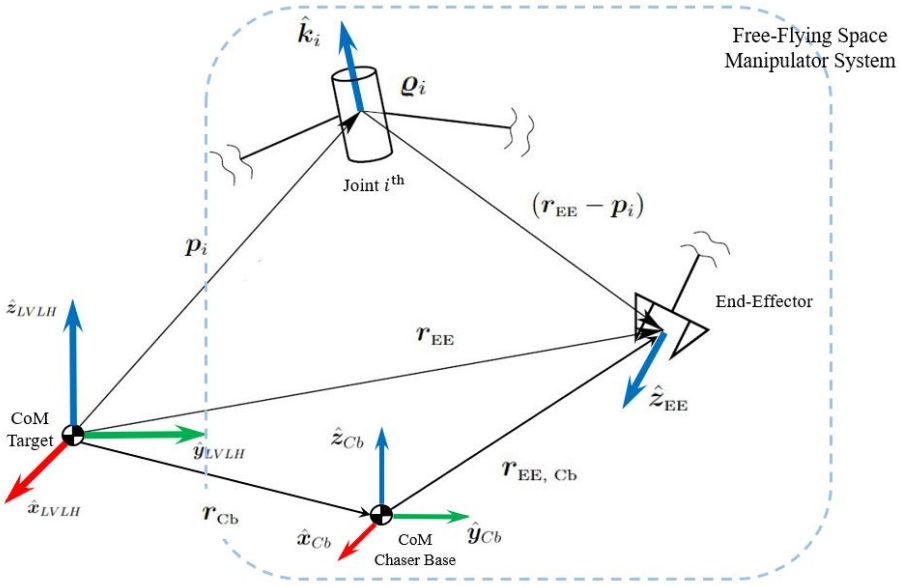


Figure 2.5: Representation of vectors needed for computing Jacobian matrices [80]

From Equations (2.25) and (2.27) it can be observed that the Jacobian matrices of a free-flying depend on the position and orientation of the chaser spacecraft base, respectively $\boldsymbol{r}_{\text{Cb}}$ and $\boldsymbol{R}_{\text{Cb}}$, and joint angles configuration $\boldsymbol{\theta}$:

$$\boldsymbol{J} = \boldsymbol{J}(\boldsymbol{\theta}, \boldsymbol{r}_{\text{Cb}}, \boldsymbol{R}_{\text{Cb}})\,; \quad \boldsymbol{J}_{\text{Cb}} = \boldsymbol{J}_{\text{Cb}}(\boldsymbol{\theta}, \boldsymbol{r}_{\text{Cb}}, \boldsymbol{R}_{\text{Cb}})\,; \quad \boldsymbol{J}_{\text{m}} = \boldsymbol{J}_{\text{m}}(\boldsymbol{\theta}, \boldsymbol{r}_{\text{Cb}}, \boldsymbol{R}_{\text{Cb}}) \tag{2.28}$$

Given the vector of the chaser base velocities $\dot{\boldsymbol{X}}_{\text{Cb}}$, the joint rates vector $\dot{\boldsymbol{\theta}}$ required to have the end-effector move at a desired linear velocity ${}^{d}\dot{\boldsymbol{r}}_{\text{EE}}$ and a desired angular velocity ${}^{d}\boldsymbol{\omega}_{\text{EE}}$ can be computed from Equation (2.24) as follow:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_{\text{m}}^{+} \left(\begin{bmatrix} {}^{d}\dot{\boldsymbol{r}}_{\text{EE}} \\ {}^{d}\boldsymbol{\omega}_{\text{EE}} \end{bmatrix} - \boldsymbol{J}_{\text{Cb}}\, \dot{\boldsymbol{X}}_{\text{Cb}} \right) \tag{2.29}$$

where $\boldsymbol{J}_{\text{m}}^{+}$ is the *pseudo-inverse* of the robotic manipulator Jacobian matrix $\boldsymbol{J}_{\text{m}}$.

2.4.3. Inverse Kinematics

The inverse kinematics problem consists in finding a joint angles configuration $\boldsymbol{\theta}$ such that the end-effector attains the *desired* position ${}^{d}\boldsymbol{r}_{\text{EE}}^{\text{LVLH}}$ with the *desired* orientation ${}^{d}\boldsymbol{R}_{\text{EE}}^{\text{LVLH}}$.
For a free-flying space manipulator system with known position and orientation of the spacecraft base, $\boldsymbol{r}_{\text{Cb}}^{\text{LVLH}}$ and $\boldsymbol{R}_{\text{Cb}}^{\text{LVLH}}$ respectively, the inverse kinematics problem, denoted here by f_{IK}, can be stated as follow: given the *desired* homogeneous transform matrix ${}^{d}\boldsymbol{T}_{\text{EE}}^{\text{LVLH}}$, find a joint angles configuration $\boldsymbol{\theta}$ that satisfies $\boldsymbol{T}_{\text{EE}}^{\text{LVLH}}(\boldsymbol{\theta}) = {}^{d}\boldsymbol{T}_{\text{EE}}^{\text{LVLH}}$:

$$\boldsymbol{\theta} = f_{IK}({}^{d}\boldsymbol{T}_{\text{EE}}^{\text{LVLH}}, \boldsymbol{r}_{\text{Cb}}^{\text{LVLH}}, \boldsymbol{R}_{\text{Cb}}^{\text{LVLH}}, \text{geometric parameters}) \tag{2.30}$$

Except for manipulator having a simple kinematic structure, generally the inverse kinematics problem shown in Equation (2.30) can not be solved analytically and then iterative numerical methods are used to find a solution [51, 80].
This behaviour is due to the highly nonlinear relationship between joint space variables $\theta_1, \theta_2, ..., \theta_N$ and task space variables $\boldsymbol{r}_{\text{EE}}^{\text{LVLH}}$ and $\boldsymbol{R}_{\text{EE}}^{\text{LVLH}}$, reported in Equation (2.22).
Let ${}^{d}\boldsymbol{X}_{\text{EE}}$ denote the desired position and the desired orientation that the end-effector has to attain and let $\boldsymbol{X}_{\text{EE}} = f_{DK}(\boldsymbol{\theta}_k)$ denote the end-effector position and orientation corresponding to a generic joint configuration $\boldsymbol{\theta}$, where f_{DK} is the direct kinematics function written in Equation (2.22).
Let $\boldsymbol{e}$ represent the error between the desired and a generic end-effector *pose* [80]:

$$\boldsymbol{e} = {}^{d}\boldsymbol{X}_{\text{EE}} - \boldsymbol{X}_{\text{EE}}(\boldsymbol{\theta}) \tag{2.31}$$

To solve the inverse kinematic problem, it must be found a joint configuration $\boldsymbol{\theta}$ such that error $\boldsymbol{e}$ is reduced within a given treshold.
A feedback control scheme can be used to reduce the error $\boldsymbol{e}$, as proposed in [80] and shown in Fig.2.6. Since the problem has to be solved in a fixed time instant, time t becomes a fictitious variable and the successive time instant assumes the meaning of successive iteration as follow:

$$\boldsymbol{\theta}(t_{k+1}) = \boldsymbol{\theta}(t_k) + \dot{\boldsymbol{\theta}}(t_k)\,(t_{k+1} - t_k) \quad \longrightarrow \quad \boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \dot{\boldsymbol{\theta}}^{(k)}\,dt \tag{2.32}$$

where interval $(t_{k+1} - t_k)$ denotes controller time step, superscripts $(\,k\,)$ and $(\,k+1\,)$ refer to successive algorithm iterations and time interval dt becomes a tuning parameter of the algorithm. A proper choice of joint rates vector $\dot{\boldsymbol{\theta}}$ at each iteration is then needed to make the error $\boldsymbol{e}$ converge to zero.

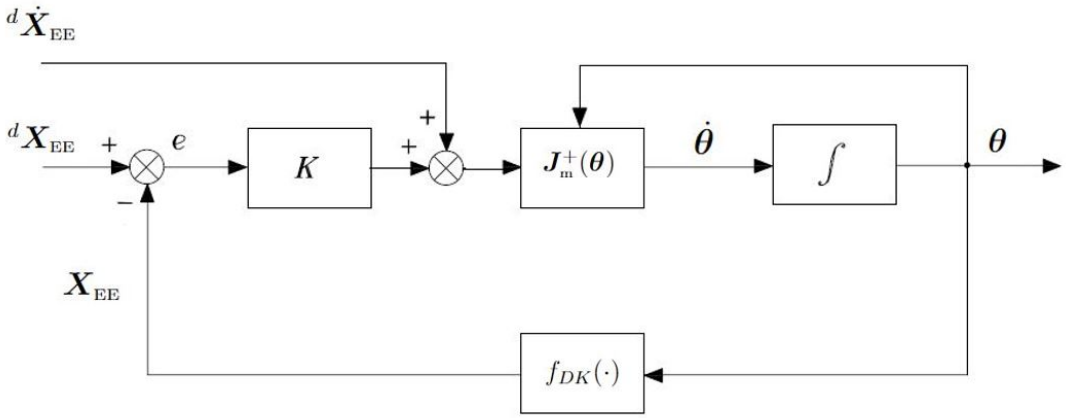


Figure 2.6: Scheme of Inverse Kinematics algorithm with Jacobian pseudo-inverse [80]

The evolution of error $\boldsymbol{e}$ over successive iterations can be described by its time derivative, as follow [80]:

$$\dot{\boldsymbol{e}} = {}^{d}\dot{\boldsymbol{X}}_{\mathrm{EE}} - \dot{\boldsymbol{X}}_{\mathrm{EE}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{2.33}$$

where ${}^{d}\dot{\boldsymbol{X}}_{\mathrm{EE}} = \left[{}^{d}\dot{\boldsymbol{r}}_{\mathrm{EE}}\,;{}^{d}\boldsymbol{\omega}_{\mathrm{EE}}\right]$ is the vector containing the desired end-effector velocities and $\dot{\boldsymbol{X}}_{\mathrm{EE}} = [\dot{\boldsymbol{r}}_{\mathrm{EE}}\,;\boldsymbol{\omega}_{\mathrm{EE}}]$ is the vector of end-effector velocities corresponding to a generic joint angle configuration $\boldsymbol{\theta}$ and generic joint angle rates $\dot{\boldsymbol{\theta}}$.
Assuming that the chaser base has negligible linear and angular speeds, differential kinematics Equation (2.24) can be written as follow:

$$\begin{bmatrix} \dot{\boldsymbol{r}}_{\mathrm{EE}} \\ \boldsymbol{\omega}_{\mathrm{EE}} \end{bmatrix} = \boldsymbol{J}_{\mathrm{m}}\ \dot{\boldsymbol{\theta}} \tag{2.34}$$

and then, substituting Equation (2.34) into Equation 2.33 it is obtained:

$$\dot{\boldsymbol{e}} = {}^{d}\dot{\boldsymbol{X}}_{\text{EE}} - \boldsymbol{J}_{\text{m}}\left(\boldsymbol{\theta}\right)\dot{\boldsymbol{\theta}} \tag{2.35}$$

The choice of joint rates vector $\dot{\boldsymbol{\theta}}$ as follow [80]:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_{\text{m}}^{+}(\boldsymbol{\theta})\,\left({}^{d}\dot{\boldsymbol{X}}_{\text{EE}} + \boldsymbol{K}\boldsymbol{e}\right) \tag{2.36}$$

leads from Equation (2.35) to the equivalent linear system:

$$\dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e} = \boldsymbol{0} \tag{2.37}$$

If the matrix $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ is *positive definite*, the system in Equation (2.37) is *asymptotically stable*. This means that the computation of $\boldsymbol{\theta}_{k+1}$ in Equation (2.32) using $\dot{\boldsymbol{\theta}}_k$ defined in Equation (2.36) drives the error $\boldsymbol{e}$ to zero over time and then over successive iterations, with a convergence rate that depends on the eigenvalues of matrix $\boldsymbol{K}$; the larger the eigenvalues, the faster the convergence [80].
The definition of error $\boldsymbol{e}$ in Equation (2.31) needs some remarks. It is convenient to define separately end-effector position error $\boldsymbol{e_P}$ and end-effector orientation error $\boldsymbol{e}_O$, such that:

$$\boldsymbol{e} = \begin{bmatrix} \boldsymbol{e}_P \\ \boldsymbol{e}_O \end{bmatrix} \tag{2.38}$$

- *position error* $\boldsymbol{e}_P$ is the difference between the desired position of the end-effector ${}^{d}\boldsymbol{r}_{\text{EE}}$ and the actual position of the end-effector $\boldsymbol{r}_{\text{EE}}$. Its expression is given by:

$$\boldsymbol{e}_P = {}^{d}\boldsymbol{r}_{\text{EE}} - \boldsymbol{r}_{\text{EE}}(\boldsymbol{\theta}) \tag{2.39}$$

- *orientation error* $\boldsymbol{e}_O$ depends on the representation of the end-effector orientation, namely Euler angles or unit quaternion. *Unit quaternion* representation is consistent with the use of angular speed vector $\boldsymbol{\omega}_{\text{EE}}$ to describe end-effector rotational motion, end orientation error can be computed as follow [80]:

$$\boldsymbol{e}_O = q_{4_{\text{EE}}}(\boldsymbol{\theta})\,{}^{d}\boldsymbol{q}_{1:3_{\text{EE}}} - {}^{d}q_{4_{\text{EE}}}\,\boldsymbol{q}_{1:3_{\text{EE}}}(\boldsymbol{\theta}) - \left[{}^{d}\boldsymbol{q}_{1:3_{\text{EE}}}\right]^{\times}\boldsymbol{q}_{1:3_{\text{EE}}}(\boldsymbol{\theta}) \tag{2.40}$$

where:

$${}^{d}\boldsymbol{q}_{\text{EE}} = \begin{bmatrix} {}^{d}\boldsymbol{q}_{1:3_{\text{EE}}} \\ {}^{d}q_{4_{\text{EE}}} \end{bmatrix} \qquad \boldsymbol{q}_{\text{EE}}(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{q}_{1:3_{\text{EE}}} \\ q_{4_{\text{EE}}} \end{bmatrix} \tag{2.41}$$

are the quaternion representation respectively of the desired end-effector orientation and of a generic end-effector orientation corresponding to joint configuration $\boldsymbol{\theta}$.
The quaternions ${}^{d}\boldsymbol{q}_{\text{EE}}$ and $\boldsymbol{q}_{\text{EE}}$ can be extracted from the corresponding rotation matrices ${}^{d}\boldsymbol{R}_{\text{EE}}$ and $\boldsymbol{R}_{\text{EE}}$ with the method proposed in [3], recalling that ${}^{d}\boldsymbol{R}_{\text{EE}}$ is known from the objective and $\boldsymbol{R}_{\text{EE}}$ is available from Equation (2.22).

The Equation (2.36) can be re-written according to the error definition in Equation (2.38) as follow [80]:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_{\text{m}}^{+} \begin{bmatrix} {}^{d}\dot{\boldsymbol{r}}_{\text{EE}} + \boldsymbol{K}_P \, \boldsymbol{e}_P \\ {}^{d}\boldsymbol{\omega}_{\text{EE}} + \boldsymbol{K}_O \, \boldsymbol{e}_O \end{bmatrix} \tag{2.42}$$

where the matrices $\boldsymbol{K}_P \in \mathbb{R}^{3\times3}$ and $\boldsymbol{K}_O \in \mathbb{R}^{3\times3}$ are such that:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_P & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & \boldsymbol{K}_O \end{bmatrix} \tag{2.43}$$

It is worth noting that the described iterative inverse kinematics algorithm needs an initial guess $\boldsymbol{\theta}^{(0)}$ to start. If the end-effector desired position ${}^{d}\boldsymbol{r}_{\text{EE}}$ and orientation, represented by ${}^{d}\boldsymbol{R}_{\text{EE}}$ or ${}^{d}\boldsymbol{q}_{\text{EE}}$, are inside the manipulator *workspace*, the iterative process tends to converge to the solution $\boldsymbol{\theta}$ that is the "closest" to the initial guess $\boldsymbol{\theta}^{(0)}$. If the initial guess is not sufficiently close to a solution, the iterative process may not converge [51].
The inverse kinematics algorithm stops when the norms of the errors is lower than a prescribed tolerance tol_{IK} or when the maximum number of iterations allowed k_{MAX} is reached.

3 | Spacecraft-Manipulator System Path Planning

This chapter presents the core idea of the this thesis work and it is defined in two main sections: in Section 3.1 the objective of the path planning problem addresses in this thesis is formalized with all the constraints considered, then the solution approach is proposed in Section 3.2. The original Near-Optimal Sampling Based path planning algorithm is presented and described in Section 3.2.3

3.1. Path Planning Problem Formalization

The goal of the path planning problem for the free-flying manipulator system proposed in this thesis work is to find a *feasible* trajectory for the spacecraft base position and orientation, represented by the homogeneous transform matrix $\boldsymbol{T}_{\text{Cb}}^{\text{LVLH}}(t)$ and the robotic arm joint variables $\boldsymbol{\theta}(t)$, driving the system from its initial conditions ${}^{0}\boldsymbol{T}_{\text{Cb}}^{\text{LVLH}}(t_0)$, ${}^{0}\boldsymbol{\theta}(t_0)$ to a *successful* capture of the target body.
For a *successful* capture in a specific time instant t_f, the end-effector of the robotic arm must match the position, the orientation and the linear and angular velocities of the grasping point reference frame. In particular:

$$\boldsymbol{r}_{\text{EE}}(t_f) = \boldsymbol{r}_{gr}(t_f) \tag{3.1}$$

$$\boldsymbol{q}_{\text{EE}}(t_f) = \boldsymbol{q}_{gr}(t_f) \tag{3.2}$$

$$\dot{\boldsymbol{r}}_{\text{EE}}(t_f) = \dot{\boldsymbol{r}}_{gr}(t_f) \tag{3.3}$$

$$\boldsymbol{\omega}_{\text{EE}}(t_f) = \boldsymbol{\omega}_{Tb}(t_f) \tag{3.4}$$

To be *feasible* the trajectory have to satisfy several *hard constraints*:

- **Collision Avoidance**
  The trajectory must be such that the chaser approaches the target spacecraft in a safe way, avoiding any accidental collision with it as well as any self-collision.

- **Thruster Limits**
  Due to the operational limits of the thrusters used to maneuver the chaser spacecraft, the trajectory must be such that the required control acceleration **u** lies within finite bounds:
  $$|u_{x,y,z}| \leq u_{\text{MAX}} \quad \text{with} \quad u_{\text{MAX}} = \frac{F_{\text{MAX}}}{m_c} \tag{3.5}$$
  where $u_{x,y,z}$ represents the generic component of the control acceleration with respect to LVLH frame, m_c represents the mass of the chaser spacecraft and u_{MAX} is the maximum available control acceleration, limited by the maximum thrust F_{MAX} that thrusters can provide along each direction in LVLH frame.

- **Sensors Field-of-View**
  For relative navigation, pose estimation and grasping point tracking the trajectory must ensure that the target body is kept within the Field of View (FoV) of sensors (camera, LiDAR) mounted on the chaser spacecraft at any time instant [105].

- **Joint Limits and Kinematic Constraints**
  The joints of the robotic manipulator typically have bounded angular displacements as well as bounded velocities and accelerations due to actuators torque limits. Then the trajectory must not violate following conditions:
  $$\theta_{i\,\text{min}} \leq \theta_i(t) \leq \theta_{i\,\text{MAX}} \quad \text{for} \quad i = 1, ..., N \tag{3.6}$$
  $$\|\dot{\theta}_i(t)\| \leq \dot{\theta}_{i\,\text{MAX}} \quad \text{for} \quad i = 1, ..., N \tag{3.7}$$
  $$\|\ddot{\theta}_i(t)\| \leq \ddot{\theta}_{i\,\text{MAX}} \quad \text{for} \quad i = 1, ..., N \tag{3.8}$$
  where $\theta_{i\,\text{min}}$ and $\theta_{i\,\text{MAX}}$ denote respectively the minimum and maximum angular displacement of i^{th} joint, $\dot{\theta}_{i\,\text{MAX}}$ represents the maximum absolute value of i^{th} joint speed and $\ddot{\theta}_{i\,\text{MAX}}$ represents the maximum absolute value of i^{th} joint acceleration.
  The motion of the robotic manipulator is described by its kinematics and then the *pose* and the velocities of the end-effector in LVLH frame are constrained respectively by direct kinematics equation (2.22) and differential kinematics equation (2.24).

There are also additional aspects that have to be taken into account to improve the quality of the trajectory:

- **Fuel Consumption**
  The fuel used by the thrusters that is stored on board the chaser spacecraft is limited, thus representing a critical resource for the mission. The trajectory must be such that to minimize its consumption. The index used to measure the control

effort is the fuel cost index $C_{f_{tot}}$, that is defined as [9]:

$$C_{f_{tot}} = \int_{t_0}^{t_f} \sqrt{\mathbf{u} \cdot \mathbf{u}}\, dt \tag{3.9}$$

where $\mathbf{u}$ represent the control acceleration vector.

- **Duration of the Maneuver**
  Time is another critical factor of the mission. Indeed, the tumbling motion of the target body and of its appendages defines time-varying safe positions in the immediate vicinity of the target to perform the capture. The grasping point is accessible from a fixed position in the LVLH frame for a limited time interval, so the trajectory must be such that to complete the maneuver within that interval [86].

- **Kinematic Singularity Avoidance**
  The Jacobian matrix of the robotic manipulator $\boldsymbol{J}_{\mathrm{m}}$ relates the vector $\dot{\boldsymbol{\theta}}$ of joint velocities to the vector of end-effector velocities and it is a function of joint configuration $\boldsymbol{\theta}$, as discussed in Sec. 2.4.2. Those configuration at which Jacobian matrix $\boldsymbol{J}_{\mathrm{m}}$ in Equation (2.27) is rank-deficient are termed *kinematic singularities*. They are generally caused by the alignment of two or more axes of motion [51].
  The design of the trajectory must take into consideration singularities for the following reasons:

  – kinematic singularities represent configurations at which the mobility of the robotic arm is reduced, since the end-effector loses the ability to move instantaneously in one or more directions;

  – in the proximity of a kinematic singularity, large joint velocities, and then large control effort, are needed to produce small velocities of the end-effector [80].

  The trajectory must be such that to prevent the manipulator from reaching singular configuration. This can be attained maximizing the *manipulability measure* M of the robotic arm that vanishes at singular configurations and it is defined as [100]:

$$M(\boldsymbol{\theta}) = \sqrt{\det\left(\boldsymbol{J}_{\mathrm{m}}(\boldsymbol{\theta})\, \boldsymbol{J}_{\mathrm{m}}^{T}(\boldsymbol{\theta})\right)} \tag{3.10}$$

3.2. Proposed Path Planning Approach

To make the path planning problem described in Sec. 3.1 more tractable, the design of the trajectory has been divided in three parts, each one addressed to deal with different

tasks and constraints:

- **Planning of Chaser Base CoM Trajectory $\boldsymbol{r}_{\mathrm{Cb}}(t)$:**
  the trajectory of the chaser base CoM must be such that to approach the target without colliding with it minimizing fuel consumption and maneuver time t_f and reach a safe final desired position ${}^{d}\boldsymbol{r}_{\mathrm{Cb}}$ from which performing the capture with the robotic manipulator;
- **Planning of Chaser Base Attitude Trajectory $\boldsymbol{R}_{\mathrm{Cb}}(t)$:**
  the trajectory of the chaser base attitude must be such that to keep the sensors pointed towards the target for navigation purposes;
- **Planning of Joint Variables Trajectory $\boldsymbol{\theta}(t)$:**
  the trajectory of the joint angles must be such that end-effector matches position, orientation and velocities of the grasping point at time t_f, avoiding collisions with the chaser base and avoiding kinematic singularities.

The proposed approach is resumed in the scheme in Figure 3.1 and described in detail in the next sections. Its flexibility is due to the fact that it allows to plan the motion of the manipulator regardless of the method used to design the trajectory of the chaser base CoM and of its attitude.

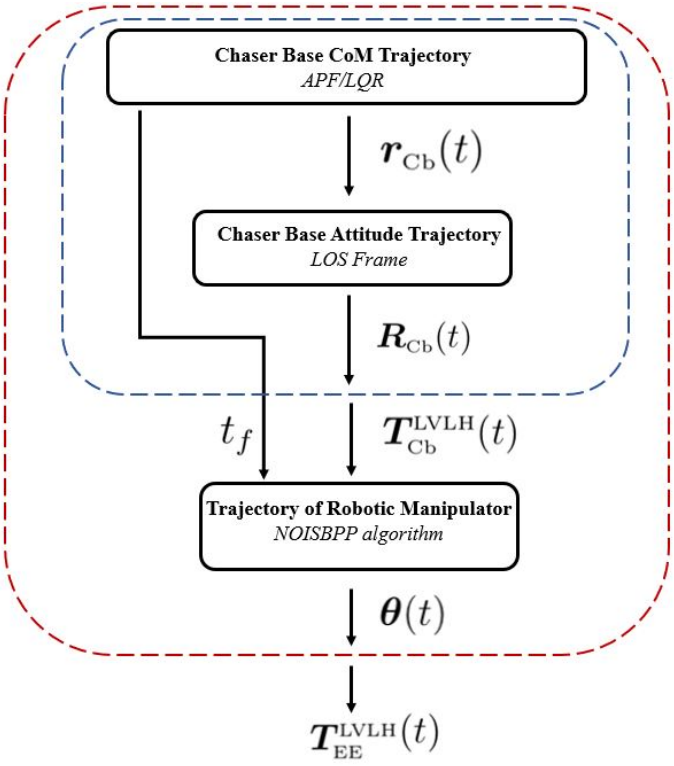


Figure 3.1: Proposed path planning approach scheme

3.2.1. Trajectory Planning of the Chaser Base CoM

The planning of the trajectory of the chaser base CoM is the first step of the proposed strategy. Its main objective is carrying the robotic manipulator in a position near the target to capture it, avoiding any collision. An algorithm based on the combination of Linear Quadratic Regulator and Artificial Potential field is used to design a collision-free trajectory and keep contained the fuel consumption.

Proposed Adaptive LQR/APF algorithm

The adaptive LQR/APF algorithm combines the the optimal control for linear system of the Linear Quadratic Regulator and the collision avoidance capabilities of the Artificial Potential Function algorithm [5]. The space manipulator system moves under the combined action of two control inputs:

$$\mathbf{u} = \mathbf{u}_{\text{LQR}} + \mathbf{u}_{\text{APF}} \tag{3.11}$$

where $\mathbf{u}$ is the total control acceleration, $\mathbf{u}_{\text{LQR}}$ is the LQR component of control acceleration that drives the chaser towards the desired final state $\mathbf{x}_d$ and $\mathbf{u}_{\text{APF}}$ is the APF component of the control acceleration that avoids that the chaser collides with the target. The two components are described in detail in the following sections.

LQR component

The LQR component of the LQR/APF control serves as the attractive force toward the goal state $\mathbf{x}_{\text{d}}$. The LQR controllers are based on finding the gain matrix $\mathbf{K}_{\text{LQR}}$, generating the optimal control acceleration $\mathbf{u}_{\text{LQR}} = -\mathbf{K}_{\text{LQR}}\mathbf{x}_{\text{e}}$ that minimizes the following quadratic cost function J_{LQR} [5]:

$$J_{\text{LQR}} = \frac{1}{2}\int_0^{+\infty} \left(\mathbf{x}_{\text{e}}^T\mathbf{Q}\mathbf{x}_{\text{e}} + \mathbf{u}^T\mathbf{R}\mathbf{u}\right)\, dt \tag{3.12}$$

where the tracking error $\mathbf{x}_{\text{e}}(t)$ is defined as the difference between the current state of the chaser base $\mathbf{x}_{\text{cw}}(t)$ and the goal state $\mathbf{x}_{\text{d}}$ as follow:

$$\mathbf{x}_{\text{e}}(t) = \mathbf{x}_{\text{cw}}(t) - \mathbf{x}_{\text{d}} \tag{3.13}$$

The gain matrix $\mathbf{K}_{\text{LQR}}$ is the one that minimizes the cost function J_{LQR} in Equation (3.12) according to a specific choice of *weighting matrices* $\mathbf{Q}$ and $\mathbf{R}$. This means that for a

different choice of weighting matrices, the form of gain matrix $\mathbf{K}_{\mathrm{LQR}}$ will be different.
Following the approach proposed in [5], the *state weighting matrix* $\mathbf{Q} \in \mathbb{R}^{6\times6}$ and the *control weighting matrix* $\mathbf{R} \in \mathbb{R}^{3\times3}$ are updated at constant time intervals Δt_{LQR} along the trajectory as a function of the distance $\mathrm{d}(t)$ between the actual position of the chaser base CoM $\boldsymbol{r}_{\mathrm{Cb}}(t)$ and its desired final position ${}^{d}\boldsymbol{r}_{\mathrm{Cb}}$; the gain matrix $\mathbf{K}_{\mathrm{LQR}}$ is updated accordingly.
Variable weighting matrices allow a faster convergence to the goal state, avoiding that the cost function slope tends to flatten in the vicinity of the goal due to the small state error values being considered [53].
In this work, the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are assumed diagonal matrices and their terms Q_{jj} and R_{jj} are respectively defined as:

$$\mathrm{Q}_{jj} = \alpha \qquad j = 1, ..., 6 \tag{3.14}$$

$$\mathrm{R}_{jj} = \frac{\beta}{u_{\mathrm{MAX}}^2} \qquad j = 1, ..., 3 \tag{3.15}$$

where the coefficients α and β are functions of the distance $\mathrm{d}(t)$ between the actual and final desired position of the chaser base CoM:

$$\mathrm{d}(t) = \|\boldsymbol{r}_{\mathrm{Cb}}(t) - {}^{d}\boldsymbol{r}_{\mathrm{Cb}}\| \tag{3.16}$$

and they are defined as follow:

$$\alpha(\mathrm{d}) = \frac{1 + \ln\left(\frac{\mathrm{d}_0}{\mathrm{d}}\right)}{\mathrm{d}_0} \qquad \beta(\mathrm{d}) = \left[1 + \exp\left(-\frac{\mathrm{d}}{\mathrm{d}_0}\right)\right] \mathrm{d} \tag{3.17}$$

where $\mathrm{d}_0 = \mathrm{d}(t_0)$ is the distance between the initial position of the chaser base CoM ${}^{0}\boldsymbol{r}_{\mathrm{Cb}} = \boldsymbol{r}_{\mathrm{Cb}}(t_0)$ and the desired final position. To avoid numerical issues when the chaser approaches to desired position, a lower limit is imposed to distance d, such that the condition $\mathrm{d} \geq 0.05\,\mathrm{m}$ is always satisfied.
The denominator of the term R_{jj} in Equation (3.15) is set to the square of the maximum available control acceleration u_{MAX}, limited by the maximum thrust F_{MAX} that thrusters can provide along each direction in LVLH frame The behaviour of weighting matrices components Q_{jj} and R_{jj} with respect to distance d is shown in Fig.3.2.

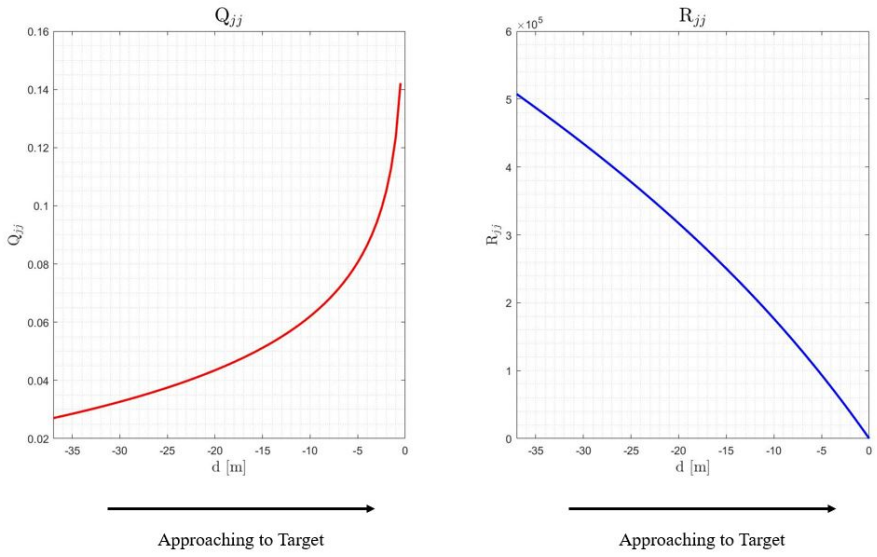


Figure 3.2: Weighting matrices components Q_{jj} and R_{jj} as function of distance d

The weights Q_{jj} increase as the chaser approaches the desired final position. This means that the importance of reducing the state error $\mathbf{x}_e$ and reaching the desired state $\mathbf{x}_d$ increases as the chaser approaches the final desired position.
The control weights R_{jj} have opposite behaviour since they decrease as the chaser approaches final desired position. The control effort is penalized less as the chaser approaches the target, since less and less control is needed to reach the final position as the chaser approaches it.

APF component

The APF component of the LQR/APF control is used to provide collision avoidance capabilities to the chaser when approaching the target.
When the chaser is approaching the target, it should avoid collisions between itself and appendages of the target, such as antennae and solar arrays. To enforce this requirement, the potential function $U(\boldsymbol{r})$ is used to define a keep-out zone around the target which contains all its components of and has only a tapered opening, denoted as *final approaching corridor*, coincident with the final desired approaching direction ${}^d\hat{\boldsymbol{r}}$, defined as:

$$ {}^d\hat{\boldsymbol{r}} = \frac{{}^d\boldsymbol{r}_{\mathrm{Cb}}}{\left\| {}^d\boldsymbol{r}_{\mathrm{Cb}} \right\|} \tag{3.18} $$

If the chaser never enters into this keep-out zone, no collision will happen and the chaser can only approach the target along final approaching corridor.

The potential function $U(\boldsymbol{r})$ is defined in LVLH coordinates as follows:

$$U(\boldsymbol{r}) = k_p \left(\|\boldsymbol{r}\| - \boldsymbol{r} \cdot {}^{d}\hat{\boldsymbol{r}} \right) \exp\left(-\frac{1}{2\,k_h} |h(\boldsymbol{r})| \right) \tag{3.19}$$

where $\boldsymbol{r} = [x, y, z]^T$ denotes the generic position of the chaser base CoM in LVLH frame, ${}^{d}\hat{\boldsymbol{r}}$ is the unit length vector pointing from target CoM to the final desired position of the chaser base, identifying the direction of the final approaching corridor, $h(\boldsymbol{r})$ is a scalar function such as $h(\boldsymbol{r}) = 0$ denotes the equation of the surface of the keep-out zone around the target [21] , $k_p > 0$ and $k_h > 0$ are positive parameters for modifying the shape of the potential function.
As shown in Fig.3.3, the potential function $U(\boldsymbol{r})$ is obtained by the product of two functions $M(\boldsymbol{r})$ and $Z(\boldsymbol{r})$, such that:

$$U(\boldsymbol{r}) = M(\boldsymbol{r})\, Z(\boldsymbol{r}) \tag{3.20}$$

where:

- the function $Z(\boldsymbol{r})$:

$$Z(\boldsymbol{r}) = \exp\left(-\frac{1}{2\,k_h} |h(\boldsymbol{r})| \right) \tag{3.21}$$

  is a Gaussian function that creates a maximum value region in correspondence of the surface of the keep-out zone around the target defined by equation $h(\boldsymbol{r}) = 0$.
  In this work a spherical keep-out zone has been used such that $h(\boldsymbol{r})$ is defined as:

$$h(\boldsymbol{r}) = \|\boldsymbol{r}\|^2 - a_{ko}^2 \tag{3.22}$$

  where a_{ko} is the radius of the keep-out zone sphere.
  The positive parameter k_h controls the width of the Gaussian function;

- the function $M(\boldsymbol{r})$:

$$M(\boldsymbol{r}) = k_p \left(\|\boldsymbol{r}\| - \boldsymbol{r} \cdot {}^{d}\hat{\boldsymbol{r}} \right) \tag{3.23}$$

  modulates the value of the Gaussian function $Z(\boldsymbol{r})$. The function $M(\boldsymbol{r})$ minimizes potential function U along the desired approaching direction ${}^{d}\hat{\boldsymbol{r}}$ making it equal to zero, creating a minimum value potential region to allow the chaser to get close to the target [21].
  The positive parameter k_p is proportional to the maximum value of the potential function $U(\boldsymbol{r})$.

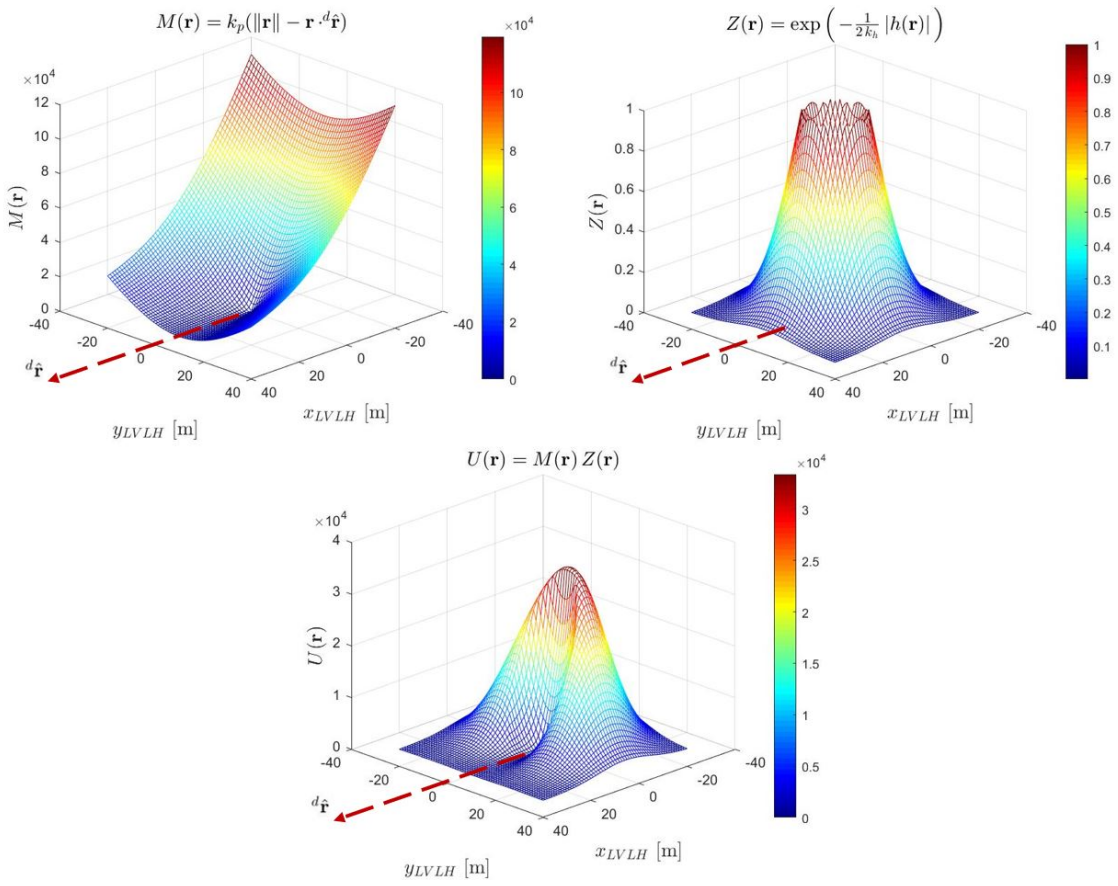


Figure 3.3: 2D representation of potential function $U(\boldsymbol{r}) = M(\boldsymbol{r})\,Z(\boldsymbol{r})$

The gradient of the potential function $U(\boldsymbol{r})$ with respect to LVLH frame coordinates can be written as:

$$\nabla U(\boldsymbol{r}) = \nabla M(\boldsymbol{r})\,Z(\boldsymbol{r}) + M(\boldsymbol{r})\,\nabla Z(\boldsymbol{r}) \tag{3.24}$$

where the gradient of function $M(\boldsymbol{r})$ is computed from Equation (3.23), obtaining the following result:

$$\nabla M(\boldsymbol{r}) = k_p \left(\frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} - {}^{d}\hat{\boldsymbol{r}} \right) \tag{3.25}$$

and the gradient of function $Z(\boldsymbol{r})$ is obtained from Equation (3.21) and it is equal to:

$$\nabla Z(\boldsymbol{r}) = -\frac{1}{2\,k_h} \exp\left(-\frac{1}{2\,k_h} |h(\boldsymbol{r})| \right) \nabla \left(|h(\boldsymbol{r})| \right) \tag{3.26}$$

Since the chaser can't cross the keep-out zone surface defined by $h(\boldsymbol{r}) = 0$, the function $h(\boldsymbol{r})$ does not change sign and then the term $\nabla\left(|h(\boldsymbol{r})|\right)$ in Equation (3.26) can simply be replaced by $\nabla h(\boldsymbol{r})$ [49].

Taking this last consideration into account and substituting Equations (3.21), (3.23),

(3.25) and (3.26) into Equation (3.24), the gradient of potential function $U(\boldsymbol{r})$ results:

$$\nabla U(\boldsymbol{r}) = c_1(\boldsymbol{r}) \left(\frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} - {}^{d}\hat{\boldsymbol{r}} \right) - c_2(\boldsymbol{r}) \nabla h(\boldsymbol{r}) \tag{3.27}$$

where $c_1(\boldsymbol{r})$ and $c_2(\boldsymbol{r})$ are scalar functions of position $\boldsymbol{r}$ and they are equal to:

$$c_1(\boldsymbol{r}) = k_p \, Z(\boldsymbol{r}) \qquad c_1(\boldsymbol{r}) > 0 \quad \forall \boldsymbol{r} \tag{3.28}$$

$$c_2(\boldsymbol{r}) = \frac{1}{2\, k_h} M(\boldsymbol{r}) \qquad c_2(\boldsymbol{r}) \geq 0 \quad \forall \boldsymbol{r} \tag{3.29}$$

The APF control acceleration component $\mathbf{u}_{\text{APF}}$ has opposite direction of ∇U reported in Equation (3.27) and it is taken as:

$$\mathbf{u}_{\text{APF}} = -k_{\text{APF}} \nabla U(\boldsymbol{r}) = k_{\text{APF}} \left(c_1(\boldsymbol{r}) \left({}^{d}\hat{\boldsymbol{r}} - \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} \right) + c_2(\boldsymbol{r}) \nabla h(\boldsymbol{r}) \right) \tag{3.30}$$

where $k_{\text{APF}} \in \mathbb{R}^+$ is a positive tuning parameter.
The APF control acceleration field $\mathbf{u}_{\text{APF}}$ in x-y plane of LVLH frame is shown in Fig. 3.4.

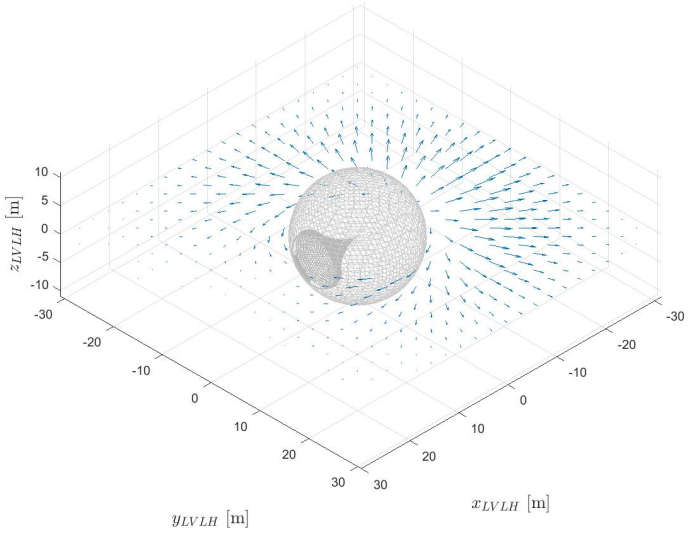


Figure 3.4: APF control acceleration $\mathbf{u}_{\text{APF}}$ vector field

As can be observed in Equation (3.30), the APF control acceleration $\mathbf{u}_{\text{APF}}$ is obtained by the sum of two vectors, modulated by scalar functions $c_1(\boldsymbol{r})$ and $c_2(\boldsymbol{r})$:

- the component directed as $\left({}^{d}\hat{\boldsymbol{r}} - \frac{\boldsymbol{r}}{\|\boldsymbol{r}\|} \right)$ tends to align the direction identified by the current position of the chaser with the desired final approaching direction;

- the component directed as $\nabla h(\boldsymbol{r})$ acts to prevent the chaser from approaching the keep-out zone surface.

To generate the trajectoy of the chaser base CoM, the APF control acceleration $\mathbf{u}_{\text{APF}}$ is updated at constant time interval Δt_{APF}.

3.2.2. Trajectory Planning of the Chaser Base Attitude

The trajectory of the chaser base attitude is defined starting from the planned trajectory of the chaser base CoM and imposing that sensors boresight axis has always to be pointed towards the target body for navigation purposes. It take as input the trajectory of the chaser base CoM $\boldsymbol{r}_{\text{Cb}}$ and it returns the evolution of the chaser base attitude over time $\boldsymbol{R}_{Cb}^{\text{LVLH}}$.

Line Of Sight Reference Frame

When the chaser maneuvers in close proximity to a non-cooperative object, relative position, relative attitude and other information of the target must be attained through the sensors of the chaser, like LiDAR, cameras or other vision systems, that have limited field of view [6]. This means that chaser's sensors needs to be pointed towards the target for getting the necessary measurements.
Such condition is attained if the chaser tracks the *Line Of Sight (LOS)* frame whose coordinate axes' directions are defined as follow [34]:

$$\hat{\boldsymbol{x}}_{LOS} = -\frac{\boldsymbol{r}}{\|\boldsymbol{r}\|}; \qquad \hat{\boldsymbol{y}}_{LOS} = \frac{\boldsymbol{r}_{xy} \times \boldsymbol{r}}{\|\boldsymbol{r}_{xy} \times \boldsymbol{r}\|}; \qquad \hat{\boldsymbol{z}}_{LOS} = \hat{\boldsymbol{x}}_{LOS} \times \hat{\boldsymbol{y}}_{LOS} \tag{3.31}$$

where $\boldsymbol{r} = [x, y, z]^T$ denotes the chaser's position vector relative to the target expressed in $\mathcal{L}_{LVLH}$ and $\boldsymbol{r}_{xy} = [x, y, 0]^T$ denotes its projection in the xy-plane of LVLH frame.
In Eqs. (3.31) it is assumed that chaser sensors boresight axis coincides with x-axis of chaser's base body frame $\hat{\boldsymbol{x}}_{Cb}$.
The orientation of the LOS frame with respect to LVLH frame can be represented by the rotation matrix $\boldsymbol{R}_{\text{LOS}}^{\text{LVLH}}$:

$$\boldsymbol{R}_{\text{LOS}}^{\text{LVLH}} = [\hat{\boldsymbol{x}}_{LOS}, \hat{\boldsymbol{y}}_{LOS}, \hat{\boldsymbol{z}}_{LOS}] \tag{3.32}$$

where columns are LOS frame's axes in $\mathcal{L}_{LVLH}$ defined in Eqs. (3.31).
The LOS frame's angular speed $\boldsymbol{\omega}_{\text{LOS}}^{\text{LVLH}}$ with respect to LVLH frame and expressed in

$\mathcal{L}_{LVLH}$ can be computed as follow [52]:

$$\dot{\boldsymbol{R}}_{\text{LOS}}^{\text{LVLH}} = [\boldsymbol{\omega}_{\text{LOS}}^{\text{LVLH}}]^{\times} \boldsymbol{R}_{\text{LOS}}^{\text{LVLH}} \quad \longrightarrow \quad [\boldsymbol{\omega}_{\text{LOS}}^{\text{LVLH}}]^{\times} = \dot{\boldsymbol{R}}_{\text{LOS}}^{\text{LVLH}} \, (\boldsymbol{R}_{\text{LOS}}^{\text{LVLH}})^{T} \tag{3.33}$$

where $[\,\cdot\,]^{\times}$ denotes the *skew symmetric matrix operator* and the matrix $\dot{\boldsymbol{R}}_{\text{LOS}}^{\text{LVLH}}$ is defined as:

$$\dot{\boldsymbol{R}}_{\text{LOS}}^{\text{LVLH}} = [\dot{\hat{\boldsymbol{x}}}_{LOS}, \, \dot{\hat{\boldsymbol{y}}}_{LOS}, \, \dot{\hat{\boldsymbol{z}}}_{LOS}] \tag{3.34}$$

whose columns can be obtained taking time derivative of Eqs. (3.31), obtaining:

$$\begin{aligned}
\dot{\hat{\boldsymbol{x}}}_{LOS} &= \left(\frac{\boldsymbol{I}_{3,3} - \hat{\boldsymbol{x}}_{\text{LOS}} \hat{\boldsymbol{x}}_{\text{LOS}}^{T}}{\|\boldsymbol{r}\|} \right) (-\dot{\boldsymbol{r}}) \\
\dot{\hat{\boldsymbol{y}}}_{LOS} &= \left(\frac{\boldsymbol{I}_{3,3} - \hat{\boldsymbol{y}}_{\text{LOS}} \hat{\boldsymbol{y}}_{\text{LOS}}^{T}}{\|\boldsymbol{r}_{xy} \times \boldsymbol{r}\|} \right) (\dot{\boldsymbol{r}}_{xy} \times \boldsymbol{r} + \boldsymbol{r}_{xy} \times \dot{\boldsymbol{r}}) \\
\dot{\hat{\boldsymbol{z}}}_{LOS} &= \left(\boldsymbol{I}_{3,3} - \hat{\boldsymbol{z}}_{\text{LOS}} \hat{\boldsymbol{z}}_{\text{LOS}}^{T} \right) (\dot{\hat{\boldsymbol{x}}}_{\text{LOS}} \times \hat{\boldsymbol{y}}_{\text{LOS}} + \hat{\boldsymbol{x}}_{\text{LOS}} \times \dot{\hat{\boldsymbol{y}}}_{\text{LOS}})
\end{aligned} \tag{3.35}$$

where $\boldsymbol{I}_{3,3}$ denotes 3×3 identity matrix, $\dot{\boldsymbol{r}} = [\dot{x}, \dot{y}, \dot{z}]^{T}$ represents chaser's relative linear speed in $\mathcal{L}_{LVLH}$ and $\dot{\boldsymbol{r}}_{xy} = [\dot{x}, \dot{y}, 0]^{T}$ is its projection in xy-plane of LVLH frame.
Two final remarks are needed to complete the definition of the LOS frame [34]:

1. when $z = 0$, chaser's position vector $\boldsymbol{r}$ lies in xy-plane of LVLH frame and it is parallel to $\boldsymbol{r}_{xy}$, then $\boldsymbol{r}_{xy} \times \boldsymbol{r} = \boldsymbol{0}$ and $\hat{\boldsymbol{y}}_{LOS}$ is not defined.
   In this case $\hat{\boldsymbol{x}}_{LOS} = [\hat{x}_{LOS_x}, \, \hat{x}_{LOS_y}, \, 0]^{T}$ and the problem can be solved redefining $\hat{\boldsymbol{y}}_{LOS}$ as $\hat{\boldsymbol{y}}_{LOS} = [-\hat{x}_{LOS_y}, \, \hat{x}_{LOS_x}, \, 0]^{T}$. Then, it results that $\hat{\boldsymbol{z}}_{LOS} = [0, \, 0, \, 1]^{T}$
2. when $x = 0$ and $y = 0$, chaser's position vector $\boldsymbol{r}$ is aligned with $\hat{\boldsymbol{z}}_{LVLH}$, $\boldsymbol{r}_{xy} = \boldsymbol{0}$ and $\hat{\boldsymbol{y}}_{LOS}$ is not defined.
   This happens when $\hat{\boldsymbol{x}}_{LOS} = [0, \, 0, \, \pm 1]^{T}$ then to solve the problem vector $\boldsymbol{r}_{xy}$ can be regarded as constant for $||\hat{x}_{LOS_z}| - 1| \leq \epsilon$ where ϵ is a small constant.

3.2.3. Trajectory Planning of the Robotic Manipulator

The proposed Near-Optimal Incremental Sampling-Based Path Planning (NOISBPP) algorithm is described in detail in this section. It takes as input the time t_f needed to reach the target, the trajectory of the chaser base CoM $\boldsymbol{r}_{\text{Cb}}(t)$, the trajectory of its attitude $\boldsymbol{R}_{\text{Cb}}^{\text{LVLH}}(t)$ and the trajectory of the grasping point $\boldsymbol{r}_{gr}(t)$ and it returns, coupled with an inverse kinematics algorithm, the trajectory of the robotic manipulator joints $\boldsymbol{\theta}(t)$. This

is the last step for planning the trajectory of the end-effector in the LVLH frame.

Proposed Near-Optimal Incremental Sampling-Based algorithm

The robotic arm starts moving at time instant t_{start} and the capture of the target is performed at time instant t_f.
The proposed algorithm divides the time interval $[t_{start} : t_f]$ into a number of sub-intervals N_{int}, each one of duration Δt_{int}, such that the vector $\boldsymbol{t} \in \mathbb{R}^{N_{int}+1}$ results:

$$\boldsymbol{t} = [\, t_1 \,, ..., t_{k-1} \,, t_k \,, ..., t_{N_{int}+1} \,]^T \tag{3.36}$$

where its components are defined as follow:

$$t_1 = t_{start} \tag{3.37}$$

$$t_{N_{int}+1} = t_f \tag{3.38}$$

$$t_k = t_{k-1} \, + \, \Delta t_{int} \tag{3.39}$$

Then, at each time instant t_k from t_2 to $t_{N_{int}}$, the algorithm computes the trajectories of the joints by incrementally selecting joint angles waypoints $\boldsymbol{\theta}^{(k)}$ and joint velocities waypoints $\dot{\boldsymbol{\theta}}^{(k)}$. The selected waypoints $\boldsymbol{\theta}^{(k)}$ and $\dot{\boldsymbol{\theta}}^{(k)}$ at instant t_k are those which, among all the possible solutions that are sampled, minimize a cost functional that takes into account multiple objectives. Each waypoint is then connected with the next one by a polynomial trajectory.
The proposed algorithm is based on the repetition of four main steps at each time instant t_k, with k that goes from 2 to N_{int}:

1. *define sampling limits*: given the current waypoint defined by $\boldsymbol{\theta}_{\text{current}}^{(k-1)}$ and $\dot{\boldsymbol{\theta}}_{\text{current}}^{(k-1)}$, define the portions of joint configuration and joint velocity spaces where searching for a solution $\boldsymbol{\theta}^{(k)}$ and $\dot{\boldsymbol{\theta}}^{(k)}$ at time t_k. The *current* condition denotes the last waypoint before the algorithm selects the new one;

2. *sample*: the defined portions of joint configuration and joint velocities space are explored through random sampling, generating multiple possible solutions;

3. *evaluate the cost of each sample and select the cost minimizing one*: each sample is evaluated by multiple cost functions and a total cost is then assigned to it. The sample with the lowest total cost is selected as a waypoint at instant t_k. The minimization of the cost functions guides the end-effector to the grasping point in LVLH frame and imposes constraints on joint angles, velocities and accelerations.

4. *update*: the selected waypoint becomes the new current waypoint and steps from 1 to 4 are repeated moving forward in time.

The last part of the trajectory, between instant $t_{N_{int}}$ and capture instant $t_{N_{int}+1}$ is computed by solving the inverse kinematics problem described in Section 2.4.3.
The main steps of the algorithm are reported in the scheme in Figure 3.5 and are here described in detail.

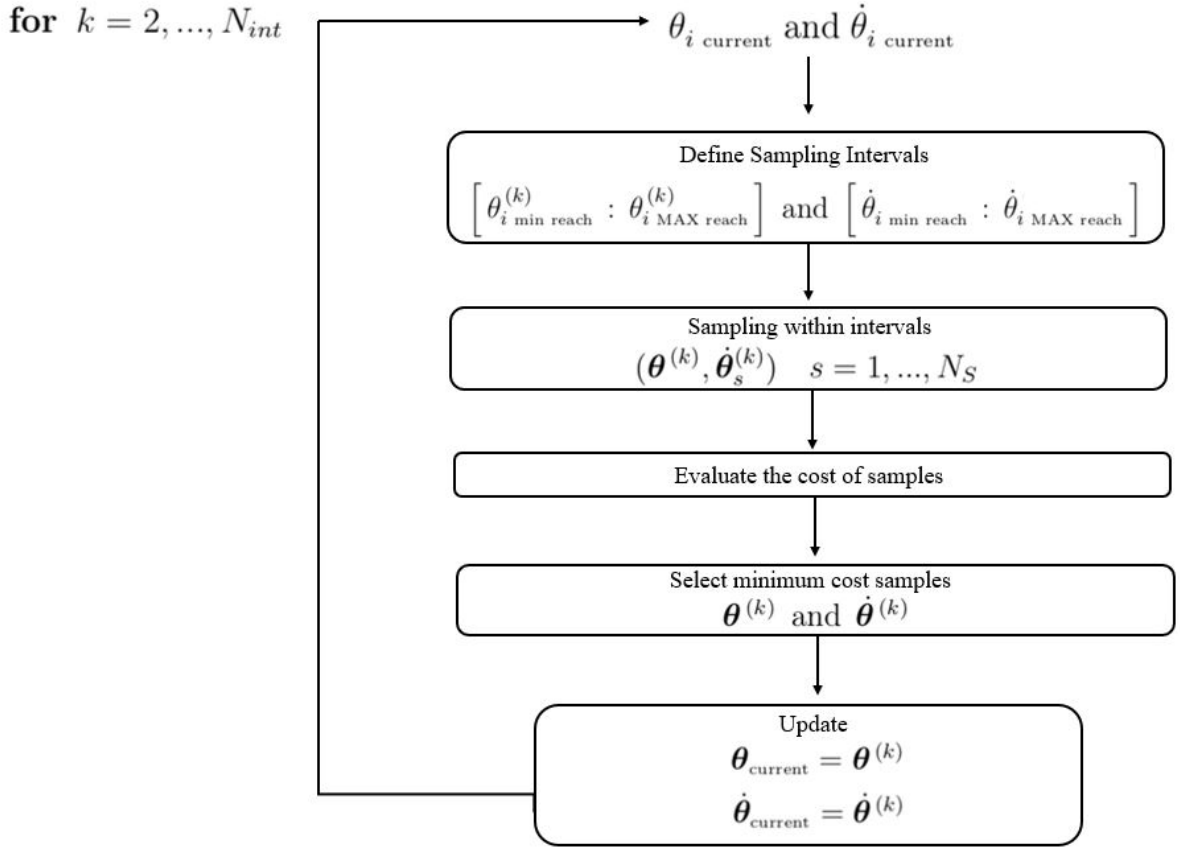


Figure 3.5: Scheme of the NOISBPP algorithm

Define Sampling Limits

Given the generic current joint angle $\theta^{(k-1)}_{i\ \text{current}}$ and its current speed $\dot{\theta}^{(k-1)}_{i\ \text{current}}$ at time instant t_{k-1}, their possible values at the successive time instant t_k, respectively $\theta^{(k)}_i$ and $\dot{\theta}^{(k)}_i$, must be in the intervals $\left[\theta^{(k)}_{i\ \text{min reach}} : \theta^{(k)}_{i\ \text{MAX reach}}\right]$ and $\left[\dot{\theta}^{(k)}_{i\ \text{min reach}} : \dot{\theta}^{(k)}_{i\ \text{MAX reach}}\right]$. These intervals, once defined for every joint i with $i = 1, ..., N$, limit the portion of the joint configuration space and joint velocities space to sample to find a solution $\boldsymbol{\theta}^{(k)}$, $\dot{\boldsymbol{\theta}}^{(k)}$ at time instant t_k. The boundary values of the intervals represents the "furthest" joint angles and speeds that can be reached after the time interval Δt_{int} from $\theta^{(k-1)}_{i\ \text{current}}$ and $\dot{\theta}^{(k-1)}_{i\ \text{current}}$ compatibly with the constraints on joint angles, speed and acceleration.

Joint Angles Sampling Limits

Each time interval Δt_{int} is divided in two consecutive sub-intervals, denoted by $\Delta t_1^{\pm}$ and Δt_2, as shown in Equation (3.40). A constant acceleration $+\ddot{\theta}_{i\,\mathrm{MAX}}$ (or deceleration $-\ddot{\theta}_{i\,\mathrm{MAX}}$) is assumed throughout the first interval Δt_1^{+} (or Δt_1^{-}), while a constant speed $+\dot{\theta}_{i\,\mathrm{MAX}}$ (or $-\dot{\theta}_{i\,\mathrm{MAX}}$) is assumed in the second interval:

$$\Delta t_{int} = \Delta t_1^{\pm} + \Delta t_2 \tag{3.40}$$

where time interval $\Delta t_1^{\pm}$ is defined as:

$$\Delta t_1^{+} = \frac{\dot{\theta}_{i\,\mathrm{MAX}} - \dot{\theta}_{i\ \mathrm{current}}^{(k-1)}}{\ddot{\theta}_{i\,\mathrm{MAX}}} \quad \text{or} \quad \Delta t_1^{-} = \frac{\dot{\theta}_{i\,\mathrm{MAX}} + \dot{\theta}_{i\ \mathrm{current}}^{(k-1)}}{\ddot{\theta}_{i\,\mathrm{MAX}}} \tag{3.41}$$

is the time interval needed to reach the maximum $+\dot{\theta}_{i\,\mathrm{MAX}}$ (or minimum $-\dot{\theta}_{i\,\mathrm{MAX}}$) speed from $\dot{\theta}_{i\ \mathrm{current}}^{(k-1)}$ with a constant maximum acceleration $+\ddot{\theta}_{i\,\mathrm{MAX}}$ (or deceleration $-\ddot{\theta}_{i\,\mathrm{MAX}}$).
Time interval Δt_2 is defined as follow:

$$\Delta t_2 = \begin{cases} \Delta t_{int} - \Delta t_1^{\pm} & \text{if} \quad \Delta t_{int} > \Delta t_1^{\pm} \\ 0 & \text{if} \quad \Delta t_{int} \leq \Delta t_1^{\pm} \end{cases} \tag{3.42}$$

The definition of Δt_2 in Equation (3.42) is due to the fact that a joint acceleration (or deceleration) is possible until the maximum (or minimum) speed imposed by actuator constraints is reached. If the interval Δt_{int} is long enough, the joint motion continues in the Δt_2 interval at a constant maximum (or minimum) speed, after reaching it in previous time interval Δt_1.
The lower limit $\theta_{i\ \mathrm{min\ reach}}^{(k)}$ of the i^{th} joint angle sampling interval is given by:

$$\theta_{i\ \mathrm{min\ reach}}^{(k)} = \max\left(\left[\theta_{i\ \mathrm{LOW\ reach}}^{(k)}\ ,\ \theta_{i\ \mathrm{min}} \right] \right) \tag{3.43}$$

where $\theta_{i\ \mathrm{min}}$ is the lower excursion limit imposed by mechanical constraint of the i^{th} joint and $\theta_{i\ \mathrm{LOW\ reach}}^{(k)}$ represents the lowest joint angle that is reachable at time instant t_k without considering constraints and considering a joint motion at constant maximum deceleration $-\ddot{\theta}_{i\,\mathrm{MAX}}$ and then at constant minimum speed $-\dot{\theta}_{i\,\mathrm{MAX}}$. It is defined as:

$$\theta_{i\ \mathrm{LOW\ reach}}^{(k)} = \theta_{i\ \mathrm{LOW\ reach\,1}} + \Delta\theta_{i\ \mathrm{LOW\ reach\,2}} \tag{3.44}$$

where $\theta_{i\ \mathrm{LOW\ reach\,1}}$ is the angle reached after time interval Δt_1^{-} assuming constant maximum

deceleration $-\ddot{\theta}_{i\,\mathrm{MAX}}$ and that is equal to [17]:

$$\theta_{i\ \mathrm{LOW\ reach\,1}} = \theta_{i\ \mathrm{current}}^{(k-1)} + (\Delta t_1^-)\,\dot{\theta}_{i\ \mathrm{current}}^{(k-1)} - \frac{1}{2}(\Delta t_1^-)^2\,\ddot{\theta}_{i\ \mathrm{MAX}} \tag{3.45}$$

and $\Delta\theta_{i\ \mathrm{LOW\ reach\,2}}$ is the angle swept in the time interval Δt_2 and it is equal to:

$$\Delta\theta_{i\ \mathrm{LOW\ reach\,2}} = -\,(\Delta t_2)\,\dot{\theta}_{i\ \mathrm{MAX}} \tag{3.46}$$

The upper limit $\theta_{i\ \mathrm{MAX\ reach}}^{(k)}$ of the i^{th} joint angle sampling interval is given by:

$$\theta_{i\ \mathrm{MAX\ reach}}^{(k)} = \min\left(\left[\theta_{i\ \mathrm{UP\ reach}}^{(k)},\ \theta_{i\ \mathrm{MAX}}\right]\right) \tag{3.47}$$

where $\theta_{i\ \mathrm{MAX}}$ is the upper excursion limit imposed by mechanical constraint of the i^{th} joint and $\theta_{i\ \mathrm{UP\ reach}}^{(k)}$ represents the uppest joint angle that is reachable at time instant t_k without considering joint excursion limits and considering a joint motion at constant maximum acceleration $+\ddot{\theta}_{i\,\mathrm{MAX}}$ and then at constant maximum speed $+\dot{\theta}_{i\,\mathrm{MAX}}$. It is defined as:

$$\theta_{i\ \mathrm{UP\ reach}}^{(k)} = \theta_{i\ \mathrm{UP\ reach\,1}} + \Delta\theta_{i\ \mathrm{UP\ reach\,2}} \tag{3.48}$$

where $\theta_{i\ \mathrm{UP\ reach\,1}}$ is the angle reached after time interval Δt_1^+ assuming constant maximum acceleration $+\ddot{\theta}_{i\,\mathrm{MAX}}$ and that is equal to [17]:

$$\theta_{i\ \mathrm{UP\ reach\,1}} = \theta_{i\ \mathrm{current}}^{(k-1)} + (\Delta t_1^+)\,\dot{\theta}_{i\ \mathrm{current}}^{(k-1)} + \frac{1}{2}(\Delta t_1^+)^2\,\ddot{\theta}_{i\ \mathrm{MAX}} \tag{3.49}$$

and $\Delta\theta_{i\ \mathrm{UP\ reach\,2}}$ is the angle swept in the time interval Δt_2 and it is equal to:

$$\Delta\theta_{i\ \mathrm{UP\ reach\,2}} = +\,(\Delta t_2)\,\dot{\theta}_{i\ \mathrm{MAX}} \tag{3.50}$$

The Equations (3.43) and (3.47) return the correct result except in the following cases:

1. if $\theta_{i\ \mathrm{LOW\ reach}}^{(k)} \geq \theta_{i\ \mathrm{MAX}}$. In this case, according to Equation (3.43), $\theta_{i\ \mathrm{min\ reach}}^{(k)}$ would assume a value higher than the maximum one allowed. To solve this issue, $\theta_{i\ \mathrm{min\ reach}}^{(k)}$ is set equal to $\theta_{i\ \mathrm{current}}^{(k-1)}$. In particular: $\theta_{i\ \mathrm{min\ reach}}^{(k)} = \theta_{i\ \mathrm{current}}^{(k-1)}$.
2. if $\theta_{i\ \mathrm{UP\ reach}}^{(k)} \leq \theta_{i\ \mathrm{min}}$. In this case, according to Equation (3.47), $\theta_{i\ \mathrm{MAX\ reach}}^{(k)}$ would assume a value lower than the minimum one allowed. To solve this issue, $\theta_{i\ \mathrm{MAX\ reach}}^{(k)}$ is set equal to $\theta_{i\ \mathrm{current}}^{(k-1)}$. In particular: $\theta_{i\ \mathrm{MAX\ reach}}^{(k)} = \theta_{i\ \mathrm{current}}^{(k-1)}$.

In Figure 3.6 is given a qualitative representation of the procedure to define the joint angle sampling interval.

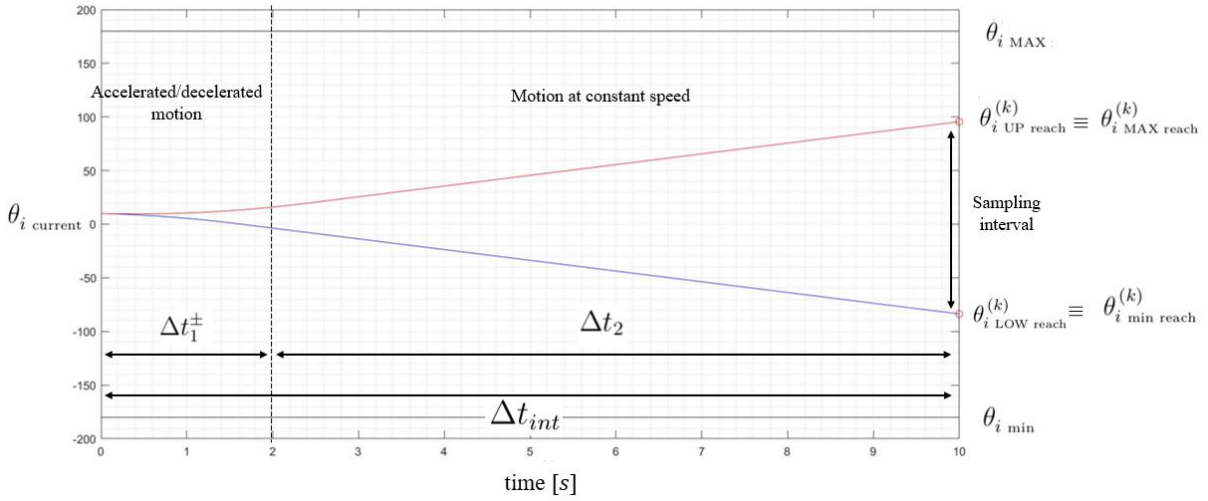


Figure 3.6: Sampling limits definition

Joint Velocities Sampling Limits

Concerning joint velocities, the lower limit $\dot{\theta}^{(k)}_{i\ \text{min reach}}$ of the i^{th} joint velocity sampling interval is given by:

$$\dot{\theta}^{(k)}_{i\ \text{min reach}} = \max\left(\left[\dot{\theta}^{(k)}_{i\ \text{LOW reach}}\ ,\ -\dot{\theta}_{i\ \text{MAX}}\right]\right) \tag{3.51}$$

where $-\dot{\theta}_{i\ \text{MAX}}$ is the minimum speed at which the i^{th} joint can rotate and $\dot{\theta}^{(k)}_{i\ \text{LOW reach}}$ represents the lowest velocity that the i^{th} joint can reach at time t_k from $\dot{\theta}^{(k-1)}_{i\ \text{current}}$ without taking into accounts constraints and considering a constant maximum deceleration $-\ddot{\theta}_{i\ \text{MAX}}$. It can be computed as follow:

$$\dot{\theta}^{(k)}_{i\ \text{LOW reach}} = \dot{\theta}^{(k-1)}_{i\ \text{current}} - \Delta t_{int}\,\ddot{\theta}_{i\ \text{MAX}} \tag{3.52}$$

The upper limit $\dot{\theta}^{(k)}_{i\ \text{MAX reach}}$ of the i^{th} joint velocity sampling interval is given by:

$$\dot{\theta}^{(k)}_{i\ \text{MAX reach}} = \min\left(\left[\dot{\theta}^{(k)}_{i\ \text{UP reach}}\ ,\ \dot{\theta}_{i\ \text{MAX}}\right]\right) \tag{3.53}$$

where $\dot{\theta}_{i\ \text{MAX}}$ is the maximum speed at which the i^{th} joint can rotate and $\dot{\theta}^{(k)}_{i\ \text{UP reach}}$ represents the highest velocity that the i^{th} joint can reach at time t_k from $\dot{\theta}^{(k-1)}_{i\ \text{current}}$ without taking into accounts constraints and considering a constant maximum acceleration $\ddot{\theta}_{i\ \text{MAX}}$. It can

be computed as follow:

$$\dot{\theta}_{i\ \text{UP reach}}^{(k)} = \dot{\theta}_{i\ \text{current}}^{(k-1)} + \Delta t_{int}\, \ddot{\theta}_{i\ \text{MAX}} \tag{3.54}$$

Since the joints are unlikely to move throughout the entire interval Δt_{int} with the maximum acceleration and the maximum speed, a safety factor can be used to decrease the effective maximum speed and the effective maximum acceleration imposed by actuators and then reduce the amplitudes of sampling intervals. The absolute values of maximum velocity $\dot{\theta}_{i\ \text{MAX}}$ and maximum acceleration $\ddot{\theta}_{i\ \text{MAX}}$ can be respectively substituted by $\dot{\theta}_{i\ \text{safe}}$ and $\ddot{\theta}_{i\ \text{safe}}$ that are defined as:

$$\dot{\theta}_{i\ \text{safe}} = \lambda_{\text{vel}}\, \dot{\theta}_{i\ \text{MAX}} \tag{3.55}$$

$$\ddot{\theta}_{i\ \text{safe}} = \lambda_{\text{acc}}\, \ddot{\theta}_{i\ \text{MAX}} \tag{3.56}$$

where $0 < \lambda_{\text{vel}} \leq 1$ and $0 < \lambda_{\text{acc}} \leq 1$ represent respectively the safety factor on the value of joint maximum speed and the safety factor on the value of joint maximum acceleration. The use of safety factors also helps to keep the motion of the joints within the constraints.

Sampling Procedure

The intervals $\left[\theta_{i\ \text{min reach}}^{(k)} : \theta_{i\ \text{MAX reach}}^{(k)}\right]$ and $\left[\dot{\theta}_{i\ \text{min reach}}^{(k)} : \dot{\theta}_{i\ \text{MAX reach}}^{(k)}\right]$ are discretized with steps respectively equal to $\delta\theta$ and $\delta\dot{\theta}$ to generate a finite population of possible i^{th} joint angles $\theta_i^{(k)}$ and a finite population of possible i^{th} joint velocities $\dot{\theta}_i^{(k)}$. Then, each of the populations thus obtained is sampled N_S times randomly.
This procedure is repeated for every i^{th} joint, with i going from 1 to N, then obtaining N_S joint angle configuration vectors $\boldsymbol{\theta}_s^{(k)} \in \mathbb{R}^N$ and N_S joint velocities vector $\dot{\boldsymbol{\theta}}_s^{(k)} \in \mathbb{R}^N$, with $s = 1, ..., N_S$.

Evaluating Cost of each Sample and Select Minimum Cost Samples

The joint angle configuration vector $\boldsymbol{\theta}^{(k)}$ and the joint velocities vector $\dot{\boldsymbol{\theta}}^{(k)}$ at time instant t_k are selected among all the samples through an *optimization process.*
A *total cost* c_s^{tot} is associated to each sampled joint configuration vector $\boldsymbol{\theta}_s^{(k)}$ and to each sampled joint velocities vector $\dot{\boldsymbol{\theta}}_s^{(k)}$. Then $\boldsymbol{\theta}^{(k)}$ and $\dot{\boldsymbol{\theta}}^{(k)}$ are selected with the following procedure:

1. select the joint angle configuration vector that minimizes total cost $c_s^{\text{tot}}(\boldsymbol{\theta}_s^{(k)})$:

$$\boldsymbol{\theta}^{(k)} = \min_{\boldsymbol{\theta}_s^{(k)}} c_s^{\text{tot}}(\boldsymbol{\theta}_s^{(k)}) \quad s = 1, ..., N_S \tag{3.57}$$

2. considering the joint angle configuration vector $\boldsymbol{\theta}^{(k)}$ found at previous step, select

the joint velocities vector that minimizes total cost $c_s^{\text{tot}}(\boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{\theta}}_s^{(k)})$:

$$\dot{\boldsymbol{\theta}}^{(k)} = \min_{\dot{\boldsymbol{\theta}}_s^{(k)}} \; c_s^{\text{tot}}(\boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{\theta}}_s^{(k)}) \quad s = 1, ..., N_S \tag{3.58}$$

The total cost c_s^{tot} of the s^{th} sample takes into account multiple cost functions f_j that have to be minimized at each time instant t_k. The cost functions for selecting joint angles configuration vector $\boldsymbol{\theta}^{(k)}$ are different from those for selecting the joint velocities vector $\dot{\boldsymbol{\theta}}^{(k)}$ but the procedure to compute total cost c_s^{tot} is the same in both cases and it is here described:

1. for each sample $\boldsymbol{\theta}_s^{(k)}$ or $\dot{\boldsymbol{\theta}}_s^{(k)}$, evaluate each cost function f_j. The cost of the s^{th} sample with respect to the j^{th} function f_j is denoted by c_s^j.

$$c_s^j = f_j(\boldsymbol{\theta}_s^{(k)}) \quad \text{or} \quad c_s^j = f_j(\dot{\boldsymbol{\theta}}_s^{(k)}) \tag{3.59}$$

2. for each function f_j, find the minimum value $c_{\min}^j$ among all the costs c_s^j referred to the same j^{th} function.

$$c_{\min}^j = \min_{\substack{c_s^j \\ \text{for fixed } j}} c_s^j \quad s = 1, ..., N_S \tag{3.60}$$

3. for each function f_j, make dimensionless the cost of each sample c_s^j referred to the same j^{th} function with respect to the minimum value $c_{\min}^j$. The dimensionless cost of the s^{th} sample with respect to the j^{th} function f_j is denoted by c_s^{*j}. It is equal to 1 if $c_s^j = c_{\min}^j$ and it is greater than 1 the more the cost c_s^j is greater than the minimum value $c_{\min}^j$.

$$c_s^{*j} = \frac{c_s^j}{c_{\min}^j} \tag{3.61}$$

4. for each samples, compute the total cost of the s^{th} sample c_s^{tot} as the weighted sum of all dimensionless costs c_s^{*j} referred to the same s^{th} sample:

$$c_s^{\text{tot}} = \sum_j c_s^{*j} \, w_s^j \tag{3.62}$$

where w_s^j represents the weight coefficient referred to dimensionless cost c_s^{*j}.
The weight w_s^j is a measure of the importance of minimizing the contribution of function f_j to the total cost c_s^{tot}.

Cost Functions and Weights

The minimization of the cost functions f_j at each time instant t_k guides progressively the end-effector to the objective and imposes the constraints on the joint trajectories.
The cost functions are divided in two different groups:

- cost functions $f_{j\theta}$ for the selection of the joint angles configuration $\boldsymbol{\theta}^{(k)}$
- cost functions $f_{j\dot{\theta}}$ for the selection of the joint velocities configuration $\dot{\boldsymbol{\theta}}^{(k)}$

Cost Functions to select Joint Angles Configuration $\boldsymbol{\theta}^{(k)}$

1. First Cost Function $f_{1\theta}$
   This is the function that guides the end-effector towards the desired final *position* for the capture of the target and it is defined as the distance between the position of the grasping point at time t_f, denoted by $\boldsymbol{r}_{gr}(t_f)$, and the position in the LVLH frame that the end-effector would have at capture time instant t_f considering the joint angle configuration $\boldsymbol{\theta}_s^{(k)}$ and the position and the orientation of the chaser base at time t_f, here denoted by $\boldsymbol{r}_{\text{EE}}(\boldsymbol{\theta}_s^{(k)}, \boldsymbol{T}_{\text{Cb}}(t_f))$. The first cost function is reported here below:
   $$f_{1\theta}(\boldsymbol{\theta}_s^{(k)}) = \|\boldsymbol{r}_{gr}(t_f) - \boldsymbol{r}_{\text{EE}}(\boldsymbol{\theta}_s^{(k)}, \boldsymbol{T}_{\text{Cb}}(t_f))\| \tag{3.63}$$
   The importance of reducing the position error of the end effector increases as the chaser approaches the target, then the weight $w_s^{1\theta}$ associated to function $f_{1\theta}$ is assumed to be proportional to the fraction of time passed until instant t_k:
   $$w_s^{1\theta} = \frac{t_k}{t_f} \quad \text{for} \quad s = 1, ..., N_S \tag{3.64}$$

2. Second Cost Function $f_{2\theta}$
   This function drives the end-effector towards the correct final *orientation*, denoted by $\boldsymbol{q}_{gr}(t_f)$ to capture the target. It is defined as the norm of the orientation error $\boldsymbol{e}_O$ introduced in Equation (2.40), between the orientation of the grasping point reference frame at time t_f, represented by $\boldsymbol{q}_{gr}(t_f)$ and the orientation that the end-effector would have at capture time instant t_f considering the joint angle configuration $\boldsymbol{\theta}_s^{(k)}$ and the position and the orientation of the chaser base at time t_f, here represented by $\boldsymbol{q}_{\text{EE}}(\boldsymbol{\theta}_s^{(k)}, \boldsymbol{T}_{\text{Cb}}(t_f))$:
   $$f_{2\theta}(\boldsymbol{\theta}_s^{(k)}) = \|\boldsymbol{e}_O\left(\boldsymbol{q}_{gr}(t_f), \boldsymbol{q}_{\text{EE}}(\boldsymbol{\theta}_s^{(k)}, \boldsymbol{T}_{\text{Cb}}(t_f))\right)\| \tag{3.65}$$
   As for the position error, the weight $w_s^{2\theta}$ on the orientation error increases as the

chaser approaches the target:

$$w_s^{2\theta} = \frac{t_k}{t_f} \quad \text{for} \quad s = 1, ..., N_S \tag{3.66}$$

3. Third Cost Function $f_{3\theta}$
   The cost function $f_{3\theta}$ is related to the *singularity avoidance* during the motion of the arm. As discussed in Section 3.1, the manipulability measure M has to be maximized to prevent the manipulator from reaching singular configurations. This can be also attained minimizing its inverse. Therefore, the third cost function $f_{3\theta}$ is defined as follow [78]:

$$f_{3\theta}(\boldsymbol{\theta}_s^{(k)}) = \frac{1}{1 + M(\boldsymbol{\theta}_s^{(k)}, \boldsymbol{T}_{\text{Cb}}(t_k))} \tag{3.67}$$

   where $M(\boldsymbol{\theta}_s^{(k)}, \boldsymbol{T}_{\text{Cb}}(t_k))$ is the manipulability measure defined in Equation (3.10) associated to the joint angle configuration $\boldsymbol{\theta}_s^{(k)}$ at time t_k.
   The weight $w_s^{3\theta}$ associated to the third cost function $f_{3\theta}$ is kept constant along the entire maneuver, since singularity avoidance is always considered an important parameter for the quality of the trajectory:

$$w_s^{3\theta} = 1 \quad \text{for} \quad s = 1, ..., N_S \tag{3.68}$$

4. Fouth cost function $f_{4\theta}$
   The cost function $f_{4\theta}$ is used to *avoid collisions* between the end-effector and the chaser base. To avoid any collisions, the distance between the end-effector and the CoM of the chaser base must be greater than a safe distance d_{safe}. This condition can be formally expressed as:

$$\|\boldsymbol{r}_{\text{EE}}(\boldsymbol{\theta}_s^{(k)}, \boldsymbol{T}_{\text{Cb}}(t_k)) - \boldsymbol{r}_{\text{Cb}}(t_k)\| \geq d_{\text{safe}} \tag{3.69}$$

   The safe distance d_{safe} represents the radius of a sphere that encloses all the edges of the chaser base and that delimits the keep-out zone for the end-effector. To impose the collision avoidance constraint, joint angle configurations corresponding to a position of the end-effector inside the keep-out sphere are much more penalized than others. The cost function $f_{4\theta}$ is defined in the same way for all the samples:

$$f_{4\theta}(\boldsymbol{\theta}_s^{(k)}) = 1 \quad \forall\, \boldsymbol{\theta}_s^{(k)} \tag{3.70}$$

and the weights $w_s^{4\theta}$ are defined as follow:

$$w_s^{4\theta}(\boldsymbol{\theta}_s^{(k)}) = \begin{cases} 0 & \text{if (3.69) is True} \\ 1000 & \text{if (3.69) is False} \end{cases} \tag{3.71}$$

Cost Functions to select Joint Velocities Vector $\dot{\boldsymbol{\theta}}^{(k)}$

1. First cost function $f_{1\dot{\theta}}$
   The cost function $f_{1\dot{\theta}}$ is used to select the joint velocities vector $\dot{\boldsymbol{\theta}}_s^{(k)}$ that minimizes the norm of the difference between the the linear speed of the grasping point at time instant t_f, denoted by $\dot{\boldsymbol{r}}_{gr}(t_f)$, and the linear speed that the end-effector would have at capture time instant t_f considering the joint angle configuration $\boldsymbol{\theta}^{(k)}$ and the joint velocities vector $\dot{\boldsymbol{\theta}}_s^{(k)}$, here denoted by $\dot{\boldsymbol{r}}_{\text{EE}}(\dot{\boldsymbol{\theta}}_s^{(k)}, \boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{X}}_{\text{Cb}}(t_f))$. The first cost function $f_{1\dot{\theta}}$ is defined as follow:

$$f_{1\dot{\theta}}(\dot{\boldsymbol{\theta}}_s^{(k)}) = \|\dot{\boldsymbol{r}}_{gr}(t_f) - \dot{\boldsymbol{r}}_{\text{EE}}(\dot{\boldsymbol{\theta}}_s^{(k)}, \boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{X}}_{\text{Cb}}(t_f))\| \tag{3.72}$$

   The weight $w_s^{1\dot{\theta}}$ is assumed to increase as the chaser approaches the target:

$$w_s^{1\dot{\theta}} = \frac{t_k}{t_f} \quad \text{for} \quad s = 1, ..., N_S \tag{3.73}$$

2. Second cost function $f_{2\dot{\theta}}$
   This cost function is used to select the joint velocities vector $\dot{\boldsymbol{\theta}}_s^{(k)}$ that minimizes the norm of the relative angular velocity between that of the target at t_f, denoted by $\boldsymbol{\omega}_{\text{Tb}}(t_f)$ and that one the end-effector would have at capture time instant t_f considering the joint angle configuration $\boldsymbol{\theta}^{(k)}$ and the joint velocities vector $\dot{\boldsymbol{\theta}}_s^{(k)}$, here denoted by $\boldsymbol{\omega}_{\text{EE}}(\dot{\boldsymbol{\theta}}_s^{(k)}, \boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{X}}_{\text{Cb}}(t_f))$. The second cost function is defined as:

$$f_{2\dot{\theta}}(\dot{\boldsymbol{\theta}}_s^{(k)}) = \|\boldsymbol{\omega}_{\text{Tb}}(t_f) - \boldsymbol{\omega}_{\text{EE}}(\dot{\boldsymbol{\theta}}_s^{(k)}, \boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{X}}_{\text{Cb}}(t_f))\| \tag{3.74}$$

   The weight $w_s^{2\dot{\theta}}$ associated to the cost function $f_{2\dot{\theta}}$ increases as the chaser approaches the target:

$$w_s^{1\dot{\theta}} = \frac{t_k}{t_f} \quad \text{for} \quad s = 1, ..., N_S \tag{3.75}$$

3. Third cost function $f_{3\dot{\theta}}$
   The third cost function $f_{3\dot{\theta}}$ is used to impose the constraints on joint angles excursions and on joint velocities, reported in Equations (3.6) and (3.7). In the time

interval $[\, t_{k-1} : t_k \,]$ a polynomial trajectory is computed and associated to each sample $\dot{\boldsymbol{\theta}}_s^{(k)}$. The generic s^{th} trajectory, denoted by $\boldsymbol{\theta}_s(t)$, has the following boundary conditions:

- boundary conditions at time instant t_{k-1}:

$$\boldsymbol{\theta}_s(t_{k-1}) = \boldsymbol{\theta}_{\text{current}}^{(k-1)} \tag{3.76}$$
$$\dot{\boldsymbol{\theta}}_s(t_{k-1}) = \dot{\boldsymbol{\theta}}_{\text{current}}^{(k-1)} \tag{3.77}$$
$$\ddot{\boldsymbol{\theta}}_s(t_{k-1}) = \mathbf{0}_{N,1} \tag{3.78}$$

- boundary conditions at time instant t_k

$$\boldsymbol{\theta}_s(t_k) = \boldsymbol{\theta}^{(k)} \tag{3.79}$$
$$\dot{\boldsymbol{\theta}}_s(t_k) = \dot{\boldsymbol{\theta}}_s^{(k)} \tag{3.80}$$
$$\ddot{\boldsymbol{\theta}}_s(t_k) = \mathbf{0}_{N,1} \tag{3.81}$$

The boundary conditions in Equations from (3.77) to (3.79) and in Equation (3.81) are the same for every trajectory $\boldsymbol{\theta}_s(t)$ for $s = 1, ..., N_S$, while boundary conditions in Equation (3.80) depends on the specific sample $\dot{\boldsymbol{\theta}}_s^{(k)}$. This clarifies how the trajectory $\boldsymbol{\theta}_s(t)$ is associated to the sample $\dot{\boldsymbol{\theta}}_s^{(k)}$. Then, those trajectories $\boldsymbol{\theta}_s(t)$ that violates constraints on joint angle excursion limits or maximum velocities are penalized compared to those that respect them. The cost function $f_{4\dot{\theta}}$ is defined as:

$$f_{4\dot{\theta}}(\dot{\boldsymbol{\theta}}_s^{(k)}) = 1 \quad \forall\, \dot{\boldsymbol{\theta}}_s^{(k)} \tag{3.82}$$

and the weights $w_s^{4\dot{\theta}}$ are defined as follow:

- $w_s^{4\dot{\theta}} = 0$ *if* $\boldsymbol{\theta}_s(t)$ satisfies (3.6) *and* (3.7) $\forall t \in [t_{k-1} : t_k]$;
- $w_s^{4\dot{\theta}} = 1000$ *if* $\exists t \in [t_{k-1} : t_k]$ s. t. $\boldsymbol{\theta}_s(t)$ does not satisfy (3.6) *or* (3.7).

Polynomial trajectory

A polynomial trajectory is computed to make the path of every joint angle $\theta_i(t)$ between two generic time instants t_{k-1} and t_k as smooth as possible. The boundary conditions for the design of the polynomial that describes the motion of each joint are known and they

are given by:

$$\begin{cases} \theta_i(t_{k-1}) = \theta_i^{(k-1)} \\ \dot{\theta}_i(t_{k-1}) = \dot{\theta}_i^{(k-1)} \\ \ddot{\theta}_i(t_{k-1}) = 0 \\ \theta_i(t_k) = \theta_i^{(k)} \\ \dot{\theta}_i(t_k) = \dot{\theta}_i^{(k)} \\ \ddot{\theta}_i(t_k) = 0 \end{cases} \tag{3.83}$$

In order to satisfy the six boundary conditions reported in Equation (3.83), a polynomial of order $N_{b.c.} - 1$ is required, where $N_{b.c.}$ is the number of boundary conditions imposed. The polynomial for the i^{th} joint is then defined as follow:

$$\theta_i(t) = a_5 t^5 + a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0 \tag{3.84}$$

where $t_{k-1} \leq t \leq t_k$ and a_j, with $j = 0, 1, ..., 5$ are the coefficients of the polynomial that have to be determined by solving the following system obtained imposing all the six conditions reported in Equation (3.83):

$$\begin{cases} \theta_i(t_{k-1}) = a_5 t_{k-1}^5 + a_4 t_{k-1}^4 + a_3 t_{k-1}^3 + a_2 t_{k-1}^2 + a_1 t_{k-1} + a_0 \\ \dot{\theta}_i(t_{k-1}) = 5a_5 t_{k-1}^4 + 4a_4 t_{k-1}^3 + 3a_3 t_{k-1}^2 + 2a_2 t_{k-1} + a_1 \\ \ddot{\theta}_i(t_{k-1}) = 20a_5 t_{k-1}^3 + 12a_4 t_{k-1}^2 + 6a_3 t_{k-1} + 2a_2 \\ \theta_i(t_k) = a_5 t_k^5 + a_4 t_k^4 + a_3 t_k^3 + a_2 t_k^2 + a_1 t_k + a_0 \\ \dot{\theta}_i(t_k) = 5a_5 t_k^4 + 4a_4 t_k^3 + 3a_3 t_k^2 + 2a_2 t_k + a_1 \\ \ddot{\theta}_i(t_k) = 20a_5 t_k^3 + 12a_4 t_k^2 + 6a_3 t_k + 2a_2 \end{cases} \tag{3.85}$$

The system in Equation (3.85) can be expressed in the matrix format as follow:

$$\begin{bmatrix} 1 & t_{k-1} & t_{k-1}^2 & t_{k-1}^3 & t_{k-1}^4 & t_{k-1}^5 \\ 0 & 1 & 2t_{k-1} & 3t_{k-1}^2 & 4t_{k-1}^3 & 5t_{k-1}^4 \\ 0 & 0 & 2 & 6t_{k-1} & 12t_{k-1}^2 & 20t_{k-1}^3 \\ 1 & t_k & t_k^2 & t_k^3 & t_k^4 & t_k^5 \\ 0 & 1 & 2t_k & 3t_k^2 & 4t_k^3 & 5t_k^4 \\ 0 & 0 & 2 & 6t_k & 12t_k^2 & 20t_k^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} \theta_i^{(k-1)} \\ \dot{\theta}_i^{(k-1)} \\ \ddot{\theta}_i^{(k-1)} \\ \theta_i^{(k)} \\ \dot{\theta}_i^{(k)} \\ \ddot{\theta}_i^{(k)} \end{bmatrix} \rightarrow \mathbf{P}\,\mathbf{a}_i = \mathbf{p}_i \tag{3.86}$$

Then, the vector of polynomial coefficients $\mathbf{a}_i$ related to i^{th} joint trajectory can be found by solving:

$$\mathbf{a}_i = \mathbf{P}^{-1}\,\mathbf{p}_i \tag{3.87}$$

Update Current Configuration

The cost minimizing joint angle configuration vector $\boldsymbol{\theta}^{(k)}$ and the cost minimizing joint velocities vector $\dot{\boldsymbol{\theta}}^{(k)}$ that have been selected among all the samples become now the new *"current"* configuration. They become the new starting point to repeat the procedure and move forward in time. The quantities denoted up to this point by (k) become those denoted by $(k-1)$ and the procedure is repeated:

$$\begin{aligned} k &\rightarrow k-1 \\ t_k &\rightarrow t_{k-1} \\ \boldsymbol{\theta}^{(k)} &\rightarrow \boldsymbol{\theta}_{\text{current}}^{(k-1)} \\ \dot{\boldsymbol{\theta}}^{(k)} &\rightarrow \dot{\boldsymbol{\theta}}_{\text{current}}^{(k-1)} \end{aligned} \tag{3.88}$$

Algorithm 3.1 Near-Optimal Incremental Sampling Based Path Planning algorithm

INPUT: t_{start}, t_f, Δt_{int}, N_S, $\boldsymbol{T}_{\text{Cb}}^{\text{LVLH}}(t)$, $\dot{\boldsymbol{X}}_{\text{Cb}}^{\text{LVLH}}(t)$, $\boldsymbol{T}_{gr}^{\text{LVLH}}(t)$, $\dot{\boldsymbol{X}}_{gr}^{\text{LVLH}}(t)$, $\boldsymbol{\theta}(t_0)$, $\dot{\boldsymbol{\theta}}(t_0)$
define $\boldsymbol{t} = [t_1, ..., t_{k-1}, t_k, t_{k+1}, ..., t_{N_{int}+1}]^T \leftarrow \{t_{start},\ t_f,\ \Delta t_{int}\}$
initialize $\boldsymbol{\theta}_{\text{current}} = \boldsymbol{\theta}(t_0)$; $\dot{\boldsymbol{\theta}}_{\text{current}} = \dot{\boldsymbol{\theta}}(t_0) \leftarrow \{\boldsymbol{\theta}(t_0),\ \dot{\boldsymbol{\theta}}(t_0)\}$
for each instant t_k, with $k = 2, ..., N_{int}$ **do**
  define sample limits $\theta_{i\ \text{min reach}}^{(k)}, \theta_{i\ \text{MAX reach}}^{(k)}$; $\dot{\theta}_{i\ \text{min reach}}^{(k)}, \dot{\theta}_{i\ \text{MAX reach}}^{(k)} \leftarrow \{\boldsymbol{\theta}_{\text{current}}, \dot{\boldsymbol{\theta}}_{\text{current}}\}$
  sample within limits: $\boldsymbol{\theta}_s^{(k)}$; $\dot{\boldsymbol{\theta}}_s^{(k)}$, with $s = 1, ..., N_S$
  for each sampled configuration $\boldsymbol{\theta}_s^{(k)}$, with $s = 1, ..., N_S$ **do**
    for each cost function $f_{j\theta}$ **do**
      evaluate cost of sample $\boldsymbol{\theta}_s^{(k)} \leftarrow \{\boldsymbol{T}_{\text{Cb}}^{\text{LVLH}}(t), \boldsymbol{T}_{gr}^{\text{LVLH}}(t)\}$
    end for
    compute total cost c_s^{tot} of sample $\boldsymbol{\theta}_s^{(k)}$
  end for
  select minimum total cost configuration: $\boldsymbol{\theta}^{(k)}$
  for each sampled velocities vector $\dot{\boldsymbol{\theta}}_s^{(k)}$, with $s = 1, ..., N_S$ **do**
    for each cost function $f_{j\dot{\theta}}$ **do**
      evaluate cost of sample $\dot{\boldsymbol{\theta}}_s^{(k)} \leftarrow \{\boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{X}}_{\text{Cb}}^{\text{LVLH}}(t), \dot{\boldsymbol{X}}_{gr}^{\text{LVLH}}(t)\}$
    end for
    evaluate cost of the trajectory between $[\boldsymbol{\theta}_{\text{current}}, \dot{\boldsymbol{\theta}}_{\text{current}}]$ and $[\boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{\theta}}_s^{(k)}]$
    compute total cost c_s^{tot} of sample $\dot{\boldsymbol{\theta}}_s^{(k)}$
  end for
  select minimum total cost velocities vector: $\dot{\boldsymbol{\theta}}^{(k)}$
  compute polynomial trajectory between $[\boldsymbol{\theta}_{\text{current}}, \dot{\boldsymbol{\theta}}_{\text{current}}]$ and $[\boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{\theta}}^{(k)}]$
  update $\boldsymbol{\theta}_{\text{current}} = \boldsymbol{\theta}^{(k)}$; $\dot{\boldsymbol{\theta}}_{\text{current}} = \dot{\boldsymbol{\theta}}^{(k)} \leftarrow \{\boldsymbol{\theta}^{(k)}, \dot{\boldsymbol{\theta}}^{(k)}\}$
end for
return joint trajectories $\boldsymbol{\theta}(t)$ between $[t_{start} : t_{N_{int}}]$

Solve Inverse Kinematics Problem

The trajectory of the joints is computed incrementally through sampling up to time instant $t_{N_{int}}$. At that instant, the end-effector should have been guided by the procedure described in Algorithm 3.1 to a position $\boldsymbol{r}_{\text{EE}}$ and to an orientation $\boldsymbol{q}_{\text{EE}}$ similar to those needed to capture the target. The same is true for the joint angle configuration $\boldsymbol{\theta}^{(N_{int})}$ at time instant $t_{N_{int}}$, that should be "close" to a final configuration $\boldsymbol{\theta}^{(N_{int}+1)}$ such that the

following conditions to capture the target are satisfied:

$$\boldsymbol{r}_{\text{EE}}(\boldsymbol{\theta}^{(N_{int}+1)},\, t_f) = \boldsymbol{r}_{gr}(t_f) \tag{3.89}$$

$$\boldsymbol{q}_{\text{EE}}(\boldsymbol{\theta}^{(N_{int}+1)},\, t_f) = \boldsymbol{q}_{gr}(t_f) \tag{3.90}$$

The configuration $\boldsymbol{\theta}^{(N_{int+1})}$ is computed by solving the inverse kinematics problem described in Section 2.4.3:

- the initial guess to start the iterative inverse kinematics algorithm is assumed to be the joint angle configuration vector $\boldsymbol{\theta}^{(N_{int})}$ at time instant $t_{N_{int}}$;
- the desired final position ${}^{d}\boldsymbol{r}_{\text{EE}}$ and the desired final orientation ${}^{d}\boldsymbol{r}_{\text{EE}}$ of the end-effector are set equal to the position and the orientation of the grasping point reference frame at the capture time instant t_f. In particular:

$$ {}^{d}\boldsymbol{r}_{\text{EE}} = \boldsymbol{r}_{gr}(t_f) \tag{3.91}$$

$$ {}^{d}\boldsymbol{r}_{\text{EE}} = \boldsymbol{q}_{gr}(t_f) \tag{3.92}$$

- the desired final linear and angular velocities of the end-effector, respectively ${}^{d}\dot{\boldsymbol{r}}_{\text{EE}}$ and ${}^{d}\boldsymbol{\omega}_{\text{EE}}$, are set equal to the linear velocity of the grasping point and to the angular speed of the target at time t_f, respectively denoted by $\dot{\boldsymbol{r}}_{gr}(t_f)$ and $\boldsymbol{\omega}_{\text{Tb}}(t_f)$. In particular:

$$ {}^{d}\dot{\boldsymbol{r}}_{\text{EE}} = \boldsymbol{r}_{gr}(t_f) \tag{3.93}$$

$$ {}^{d}\boldsymbol{\omega}_{\text{EE}} = \boldsymbol{\omega}_{\text{Tb}}(t_f) \tag{3.94}$$

Once found joint angle configuration vector $\boldsymbol{\theta}^{(N_{int}+1)}$, the joint velocities vector $\dot{\boldsymbol{\theta}}^{(N_{int}+1)}$ can be obtained by using Equation (2.29). Then a polynomial trajectory is computed between time instants $t_{N_{int}}$ and $t_{N_{int}+1}$ with boundary conditions respectively equals to $\boldsymbol{\theta}^{(N_{int})}$, $\dot{\boldsymbol{\theta}}^{(N_{int})}$ and $\boldsymbol{\theta}^{(N_{int}+1)}$, $\dot{\boldsymbol{\theta}}^{(N_{int}+1)}$.

4 | Results

In this Chapter the algorithm proposed in Section 3.2 is validated and its behaviour with respect a change of parameters is analyzed. Firstly the simulation setup is introduced, including a description of the model of the target body and the model of the free-flying manipulator system with its thrust and joint constraints. Simulation results are reported with the corresponding input parameter. Then the performance of the algorithm are tested with respect to a parameters change and critical observation are formulated.

4.1. Simulation Setup

In this section the parameters used in numerical simulations that generate the results reported in section 4.2 are summarized. The section is organized in two main parts:

1. the subsection 4.1.1 summarizes the parameters related to the models of the target body, of the chaser spacecraft and of the robotic manipulator that have been assumed.
2. the subsection 4.1.2 summarizes the input parameters of the proposed path planning algorithm introduced in section 3.1.

4.1.1. Target and Chaser Spacecraft Models

Target Body Model

For simulation purposes, the target body is assumed to be an uncontrolled small satellite with a cubic main body and equipped with two solar panels, as shown in Fig. 4.1 where the position and the orientation of the grasping point frame are also highlighted. The dimensions and the inertia properties of the target body, along with information about its orbit are reported in Tab.4.1 while the grasping point data are reported in Tab. 4.2.

Target Body	
Parameter	**Value**
orbit altitude $[km]$	400
main body mass m_{Tb} $[kg]$	100
solar panel mass m_{sp} $[kg]$	15
main body dimensions $[m]$	$1 \times 1 \times 1$
solar panel dimensions $[m]$	$0.01 \times 3 \times 2$
inertia matrix $\mathbb{J}$ $[kg\,m^2]$	$\begin{bmatrix} 236.67 & 0 & 0 \\ 0 & 26.67 & 0 \\ 0 & 0 & 226.67 \end{bmatrix}$

Table 4.1: Target Body orbit, dimensions and inertia properties

Grasping Point	
Parameter	**Value**
position vector $\boldsymbol{r}_{gr}^{\text{Tb}}$ $[m]$	$[0.5,\ -0.25,\ 0.5]^T$
orientation $\boldsymbol{R}_{gr}^{\text{Tb}}$	$\begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

Table 4.2: Grasping Point position and orientation with respect Target Body

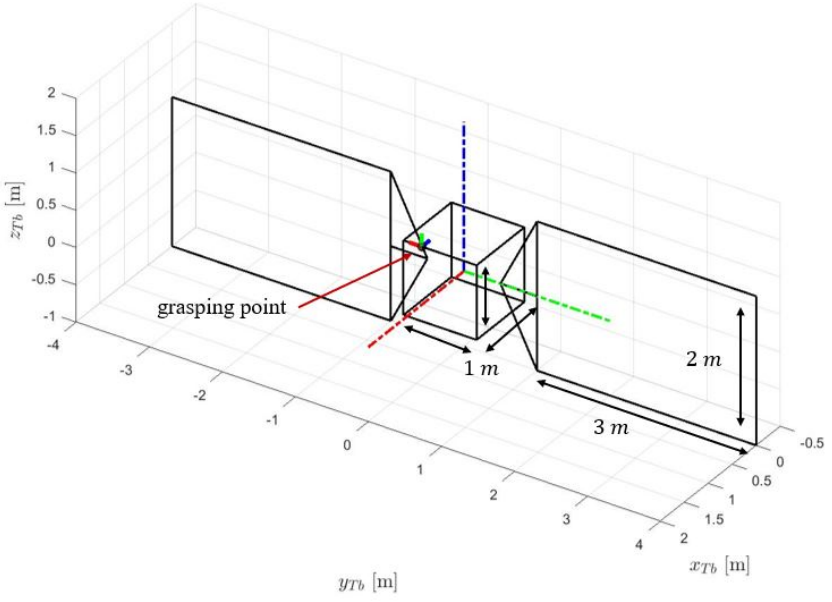


Figure 4.1: Target body and grasping point geometry

Chaser Spacecraft Model

The chaser consist of a base spacecraft equipped with a robotic manipulator. The chaser base is assumed to be a small cubic spacecraft provided with thrusters and an attitude control system to maneuver in LVLH frame. The main characteristics of the chaser base spacecraft are reported in Tab.4.3.
As depicted in Figure 4.2, the robotic manipulator and the navigation sensors are assumed to be mounted on the panel of the base spacecraft with the outgoing normal directed as $\hat{\boldsymbol{x}}_{Cb}$. The sensors boresight axis it is assumed to be aligned with $\hat{\boldsymbol{x}}_{Cb}$.

Chaser Spacecraft	
Parameter	**Value**
base mass m_c $[kg]$	500
base dimensions $[m]$	$2 \times 2 \times 2$
max. thrust T_{MAX} $[N]$	5
max. acceleration u_{MAX} $[m/s^2]$	0.01

Table 4.3: Chaser base main properties

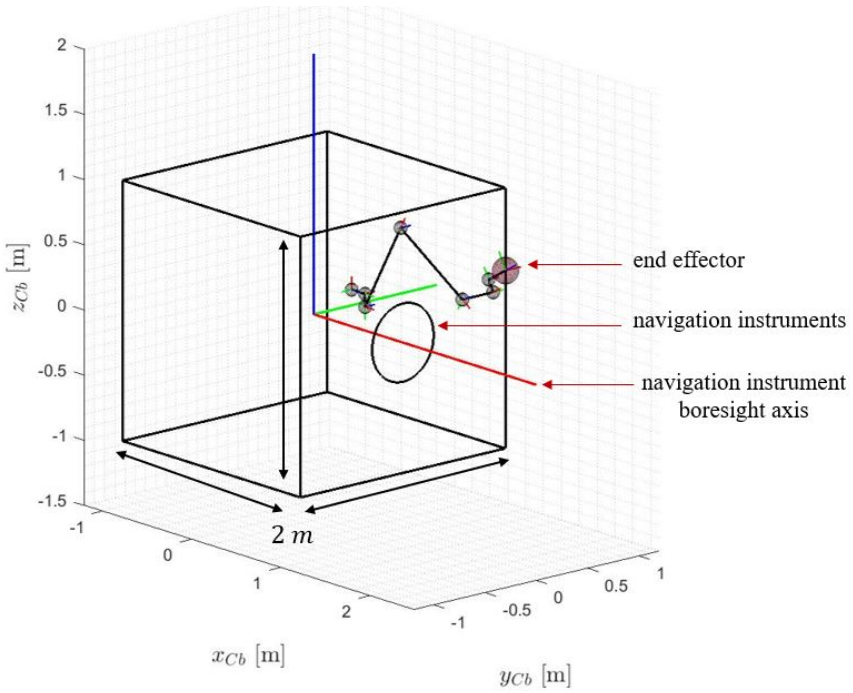


Figure 4.2: Chaser spacecraft and robotic manipulator

The robotic arm mounted on the spacecraft base is assumed to be a *redundant* manipulator characterized by $N = 7$ revolute joints providing it 7 DoF. It is assumed that the

manipulator is a SSRMS-type manipulator since the joints arrangement is similar to that of Canadarm2, with different customized dimensions. As discussed in Section 2.4.1, the joints are numbered from 1 to N starting from the one attached to the spacecraft base and they are arranged as follow [95]:

- three shoulder joints, in particular shoulder roll, yaw and pitch joints, whose variables are respectively angles θ_1, θ_2 and θ_3;
- one elbow pitch joint whose variable is angle θ_4;
- three wrist joints, in particular wrist pitch, roll and yaw joints, whose variables are respectively angles θ_5, θ_6 and θ_7;

The Denavit-Hartenberg parameters for the robotic manipulator used in the simulations are summarized in Table 4.4.

Joint	**DH Parameters**		
	$\boldsymbol{\alpha_i}$ [$^\circ$]	$\boldsymbol{c_i}$ $[m]$	$\boldsymbol{d_i}$ $[m]$
1	90	0	0.15
2	90	0	0.10
3	0	0.80	0
4	0	0.80	0
5	90	0	0.30
6	90	0	0.10
7	90	0	0.22

Table 4.4: DH parameters for customized SSMRS-type manipulator

The values of the position vector $\boldsymbol{b}_1^{Cb}$ and the rotation matrix $\boldsymbol{R}_{\mathcal{J}_1}^{\mathrm{Cb}}$ of the reference frame of joint 1 with respect to Chaser Body frame $\mathcal{B}_{Cb}$ are reported in Table 4.5, along with the position vector $\boldsymbol{b}_{\mathrm{EE}}^{\mathcal{J}_{N+1}}$ and the rotation matrix $\boldsymbol{R}_{\mathrm{EE}}^{\mathcal{J}_{N+1}}$ between the fictitious joint frame $\mathcal{J}_{N+1}$ and the end-effector frame. The end-effector has the same origin of the joint frame $\mathcal{J}_{N+1}$ but it is rotated so that its z-axis $\hat{\boldsymbol{z}}_{\mathrm{EE}}$ coincides with the direction of approach of the gripping device.

Parameter	Value
joint 1 position vector $\boldsymbol{b}_1^{Cb}$ $[m]$	$[1, -0.5, 0.5]^T$
joint 1 orientation $\boldsymbol{R}_{\mathcal{J}_1}^{\text{Cb}}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
end-effector position vector $\boldsymbol{b}_{\text{EE}}^{\mathcal{J}_{N+1}}$ $[m]$	$[0, 0, 0]^T$
end-effector orientation $\boldsymbol{R}_{\text{EE}}^{\mathcal{J}_{N+1}}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$

Table 4.5: Robotic manipulator additional parameters

The joint angles excursion limits are listed in Table 4.6. It is assumed that all the joints have equal maximum velocity and accelerations that are reported in Table 4.7.

Joint Angle	**min. angle** [$^\circ$]	**MAX. angle** [$^\circ$]
θ_1	-180	180
θ_2	0	180
θ_3	-90	90
θ_4	0	180
θ_5	-90	90
θ_6	-90	90
θ_7	-180	180

Table 4.6: Joint angles excursion limits

Parameter	Value
joints max. velocity $\dot{\theta}_{\text{MAX}}$ $[^\circ/s]$	10
joints max. acceleration $\ddot{\theta}_{\text{MAX}}$ $[^\circ/s^2]$	5

Table 4.7: Joints maximum velocity and acceleration

4.1.2. Path Planning Algorithm Parameters

LQR/APF Proposed Algorithm Input Parameters

In table 4.8 are reported the input parameters of the adaptive LQR/APF algorithm. The choice of the desired final position ${}^{d}\boldsymbol{r}$ must take into account the rotational motion of the target body and in particular the region of space in which the motion of grasping is bounded. Indeed ${}^{d}\boldsymbol{r}$ has to be selected so that the the grasping point is inside the manipulator workspace. The chaser is considered arrived to the final desired position when the distance from it is below the tolerance tol_d. The LQR update interval Δt_{LQR} is set equal to $10s$, a compromise between obtaining a short maneuver and limiting computational effort.
The coefficients k_p, k_h and k_{APF} reported in Table 4.8 has been obtained by hand-tuning and the update interval of the APF control component Δt_{APF} is assumed equal to $1s$ to be compatible with realistic instruments update frequency.

Input LQR/APF algorithm	
LQR	
Parameter	**Value**
desired final chaser CoM position ${}^{d}\boldsymbol{r}$ $[m]$	$[2.75,\ 0,\ 0]^T$
desired final chaser CoM speed ${}^{d}\dot{\boldsymbol{r}}$ $[m/s]$	$[0,\ 0,\ 0]^T$
tolerance on final position tol_d $[m]$	0.05
LQR gain update interval Δt_{LQR} $[s]$	10
APF	
Parameter	**Value**
APF proportional constant k_p	1650
APF width constant k_h	125
keep-out zone radius a_{ko} $[m]$	6
APF control coefficient k_{APF}	$5 \cdot 10^{-6}$
APF control update interval Δt_{APF} $[s]$	1

Table 4.8: Adaptive LQR/APF algorithm input parameters

NOISBPP Algorithm Input Parameters

The user-defined input parameter of the proposed Near-Optimal Incremental Sampling Based algorithm are reported in Table 4.9. The instant at which the robotic manipulator starts moving from its stowed condition is arbitrary and it is here fixed at 360 seconds before performing the capture. A safety factor has been used for both maximum joint velocity and maximum joint acceleration as reported in Equations (3.55) and (3.56) to

reduce the risk of constraints violations. The algorithm selects a cost minimizing waypoint every $\Delta t_{int} = 60s$ among $N_S = 50$ possible solutions. A critical analysis on the performance of the proposed algorithm with respect to these two parameters is conducted in section 4.3.2.

In Table 4.10 are reported the user-defined input parameters of the inverse kinematics algorithm discussed in section 2.4.3. The inverse kinematics algorithm is used to define precisely the joints configuration to reach the grasping point at t_f. The initial joints configuration guess to start the iterative inverse kinematics algorithm is set equal to the last waypoint configuration $\boldsymbol{\theta}^{(N_{int})}$ defined by NOISBPP algorithm.

Input NOISBPP algorithm	
Parameter	**Value**
robotic arm maneuver span $t_f - t_{start}$ $[s]$	360
time between sampling instants Δt_{int} $[s]$	60
number of samples N_S	50
joint angles discretization step $\delta\theta$ $[^\circ]$	1
joint velocities discretization step $\delta\dot{\theta}$ $[^\circ/s]$	1
joint velocity safety factor λ_{vel}	0.25
joint acceleration safety factor λ_{acc}	0.25

Table 4.9: Near-Optimal Incremental Sampling Based algorithm input parameters

Input Inverse Kinematics algorithm	
Parameter	**Value**
final maximum error norm tol_{IK}	$1 \cdot 10^{-4}$
maximum number of iterations k_{MAX}	10000
time interval tuning parameter dt $[s]$	0.01
position error gain matrix $\boldsymbol{K}_P$	$\begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 40 \end{bmatrix}$
orientation error gain matrix $\boldsymbol{K}_O$	$\begin{bmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 50 \end{bmatrix}$

Table 4.10: Inverse Kinematics algorithm input parameters

4.2. Simulation Results

In this section are reported the most significant results obtained with the proposed path planning approach. Firstly, the initial condition referred to the rotational motion of the target body, the chaser CoM initial position and the robotic manipulator initial configuration are reported in Table 4.11.
The target body is initially assumed to be aligned with the LVLH reference frame, rotating around its major inertia axis. The trajectory of the grasping point is bounded in a region of LVLH frame that is along the positive direction of the x-axis. Starting from the assumed initial position, to access this region and perform the capture the chaser has to steer the trajectory to avoid a collision with the target. In this way the APF algorithm is tested. The robotic manipulator is assumed to be stowed at the beginning of the approaching maneuver at time t_0 and it remains stowed until t_{start}.

Initial Conditions	
Target Body	
Parameter	**Value**
angular speed $\boldsymbol{\omega}_{\mathrm{Tb}}^{\mathrm{Tb}}(t_0)$ $[^\circ/s]$	$[2,\ 0,\ 1]^T$
attitude quaternion $\boldsymbol{q}_{\mathrm{Tb}}(t_0)$	$[0,\ 0,\ 0,\ 1]^T$
Chaser CoM	
Parameter	**Value**
CoM position $\boldsymbol{r}_{\mathrm{Cb}}(t_0)$ $[m]$	$[-30,\ -15,\ -15]^T$
CoM velocity $\dot{\boldsymbol{r}}_{\mathrm{Cb}}(t_0)$ $[m/s]$	$[0,\ 0,\ 0]^T$
Robotic Arm	
Parameter	**Value**
joint angles $\boldsymbol{\theta}(t_0)$ $[^\circ]$	$[180,\ 90,\ -90,\ 180,\ 90,\ -90,\ 0]^T$
joint velocities $\dot{\boldsymbol{\theta}}(t_0)$ $[^\circ/s]$	$[0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^T$

Table 4.11: Numerical simulation inititial conditions

The resulting trajectory of the end-effector in LVLH frame is shown with the trajectory of the chaser base CoM in Figure 4.3.

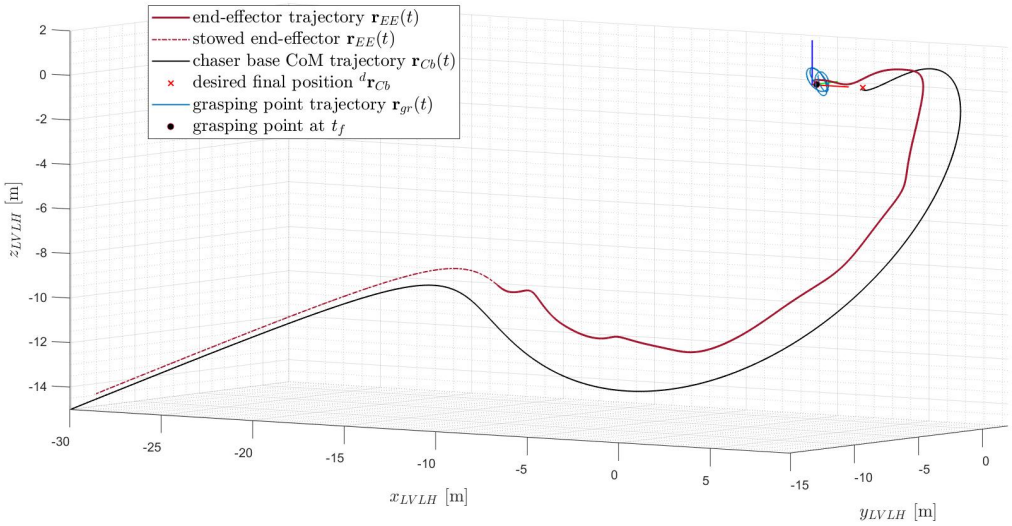


Figure 4.3: Planned trajectory of the end-effector position in LVLH frame

The chaser base moves in LVLH frame under the action of control acceleration $\mathbf{u}$, whose components in LVLH frame are reported in Figure 4.4.
As can be observed in Figure 4.3, the chaser is initially guided towards the desired final position on a straight trajectory by the predominant LQR component $\mathbf{u}_{\mathrm{LQR}}$ of the control acceleration. As the chaser approaches the target, the APF control component $\mathbf{u}_{\mathrm{APF}}$ becomes more and more significant, until it overcomes the LQR component and it curves the trajectory avoiding that the chaser enters in the keep-out surface defined around the target and collides with it. Then, the APF control component guides the chaser tangentially to the keep-out surface until the chaser position $\boldsymbol{r}_{\mathrm{Cb}}$ aligns with the desired final position ${}^{d}\boldsymbol{r}_{\mathrm{Cb}}$: at this point the APF control components is null and the chaser finally gets to the desired final position with zero relative velocity thanks to the LQR control component. The chattering behaviour of the control acceleration components shown in Figure 4.4 is due to the updating at regular time intervals Δt_{LQR} of LQR control gain matrix $\mathbf{K}_{\mathrm{LQR}}$. In Figure 4.4 can be also observed that all the control acceleration components respect the constraint on maximum control acceleration available.

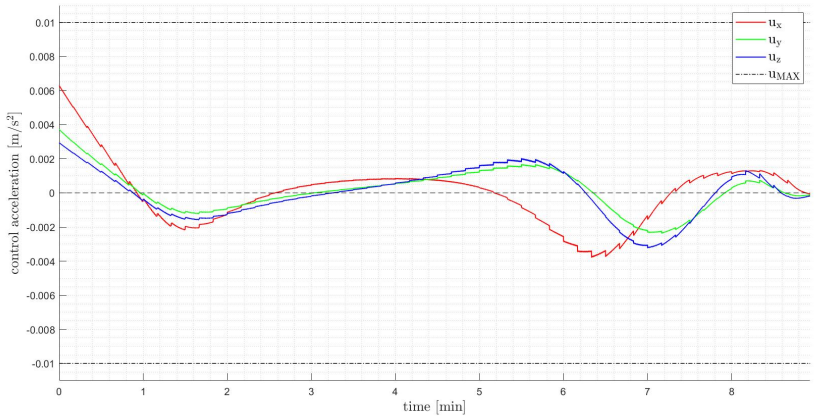


Figure 4.4: Chaser base control acceleration components in LVLH frame

In Table 4.12 are reported the duration and the total fuel cost index of the maneuver.

Parameter	**Results**
maneuver duration t_f	8min 40s
maneuver total fuel cost index $C_{f_{tot}}$	1.14 m/s

Table 4.12: Planned maneuver duration and total fuel consumption

The figure 4.5 shows the planned trajectory of the end-effector in Chaser base Body frame and depicts also the final configuration of the robotic manipulator at the instant of the capture t_f. It can be observed how the robotic arm moves with respect to the chaser base: its movements allow to adjust the position and the orientation of the end-effector for the final capture, avoiding singular configuration as shown in Figure 4.6.

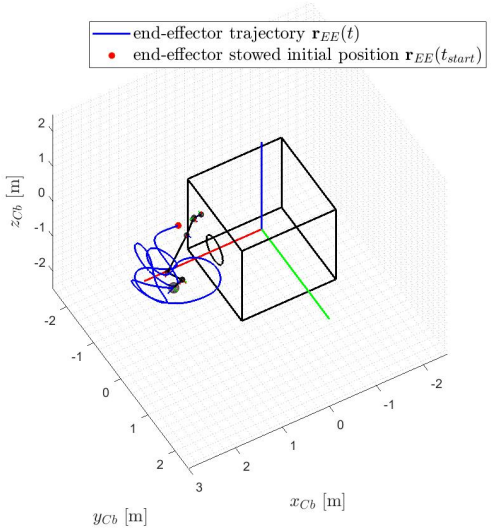


Figure 4.5: Planned trajectory of the end-effector position in Chaser base Body frame

In Figure 4.6 is shown the behaviour of the manipulability measure over time. It can be observed that the manipulability index is equal to zero only at initial instant t_{start} of the maneuver, when the arm is stowed. This means that the algorithm can provide a solution without kinematic singularities.

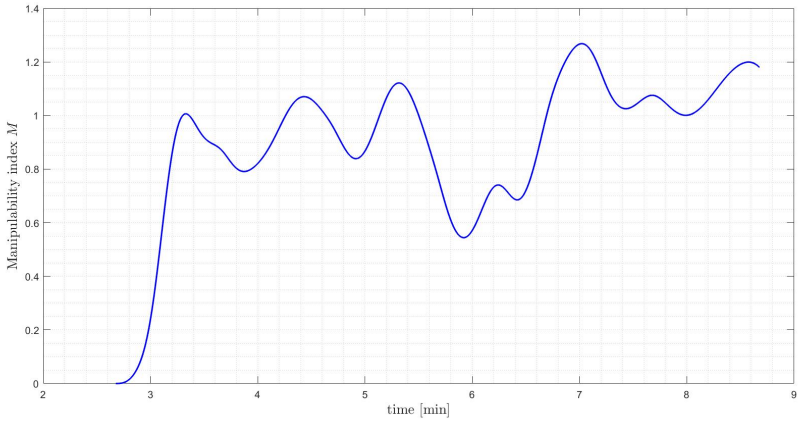


Figure 4.6: Manipulability index M time history

The planned time histories of the joints, from which the trajectory of the end effector originates, are shown in detail in Figures from 4.7 to 4.9. In particular Figure 4.7 groups the trajectories of the first three shoulder joints, Figure 4.8 depicts the trajectories of the elbow joint and finally Figure 4.9 shows the time histories of the last three wrist joints.

All the trajectories here reported respect constraint on joint angles, joint velocities and joint acceleration.

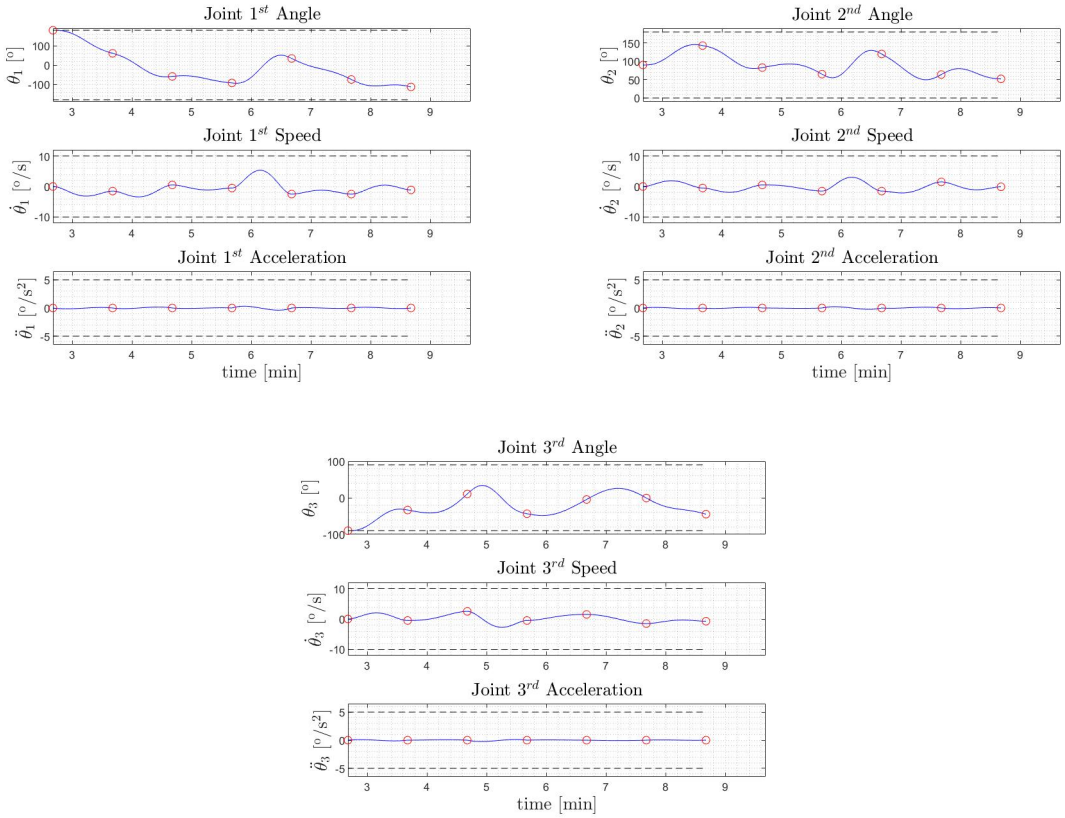


Figure 4.7: Shoulder joints $i = 1, 2, 3$ time histories

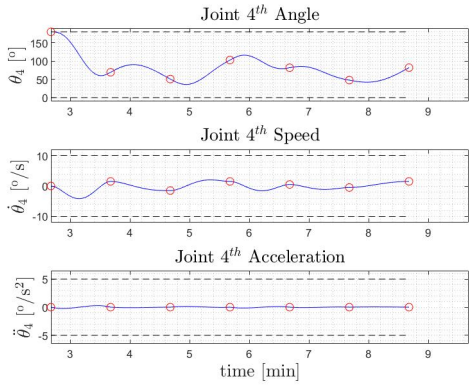


Figure 4.8: Elbow joint $i = 4$ time history

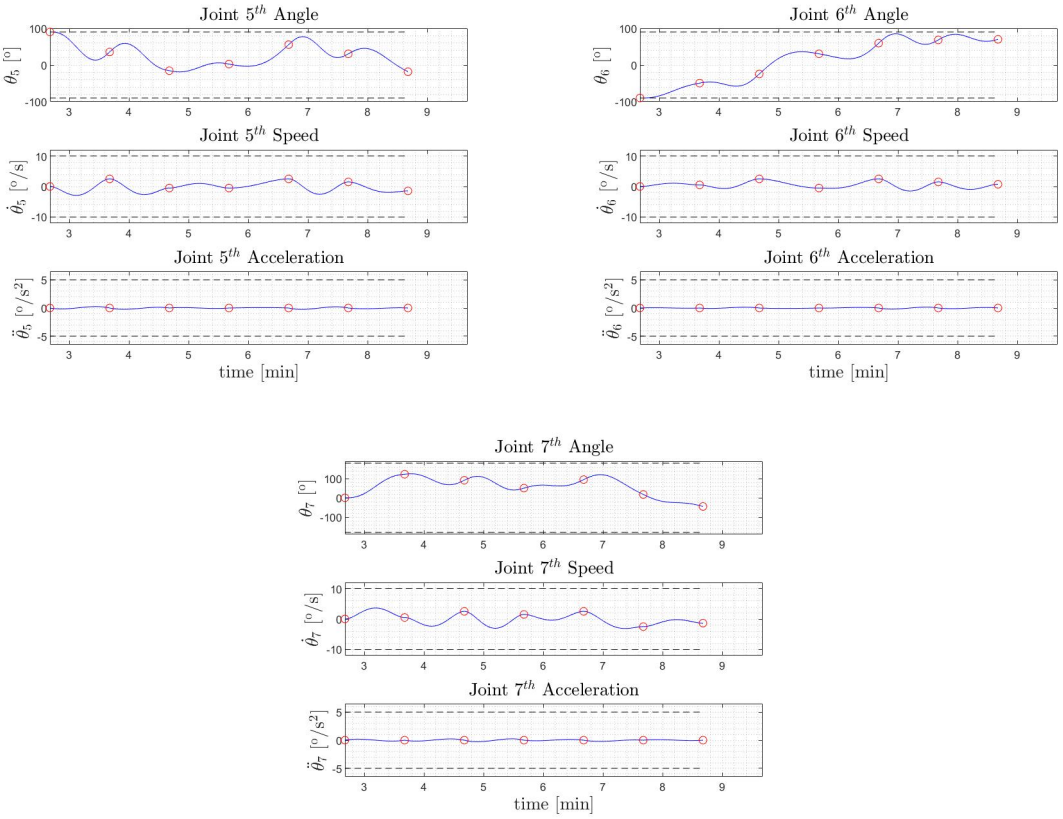


Figure 4.9: Wrist joints $i = 5, 6, 7$ time histories

4.3. Analysis on Variation of Parameters

This section is is dedicated to analyzing the results returned by the path planning approach proposed with respect to a variation of the parameters. In particular:

- for what concerns the LQR/APF algorithm, the effects of a variation of the update LQR gain interval Δt_{LQR} on maneuver time and fuel consumption are discussed;
- for what concerns the NOISBPP algorithm, the effects of a variation of the interval between two consecutive samples Δt_{int} and of number of samples at each time instant t_k are discussed in terms of feasibility of the trajectory

4.3.1. LQR/APF Proposed Algorithm

Several numerical simulations are performed keeping constant the initial conditions and all input parameters, except for the update LQR gain time inteval Δt_{LQR} that is different for each simulation. The results of the simulations are reported in Figures 4.10 and

4.11, where the solutions related to a zero updating interval $\Delta t_{\text{LQR}} = 0s$ is plotted in correspondence of the final time of the maneuver. The following observations can be formulated:

- Figure 4.10 shows a clear pattern between the update interval Δt_{LQR} and the time t_f to reach the desired final position: higher the update frequency, shorter is the time to reach the target. Updating the LQR gain has clear advantages in terms of time with respect a solution with constant gain;
- Figure 4.11 doesn't show clear advantages of updating or not LQR gain in terms of fuel consumption.

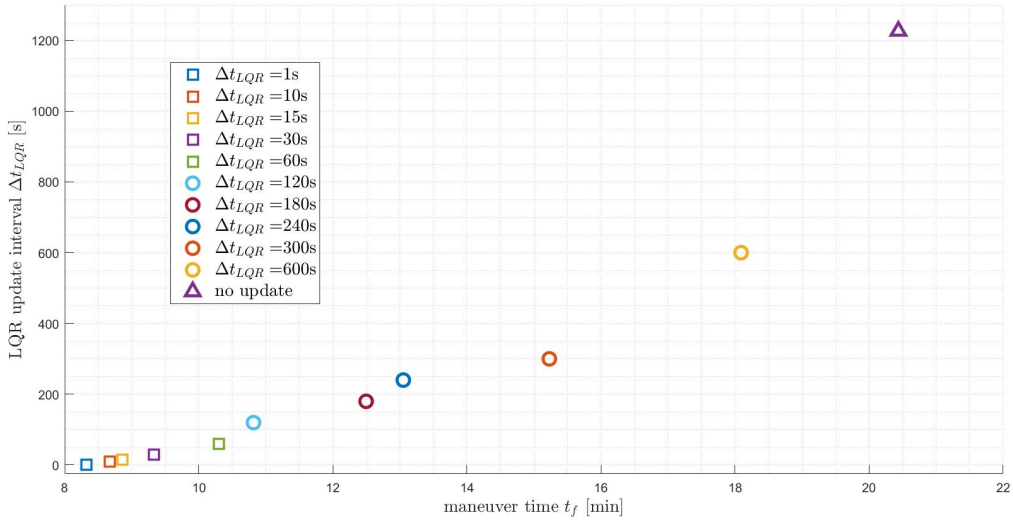


Figure 4.10: Maneuver duration t_f varying LQR gain update interval Δt_{LQR}

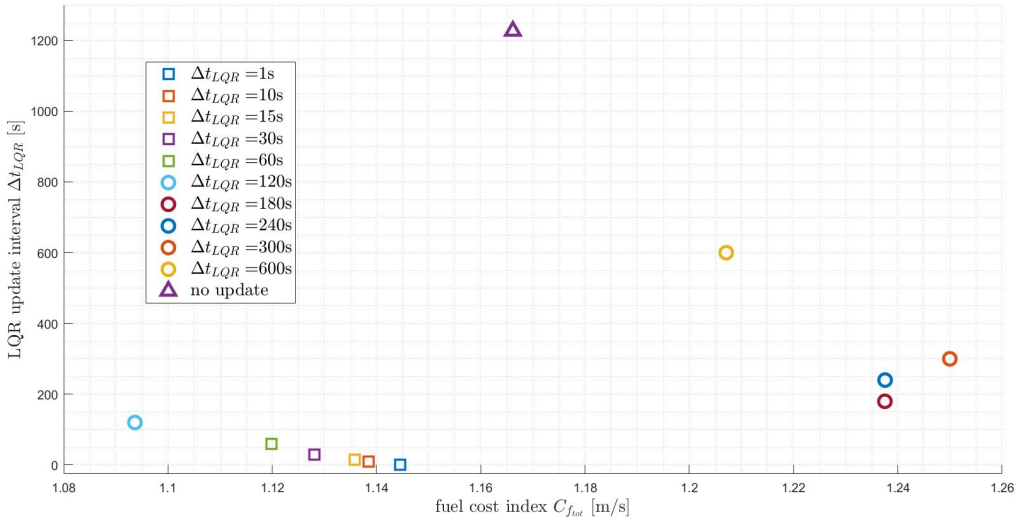


Figure 4.11: Maneuver fuel consumption $C_{f_{tot}}$ varying LQR gain update interval Δt_{LQR}

Summarizing, updating LQR gain increases the computational effort of the algorithm and has negligible benefits in terms of fuel consumption but offers great advantages in terms of maneuver duration: time to reach the target is a crucial parameter since safe regions in the close proximity of the target remain safe for a limited time window due to the tumbling motion of the target.

4.3.2. Near-Optimal Incremental Sampling Based Algorithm

The results obtained with the proposed Near-Optimal Incremental Sampling Based Algorithm are analyzed with respect to two main parameters:

- the *duration of each sub-interval* Δt_{int}, that represents the time interval between two consecutive waypoints;
- the *number of samples* N_S taken within selecting each waypoint.

The objective is characterizing the performance of the algorithm with respect to this two parameters in terms of violations of joint angles and velocities constraints reported in Equations (3.6) and (3.7). Three different cases are considered:

1. *no violations*: in this case are included all the simulations without any violation of the constraints on joint angles and joint speeds. In other terms:

$$\text{no violations: } \forall t \in [t_{start} : t_f] \quad (3.6) \text{ and } (3.7) \text{ are satisfied} \tag{4.1}$$

2. *violations in* $[t_{start} : t_{N_{int}}]$: in this case are included all the simulations with at least one violations of constraints on joint angles or joint speeds in time interval $[t_{start} : t_{N_{int}}]$. These violations are addressed to a wrong behaviour of the proposed algorithm, since it fails to find a trajectory within imposed constraint. They are classified as *serious* errors. This case can be formally expressed as:

$$\text{violations } [t_{start} : t_{N_{int}}] : \exists t \in [t_{start} : t_{N_{int}}] \text{ (3.6) or (3.7) are not satisfied} \tag{4.2}$$

3. *violations in* $[t_{N_{int}} : t_f]$: in this case are included all the simulations with at least one violations of constraints on joint angles or joint speeds in the last time interval $[t_{N_{int}} : t_f]$. These violations are addressed to the method used to solve the inverse kinematics and to the order of polynomial used to compute the trajectory between the waypoint at $t_{N_{int}}$ and final point at t_f. They are considered *minor* errors since no constraints are imposed in this time interval. This case can be formally expressed as:

$$\begin{aligned} &\text{if (4.2) is not verified,} \\ &\text{violations } [t_{N_{int}} : t_f] : \exists t \in [t_{N_{int}} : t_f] \text{ s.t. (3.6) or (3.7) are not satisfied} \end{aligned} \tag{4.3}$$

Several simulations are performed considering different durations of the interval Δt_{int} and constant number of samples N_S. Each simulation consists of $2N$ time histories: N time histories for the joint angles and their correspondent N time histories for the joint velocities. The results of the simulations are shown in Table 4.13.

	$\boldsymbol{N_S = 50}$		
	$\Delta t_{int} = 15s$	$\Delta t_{int} = 30s$	$\Delta t_{int} = 60s$
N°of simulations	30 (100%)	30 (100%)	30 (100%)
no violations	1 (3.33%)	12 (40%)	8 (26.67%)
violations $[t_{start} : t_{N_{int}}]$	2 (6.67%)	3 (10%)	5 (16.67%)
violations $[t_{N_{int}} : t_f]$	27 (90%)	15 (50%)	17 (56.66%)

Table 4.13: Simulations results with $N_S = 50$

From data reported in 4.13 results that, for a low number of samples N_S, an intermediate interval returns feasible results with higher probability: by its derivation, the sampling interval can be considered proportional to the time interval Δt_{int}. Then, a low number of samples can be enough to efficiently explore a not too wide sampling interval. However,

there seem to be no advantages in a smaller sampling interval (correspondent to the shorter Δt_{int} considered) that can be better explored with the same number of samples. A smaller sampling interval does not allow to explore the entire configuration space and the arm remains stuck in a non-optimal configuration to perform the capture. This behaviour can be observed in Figure 4.12a, in contrast with a solution obtained with a longer time interval Δt_{int} that allows a better exploration of the whole configuration space and wider movements to reach an optimal configuration for the capture.
To keep track of the amount of constraints violations, each time history of the single joint angle and of the single joint velocity has been classified with the same criteria mentioned above. The results referred to joint angles time histories are shown in Table 4.14 and those referred to joint velocities time histories are reported in Table 4.15.

	$\boldsymbol{N_S = 50}$		
	$\Delta t_{int} = 15s$	$\Delta t_{int} = 30s$	$\Delta t_{int} = 60s$
N°of time histories	210 (100%)	210 (100%)	210 (100%)
no violations	170 (80.95%)	187 (89.05%)	176 (83.81%)
violations $[t_{start} : t_{N_{int}}]$	2 (0.95%)	3 (1.43%)	6 (2.86%)
violations $[t_{N_{int}} : t_f]$	38 (18.1%)	20 (9.52%)	28 (13.3%)

Table 4.14: Joint angles time histories classification with $N_S = 50$

	$\boldsymbol{N_S = 50}$		
	$\Delta t_{int} = 15s$	$\Delta t_{int} = 30s$	$\Delta t_{int} = 60s$
N°of time histories	210 (100%)	210 (100%)	210 (100%)
no violations	155 (73.81%)	194 (92.38%)	210 (100%)
violations $[t_{start} : t_{N_{int}}]$	0 (0.0%)	0 (0.0%)	0 (0.0%)
violations $[t_{N_{int}} : t_f]$	55 (26.19%)	16 (7.62%)	0 (0.0%)

Table 4.15: Joint velocities time histories classification with $N_S = 50$

Comparing results reported in Table 4.13 with those in Tables 4.14 and 4.15 it can be observed that:

- *serious errors* only occur in joint angles time histories and not in joint velocities time histories;
- if a *serious error* occurs in a simulation, it occurs in most the cases in only one of the N joint angles time histories;

- increasing the interval Δt_{int} seems to reduce the number of errors occurring in joint velocities time histories, but not those occurring to the joint angles time histories.

Another set of simulations was performed by varying the number of samples N_S taken at each instant t_k to report how this parameter affects the solutions. The number of samples N_S has been doubled with respect to previous simulations and the results are classified in Tables 4.16, 4.17 and 4.18.

	$\boldsymbol{N_S = 100}$		
	$\Delta t_{int} = 15s$	$\Delta t_{int} = 30s$	$\Delta t_{int} = 60s$
N°of simulations	30 (100%)	30 (100%)	30 (100%)
no violations	1 (3.33%)	11 (36.67%)	11 (36.67%)
violations $[t_{start} : t_{N_{int}}]$	1 (3.33%)	2 (6.67%)	7 (23.33%)
violations $[t_{N_{int}} : t_f]$	28 (93.34%)	17 (56.66%)	12 (40%)

Table 4.16: Simulations results with $N_S = 100$

	$\boldsymbol{N_S = 100}$		
	$\Delta t_{int} = 15s$	$\Delta t_{int} = 30s$	$\Delta t_{int} = 60s$
N°of time histories	210 (100%)	210 (100%)	210 (100%)
no violations	169 (80.48%)	187 (89.05%)	180 (85.71%)
violations $[t_{start} : t_{N_{int}}]$	1 (0.48%)	2 (0.95%)	7 (3.33%)
violations $[t_{N_{int}} : t_f]$	40 (19.04%)	21 (10%)	23 (10.96%)

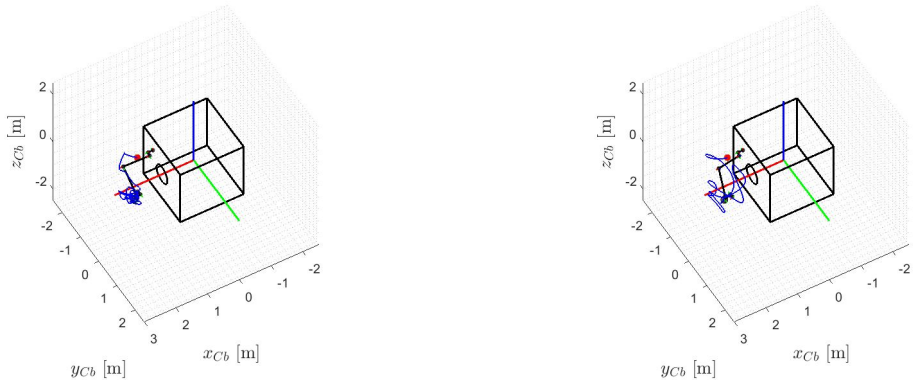
Table 4.17: Joint angles time histories classification with $N_S = 100$

	$\boldsymbol{N_S = 100}$		
	$\Delta t_{int} = 15s$	$\Delta t_{int} = 30s$	$\Delta t_{int} = 60s$
N°of time histories	210 (100%)	210 (100%)	210 (100%)
no violations	152 (72.38%)	196 (93.33%)	208 (99.05%)
violations $[t_{start} : t_{N_{int}}]$	0 (0.0%)	0 (0.0%)	0 (0.0%)
violations $[t_{N_{int}} : t_f]$	58 (27.62%)	14 (6.67%)	2 (0.95%)

Table 4.18: Joint velocities time histories classification with $N_S = 100$

Comparing data reported in Table 4.13 with those reported it can be observed that an higher number of samples N_S advantages simulations with longer intervals Δt_{int} and it

has no effects on simulations with intermediate and short intervals: the percentage of feasible simulations increases for $\Delta t_{int} = 60s$ and remains almost equal for $\Delta t_{int} = 30s$ and $\Delta t_{int} = 15s$. This result is expected. As discussed above, a small time interval Δt_{int} drives to a small sampling interval that can be explored with a low number of samples. On the other hand, a longer time interval Δt_{int} drives to a wider sampling interval that can be better explored with an higher number of samples.



(a) $\Delta t_{int} = 15s$ (b) $\Delta t_{int} = 60s$

Figure 4.12: End-effector trajectory varying interval Δt_{int}

5 | Conclusions

This thesis work proposes a flexible strategy to plan the motion of a free-flying space manipulator system and capture a target body. The proposed strategy consists in a reworking of methods already proposed in literature such as the adaptive LQR/APF algorithm and the Line-Of-Sight reference frames and in a new incremental sampling-based algorithm for planning the trajectory of a robotic manipulator on a free-flying base in the configuration space. This work has shown, through numerical simulations, the validity of the strategy and of the new algorithm, that, under some hypothesis on the motion of the target body and on the operative modes of the space robot, are able to deal with common constraints and aspects of an on-orbit servicing mission with a robotic manipulator, such as collision avoidance, limited thrust, bounded joint displacements and kinematic-singularities avoidance. The performances of the algorithm has been discussed as well as its failures. Future developments would aim to solve failures and reduce the number of hypothesis on which the work is based.

5.1. Future Developments

Some improvements can be made in the near future to improve the quality of the solution:

- In the proposed strategy the chaser has access to the grasping point from a fixed position in LVLH frame. This is possible only for target bodies tumbling at limited angular speeds. If the target rotate faster, the chaser must synchronize with the motion of the target, tracking a desired position fixed with respect to the target. This can be attempt through an LQR-tracking controller.
- Dynamics coupling between the chaser base and the robotic arm has not been considered. A new cost function that minimizes this quantity, based for instance on Reaction Null Space can be added.
- an improved algorithm for the solution of the inverse kinematics can be implemented to obtain more feasible solutions

Bibliography

[1] R. O. Ambrose, H. Aldridge, R. S. Askew, R. R. Burridge, W. Bluethmann, M. Diftler, C. Lovchik, D. Magruder, and F. Rehnmark. Robonaut: Nasa's space humanoid. *IEEE Intelligent Systems and Their Applications*, 15(4):57–63, 2000.

[2] J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, et al. The oos-sim: An on-ground simulation facility for on-orbit servicing robotic operations. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2854–2860. IEEE, 2015.

[3] I. Y. Bar-Itzhack. New method for extracting the quaternion from a rotation matrix. *Journal of guidance, control, and dynamics*, 23(6):1085–1087, 2000.

[4] J. R. Benevides and V. Grassi. Autonomous path planning of free-floating manipulators using rrt-based algorithms. In *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, pages 139–144. IEEE, 2015.

[5] R. Bevilacqua, T. Lehmann, and M. Romano. Development and experimentation of lqr/apf guidance and control for autonomous proximity maneuvers of multiple spacecraft. *Acta Astronautica*, 68(7-8):1260–1275, 2011.

[6] R. Biesbroek. *Active Debris Removal in Space: How to Clean the Earth's Environment from Space Debris*. Create Space, 2015.

[7] R. Biesbroek, L. Innocenti, A. Wolahan, and S. M. Serrano. e. deorbit-esa's active debris removal mission. In *Proceedings of the 7th European Conference on Space Debris*, volume 10. ESA Space Debris Office, 2017.

[8] C. Bonnal, J.-M. Ruault, and M.-C. Desjean. Active debris removal: Recent progress and current trends. *Acta Astronautica*, 85:51–60, 2013.

[9] L. Cao and B. Xiao. Suboptimal obstacles avoidance control of spacecraft ren-

dezvous. *Transactions of the Institute of Measurement and Control*, 44(1):40–49, 2022.

[10] J. J. Craig. *Introduction to robotics: mechanics and control*. Pearson Educacion, 2005.

[11] H. Cruijssen, M. Ellenbroek, M. Henderson, H. Petersen, P. Verzijden, and M. Visser. The european robotic arm: a high-performance mechanism finally on its way to space. In *The 42nd Aerospace Mechanism Symposium*, 2014.

[12] N. J. Currie and B. Peacock. International space station robotic systems operations-a human factors perspective. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 46, pages 26–30. SAGE Publications Sage CA: Los Angeles, CA, 2002.

[13] H. Curtis. *Orbital Mechanics for Engineering Students: Revised Reprint*. Butterworth-Heinemann, 2020.

[14] F. Cusumano, R. Lampariello, and G. Hirzinger. Development of tele-operation control for a free-floating robot during the grasping of a tumbling target. In *International Conference on Intelligent Manipulation and Grasping*, 2004.

[15] N. Davinic, A. Arkus, S. Chappie, and J. Greenberg. Cost-benefit analysis of on-orbit satellite servicing. *Journal of Reducing Space Mission Cost*, 1(1):27–52, 1998.

[16] T. Debus and S. Dougherty. Overview and performance of the front-end robotics enabling near-term demonstration (frend) robotic arm. In *AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference*, page 1870, 2009.

[17] A. Del Prete. Joint position and velocity bounds in discrete-time acceleration/torque control of robot manipulators. *IEEE Robotics and Automation Letters*, 3(1):281–288, 2017.

[18] M. A. Diftler, J. Mehling, M. E. Abdallah, N. A. Radford, L. B. Bridgwater, A. M. Sanders, R. S. Askew, D. M. Linn, J. D. Yamokoski, F. Permenter, et al. Robonaut 2-the first humanoid robot in space. In *2011 IEEE international conference on robotics and automation*, pages 2178–2183. IEEE, 2011.

[19] DLR. Older closed space robotics missions: Tecsas/deos. URL `https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3827/5969_read-8759/`.

[20] W. R. Doggett, J. T. Dorsey, T. C. Jones, and B. King. Development of a tendon-

actuated lightweight in-space manipulator (talisman). In *The 42nd Aerospace Mechanism Symposium*, 2014.

[21] H. Dong, Q. Hu, and M. R. Akella. Safety control for spacecraft autonomous rendezvous and docking under motion constraints. *Journal of Guidance, Control, and Dynamics*, 40(7):1680–1692, 2017.

[22] S. Dubowsky and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *IEEE Transactions on robotics and automation*, 9(5):531–543, 1993.

[23] S. Dubowsky and M. Torres. Path planning for space manipulators to minimize the use of attitude control fuel. In *Proc. of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (I-SAIRAS)*, 1990.

[24] S. Dubowsky and M. A. Torres. Path planning for space manipulators to minimize spacecraft attitude disturbances. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 2522–2528. Citeseer, 1991.

[25] A. Ellery, J. Kreisel, and B. Sommer. The case for robotic on-orbit servicing of spacecraft: Spacecraft reliability is a myth. *Acta Astronautica*, 63(5-6):632–648, 2008.

[26] ESA. Space debris by the numbers, 2022. URL `https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers`.

[27] W. Fehse. *Automated rendezvous and docking of spacecraft*, volume 16. Cambridge university press, 2003.

[28] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in aerospace sciences*, 68:1–26, 2014.

[29] R. B. Friend. Orbital express program summary and mission overview. In *Sensors and Systems for space applications II*, volume 6958, pages 11–21. SPIE, 2008.

[30] C. Froissart and P. Mechler. On-line polynomial path planning in cartesian space for robot manipulators. *Robotica*, 11(3):245–251, 1993.

[31] A. García, A. Lamb, A. Sleptsov, C. Moreno, M. Victorova, N. Glazkova, and V. Shteyngardt. Post-iss plans: What should be done? *REACH*, 1:63–73, 2016.

[32] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Rotex-the first remotely controlled robot in space. In *Proceedings of the 1994 IEEE international conference on robotics and automation*, pages 2604–2611. IEEE, 1994.

[33] R. P. Hoyt. Spiderfab: An architecture for self-fabricating space systems. In *AIAA Space 2013 conference and exposition*, page 5509, 2013.

[34] Q. Hu, Y. Liu, and Y. Zhang. Control of non-cooperative spacecraft in final phase proximity operations under input constraints. *Control Engineering Practice*, 87: 83–96, 2019.

[35] P. Huang, K. Chen, and Y. Xu. Optimal path planning for minimizing disturbance of space robot. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6. IEEE, 2006.

[36] Y. K. Hwang, N. Ahuja, et al. A potential field approach to path planning. *IEEE transactions on robotics and automation*, 8(1):23–32, 1992.

[37] S. Jacobsen, C. Lee, C. Zhu, and S. Dubowsky. Planning of safe kinematic trajectories for free flying robots approaching an uncontrolled spinning satellite. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 36533, pages 1145–1151. American Society of Mechanical Engineers, 2002.

[38] S. Jaekel, R. Lampariello, G. Panin, M. Sagardia, B. Brunner, O. Porges, E. Kraemer, M. Wieser, R. Haarmann, and R. Biesbroek. Robotic capture and de-orbit of a heavy, uncooperative and tumbling target in low earth orbit. 2015.

[39] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986.

[40] K. Kumar, N. Ortiz Gómez, M. Jankovic, J. Romero Martín, F. Topputo, S. Walker, F. Kirchner, and M. Vasile. Agora: Mission to demonstrate technologies to actively remove ariane rocket bodies. In *Proceedings of the International Astronautical Congress, Jerusalem, Israel*, volume 6, page 6, 2015.

[41] R. Lampariello. Motion planning for the on-orbit grasping of a non-cooperative target satellite with collision avoidance. *i-SAIRAS 2010*, 2010.

[42] R. Lampariello and G. Hirzinger. Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5652–5659. IEEE, 2013.

[43] S. M. LaValle. *Planning algorithms*. Cambridge university press, 2006.

[44] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and

prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan. *Algorithmic and Computational Robotics*, pages 303–307, 2001.

[45] G. Leisman, A. Wallen, S. Kramer, and W. Murdock. Analysis and preliminary design of on-orbit servicing architectures for the gps constellation. In *Space Technology Conference and Exposition*, page 4425, 1999.

[46] W.-J. Li, D.-Y. Cheng, X.-G. Liu, Y.-B. Wang, W.-H. Shi, Z.-X. Tang, F. Gao, F.-M. Zeng, H.-Y. Chai, W.-B. Luo, et al. On-orbit service (oos) of spacecraft: A review of engineering developments. *Progress in Aerospace Sciences*, 108:32–120, 2019.

[47] N. Y. Lii, D. Leidner, A. Schiele, P. Birkenkampf, R. Bayer, B. Pleintinger, A. Meissner, and A. Balzer. Simulating an extraterrestrial environment for robotic space exploration: The meteron supvis-justin telerobotic experiment and the solex proving ground. 2015.

[48] N. Y.-S. Lii, C. Riecke, D. Leidner, S. Schätzle, P. Schmaus, B. Weber, T. Krueger, M. Stelzer, A. Wedler, and G. Grunwald. The robot as an avatar or co-worker? an investigation of the different teleoperation modalities through the kontur-2 and meteron supvis justin space telerobotic missions. In *Proceedings of the International Astronautical Congress, IAC*, 2018.

[49] J. Liu and H. Li. Artificial potential function safety and obstacle avoidance guidance for autonomous rendezvous and docking with noncooperative target. *Mathematical Problems in Engineering*, 2019, 2019.

[50] S. Liu, Q. Zhang, and D. Zhou. Obstacle avoidance path planning of space manipulator based on improved artificial potential field method. *Journal of The Institution of Engineers (India): Series C*, 95(1):31–39, 2014.

[51] K. M. Lynch and F. C. Park. *Modern robotics*. Cambridge University Press, 2017.

[52] F. L. Markley and J. L. Crassidis. *Fundamentals of spacecraft attitude determination and control*. Springer, 2014.

[53] S. B. McCamish, M. Romano, S. Nolet, C. M. Edwards, and D. W. Miller. Flight testing of multiple-spacecraft control on spheres during close-proximity operations. *Journal of Spacecraft and Rockets*, 46(6):1202–1213, 2009.

[54] P. Melroy, L. Hill, E. E. Fowler, R. Hunter, J. Eagen, B. R. Sullivan, P. Will, and

J. Palmer. Darpa phoenix satlets: Progress towards satellite cellularization. In *AIAA SPACE 2015 Conference and Exposition*, page 4487, 2015.

[55] S. A. A. Moosavian and E. Papadopoulos. On the kinematics of multiple manipulator space free-flyers and their computation. *Journal of Robotic Systems*, 15(4): 207–216, 1998.

[56] S. A. A. Moosavian and E. Papadopoulos. Free-flying robots in space: an overview of dynamics modeling, planning and control. *Robotica*, 25(5):537–547, 2007.

[57] Z. Mu, W. Xu, and B. Liang. Avoidance of multiple moving obstacles during active debris removal using a redundant space manipulator. *International Journal of Control, Automation and Systems*, 15(2):815–826, 2017.

[58] R. Mukherjee and Y. Nakamura. Nonholonomic redundancy of space robots and its utilization via hierarchical liapunov functions. In *1991 American Control Conference*, pages 1491–1496. IEEE, 1991.

[59] J. Munoz, G. Boyarko, and N. Fitz-Coy. Rapid path-planning options for autonomous proximity operations of spacecraft. In *AIAA/AAS Astrodynamics Specialist Conference*, page 7667, 2010.

[60] Y. Nakamura and R. Mukherjee. Nonholonomic path planning of space robots via bi-directional approach. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 1764–1769. IEEE, 1990.

[61] NASA. Osam-1 mission on-orbit servicing, assembly, and manufacturing 1. URL `https://nexis.gsfc.nasa.gov/osam-1.html`.

[62] NASA. Nasa's robotic refueling mission practices new satellite-servicing tasks, 2017. URL `https://www.nasa.gov/mission_pages/station/research/news/rrm_practice.html`.

[63] D. N. Nenchev, K. Yoshida, P. Vichitkulsawat, and M. Uchiyama. Reaction null-space control of flexible structure mounted manipulator systems. *IEEE Transactions on Robotics and Automation*, 15(6):1011–1023, 1999.

[64] M. Oda, K. Kibe, and F. Yamagata. Ets-vii, space robot in-orbit experiment satellite. In *Proceedings of IEEE international conference on robotics and automation*, volume 1, pages 739–744. IEEE, 1996.

[65] E. Papadopoulos, I. Tortopidis, and K. Nanos. Smooth planning for free-floating

space robots using polynomials. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4272–4277. IEEE, 2005.

[66] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello. Robotic manipulation and capture in space: A survey. *Frontiers in Robotics and AI*, page 228, 2021.

[67] L. F. Penin. Teleoperation with time delay-a survey and its use in space robotics. *Technical Report of National Aerospace Laboratory*, 2002.

[68] R. A. Peters, C. L. Campbell, W. J. Bluethmann, and E. Huber. Robonaut task learning through teleoperation. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 2806–2811. IEEE, 2003.

[69] P. Piersigilli, I. Sharf, and A. Misra. Reactionless capture of a satellite by a two degree-of-freedom manipulator. *Acta Astronautica*, 66(1-2):183–192, 2010.

[70] R. Rathee and P. M. Pathak. Dual arm free flying space robot trajectory planning using polynomial. *Journal of Robotics*, 2015, 2015.

[71] D. Reintsema, J. Thaeter, A. Rathke, W. Naumann, P. Rank, and J. Sommer. Deos–the german robotics approach to secure and de-orbit malfunctioned satellites from low earth orbits. In *Proceedings of the i-SAIRAS*, pages 244–251. Japan Aerospace Exploration Agency (JAXA) Japan, 2010.

[72] T. Rybus. Obstacle avoidance in space robotics: Review of major challenges and proposed solutions. *Progress in Aerospace Sciences*, 101:31–48, 2018.

[73] T. Rybus. Point-to-point motion planning of a free-floating space manipulator using the rapidly-exploring random trees (rrt) method. *Robotica*, 38(6):957–982, 2020.

[74] T. Rybus and K. Seweryn. Application of rapidly-exploring random trees (rrt) algorithm for trajectory planning of free-floating space manipulator. In *2015 10th International Workshop on Robot Motion and Control (RoMoCo)*, pages 91–96. IEEE, 2015.

[75] C. Sallaberger, S. P. T. Force, and C. S. Agency. Canadian space robotic activities. *Acta astronautica*, 41(4-10):239–246, 1997.

[76] A. Saplan. Robotic servicing of geosynchronous satellites (rsgs). URL `https://www.darpa.mil/program/robotic-servicing-of-geosynchronous-satellites`.

[77] N. Sato and Y. Wakabayashi. Jemrms design features and topics from testing. In *6th International symposium on artificial intelligence, robotics and automation in space (iSAIRAS), Quebec*, volume 214, 2001.

[78] A. Seddaoui and C. M. Saaj. Collision-free optimal trajectory generation for a space robot using genetic algorithm. *Acta Astronautica*, 179:311–321, 2021.

[79] A. Seddaoui, C. M. Saaj, and M. H. Nair. Modeling a controlled-floating space robot for in-space services: A beginner's tutorial. *Frontiers in Robotics and AI*, 8, 2021.

[80] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2009.

[81] M. Smith, D. Craig, N. Herrmann, E. Mahoney, J. Krezel, N. McIntyre, and K. Goodliff. The artemis program: an overview of nasa's activities to return humans to the moon. In *2020 IEEE Aerospace Conference*, pages 1–10. IEEE, 2020.

[82] J. Sullivan, S. Grimberg, and S. D'Amico. Comprehensive survey and assessment of spacecraft relative motion dynamics models. *Journal of Guidance, Control, and Dynamics*, 40(8):1837–1859, 2017.

[83] B. Taylor, G. Aglietti, S. Fellowes, S. Ainley, T. Salmon, I. Retat, C. Burgess, A. Hall, T. Chabot, K. Kanan, et al. Remove debris mission, from concept to orbit. In *SmallSat 2018-32nd Annual AIAA/USU Conference on Small Satellites*, pages 1–10, 2018.

[84] D. Vallado. *Fundamentals of Astrodynamics and Applications*. McGraw Hill, 1997.

[85] J. Virgili-Llop, C. Zagaris, R. Zappulla, A. Bradstreet, and M. Romano. Laboratory experiments on the capture of a tumbling object by a spacecraft-manipulator system using a convex-programming-based guidance. In *AAS/AIAA Astrodynamics Specialist Conference, Stevenson*, pages 20–24, 2017.

[86] J. Virgili-Llop, C. Zagaris, R. Zappulla, A. Bradstreet, and M. Romano. A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator. *The International journal of robotics research*, 38(1):40–72, 2019.

[87] D. M. Waltz. *On-orbit servicing of space systems*. Krieger Publishing Company, 1993.

[88] X.-P. Wei, J.-X. Zhang, D.-S. Zhou, and Q. Zhang. Optimal path planning for minimizing base disturbance of space robot. *International Journal of Advanced Robotic Systems*, 13(2):41, 2016.

[89] D. Wenberg, A. Hardy, T. Lai, C. Wellins, and J. Kang. Advancing on-orbit assembly with isar. 2018.

[90] D. L. Wenberg, B. P. Keegan, M. E. Lange, E. A. Hanlon, and J. S. Kang. Rsat flight qualification and test results for manipulable robotic appendages installed on 3u cubesat platform. 2016.

[91] R. L. Werneth. Lessons learned from hubble space telescope extra vehicular activity supportability. In *AIP Conference Proceedings*, volume 552, pages 176–188. American Institute of Physics, 2001.

[92] W. Whittaker, C. Urmson, P. Staritz, B. Kennedy, and R. Ambrose. Robotics for assembly, inspection, and maintenance of space macrofacilities. In *AIAA Space*, volume 2000, pages 1–6, 2000.

[93] M. Wilde, S. Kwok Choon, A. Grompone, and M. Romano. Equations of motion of free-floating spacecraft-manipulator systems: an engineer's tutorial. *Frontiers in Robotics and AI*, 5:41, 2018.

[94] W. Xu, B. Liang, C. Li, Y. Xu, and W. Qiang. Path planning of free-floating robot in cartesian space using direct kinematics. *International Journal of Advanced Robotic Systems*, 4(1):4, 2007.

[95] W. Xu, Y. She, and Y. Xu. Analytical and semi-analytical inverse kinematics of ssrms-type manipulators with single joint locked failure. *Acta Astronautica*, 105(1): 201–217, 2014.

[96] Y. Yanoshita and S. Tsuda. Space robot path planning for collision avoidance. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 2. Citeseer, 2009.

[97] K. Yoshida. Ets-vii flight experiments for space robot dynamics and control. In *Experimental Robotics VII*, pages 209–218. Springer, 2001.

[98] K. Yoshida and B. Wilcox. *Space Robots and Systems*, pages 1031–1063. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[99] K. Yoshida, K. Hashizume, and S. Abiko. Zero reaction maneuver: flight validation with ets-vii space robot and extension to kinematically redundant arm. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 1, pages 441–446. IEEE, 2001.

[100] T. Yoshikawa. Manipulability and redundancy control of robotic mechanisms. *Pro-*

ceedings. 1985 IEEE International Conference on Robotics and Automation, 2:1004–1009, 1985.

[101] R. I. Zappulla, H. Park, J. Virgili-Llop, and M. Romano. Experiments on autonomous spacecraft rendezvous and docking using an adaptive artificial potential field approach. 2016.

[102] L. Zhang, Q. Jia, G. Chen, and H. Sun. Pre-impact trajectory planning for minimizing base attitude disturbance in space manipulator systems for a capture task. *Chinese Journal of Aeronautics*, 28(4):1199–1208, 2015.

[103] X. Zhang, J. Liu, and J. Liu. Optimal coordinated planning strategy for space robots grasping targets. In *2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 288–293. IEEE, 2017.

[104] X. Zhihui, L. Jinguo, W. Chenchen, and T. Yuchuang. Review of in-space assembly technologies. *Chinese Journal of Aeronautics*, 34(11):21–47, 2021.

[105] D. Zhou, Z. Yu, Y. Zhang, and S. Li. Translational and rotational motion planning for spacecraft close proximity using sampling-based methods. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(10):3680–3699, 2019.